UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

DISEÑO DE UN CONTROL EN LAZO CERRADO DE INSULINA Y UN SISTEMA DE DOSIFICACIÓN DE INSULINA

.

Estéfano Andrés Gómez Mosquera

Ingeniería Electrónica

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero Electrónico

Quito, 13 de mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

DISEÑO DE UN CONTROL EN LAZO CERRADO DE INSULINA Y UN SISTEMA DE DOSIFICACIÓN DE INSULINA

.

Estéfano Andrés Gómez Mosquera

Nombre del profesor, Título académico	Alberto Sanchez, Ph.D.
Nombre del profesor, Título académico	Malena Loza, MSc.

Quito, 13 de mayo de 2020

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Nombres y apellidos: Estéfano Andrés Gómez Mosquera

Código: 00124022

Cédula de identidad: 1722861083

Lugar y fecha: Quito, 13 de mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

La diabetes tipo 1, es una enfermedad que afecta a más de 422 millones de personas en todo el mundo. Actualmente, existen muchas soluciones para combatir la enfermedad. Las soluciones más comunes son el uso de la terapia de inyecciones múltiples y el uso de bombas de insulina. Debido a la prevalencia de la enfermedad, se diseña un control en lazo cerrado de insulina y un sistema de dosificación de insulina. En el presente trabajo, se presenta el diseño de estos dos sistemas y se muestran sus resultados. En primer lugar, se modela un paciente virtual, utilizando el modelo mínimo de Bergman. A continuación, se implementa un controlador PID que actúa sobre el paciente virtual. Por último, se diseña un sistema de dosificación de insulina, donde se muestra su programación, conjuntamente con los diagramas mecánicos, electrónicos y una requisición de materiales para su construcción.

Palabras clave: bomba, diabetes, control lazo cerrado, dosificación de insulina, sistema

ABSTRACT

Type 1 diabetes is a disease that affects more than 422 million people around the world. Now a days, there are many solutions to fight the disease. The most common solutions are the use of multiple dose insulin injection therapy and the use of insulin pumps. Due to the prevalence of the disease, a closed loop insulin control was designed along an insulin dosing system. In the following study, the design of these systems is carried out and its results are presented. Firstly, a virtual patient is modelled using the Bergman minimal model. Using the virtual patient, a PID controller is implemented. Finally, an insulin dosing system is designed, where its programming is shown along the mechanical and electronic diagrams in addition to a material requisition for its construction.

Key words: pump, diabetes, closed loop control, insulin dosing, system

TABLA DE CONTENIDO

INTRODUCCIÓN

La diabetes es una enfermedad la cual produce altos niveles de glucosa en el cuerpo (Sperling, 2014). La forma en la que el cuerpo regula los altos niveles de glucosa es usando la hormona insulina. Cuando una persona padece de diabetes no produce insulina o no produce la suficiente cantidad de esta hormona para regular los altos niveles de glucosa. Con la ayuda de la insulina el cuerpo es capaz de absorber la glucosa y usarla como fuente de energía. Ya que este proceso no ocurre con las personas que padecen de esta condición, la glucosa permanece en el cuerpo causando muchas complicaciones para la persona. La Organización Mundial de la Salud (2014), afirma que la diabetes es la causa principal de la ceguera, de la insuficiencia renal y de las amputaciones.

Una solución para controlar la diabetes tipo 1 es a través de varias inyecciones de insulina que se suministran a lo largo del día (Sperling, 2014). Esta terapia de inyecciones múltiples consiste en inyectarse una dosis de insulina de acción lenta una o dos veces al día para actuar como una dosificación de segundo plano. Por otro lado, se necesitan inyecciones extra de insulina de acción rápida para cada comida (Sperling, 2014).

Otro método para tratar la diabetes tipo 1 es a través de una terapia de infusión continua de insulina. La forma de realizar esta terapia es mediante el uso de bombas de insulina. Este tipo de dispositivos suministran insulina a partir de las dosificaciones previamente mencionadas. Una dosificación que se le conoce como dosificación basal, la cual es la dosificación de insulina de segundo plano. El segundo tipo de dosificación, la dosificación bolus, es la dosificación que se necesita cuando la persona consume alimentos.

Observando el uso de las bombas de insulina para tratar la diabetes, se propone el diseño de un control en lazo cerrado de insulina y un sistema de dosificación de insulina. Antes de diseñar estos sistemas hay que tomar en cuenta las limitaciones del diseño de estas. Al hablar de un dispositivo medico siempre existe una organización que regula la creación y la aprobación de estos dispositivos. En Ecuador la construcción de dispositivos médicos se regula a través de la agencia nacional de regulación, control y vigilancia sanitaria (2019). Sin embargo, el documento no presenta la regulaciones de este organismo ya que, la bibliografía de estas regulaciones no es del todo clara.

Se presentan las regulaciones de la Agencia de Medicamentos y Alimentación de los Estados Unidos para la construcción de un dispositivo médico (FDA, 2020). A partir de estas regulaciones se realizan pruebas de los sistemas diseñados de una manera superficial. Se acato estas regulaciones de manera superficial ya que diseñar e implementar un dispositivo medico bajo las regulaciones de la FDA es un tarea que toma muchos años en llevarse a cabo.

A continuación, se presenta un dossier de ingeniería el cual incluye el diseño de un control en lazo cerrado de insulina y un sistema de dosificación de insulina. Para la realización del control en lazo cerrado de insulina se modela un paciente virtual a partir del modelo mínimo de Bergman. A partir del paciente virtual se implementa un controlador PID. Por otro lado, se presenta el sistema de dosificación de insulina mostrando su programación, sus diagramas mecánicos, electrónicos y una requisición de materiales para su construcción.

DOSSIER DE INGENIERÍA

DISEÑO DE UN CONTROL EN LAZO CERRADO DE INSULINA Y UN SISTEMA DE DOSIFICACIÓN DE INSULINA	
Departamento:	Ingeniería Eléctrica y Electrónica
Documento:	DOSSIER DE INGENIERÍA
Código del Documento:	USFQ-DI-001

Revisión:	**Fecha:**	**Próxima Revisión:**	**Elaborado por:**	**Revisado por:**	**Aprobado por:**
0	08/05/2020	1	Estéfano Gómez Mateo Vega	Malena Loza	Alberto Sánchez
1		N/A			

Historial de Revisiones

No. Revisión:	**Fecha:**	**Páginas Revisadas:**	**Motivo de la Revisión:**
0	08/05/2020	TODAS	Revisión alcance
1			

Control de Distribución		
Ubicación del Documento:	**Controlada**	**No Controlada**
Control de Documentos	**X**	

1. OBJETIVO

El documento descrito a continuación tiene como objetivo el diseño de un control en lazo cerrado de insulina y un sistema de dosificación de insulina. Este documento presenta el diseño de un control en lazo cerrado de insulina a partir de un controlador PID y un paciente virtual. El diseño del sistema de suministro de insulina se lo realiza para que este trabaje a partir de una interfaz gráfica y a partir de una aplicación de cálculo y suministro de dosificaciones. Muchos de los valores utilizados para probar los sistemas se los tomo de los datos médicos de un paciente de 9 años.

2. DESCRIPCIÓN PROBLEMA

Diabetes mellitus es un síndrome caracterizado por altos niveles de glucosa en la sangre los cuales no pueden ser procesados por el cuerpo (Sperling, 2014). Esta condición es causada por la deficiencia parcial o absoluta de insulina. El cuerpo necesita tanto de la insulina como del glucagón para mantener los niveles de glucosa en la sangre estables. Tanto la insulina como el glucagón son hormonas secretadas por el páncreas a través de las células beta y de las células alfa respectivamente (Ghista, 2008). La insulina regula los niveles altos de glucosa mientras que el glucagón regula los niveles bajos de la misma. Ya que la glucosa es la principal fuente de energía para el ser humano, es necesario mantener un nivel de glucosa adecuado, lo cual no sucede cuando se padece de diabetes. Las células usan la glucosa como fuente de energía, pero estas no pueden usar la glucosa sin la ayuda de la insulina. La insulina ayuda a las células a absorber la glucosa manteniendo sus niveles en un rango estable. Al no poder secretar insulina, el cuerpo presenta niveles altos de glucosa los cuales son perjudiciales para la salud.

Principalmente existen tres tipos de diabetes: la diabetes tipo 1, diabetes tipo 2 y diabetes gestacional. La característica principal de la diabetes tipo 1 es que el cuerpo produce muy poca insulina o casi nada (Sperling, 2014). Por otro lado, la diabetes tipo 2 afecta la manera en la cual el cuerpo usa la insulina. En este tipo de diabetes las células no responden de manera correcta a la insulina teniendo una resistencia hacia ella (Sperling, 2014). En este tipo de diabetes también puede existir una falta de insulina. Finalmente, la diabetes gestacional ocurre en mujeres embarazadas donde el cuerpo es más sensible a la insulina. Generalmente ocurre en muy pocos casos y se resuelve después del parto.

La diabetes es una de las condiciones médicas más comunes en el mundo que afecta a más de 422 millones de personas siendo la causa principal de la ceguera, de la insuficiencia renal y de las amputaciones (WHO, 2014). La necesidad de encontrar una solución a la enfermedad ha llevado a médicos, científicos e ingenieros a proponer diferentes tipos de tratamientos innovadores en los últimos años (WHO, 2014). Observando la prevalencia de esta enfermedad en todo el mundo cualquier solución para combatir la enfermedad es de gran ayuda para la humanidad. Debido a la cantidad de personas que padecen esta enfermedad y observando el crecimiento de esta nos vimos motivados a crear una solución para tratar la enfermedad a través de un sistema de suministro de insulina.

2.1. Diabetes tipo 1

Para lograr un diseño exitoso tanto del control de insulina en lazo cerrado como del sistema de dosificación de insulina hay que entender de manera minuciosa como actúa la enfermedad en el cuerpo. La diabetes tipo 1 es una enfermedad autoinmune la cual destruye células beta encontradas en el páncreas. Las células beta son responsables de sintetizar la hormona insulina que controla los altos niveles de glucosa del cuerpo. La razón por la cual estas células son destruidas no es entendida del todo por los médicos (Sperling, 2014). Ya que el cuerpo destruye las células beta, el páncreas no puede producir insulina, de esta manera la glucosa que se encuentra en el cuerpo no puede ser utilizada por las células como energía (Sperling, 2014).

La secreción de la insulina es una interacción entre nutrientes, hormonas y el sistema nervioso. La glucosa como otros azúcares estimula la secreción de la insulina modulada por el sistema nervioso. Esta secreción es constantemente controlada por la cantidad, calidad y frecuencia de los nutrientes consumidos. Cuando se consumen carbohidratos y proteínas estos generan señales hormonales las cuales envían impulsos para la secreción de insulina. La insulina una vez que es secretada actúa en las células de los tejidos del hígado (Sperling, 2014). Esta glucosa con la ayuda de la insulina es transformada en energía para el cuerpo.

La forma en la que se maneja la diabetes tipo 1 se la puede dividir en tres fases. Un periodo inicial en la cual se trata la cetoacidosis diabética seguido por un periodo de control metabólico post-acidótico y finalmente con una rutina de regímenes de insulina acompañados con una dieta y ejercicio adecuado (Sperling, 2014). Esta rutina de regímenes de insulina se la conoce como terapia de reemplazo de insulina. Para establecer esta rutina de reemplazo es necesario analizar el estilo de vida del paciente. Generalmente este análisis es de suma importancia hacerlo cuando el paciente es un niño o un adolescente, ya que acoplar algún método de suministro de insulina con el estilo de vida de un paciente de estas características es una tarea complicada. En esta tarea se ven involucrados tanto diabetólogos, educadores, psicólogos y nutriólogos (Sperling, 2014).

Una de las metas de la terapia de reemplazo de insulina es mantener un nivel de glucosa de manera casi normal en el cuerpo. El objetivo de la terapia de reemplazo de insulina es imitar a los patrones de secreción de insulina lo más cercano posible a lo normal como se indica en la figura 1. Este gráfico es de suma importancia para el diseño de los sistemas que se presentan en este documento, ya que uno de los objetivos es replicar esta curva a través de una terapia de insulina exógena.

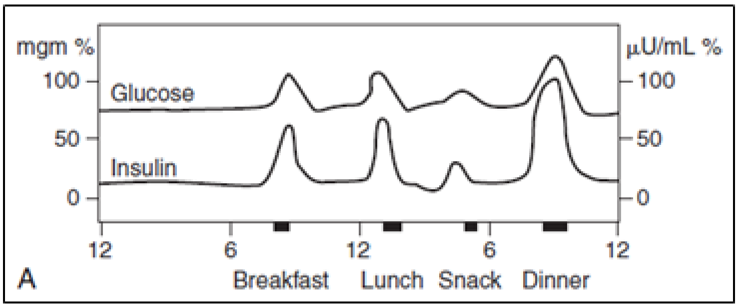


Figura. 1: Patrones de insulinas normales (Sperling, 2014)

Las terapias de reemplazo consisten en replicar las dosificaciones de insulina que el cuerpo realiza de manera endógena. Existen dos tipos de dosificaciones que se utilizan en las terapias de reemplazo. Una dosificación de segundo plano o de ayuno conocida como dosificación basal. Esta dosificación hace referencia a la cantidad de insulina que se necesita cuando no se está comiendo a lo largo del día, noche y entre comidas para mantener un nivel estable de glucosa. Por otro lado, la dosificación bolus es la cantidad de insulina que se necesita cuando se consume algún alimento durante el día.

Ya que la administración de la insulina es exógena, a través del tejido subcutáneo, la proporción con la cual es absorbida por el cuerpo varía continuamente. Las dosis que son administradas siempre son aproximaciones a la acción de la insulina real por lo tanto ningún régimen de reemplazo de insulina reproducirá los patrones de la secreción de insulina de manera exacta. Con estas aproximaciones pueden existir periodos donde se tenga tanto niveles bajos como niveles altos de glucosa. Por consiguiente, la meta de los regímenes de insulina es minimizar la frecuencia y la severidad entre los niveles altos y bajos de glucosa. Para minimizar la frecuencia es necesario una correcta administración de la insulina al igual que un cálculo lo más aproximado de los carbohidratos consumidos para administrar una dosis correcta (Sperling, 2014).

Como se mencionó anteriormente las dosis administradas de insulina tienen como objetivo mantener los niveles de glucosa en la sangre siguiendo la curva de la figura 1. A esta medida de la concentración de la glucosa en la sangre se la conoce como glucemia la cual generalmente se la mide en unidades de miligramos por decilitro [mg/dl] o milimoles por litro [mmol/l] (Sperling, 2014). En este documento se utiliza la medida de [mg/dl] para la concentración de glucosa en la sangre.

El objetivo de los sistemas presentados en este documento es mantener los niveles de glucosa en la sangre estables. Para lograr este objetivo necesitamos saber los valores de glucemia de una persona sana y de una persona con diabetes. Se le conoce como normo-glucemia a la concentración normal de glucosa en la sangre la cual está entre 70-120 [mg/dl] en ayuno (Gomez, 2017). Esta medida aumenta cuando se consumen alimentos hasta valores de 140 [mg/dl] y luego bajan a un nivel normal en una persona sana. Por otro lado, una persona con diabetes puede experimentar niveles altos de glucosa conocidos como hiperglucemia la cual tiene valores que superan los 180 [mg/dl] (Gomez, 2017). De igual manera puede experimentar niveles bajos de glucosa conocidos como hipoglucemia la cual tiene valores menores de 70 [mg/dl] (Gomez, 2017).

2.2. Soluciones actuales

Hoy en día los métodos más usados para una terapia de reemplazo de insulina son las terapias de inyecciones múltiples y la terapia de reemplazo a través de una bomba de insulina (Sperling, 2014). Las inyecciones múltiples tratan de replicar la acción normal de la secreción de insulina a través del uso de una insulina de acción lenta para reemplazar las necesidades de la dosificación basal. Por otro lado, también se necesitan inyecciones con dosificaciones mayores para cubrir las necesidades del consumo de alimentos.

Lamentablemente la terapia de inyecciones múltiples presenta muchos problemas en los pacientes de diabetes. La principal molestia al usar este tipo de terapia es la necesidad de suministrar múltiples inyecciones durante el día lo cual es inconveniente para el paciente. Por otro lado, las dosis bolus suministradas siempre se las calcula entre la relación de carbohidratos que se van a consumir y la insulina necesaria. Esta relación es lo suficientemente satisfactoria en la mayoría de los casos sin embargo muchas veces el paciente necesita una cantidad distinta de insulina. Generalmente los pacientes que utilizan esta técnica están atados a utilizar un glucómetro el cual después de una muestra de sangre pueden observar el nivel de glucosa y a partir de esto realizar el cálculo entre los carbohidratos consumidos y la insulina necesaria.

Todos los problemas mencionados anteriormente pueden ser resueltos a través del uso de una bomba de insulina (Sperling, 2014). Las bombas de insulina usan el principio de infusión de insulina continua subcutánea. Donde se administra insulina de manera continua cuando se está en un estado basal y se programa de forma manual la cantidad de insulina necesaria en un estado bolus. Sperling (2014) afirma que usando una bomba de insulina la frecuencia en la que existen niveles altos de glucosa se disminuye considerablemente al compararlo con la terapia de múltiples inyecciones. Otra ventaja que proporcionan las bombas de insulina es que los ritmos del estado basal son programables.

En lugar de varias inyecciones por día los dispensadores de insulina de las bombas pueden almacenar insulina para 2 o 3 días. Existen calculadoras programadas en estos dispositivos en los cuales solo se ingresan los carbohidratos a consumir y el dispositivo sabe cuál es la dosis requerida. La programabilidad de estos dispositivos es de gran utilidad, ya que no siempre se necesita la misma cantidad de insulina basal. Finalmente, la ventaja de una bomba de insulina es que esta guarda la información de cómo fue dispensada la insulina y tiene estos registros organizados por días.

2.3. Regulaciones FDA

Elleri, Dunger y Hovorka (2011) afirman que la solución actual que presenta resultados eficientes contra la diabetes tipo 1 radica en la construcción de un dispositivo de suministro de insulina en lazo cerrado. Estos dispositivos se los puede construir con una bomba de insulina acoplada con un DMCG (dispositivo de monitoreo continuo de glucosa) los cuales eliminan la necesidad de depender de un glucómetro y de las múltiples inyecciones durante el día. Al acoplar estos dos sistemas y realizar un suministro en lazo cerrado se elimina la necesidad de programar manualmente las secreciones del dispositivo como es en el caso con las bombas de insulina convencionales. Uno de los obstáculos después de diseñar y construir este sistema acoplado son las pruebas de funcionamiento del dispositivo y el uso de este en pacientes.

La mejor forma para realizar pruebas de funcionamiento es rigiéndose a las regulaciones de la FDA (administración de alimentos y medicamentos) de los Estados Unidos. La FDA regula alimentos, medicamentos, cosméticos, productos biológicos y dispositivos médicos para que estos puedan estar en el mercado de manera segura para el consumidor. Cuando se trata de dispositivos médicos, la FDA clasifica a estos en tres categorías dependiendo del riesgo que suponen para el paciente. Siendo la clase I la de menor riesgo y la clase III la más riesgosa (FDA, 2020). Al hablar de una bomba de insulina independientemente que esta sea acoplada con un DMCG o simplemente una bomba manual, inmediatamente la FDA da una categoría de clase II al dispositivo (FDA, 2020).

Para que un dispositivo de clase II salga al mercado en este caso una bomba de insulina, esta debe cumplir las siguientes regulaciones. En primer lugar, mostrar datos de todas las funciones del dispositivo en funcionamiento, mostrar datos robustos de una validación clínica de que el dispositivo funciona para combatir la diabetes tipo I, demostrar que una muestra significativa de usuarios entiende el funcionamiento del dispositivo y que estos saben qué hacer en caso de un mal funcionamiento y mostrar datos de fiabilidad del dispositivo con el uso (FDA, 2020). Todos estos puntos resumen de una manera superficial las regulaciones reales de un dispositivo de clase II categorizado por la FDA. Si se tratase de un dispositivo de clase III, como en un dispositivo de suministro de insulina en lazo cerrado, todas las regulaciones mencionadas anteriormente se tienen que acatar además de incluir varios estudios clínicos y no clínicos de una manera detallada.

Como se observa crear un dispositivo de suministro de insulina independientemente si es uno manual o uno en lazo cerrado es una tarea compleja en su diseño y construcción. La cual requiere de un equipo de ingenieros y médicos para que este salga al mercado y sea seguro para los pacientes. En este proyecto se detalla el diseño de un control en lazo cerrado de insulina además de presentar un prototipo de un sistema de dosificación de insulina.

A partir de las regulaciones de la FDA que se presentaron, se realizaron pruebas de funcionamiento del sistema de dosificación de insulina a un nivel de un dispositivo clase II acatándose solo a ciertas regulaciones de esta clase de manera superficial. Se realizo pruebas del dispositivo funcionando de manera continua para el suministro de insulina que posee. Se observo al dispositivo dispensando las dosis de manera adecuada tanto en su dosificación basal y bolus a través de la conexión con la aplicación y a través su interfaz gráfica.

3. CONTROL EN LAZO CERRADO DE INSULINA

La forma más segura y efectiva de cualquier régimen de insulina depende de un monitoreo frecuente de los niveles de glucosa en la sangre (Sperling, 2014). Con el paso del tiempo los glucómetros han sido reemplazados por dispositivos de monitoreo continuo de glucosa los cuales presentan en tiempo real los niveles de glucosa en el cuerpo de una manera permanente. De esta manera se tiene información de la tendencia de la glucosa en el cuerpo y cómo esta cambia con el tiempo. Con los DMCG los usuarios pueden manejar los niveles altos y bajos de glucosa de una forma más eficiente (Sperling, 2014).

Combinando los dispositivos de monitoreo continuo de glucosa con las bombas de insulina se han desarrollado varios dispositivos de suministro de insulina en lazo cerrado (Sperling 2014). Estos dispositivos solo son prototipos, ya que en el mercado existen pocos sistemas de dosificación en lazo cerrado aprobados por la FDA (FDA, 2019). Los dispositivos en lazo cerrado utilizan la información del sensor de glucosa para suministrar la cantidad adecuada de insulina durante el día a través de su sistema inteligente (Tandem Diabetes Care, 2020).

Una vez explorada la diabetes tipo 1 en sus causas, en su prevención y en sus tipos de tratamientos, un sistema de suministro de insulina acoplado con un DMCG es una buena opción para tratar la diabetes tipo 1. Por esta razón, se propone un control en lazo cerrado de insulina siguiendo los conceptos de las dosificaciones basal y bolus. En las siguientes secciones se presenta un modelo de un paciente virtual a partir del modelo mínimo de Bergman (1981). A partir de este paciente virtual se implementó un controlador PID para crear un sistema de suministro de insulina en lazo cerrado.

3.1. Diseño paciente virtual

Existen modelos matemáticos los cuales se los utiliza para modelar la relación de glucosa-insulina en el organismo. Estos modelos son de gran utilidad, ya que funcionan para diagnosticar y tratar la diabetes tipo 1 (Palma, 2013). Al hablar de una relación entre glucosa-insulina y cómo esta actúa en distintos pacientes, es necesario usar un modelamiento farmacocinético y farmacodinámico (Palma, 2013). El modelado farmacocinético se encarga de observar la absorción, la distribución metabólica y la eliminación de medicamentos y otros compuestos en el organismo. Por otro lado, el modelamiento farmacodinámico se encarga de observar los efectos de estos medicamentos en los procesos biológicos (Palma, 2013).

Al entender cómo la administración de un medicamento afecta su concentración dinámica, los médicos pueden observar el nivel de toxicidad del medicamento. Pueden observar las dosis requeridas y ajustarlas para evitar complicaciones graves y extrapolar datos entre modelos animales y humanos (Palma, 2013). A partir del modelamiento farmacocinético y farmacodinámico se puede desarrollar un modelamiento epidemiológico. El objetivo de este modelamiento es observar el efecto de las terapias de medicamentos a un nivel individual y observar el desarrollo de complicaciones a largo plazo. Con el modelamiento epidemiológico podemos evaluar los efectos de un medicamento en una población de pacientes (Palma, 2013).

Hovorka (2002), Sorensen (1985) y Cobelli (2014) crearon varios modelos matemáticos los cuales utilizan los conceptos del modelamiento epidemiológico, acoplando modelamiento farmacocinético y farmacodinámico para simular la relación glucosa-insulina del cuerpo humano. Estos modelos son los suficientemente robustos, ya que incluyen una gran cantidad de variables involucradas en la relación glucosa-insulina del cuerpo humano y por ende lo replican de forma precisa. Por otro lado, Bergman (1981) creó el modelo mínimo de Bergman el cual es ampliamente usado para el modelamiento farmacocinético y farmacodinámico de la relación glucosa-insulina.

A partir de estos modelos se han creado controladores para tratar la diabetes tipo 1 a través de simulaciones. En el trabajo de Parker (1996) se observa un controlador de modelo predictivo. Esta estrategia de control predice las salidas del sistema basado en un modelo lineal. El controlador calcula las acciones futuras de control necesarias para una salida deseada a través de un algoritmo de optimización. Este controlador a pesar de mostrar buenos resultados no es del todo estable cuando se lo utiliza para distintos tipos de pacientes (Parker, 1996).

Por otro lado, Schlotthauer (2002) para mitigar la inestabilidad de los resultados de un control predictivo, propone un control predictivo no lineal que predice de mejor manera la insulina necesaria para el paciente. Otros tipos de controladores que se han propuesto son los modelos de redes neuronales de Bamgbose (2017), un controlador Fuzzy de Maleki (2011), un controlador Fuzzy predictivo de Hachimi (2018) y muchos otros con resultados óptimos.

Todos estos modelos de glucosa-insulina y todos los controladores que utilizan estos modelos, para simular la inyección de insulina exógena, han sido evidencia de que un sistema de suministro de insulina con control en lazo cerrado es inminente. Sin embargo, ninguno de estos modelos, a pesar de tener resultados sorprendentes, ha sido implementado para su uso en pruebas clínicas. Esto quiere decir que la realización de un sistema de suministro de insulina con control en lazo cerrado es una tarea difícil y requiere la prueba no solo de diversos controladores, sino de diversos modelos que repliquen el cuerpo humano de una manera precisa y exacta. Para la realización de este dispositivo se necesita la ayuda de diabetólogos e ingenieros. En primer lugar, para desarrollar un modelo íntegro del cuerpo humano a partir de la modelación matemática. Por otro lado, para desarrollar un controlador que tome en cuenta todas las necesidades de un paciente con diabetes tipo 1.

A pesar de que existen varios modelos lo suficientemente robustos para simular la relación glucosa-insulina del cuerpo humano, se utilizó el modelo mínimo de Bergman para el desarrollo del paciente virtual. En los trabajos de Gillis et al. (2007), Jensen (2007) y Palma (2013) se evidencia al modelo mínimo de Bergman como el modelo con más prevalencia para tratar la diabetes tipo 1. Este modelo ha sido de gran utilidad ya que es lo suficientemente simplificado y robusto para describir el metabolismo de glucosa-insulina. Nalini, Balaji y Gayathiri (2016) utilizaron el modelo para el desarrollo de un control en lazo cerrado por lo tanto se evidencia su uso en este tipo de sistemas.

El modelo mínimo de Bergman es un modelo mono compartimental. Esto significa que el cuerpo se modela como un compartimiento con una concentración basal de glucosa e insulina (Bergman, 1981). Este modelo mínimo contiene dos submodelos mínimos, uno para describir la cinética de la glucosa y uno para describir la cinética de la insulina. El primer modelo mínimo describe cómo la concentración de glucosa reacciona a la concentración de la insulina en la sangre. Mientras que el segundo modelo describe cómo la concentración de la insulina en la sangre reacciona a la concentración de glucosa.

Este modelo se desarrolló para observar la relación de glucosa-insulina a partir de una prueba PTGIV (prueba de tolerancia a la glucosa intravenosa) (Bergman, 1981). Esta prueba se la realiza para observar la secreción de insulina en pacientes sanos. Por lo tanto, el modelo como lo planteo Bergman tiene como objetivo describir la cinética de la relación de glucosa-insulina a partir de la prueba PTGIV. Como se observa el modelo original no es de gran utilidad para simular tratamientos para la diabetes tipo 1. Sin embargo, con el modelo propuesto por Palma (2013) podemos obtener un modelo extendido de Bergman el cual puede describir la ingesta de alimentos y una infusión exógena de insulina.

Al tratarse de un control en lazo cerrado de insulina, el modelo extendido de Bergman es de gran utilidad ya que, usamos insulina exógena para controlar los niveles de glucosa. Además, podemos simular cómo se comporta el controlador cuando existe una ingesta de alimentos.

El modelo mínimo de Bergman extendido propuesto por Palma (2013) consiste en un modelado con tres compartimientos. Un compartimiento G que describe la concentración de glucosa en plasma, un compartimiento X que describe la acción remota de la insulina y un tercero I que corresponde a la insulina en plasma. A partir de estos tres compartimientos se tienen las ecuaciones 1-3 que modelan la relación entre glucosa e insulina tomando en cuenta el efecto de la ingesta de alimentos en las concentraciones de glucosa. Por otro lado, las ecuaciones 4-6 describen un modelo validado de la ingesta de alimentos (Palma, 2013).

$$\dot{G} = -p_1(G - G_b) - S_i X G + \frac{f k_{abs}}{V_G} G_{gut} \quad (1)$$

$$\dot{X} = -p_2(X - I - I_b) \quad (2)$$

$$\dot{I} = u - k_e I \quad (3)$$

$$\dot{q}_1 = u - k_{emp} q_1 \quad (4)$$

$$\dot{q}_2 = k_{emp}(q_1 - q_2) \quad (5)$$

$$\dot{G}_{gut} = k_{emp} q_2 - k_{abs} G_{gut} \quad (6)$$

La ecuación 1 toma en cuenta p_1 como la efectividad de la glucosa, G_b la glucosa en plasma en estado estable, S_i la efectividad de la insulina, f la fracción de los carbohidratos de alimentos ingeridos que están disponibles para la absorción en el estómago, k_{abs} la proporción de los carbohidratos absorbidos en el torrente sanguíneo desde el estómago y V_G el volumen de la distribución de la glucosa en plasma. La ecuación 2 toma en cuenta p_2 como la proporción fraccionaria de insulina depurada remota e Ib como la concentración basal de la insulina en plasma. La ecuación 3 toma en cuenta u como la proporción de infusión de insulina y k_e la proporción de la insulina en plasma depurada.

Por otro lado, para las ecuaciones de la ingesta de alimentos se tiene en la ecuación 4 a q_1 como la masa de carbohidrato en el compartimiento del estómago, u la ingesta de alimento, k_{emp} la proporción constante de vaciamiento gástrico. La ecuación 5 toma en cuenta q_2 como la masa de carbohidratos en el estómago en el segundo compartimiento. Finalmente, la ecuación 6 toma en cuenta k_{abs} como la proporción constante de absorción de carbohidrato desde el estómago y G_{gut} como la masa de carbohidratos en el estómago.

3.2. Diseño controlador PID

El modelo anteriormente descrito toma el modelo mínimo de Bergman y lo expande para poder usarlo en simulaciones de ingesta de comida. Este nuevo modelo extendido servirá para probar un controlador y observar como éste se comporta cuando existe una ingesta de comida. De esta manera, podemos observar cómo el controlador PID acoplado con el modelo de Palma (2013) dispensa insulina dependiendo de los carbohidratos consumidos y del nivel de glucosa.

A partir del modelo mínimo de Bergman extendido se utiliza el controlador propuesto por Hedengren (2020) y los conceptos de Mahmud (2017) para simular un controlador PID. El controlador de Hedengren (2020) es un controlador PID que trabaja en lazo cerrado a partir de las ecuaciones descritas anteriormente. Con este controlador se ajusta sus términos P, I y D para trabajar de acorde a las necesidades del paciente presentadas a continuación.

Para las acciones de control del controlador PID se toma en cuenta los niveles de glucemia de un paciente de 9 años. El nivel ideal de glucosa en la sangre del paciente es de 120 [mg/dl]. Se utiliza este valor para definir el setpoint del controlador. Por otro lado, la dosificación basal sigue las siguientes medidas de acuerdo con la hora del día. De 8pm a 8am se requieren 0.75 unidades por hora mientras que de 8am a 8pm se requiere 1 unidad por hora. De acuerdo con las indicaciones del médico se puede trabajar con 3 pulsos por hora. Para la dosificación bolus se puede suministrar 1 unidad por pulso dependiendo de las necesidades de los carbohidratos ingeridos.

Se modela todo el sistema en Simulink como se observa en la figura 2. Este modelo en Simulink toma las ecuaciones del modelo mínimo y sus parámetros nominales para crear una función S. Esta función actúa como un paciente virtual al cual se le inyecta insulina de manera automática dependiendo del nivel de glucosa. El código de la función S se lo encuentra en la sección de anexos del informe. Como se indica en la figura 2 el controlador tiene límites de operación del suministro de insulina dependiendo del nivel de glucosa. Los rangos de suministro de insulina están entre 0-10 $\mu U/min$. Por otro lado, los rangos de glucosa del controlador están entre 100 y 140 [mg/dl]. El setpoint al cual el controlador está ajustado es 120 [mg/dl]. Estos límites se los estableció a partir de los datos del paciente y a partir de los rangos de glucosa sugeridos por Gomez (2017).

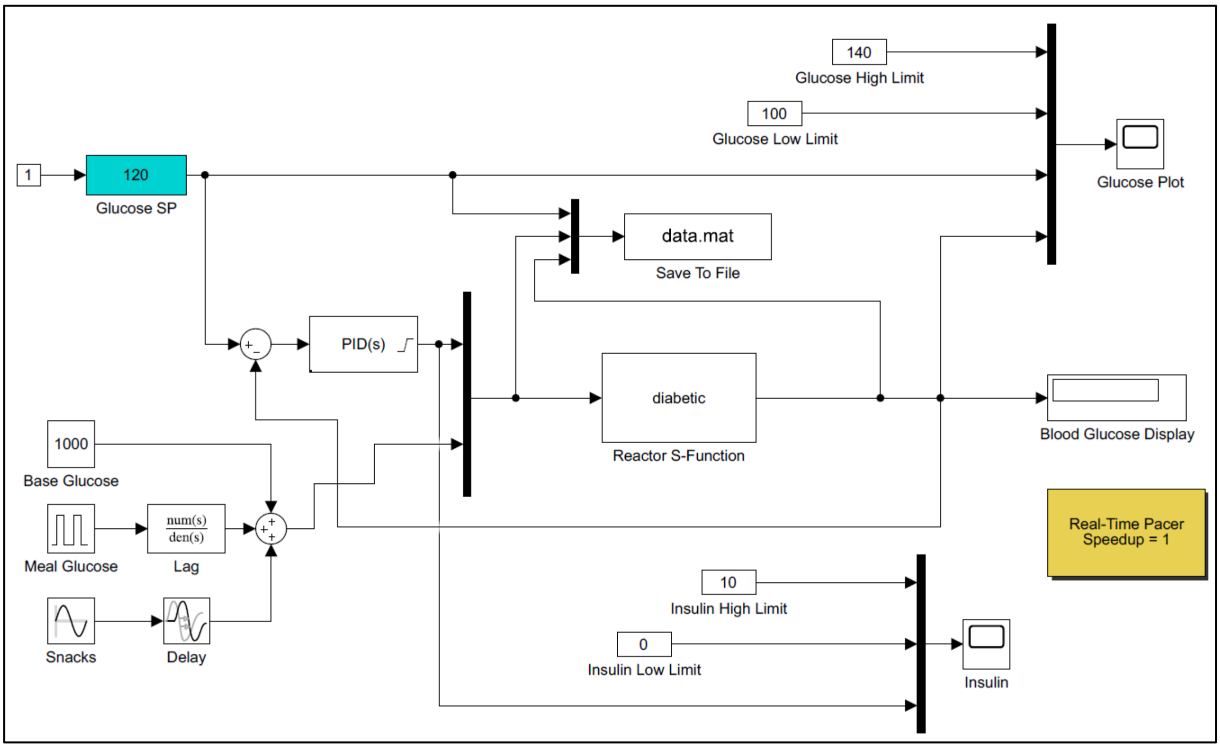


Figura. 2: Simulación del controlador PID en lazo cerrado

El término P del controlador ajusta el suministro de insulina a partir del nivel de glucosa medido, el termino I ajusta el suministro de insulina a partir del área bajo la curva entre la diferencia del nivel de glucosa medido y el setpoint, el termino D suministra insulina en respuesta del cambio del nivel de glucosa en el tiempo. El sistema simula las 24 horas del día y una ingesta de tres comidas en tres horas distintas. Una a las 7:00 am para simular el desayuno, una a la 1 pm para simular el almuerzo y una a las 7 pm para simular la cena. A partir de esto se ajusta el controlador PID con la herramienta de MATLAB PID tuner como se muestra en la figura 3. A partir de los valores del tiempo de subida, el tiempo de asentamiento y la sobre elongación de la respuesta al paso del sistema se ajustan los valores del controlador.

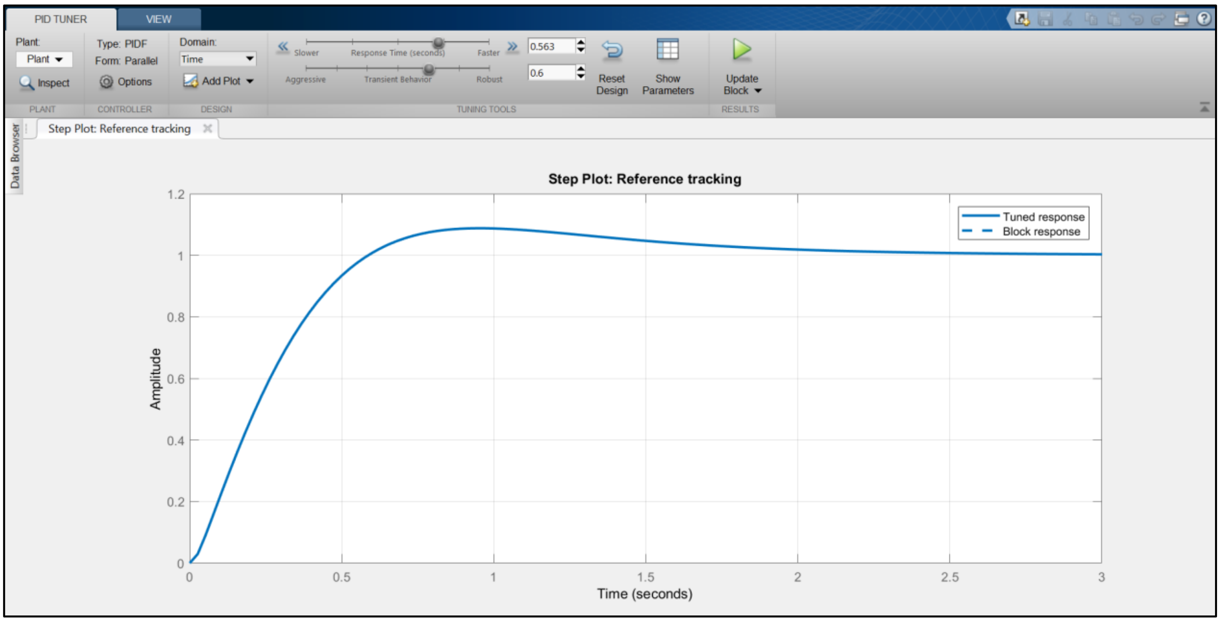


Figura. 3: Respuesta del sistema ante una entrada paso

Una vez que se ajusta el controlador se exporta su respuesta al bloque del diagrama en Simulink de la figura 2 y se obtienen los valores de la figura 4. Ya que el controlador PID esta ajustado para la respuesta deseada se procede a realizar la simulación para 24 horas del controlador.

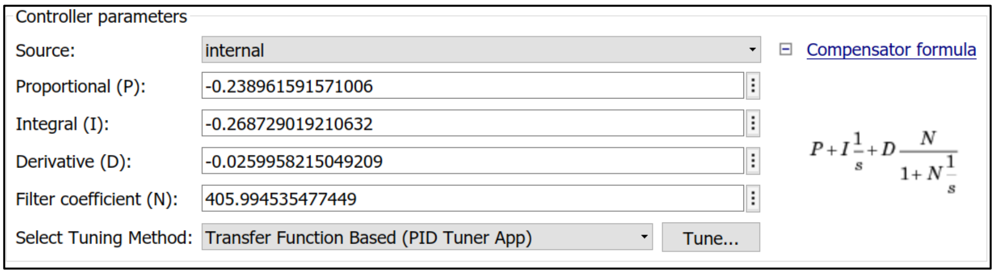


Figura. 4: Parámetros PID

En la figura 5 se observa la respuesta del controlador la cual muestra la insulina suministrada, la glucosa que se digiere y la concentración de la glucosa en el cuerpo. Se observa que el controlador suministra insulina dependiendo de la ingesta de alimentos durante el día y que los valores de glucosa que presenta el paciente no superan los rangos determinados en el diagrama de bloques. El controlador PID se comporta de manera correcta a partir de los límites de insulina de suministro y los límites de glucosa del paciente. Los valores de glucosa que el paciente alcanza durante el día se encuentran en un rango seguro de 105-137 [mg/dl].

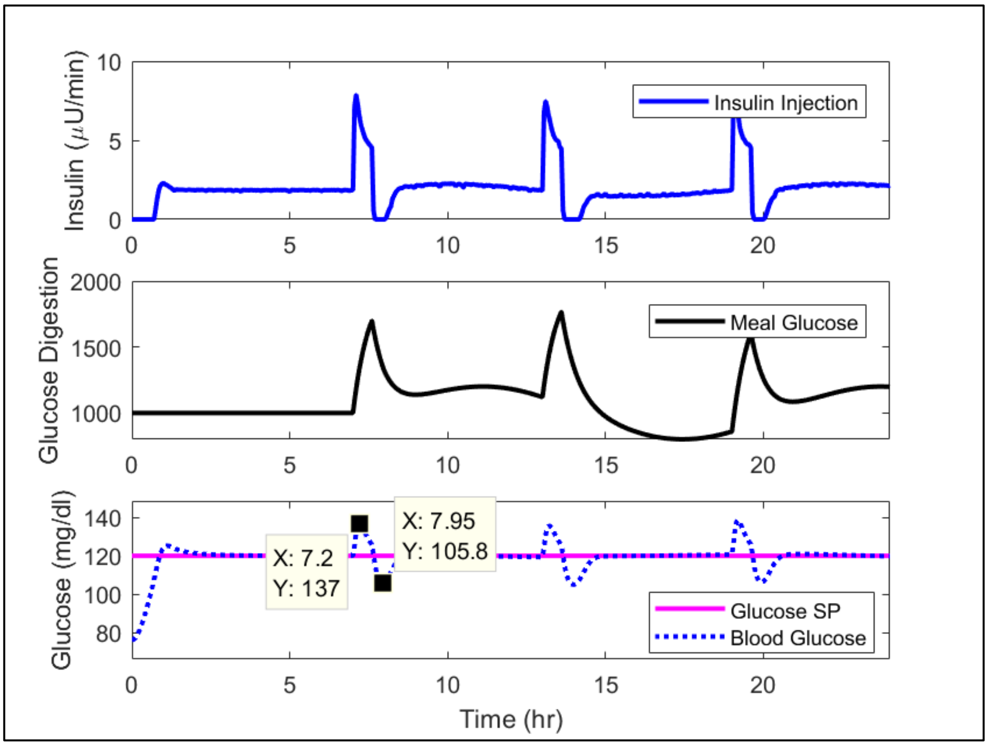


Figura. 5: Respuesta del controlador

El controlador logra cumplir con la dosificación basal manteniéndose en el setpoint de 120 [mg/dl] durante las 24 horas. Por otro lado, cuando existe una ingesta de alimentos el controlador entra en la dosificación bolus. El controlador en esta dosificación logra establecer un rango de glucosa de 105-137 [mg/dl]. Con esta respuesta del controlador se evidencia un correcto funcionamiento del control de insulina en lazo cerrado.

A pesar de que el controlador presenta los resultados deseados para que el paciente no sufra de hipoglucemia y de hiperglucemia este tiene algunos problemas. La forma de ajustar los parámetros del controlador fue hecha de manera empírica y al realizar más pruebas con el controlador inevitablemente existe la posibilidad de hipoglucemia como se ha observado en el trabajo de Mahmud (2017).

Por otro lado, se observa que el controlador propuesto esta ajustado a los parámetros de un paciente y por lo tanto la versatilidad del controlador para otros pacientes se la tiene que realizar de manera manual. La forma en la que se probó el controlador propuesto fue a partir de un paciente virtual el cual fue creado con el modelo mínimo de Bergman extendido. A pesar de que este modelo engloba muchas variables de la relación de glucosa-insulina no es lo suficientemente robusto para utilizarlo en pruebas clínicas.

Otros controladores PID se han desarrollado para el control en lazo cerrado de un sistema de glucosa-insulina como el controlador propuesto por Jensen (2007). El controlador PID de Jensen (2007) utiliza un modelo de Bergman extendido que presenta mejores resultados que el controlador propuesto. Esto se debe a que el modelo de Bergman extendido de Jensen es uno que modela mejor a la relación glucosa-insulina del cuerpo humano. Sin embargo, la implementación de este es más complicada si se quiere simular su comportamiento en Simulink.

4. SISTEMA DE DOSIFICACIÓN DE INSULINA

Al realizar el diseño de un sistema de dosificación de insulina se reducen la cantidad de componentes si se los compara con un sistema de dosificación de insulina con control en lazo cerrado. En el caso del sistema de dosificación de insulina propuesto, como se observa en la figura 6, no se utiliza un DMCG. El DMCG en el control de lazo cerrado era el encargado de proveer la señal de retroalimentación necesaria para ajustar el controlador. En este caso este elemento no es necesario ya que el usuario al trabajar con un sistema de dosificación de insulina utiliza un glucómetro y a partir de la lectura de este ajusta la insulina a suministrarse.

En el siguiente diseño del sistema de dosificación de insulina se presenta el diseño del sistema y su código funcional en el ambiente de desarrollo ECLIPSE. En este código se encuentra todo el funcionamiento del suministro de insulina para su funcionamiento con la aplicación de cálculo y suministro de dosificaciones y su funcionamiento con su interfaz gráfica. La interfaz gráfica creada se la despliega en una pantalla OLED la cual es accesible para el usuario para configurar y suministrar insulina.

En las secciones subsecuentes se explica con detalle el código implementado al igual que su funcionamiento con la aplicación para calculo y suministro de dosificaciones. De igual manera, se presenta en las siguientes secciones todas las opciones de sistemas para el dispositivo y las opciones de actuadores los cuales suministran la insulina. Finalmente, se presenta el diseño electrónico necesario para la construcción de la placa del sistema. Además, se presenta un diseño mecánico de todo el dispositivo el cual incluye el sistema embebido elegido, el actuador elegido, la pantalla, la batería, los botones físicos, el módulo de reloj, el driver del actuador y el vial de insulina.

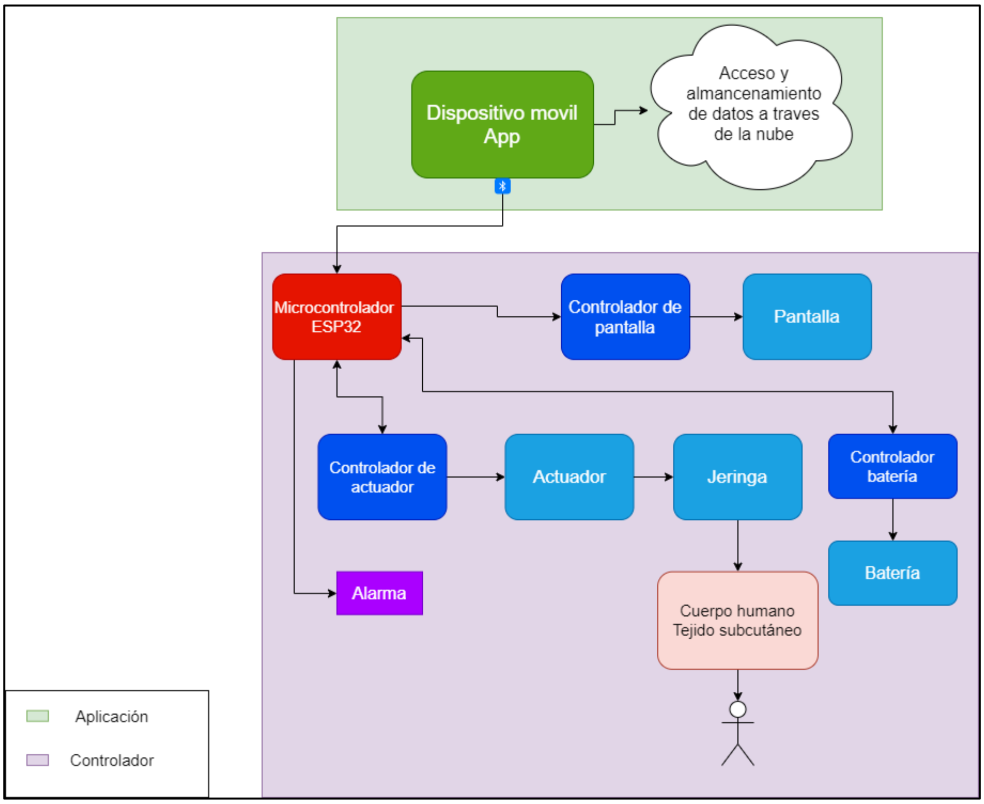


Figura. 6: Diagrama de bloques control en lazo abierto

4.1. Diseño sistema de dosificación de insulina

Para el diseño del sistema de dosificación de insulina se utiliza el sistema embebido ESP32. La elección de este sistema embebido se lo discute en secciones subsecuentes. Se crea un código del sistema para desplegarlo en una interfaz gráfica en una pantalla OLED. A partir de esta interfaz gráfica el usuario puede acceder a todas las funciones del sistema. La interfaz gráfica despliega opciones para configurar y dispensar las dosificaciones basal y bolus. El sistema integra la interfaz gráfica con las acciones del motor paso a paso para suministrar la cantidad de insulina requerida.

El código realizado para manejar todo el sistema de dosificación de insulina se lo puede explicar a través de distintas máquinas de estados. Este tiene muchas funcionalidades dependiendo en qué estado este y dependiendo de las acciones del usuario. La forma más fácil y accesible de entender todo el diseño realizado es analizando las máquinas de estados construidas. Antes de explicar las máquinas de estados se tienen que entender ciertas funcionalidades básicas del código.

En primer lugar, se definen todos los pines necesarios para las conexiones de todos los elementos del dispositivo. Se definen todas las librerías necesarias: como la librería del circuito integrado que se usa para controlar el motor, la librería para la pantalla, la librería bluetooth para realizar la conexión de la aplicación, una librería para los botones utilizados y una librería del reloj.

Por otro lado, el programa también tiene implementados módulos de reloj los cuales siempre establecen un tiempo real a pesar de un corte de energía. La memoria EEPROM del dispositivo funciona para almacenar datos los cuales no se pierden si existe un corte de energía. El programa tiene implementado timers de wakeup los cuales se los utiliza para que el dispositivo entre en una función de ultra ahorro de energía. Finalmente, el programa lee el nivel de carga de la batería y despliega su valor en la pantalla inicial.

Lo que el código realiza es crear una interfaz gráfica la cual puede ser accedida por el usuario. A partir de las modificaciones del usuario, el motor por pasos realiza distintas acciones que corresponden a las dosificaciones de insulina requeridas. Se crea una máquina de estados de todos los menús de la interfaz gráfica y de las acciones de estos. Esta máquina de estados transiciona a partir de 4 inputs, los cuales son los 4 botones que el dispositivo posee.

Estos 4 inputs corresponden a 4 acciones básicas. En el código se definió al input A como la acción de bajar, al input B como la acción de subir, al input * como la acción de aceptar y al input # como la acción de declinar o regresar. A partir de estos 4 inputs el usuario puede navegar por toda la interfaz gráfica modificando todas las opciones disponibles. La máquina de estados tiene un quinto input que se accede a partir de un botón para despertar al dispositivo el cual será explicado en los siguientes parrafos. De igual manera se puede observar la forma en cómo opera este botón si es activado, en el diagrama de proceso observado en la figura 9.

Los estados de la maquina se definen a partir del nombre PIC y un número de identificación. Estos estados corresponden a todas las opciones disponibles de la interfaz gráfica que son opciones elegibles por él usuario. A partir de los 4 inputs disponibles, cada PIC transiciona a otro estado dependiendo del input ingresado. En la figura 10 se observa una máquina de estados simplificada que describe todo el proceso de los PIC usados en el código para crear la interfaz gráfica. Como se observa existe una cantidad elevada de PIC por lo que se describirá a la mayoría de PIC de forma textual mas no numérica en los siguientes párrafos.

Es importante recalcar que si se utiliza los inputs A y B los PIC transicionan de manera secuencial es decir si estamos en el PIC 1 y se tiene un input B, input con acción de bajar, se transicionara al PIC 2. Como existen menús principales y submenús al presionar los inputs A y B se transicionara entre las opciones de los menús. Por otro lado, los inputs * y # tienen la utilidad de entrar y salir en menús y submenús. Si se presiona el input *, acción de entrar, lo que el código realiza es la operación: $\boldsymbol{pic = pic * 10 + 1}$ en la mayoría de los casos. Si se presiona el input #, lo que el código realiza es regresar al PIC anterior. Por ejemplo, si estamos en el PIC 1 y se tiene un input * lo que el código realiza es transicionar al PIC 11 siguiendo la operación presentada anteriormente. De esta manera, el código puede transicionar entre todas las opciones de la interfaz gráfica asegurándonos de estar en distintos estados siguiendo las acciones de los inputs.

Para entender de mejor manera las transiciones y sus estados respectivos se simulo como se presenta la interfaz gráfica en la pantalla OLED. En las siguientes figuras se observan todos los menús y submenús que la interfaz gráfica contiene. En resumen, se tiene una primera pantalla de portada y una pantalla de inicio que indica 4 opciones principales como se observa en la figura 7 y 8 respectivamente. En la figura 8 se observa las 4 opciones principales las cuales al ingresar a ellas se tienen submenús para su configuración respectiva. Como se observa estas opciones son: Basal, Bolus, Emergencia y Deepsleep.

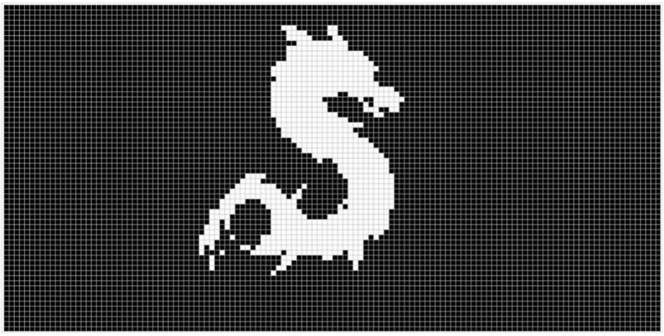


Figura. 7: Pantalla de portada

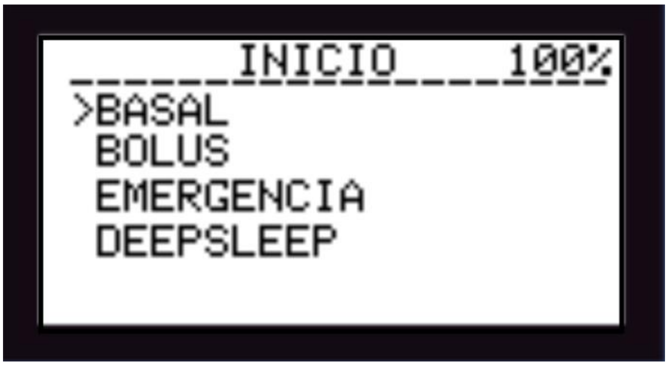


Figura. 8: Pantalla de inicio

Como se explicó anteriormente cada una de las opciones de la pantalla de inicio corresponde a un PIC. En este caso tenemos los PIC 1 - PIC 4 los cuales son accesibles por el usuario a través de los inputs A y B. Si presionamos el input * en cualquier de los PIC entonces se realiza la operación previamente explicada y el código transiciona a un PIC nuevo.

Si el usuario presiona el input * en el PIC 3 que corresponde al PIC de emergencia entonces se transiciona al PIC 31 que se muestra en la figura 11. En este PIC el usuario puede observar datos los cuales pueden ser de ayuda en una emergencia que son configurables desde el código. Por otro lado, si se transiciona al PIC de Deepsleep se muestra la pantalla de la figura 12. Si se activa el modo Deepsleep el sistema embebido ESP32 entra un modo el cual deshabilita varias opciones para un ahorro de batería. El programa a través de un timer vuelve a activarse y una vez que cumple su función de suministro vuelve a su estado de Deepsleep. Todo este proceso del código se lo puede observar en la figura 9.

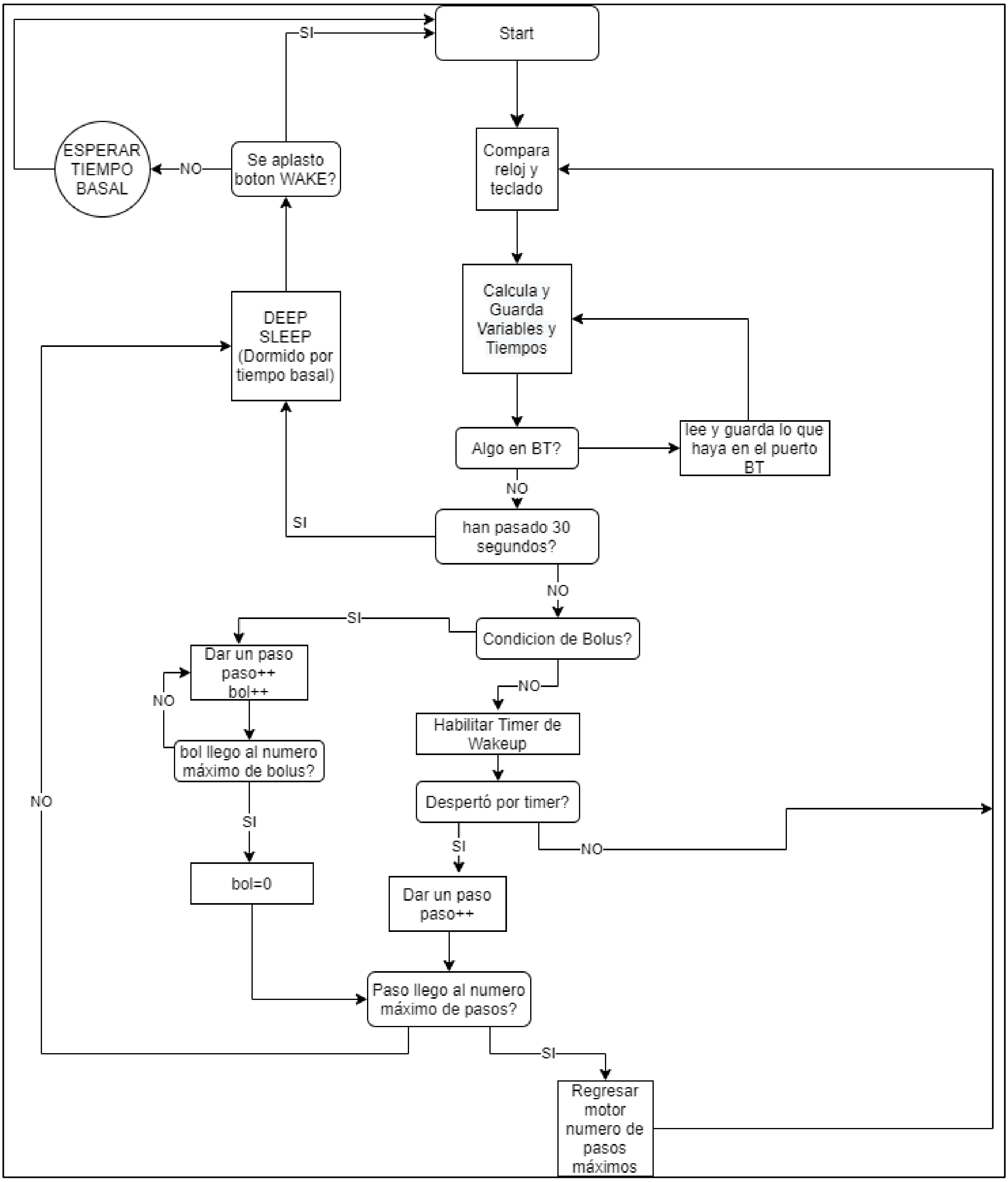


Figura. 9: Diagrama de proceso del código

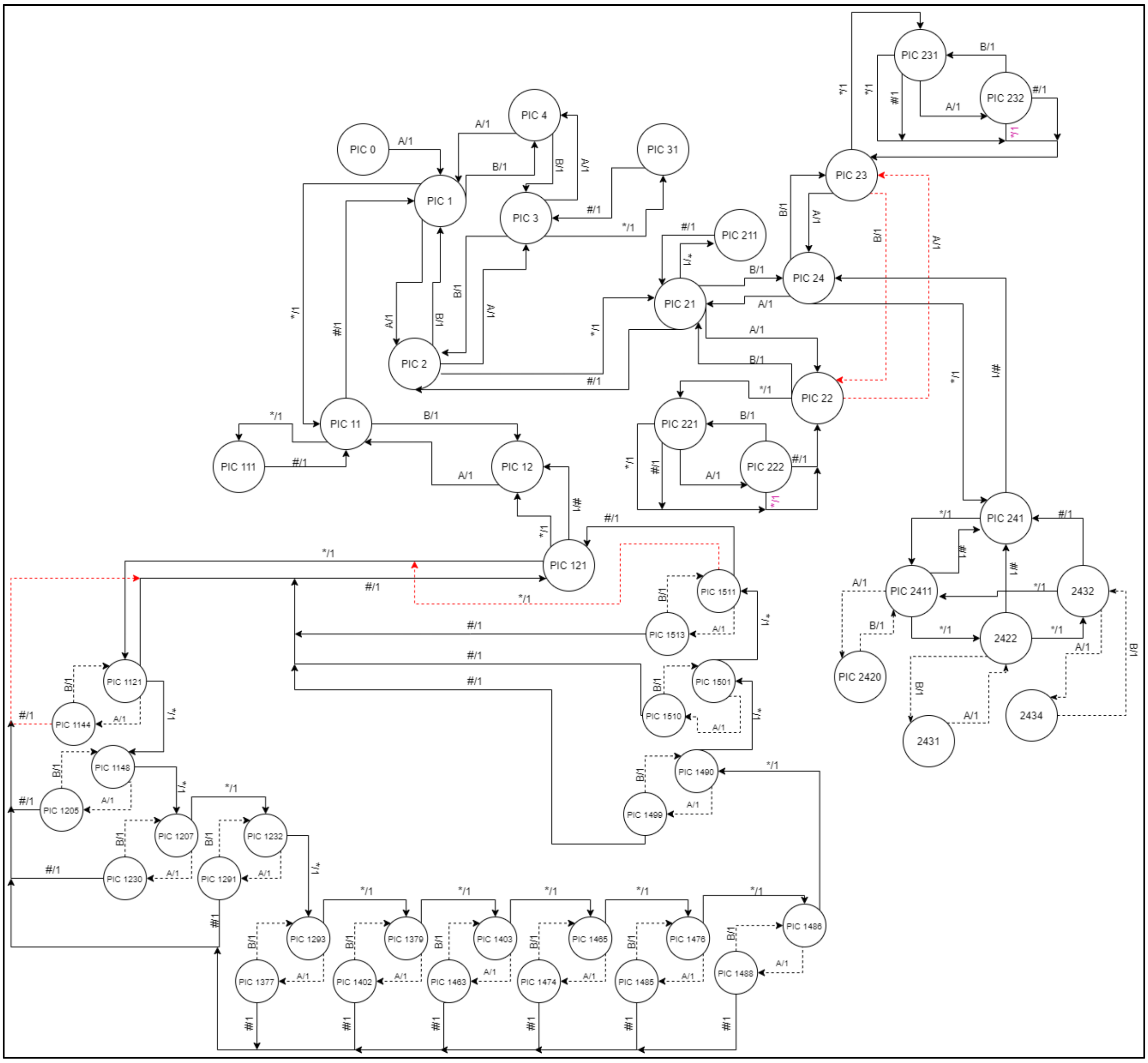


Figura. 10: Máquina de estados interfaz gráfica

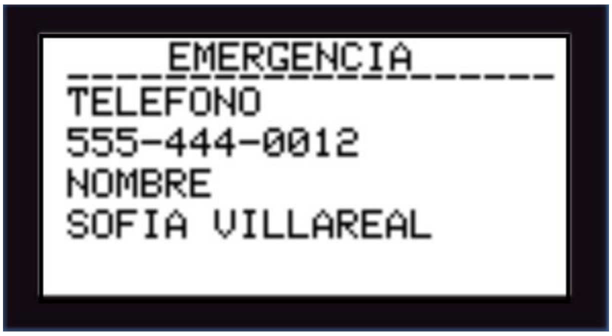


Figura. 11: Pantalla de emergencia

Figura. 12: Pantalla de Deepsleep

Una vez entendidos estos estados podemos pasar a los estados fundamentales del sistema de dosificación de insulina. Estos estados son los estados de suministro basal y bolus. En primer lugar, si transicionamos del estado basal a su submenú nos encontramos en el PIC 11 el cual se muestra en la figura 13.

A partir de este estado tenemos dos opciones una opción de configuración y una opción de información que se observa en la figura 14 y 15 respectivamente. Como se observa la pantalla de información muestra lo mismo que la pantalla de configuración. Una vez que el usuario a establecido las opciones en la pantalla de configuración estos datos se copian en la pantalla de información. La pantalla de información funciona para que el usuario revise la configuración realizada previamente. En esta pantalla el usuario no puede modificar ningún valor simplemente es una pantalla para comprobar los valores configurados.

Figura. 13: Pantalla de inicio basal

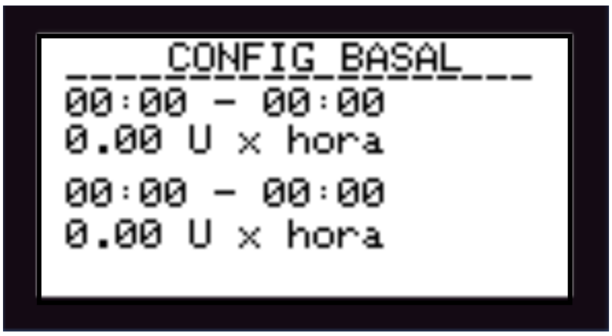


Figura. 14: Pantalla de configuración basal

Figura. 15: Pantalla de información basal

Para configurar los dos horarios y la cantidad de insulina a dispensar de la figura 14 el usuario transiciona entre varios PIC. El usuario al presionar el input *, cuando está en la pantalla de configuración, pasa al PIC que configura las horas de la primera hora del primer horario. Aquí el usuario utilizando los inputs A y B transiciona entre 24 PIC los cuales corresponde a las 24 horas del día.

Una vez que el usuario configuro la primera hora este presiona el input * y el PIC transiciona al PIC siguiente donde se configuran los minutos de la primera hora del primer horario. De igual manera el usuario usando los inputs A Y B transiciona entre 60 PIC los cuales corresponden a los 60 minutos en una hora seleccionando los minutos deseados. Este proceso se lo realiza de igual manera para las horas y los minutos de la segunda hora del primer horario.

Teniendo en cuenta que el rango del primer horario ha sido configurado el usuario al presionar * transiciona al segundo horario. Al configurar el rango de tiempo para el primer horario el código automáticamente configura las horas restantes del primer horario como el segundo horario. Solo se introduce manualmente para que el usuario pueda observar los horarios en la sección de información.

Ya que se configuraron ambos horarios se transiciona a la configuración de la cantidad de unidades que el usuario desea dispensar por hora. Al configurar la cantidad de unidades el usuario transiciona entre tres estados y configura la cantidad de unidades usando los inputs A y B que funcionan de la misma manera que al configurar las horas. En este caso las unidades tienen el rango de 0-9, las decimas rango de 0-9 y en las centésimas el usuario puede elegir entre 0 y 5. Se realizo esta forma de configuración para asegurar que las unidades dispensadas no sean difíciles de suministrar por el motor paso a paso.

El usuario una vez que configuro los dos horarios y las dos cantidades de insulina a suministrar este presiona el input # el cual tiene la acción de regresar y el sistema de dosificación empieza a dispensar la cantidad de insulina configurada.

La forma en la cual el sistema de dosificación de insulina da las señales de suministro es a través del paso mínimo del motor paso a paso. El motor paso a paso tiene un paso mínimo el cual corresponde a una cantidad mínima de insulina. Lo que realiza el sistema de dosificación es tomar los horarios configurados y a partir del horario configurado este sabe la cantidad de insulina a dispensar. El sistema de dosificación sabe la cantidad de pasos necesarios para cumplir con la dosis configurada en un rango de 1 hora a partir del paso mínimo del motor.

Por ejemplo, si se configuro 1 unidad por hora y el paso mínimo del motor es 0.166 unidades entonces el sistema de dosificación dispensa 0.166 unidades cada 10 minutos para completar en 6 pasos la dosis configurada. Por lo tanto, la forma en la cual se controla la cantidad de insulina suministrada es a partir de la variable que configura el tiempo necesario entre pasos del motor para suministrar la cantidad en una hora.

El estado basal al ser una dosificación que tiene que ser continuamente suministrada utiliza una gran cantidad de energía mientras el sistema embebido está prendido. Para mitigar el uso excesivo de batería se hace uso del modo Deepsleep acoplado con el módulo de reloj. Existe un timer de wake up el cual se configura a partir del horario basal del sistema embebido.

El sistema embebido siempre está en modo Deepsleep pero a partir del timer el dispositivo pasa a su modo normal, dispensa la cantidad de insulina necesaria y vuelve a sus estado Deepsleep. Por lo tanto, tomando en cuenta el horario del modo basal y el tiempo entre pasos del motor, el dispositivo se prende para dispensar lo requerido y vuelve al estado de Deepsleep. Este proceso se lo puede observar en la figura 9 previamente presentada.

La segunda dosificación del sistema es la dosificación bolus. El menú de la dosificación bolus se lo observa en la figura 16. A partir de este menú se puede acceder a 4 submenús: el de información, el de configuración, el de suministrar y el modo fiesta. En la figura 17 se observa el submenú de información el cual al igual que en el suministro basal simplemente se copia lo que el usuario configuro.

En la figura 18 se observa la configuración de la cantidad de unidades de insulina que se quiere suministrar. Al igual que en la dosificación basal el usuario transiciona entre tres PIC en los cuales configura la cantidad requerida haciendo uso de los inputs A y B. De igual manera las unidades se las puede configurar en un rango de 0-9, las décimas en un rango de 0-9 y las centésimas entre 0 o 5.

Figura. 16: Pantalla de inicio bolus

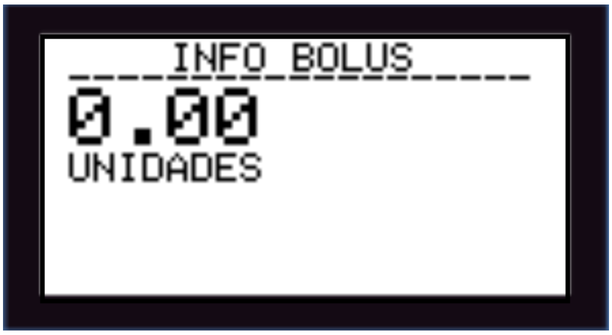


Figura. 17: Pantalla de información bolus

Para suministrar la cantidad que el usuario configuró este regresa al menú principal y entra al menú de suministrar como se observa en la figura 19. En este submenú el usuario elije si quiere o no suministrar. Si el usuario presiona "no" se regresa al menú principal. Si el usuario presiona "si" este PIC activa una función del motor por pasos que suministra la cantidad configurada.

Ya que la forma de suministrar insulina es a partir del paso mínimo del motor, el cual suministra una cantidad mínima de insulina. El sistema de dosificación en el suministro bolus toma este paso mínimo y lo multiplica hasta obtener la cantidad de unidades configuradas por el usuario. Por ejemplo, el usuario configuro 5 unidades y el paso mínimo del motor es 0.166 unidades, por lo tanto, el controlador realiza 30 pasos de motor para suministrar las 5 unidades.

Figura. 18: Pantalla de configuración

Figura. 19: Pantalla de suministro bolus

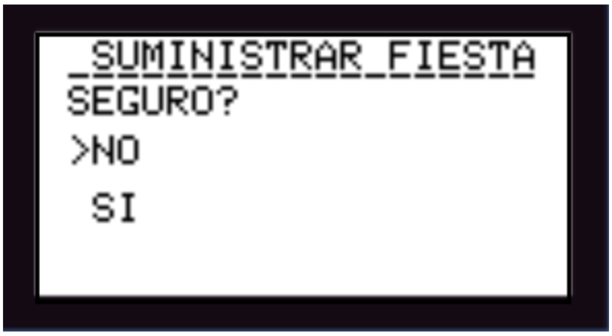


Figura. 20: Pantalla de suministro bolus fiesta

Finalmente, tenemos en el menú principal un PIC llamado fiesta. Este PIC es una opción extra que el usuario puede utilizar cuando consume pequeñas cantidades de alimentos de manera continua. El usuario tiene la opción de activar este modo como se observa en la figura 20. Si el usuario elije activar el modo fiesta el sistema de dosificación de insulina agrega un paso más cada cierta cantidad de pasos del suministro basal. Sin embargo, este modo no se habilito ya que no se encontró bibliografía del modo de administración. Cuando se realizó la prueba de la aplicación se habilito este modo simplemente para observar su comportamiento. A pesar de esto se optó por no habilitar este tipo de suministro para la protección del usuario.

4.2. Aplicación de cálculo y de suministro de dosificaciones

La aplicación para calculo y suministro de dosificaciones se la puede descomponer en tres elementos: su interfaz gráfica, su programación interna en diagrama de bloques y la programación que conecta la aplicación con el sistema de dosificación. Se utiliza el ambiente de programación MIT APP Inventor para la construcción de la interfaz gráfica y para las funciones internas de la aplicación a través de su programación por bloques. El código del sistema de dosificación de insulina previamente presentado tiene varias funciones las cuales son usadas por la aplicación de dosificaciones.

La aplicación construida esta optimizada para una tablet de tamaño pequeño sin embargo funciona para cualquier dispositivo móvil. El nombre de la aplicación es Insulinix y esta está disponible en la galería de la página de MIT APP Inventor. Posteriormente, la aplicación puede ser exportada a la tienda de Google Playstore. En las siguientes secciones se explica el funcionamiento de la aplicación y se presenta un ejemplo de uso con todas sus funciones.

La aplicación construida presenta 5 pantallas principales que incluyen: una pantalla de inicio, una pantalla de suministro basal, una pantalla de suministro bolus, una calculadora de carbohidratos e insulina y una página de indicaciones generales. En la figura 21 se observa la pantalla de inicio que incluye la descripción de todas las funcionalidades de la aplicación. Esta pantalla es la pantalla que se despliega cuando se inicializa la aplicación. A partir de esta pantalla se puede acceder a las otras 4 pantallas restantes. En esta primera pantalla la programación lo único que realiza es desplazarse a través de las 4 pantallas con los botones que se encuentran en la parte inferior.

Figura. 21: Pantalla de inicio

Una vez que el usuario reviso la pantalla de inicio este puede elegir cualquiera de las pantallas restantes. A pesar de esto, se recomienda que el usuario primero acceda a la pantalla de configuración ya que esta proporciona información importante para el manejo de la aplicación. En la figura 22 se observa la pantalla de configuración con toda la información relevante para el usuario. Como se indica esta aplicación es una segunda opción de suministro de insulina para la bomba en lazo abierto. Sin embargo, el usuario puede utilizar la aplicación para suministrar insulina como una primera opción si lo desea. Se realizo esta aclaración ya que la bomba tiene su interfaz gráfica la cual es de acceso más rápido y conveniente que la aplicación desarrollada.

Por otro lado, se indica que el usuario tiene que establecer una conexión bluetooth siempre que quiera dispensar cualquiera de las dosificaciones. Se diseño la aplicación de esta manera para la protección del usuario ya que este puede suministrar insulina por error. Esta elección puede ser una no tan amigable con el usuario, pero es la elección que se realizó para la construcción de la aplicación.

La calculadora de la aplicación contiene una cantidad de alimentos y su respetiva información nutricional. Esta información nutricional está dada a partir de las regulaciones del departamento de agricultura de los Estados Unidos y la información presentada equivale a una porción estándar para una persona adulta. De igual manera el usuario puede ingresar la cantidad de carbohidratos requeridos y se calcula la insulina necesaria a partir del valor ingresado.

Finalmente se indica que la aplicación no posee una base de datos propia si no una que se accede de manera externa a través de dos botones en cada una de las dosificaciones. Las bases de datos de la aplicación incluyen un documento de Google sheets en donde se almacenan los datos de manera numérica y un reporte de Google datastudio donde se crean gráficos a partir de los datos numéricos.

Debido a la construcción del ambiente de MIT APP Inventor integrar estas bases datos a la aplicación se la podía realizar solo de forma limitada. Ya que, almacenar datos de todos los suministros y graficarlos suponía que la aplicación se exceda de la capacidad de almacenamiento dispuesta por el ambiente. Por esta, razón se optó por realizar una base de datos externa.

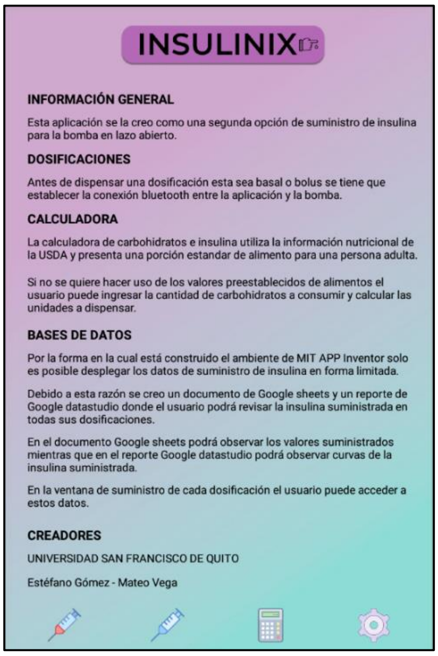


Figura. 22: Pantalla de configuración

Una vez que el usuario leyó todas las indicaciones generales este pasa a configurar y suministrar las dosificaciones requeridas. Como se explicó en la introducción cuando se trabaja con un sistema de dosificación de insulina este posee dos formas de dosificación. Una dosificación basal que hace referencia a la insulina de segundo plano y la dosificación bolus que es la insulina necesaria para cubrir los alimentos ingeridos. Los pacientes que utilizan dispositivos de suministro de insulina siempre tienen que configurar la insulina basal en una primera instancia ya que esta insulina es la necesaria para un funcionamiento normal.

En la figura 23 se observa la interfaz gráfica de la pantalla de suministro de insulina basal. Como se explicó anteriormente lo primero que el usuario tiene que realizar es establecer una conexión bluetooth con la bomba. Como se observa en la figura 23, abajo del icono bluetooth, la aplicación señala que esta no está conectada con el dispositivo. Si el usuario presiona el icono bluetooth la aplicación busca todos los dispositivos bluetooth cerca y se despliega la pantalla de la figura 24. En esta pantalla se observan todos los dispositivos disponibles y el usuario elige el dispositivo correspondiente.

Figura. 23: Pantalla suministro basal

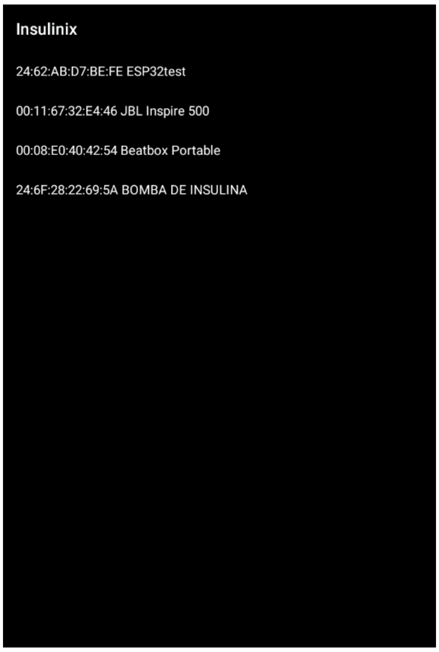


Figura. 24: Pantalla dispositivos bluetooth

Si se establece la conexión como se observa en la figura 25 la pantalla muestra que el dispositivo está conectado. Por otro lado, si se elige algún dispositivo y no se establece la conexión la aplicación muestra un mensaje de error donde se indica que no se puede conectar y pregunta si el dispositivo este prendido, esto se lo observa en la figura 26. Una vez que se estableció la conexión de la aplicación y el sistema de suministro de insulina el usuario puede configurar los rangos de tiempo y la cantidad de unidades que se desea dispensar.

Generalmente cuando se trabaja con una dosificación basal en un sistema de suministro de insulina se establecen dos horarios de trabajo (Dinsmoor, 2006). El primer horario de trabajo corresponde al horario donde la persona es más activa. Por otro lado, el segundo horario de trabajo corresponde al horario menos activo de la persona, generalmente durante todo el ciclo de sueño. Cuando el usuario ingresa las unidades de insulina a suministrarse estas hacen referencia a las unidades por hora que se dispensan.

Figura. 25: Pantalla dispositivo conectado

Si el usuario ingresa la cantidad de unidades que se desean dispensar del primer horario y este presiona el botón de configurar la aplicación realiza varias acciones. En primer lugar, la aplicación toma el valor ingresado en la caja de texto y lo asigna a una variable y se guarda su valor. Por otro lado, la aplicación envía esta misma variable al sistema de suministro para configurar la cantidad de suministro del primer horario de la dosificación basal. Esta variable es única y por lo tanto siempre se asigna a la cantidad de unidades del primer horario de la dosificación basal.

Por otro lado, lo mismo sucede cuando se ingresa la cantidad de unidades del segundo horario. Si se presiona configurar en el segundo horario se guarda el valor en una variable y también se envía la variable al sistema de suministro. Todas las acciones realizadas anteriormente solo se pueden ejecutar si la aplicación tiene establecida la conexión con la bomba. Si no existe tal conexión el programa devuelve el mismo mensaje de la figura 26.

Figura. 26: Pantalla error de conexión

El usuario al ingresar la cantidad de unidades deseadas puede hacerlo en el rango de 0.00-9.95. Este puede ingresar cualquier número del 0-9 en las unidades y las decimas y en las centésimas el numero 0 o el 5. Esta configuración se la realiza para todas las dosificaciones de la aplicación.

Luego de realizar la configuración de la cantidad de unidades del primer horario y segundo horario se configura los horarios de suministro. Se ingresan las horas deseadas por el usuario y una vez que se presiona configurar la aplicación envía estas 4 variables al sistema de suministro. Estas variables se asignan como hora 1, minuto 1, hora 2 y minuto 2. Siendo la hora 1 y minuto 1 correspondiente a la primera hora y al primer minuto del primer horario y la hora 2 y minuto 2 correspondiente a la segunda hora y al segundo minuto del segundo horario.

Si el usuario ingresa 8, 50, 21,50 se asigna el primer horario entre las 8:50-21:50. Cómo la bomba dispone solo de dos horarios para configurar el suministro de la dosificación basal. Una vez que se ingresa el primer horario automáticamente el código del sistema de suministro asigna el segundo horario a las horas restantes del día. En este caso el segundo horario corresponde a las horas de 21:50-8:50.

Ya que se configuro el horario y la cantidad de suministro del segundo horario el dispositivo empieza a suministrar. Cuando el dispositivo empieza esta acción, el sistema de suministro envía el valor de la cantidad de insulina suministrada a la aplicación. La cantidad de insulina suministrada depende de la precisión del motor paso a paso y por esta razón puede tener errores y suministrar una cantidad de insulina imprecisa.

Este valor de insulina es leído por la aplicación y comparada con la variable de la cantidad de insulina configurada por el usuario. Si el valor comparado está por arriba o por debajo del 15% de la cantidad configurada por el usuario este valor es enviado a la base de datos. Esto en un principio garantiza que los datos guardados en la base de datos correspondan con la insulina de suministro de manera general. Si los datos que se envían del sistema de suministro superan el 15% de error por un tiempo prolongado la aplicación devuelve un error. Donde se indica que la bomba está dispensando la cantidad de insulina incorrecta y se tiene que reconfigurar el sistema de suministro.

Finalmente, si se presiona el botón de resetear sistema se asigna un cero a todas las variables configuradas anteriormente. Sin embargo, el botón de resetear solo funciona para reestablecer todas las variables configuradas en la aplicación a un valor de 0 y no en el sistema de suministro. Esto significa que todo lo configurado anteriormente vuelve a un valor de cero y por lo tanto los valores enviados a la base de datos dejan de guardarse.

Los valores previamente configurados en el sistema de suministro siguen siendo los mismos hasta que se los configure con nuevos valores. El botón de resetear resetea solo la aplicación ya que es peligroso para el usuario parar cualquier tipo de suministro previamente configurado. Los últimos botones de esta pantalla corresponden a la base de datos que el usuario puede acceder si pulsa cualquier de las dos opciones. La primera base de datos guarda todos los valores suministrados en un documento Google sheets mientras que la segunda base de datos toma estos valores y los presenta en modo de gráficos. En el ejemplo de uso de la aplicación se aprecia todo lo descrito en el uso de la pantalla del suministro de insulina basal.

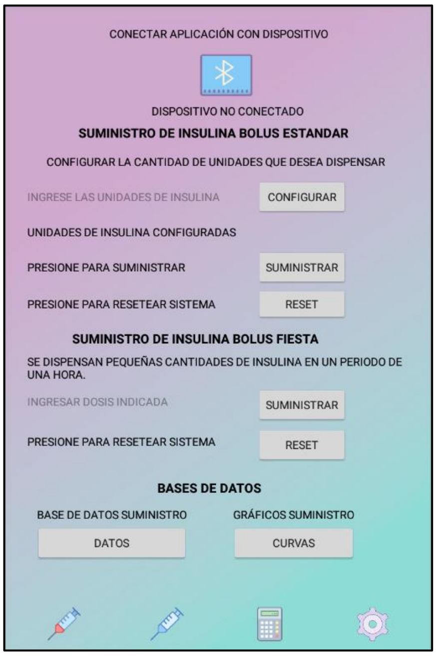


Figura. 27: Pantalla suministro bolus

Cuando se accede a la pantalla de suministro bolus como en la figura 27 se observa una interfaz gráfica similar a la pantalla de suministro basal. De igual manera que en el suministro basal se tiene que establecer una conexión bluetooth antes de configurar y suministrar la insulina como se observa en la figura 28. Una vez que se estableció la conexión se puede hacer uso de las funcionalidades de esta pantalla.

Si se quiere dispensar una cantidad de insulina bolus, se ingresa la cantidad de unidades en el rango 0.00-9.95 y se presiona configurar. Cuando se presiona configurar la variable sigue el mismo proceso que en el suministro basal y se despliega este valor en la sección de unidades de insulina configuradas de la aplicación. Una vez configurada la cantidad de unidades se puede presionar el botón de suministrar el cual activa la opción “si” en la función suministrar del sistema de suministro. El sistema de suministro antes de dispensar una dosificación bolus pregunta al usuario si en realidad quiere dispensar. Por lo tanto, la aplicación al presionar el botón suministrar establece que “si” se desea suministrar la dosificación indicada.

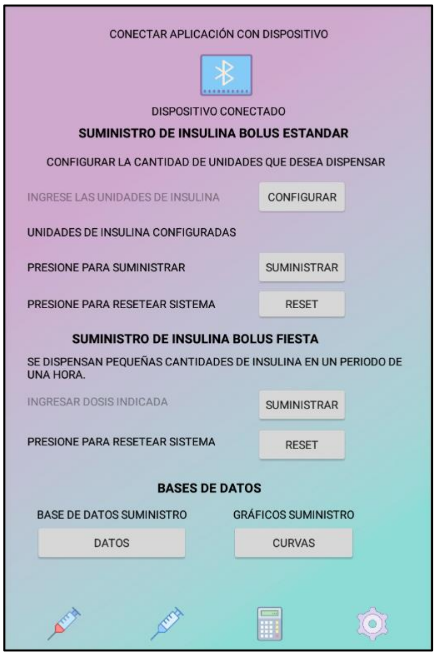


Figura. 28: Pantalla dispositivo conectado

Como se observa en la figura 28 esta pantalla cuenta con una opción de suministro de bolus fiesta. El suministro de bolus fiesta cómo su nombre lo indica es una dosificación que se utiliza generalmente cuando el usuario consume pequeñas cantidades de alimentos en un rango de tiempo continuo, como en una fiesta. Esta opción no se la habilito en el código final ya que no se encontró con la bibliografía necesaria para justificar la forma de suministro. Sin embargo, se realizó una prueba de funcionamiento simplemente para observar el comportamiento de esta opción en el sistema de suministro.

Todos los suministros de insulina del sistema funcionan bajo una ecuacion donde se hace referencia a la precisión y al paso mínimo del motor paso a paso. A partir de estas condiciones el sistema de suministro manda una señal al motor paso a paso para suministrar insulina. Por lo tanto, en la dosificación basal existe una cantidad x de pasos para completar las unidades que se tiene que dispensar, la dosificación bolus funciona de la misma manera.

Cuando se activa la opción de bolus fiesta lo que el sistema de suministro realiza es agregar pasos extra a los pasos realizados por el suministro basal, por lo tanto, la cantidad de insulina suministrada es un poco mayor a la cantidad de insulina basal estándar. El sistema de suministro toma la cantidad de insulina suministrada en el horario que se activó el modo bolus fiesta y agrega pasos a esta función.

Por ejemplo, si el usuario configuro su primer horario basal de 8:00-20:00 con una cantidad de 1 unidad por hora. Y el usuario activa el modo fiesta a las 19:00 entonces el controlador toma esta 1 unidad de referencia y aumenta pasos a esta cantidad. El modo fiesta tiene una duración de 1 hora por lo tanto el sistema de suministro dispensa 1 unidad más la cantidad extra de insulina y la reparte en toda la hora de suministro equitativamente.

Toda la explicación del suministro de todas las configuraciones se expandirá cuando se explique el funcionamiento del sistema de suministro. Finalmente, el botón de reset de la pantalla y las bases de datos funcionan de la misma manera que en la pantalla basal y bolus.

Por último, observamos la pantalla de la calculadora de carbohidratos e insulina en la figura 29. En un principio el usuario tiene que configurar su relación de insulina a carbohidratos. Esta relación hace referencia a cuanta insulina se necesita para contrarrestar una cantidad preestablecida de carbohidratos. Generalmente esta relación viene dada a partir de 1 unidad y cuantos carbohidratos pueden ser contrarrestados con esta unidad (The University of Iowa, 2020). Esta relación es una que el medico determina a partir de todos los datos fisiológicos del paciente y su respuesta a la insulina.

Una vez que el usuario ingresa su relación insulina a carbohidratos y presiona el botón de ingresar se guarda el valor ingresado en una variable. El usuario puede pasar a elegir los alimentos de las listas predeterminadas presionando en los iconos de alimento. Cada vez que el usuario presiona un alimento se crea una variable donde se guarda el valor de carbohidratos de los alimentos. Si el usuario ya eligió todos los alimentos deseados este presiona el botón de calcular y la aplicación suma todas las variables de los alimentos para obtener la cantidad de carbohidratos que se van a consumir.

Finalmente, el usuario tiene que presionar el botón de calcular, debajo de la caja de texto de unidades de insulina bolus, y la aplicación calcula las unidades de insulina necesarias para los alimentos ingresados. La aplicación devuelve la cantidad de unidades redondeadas a dos decimales para que se ajuste a los rangos de unidades que el usuario puede ingresar en las pantallas de cada suministro.

Esta calculadora también puede ser usada de forma manual. Si el usuario desea una cantidad cualquiera de carbohidratos diferentes a los que se presentan en la aplicación. Este puede ingresar en la caja de texto de carbohidratos totales la cantidad de carbohidratos a consumir. Si presiona calcular la aplicación calculara la cantidad de unidades de insulina a dispensar a partir del valor ingresado por el usuario.

Figura. 29: Pantalla calculadora de carbohidratos e insulina

4.3. Prueba sistema de dosificación de insulina

Una vez que se explicó la construcción y el funcionamiento de la aplicación para calculo y suministro de dosificaciones se observa un ejemplo de uso. Para este ejemplo se utilizaron los siguientes datos de un paciente artificial. Para la configuración basal el primer horario correspondía de 8:00-20:00 y por consiguiente el segundo horario tenía el rango de 20:00-8:00. Para el primer horario se programó la aplicación para que dispense 1 unidad cada hora y para el segundo horario 0.75 unidades por hora.

Por otro lado, el paciente ingería alimentos 5 veces al día. Un desayuno a las 9:30 utilizando 5 unidades de insulina, un snack a las 12:00 pm utilizando 3 unidades de insulina, un almuerzo a las 14:00 utilizando 5 unidades de insulina, un segundo snack a las 17:00 pm utilizando 3 unidades de insulina y una cena a las 20:30. Finalmente el paciente hizo uso del modo bolus fiesta activando su suministro a las 19:00.

Para este ejemplo se configuro todos los datos previamente mencionados y se utilizó el sistema de dosificación de insulina para que este devuelva valores de insulina suministrados. El sistema de dosificación de insulina en este ejemplo devuelve datos basándose en una precisión no ideal para el motor paso a paso. La forma en la que el sistema de dosificación de insulina suministra insulina depende de la precisión del motor y del paso mínimo de este.

Para analizar los datos dispensados por el sistema de dosificación de insulina se hace uso de la base de datos configurada y se observa sus valores. Como se mencionó anteriormente existen dos bases de datos una donde se observa los datos en una hoja de cálculo y una donde se observan gráficos realizados a partir de los datos de la hoja de cálculo. La hoja de cálculo no tiene mucha relevancia por lo tanto se omite presentarla.

En las figuras 30, 31, 32 y 33 se observan los 4 gráficos generados a partir de los datos de la hoja de cálculo. Estas graficas presentan toda la información acumulada dependiendo de la cantidad de días que se hizo uso de la aplicación y el sistema de dosificación. En este caso se simulo 24 horas en los rangos previamente mencionados. Por lo tanto, si el usuario hace uso de la aplicación para otro día estos gráficos se actualizarán con la nueva información proporcionada.

En el gráfico 30 se observa el suministro del primer horario de la dosificación basal. Se observa que esta empieza y termina en los rangos indicados. Por otro lado, si observamos la hoja de cálculo observamos 10 suministros de insulina por hora es decir el controlador opto por realizar 10 pasos de motor los cuales suministraban 0.1 unidades de insulina por paso. Ya que se necesita suministrar esta insulina en una hora cada paso de 0.1 unidades se lo realizaba cada 6 minutos para completar con la dosificación.

Figura. 30: Gráfico dosificación basal primer horario

Como se observa en el gráfico la insulina dispensada es variable es decir no siempre se suministró 0.1 unidades por paso. Esto se debe a los datos de la precisión del motor, que se ajustaron para que no siempre sean ideales. Finalmente, al analizar la gráfica se observa que el rango de insulina suministrada entra en los parámetros establecidos por la aplicación. Previamente se mencionó que los datos que se guardan en la base de datos son aquellos que máximo tienen un 15% de error. En este caso se observa exactamente que todos los datos cumplen con esta condición y por tal razón se grafican todos.

Figura. 31: Gráfico dosificación basal segundo horario

En la figura 31 se observa el segundo horario de la dosificación basal. En este horario la cantidad de insulina configurada es de 0.75 unidades de insulina. Revisando la hoja de cálculo observamos que el controlador dispenso 0.15 unidades cada 12 minutos para completar con la dosificación. Observamos que la insulina suministrada entra en los rangos del 15% de error y esta se dispensa en los rangos configurados.

Figura. 32: Gráfico dosificación bolus fiesta

Por otro lado, observamos en la figura 32 el gráfico de la dosificación bolus fiesta. En esta dosificación analizando la hoja de cálculo se observa que más o menos se dispenso 1.25 unidades de insulina. Como se empezó el modo bolus fiesta a las 19:00, este coincidía con el primer horario de suministro el cual dispensa 1 unidad por hora. El sistema de dosificación toma la cantidad de insulina dependiendo del horario y dispensa a partir de este valor como se observa en el gráfico. Como se mencionó anteriormente la opción del modo bolus fiesta no se implementó en el código final del dispositivo por falta de bibliografía. El ejemplo presentado es uno simplemente para observar el comportamiento del sistema de dosificación de insulina si se hubiese habilitado la opción de suministro bolus fiesta.

Figura. 33: Gráfico dosificación bolus estándar

Finalmente, tenemos la gráfica del modo bolus estándar. Como se observa en la figura 33 se dispenso insulina para 5 comidas distintas las cuales son las que se programó en la aplicación. Estos suministros cumplen con el 15% de error programados en la aplicación. Se observa también en la figura 33 que existe un rango de tiempo considerable entre la penúltima comida y la última comida. Si acercamos el cursor en cada suministro de insulina observamos que los suministros si se dispensaron en las horas programadas. El rango de tiempo que se observa en la gráfica se debe a que en ese rango el usuario activo el modo bolus fiesta y como las bases de datos bolus se agrupan en una misma hoja de cálculo la gráfica se observa de esa manera.

4.4. Selección placa de desarrollo

Para la construcción del dispositivo de dosificación de insulina se tomó en cuenta 4 placas de desarrollo las cuales se resumen la tabla 1. Esta tabla compara las características necesarias para la construcción del dispositivo. Las características que se comparan son el procesador, el costo, el consumo mínimo en Deepsleep, los inalámbricos integrados, la memoria RAM, los puertos I2C, el tamaño y el prototipaje de cada una de las placas de desarrollo. En las figuras 35, 36, 37 y 38 se observan las figuras de cada placa de desarrollo.

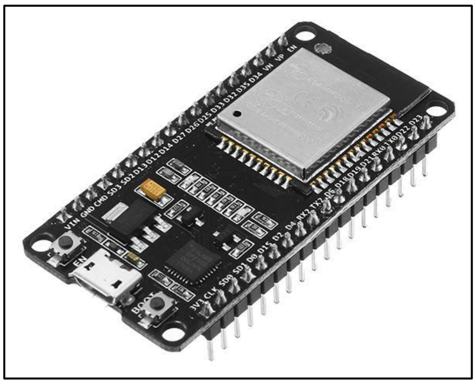


Figura. 34: Placa de desarrollo ESP32

Placas	Procesador	Costo*	Consumo Mínimo en Deepsleep	Inalámbricos Integrados	Memoria/RAM	Puertos I2C	Tamaño[mm]	Fácil Prototipaje
ESP32-S	Xtensa LX6	$7,50	4uA	Si	16Mb/520Kb	Si	54.4x27.9	Si
Arduino nano	ATMEGA328	$22,00	0,4mA	No	32Kb/1Kb	Si	43.18x17.78	Si
LaunchPad	MSP430	$23,40	1,4uA	No	512Kb/66Kb	Si	51x67	Si
MINI-32	PIC32MX	$29,00	47uA	No	64Kb/16Kb	Si	50.8x17.78	No
*Los costos mostrados en la tabla son precios establecidos por cada fabricante								

Tabla. 1: Comparación placas de desarrollo

Al comparar las 4 placas de desarrollo se eligió el sistema embebido ESP32. Se eligió este sistema ya que se adecua de mejor manera al proyecto y futuros avances de este. Este sistema tiene dimensiones adecuadas para un sistema de suministro de insulina. Sus dimensiones son de 54.4 x 27.9 mm las cuales son perfectas para el sistema diseñado tomando en cuenta todas las funcionalidades que el dispositivo posee. El sistema diseñado usa las funcionalidades de conexión bluetooth y ahorro de batería. A partir de estas funcionalidades observamos que la placa de desarrollo ESP32-S presenta las mejores características en estos ámbitos.

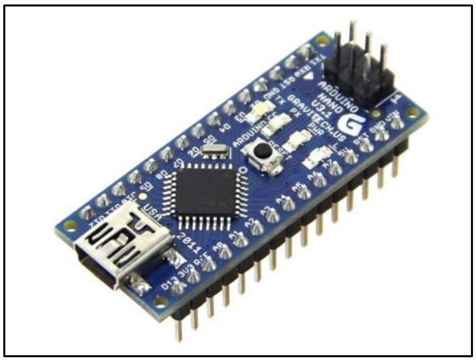


Figura. 35: Placa de desarrollo Arduino Nano

El modo Deepsleep al cual se hace referencia en la tabla 1 corresponde a un modo de bajo consumo de batería. Como se observa el sistema ESP32-S presenta el menor consumo con 4uA. Por otro lado, se hace referencia a los inalámbricos que poseen las placas de desarrollo. Esta funcionalidad de las placas hace referencia a las distintas características de módulos integrados en la placa. Algunas de estas características son los módulos bluetooth y WiFi. Estos dos módulos están integrados en el sistema ESP-32S. Ya que, usamos bluetooth para una conexión entre el sistema y la aplicación móvil la placa de desarrollo elegida es la ideal.

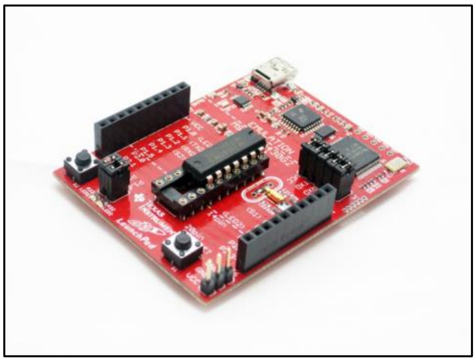


Figura. 36: Placa de desarrollo Launchpad MSP430

Finalmente, este sistema consume poca energía y su memoria es adecuada para guardar datos de display. El criterio de facilidad de prototipaje se eligió tomando en cuenta aspectos como facilidad de entorno de desarrollo y su portabilidad. Refiriéndose a portabilidad el dispositivo ESP32-S puede alimentarse a través de baterías simples y pequeñas. De igual manera, no se necesitan módulos extra para realizar una conexión bluetooth la cual es necesaria en el diseño presentado.

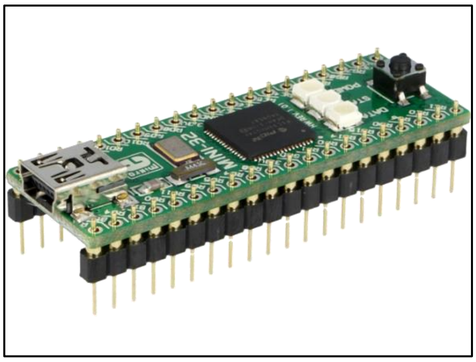


Figura. 37: Placa de desarrollo MINI-32

4.5. Selección actuador

A partir del diseño del sistema de dosificación de insulina y de la elección de la placa de desarrollo, para manejar todo el dispositivo es necesario elegir un actuador el cual dispense las dosificaciones de insulina. Existen varias formas de suministrar insulina en un dispositivo de dosificación. Se puede accionar insulina mediante una bomba la cual a través de pulsos suministra una cantidad especifica de insulina. Por otro lado, podemos usar un actuador lineal el cual empuje el reservorio de insulina y de esta manera a través del equipo de infusión se suministra insulina al paciente. De igual manera, esta forma de suministro se la puede lograr a través de un motor paso a paso y un mecanismo el cual a través del movimiento del motor empuje el reservorio de insulina. En la siguiente tabla se presenta una comparación entre estos tres tipos de dispositivos y se elige el más adecuado para el sistema de dosificación de insulina diseñado.

Actuador	Precio	Paso mínimo	Fuerza*	Adaptabilidad	Voltaje [V]	Potencia [W]	Tamaño[mm]
MasterFlex Micropump	$443	20uL	5psi	No	12*	N/A	3.65x6.35x2.54
M-232	$2.000	0.1um	40N	N/A*	12	1.78	80.5x49x20
AMD1020	$182,05	N/A*	0.2 mNm	Si	3	0.54	10x24 (cilíndrico)

Tabla. 2: Comparación actuadores

Como se indica en la tabla hay varias consideraciones que hay que tomar en cuenta. En primer lugar, los precios de todas los dispositivos son precios tomados a través de la información del fabricante. En la tabla se indica una característica de paso. Esta característica hace referencia al paso mínimo que tiene el actuador. Como se mencionó anteriormente existen varias formas de dispensar insulina a través de actuadores o de bombas. Las bombas por su mecanismo de acción pueden dispensar cualquier tipo de líquido en una precisión en unidades de volumen. En este caso la precisión de la bomba Masterflex es de 20 uL (Cole-Palmer, 2020).

Figura. 38: Bomba Masterflex Micro Pump

Por otro lado, si comparamos este paso mínimo con el paso mínimo del actuador lineal M-232, este tiene una precisión de 0.1 μm (Physik Instruments, 2020). Esta precisión viene dada a partir de una unidad de longitud por lo cual no se pueden comparar ambas medidas. Para realizar una comparación entre la precisión de cada dispositivo se establece cual es el volumen que puede dispensar el actuador lineal mediante el sistema de suministro previamente explicado. El actuador lineal empuja un reservorio de insulina el cual tiene un longitud de 25.4mm y un radio de 6mm. Entonces se observa cuanta insulina dispensara si el actuador empuja este reservorio a través de su paso mínimo de 0.1 μm.

Figura. 39: Actuador lineal M-232

A partir de estos datos se calcula el volumen que 0.1 μm dispensara al empujar el vial de insulina. En la figura 40 se observa el cálculo de volumen con estos datos. Como se observa el actuador lineal dispensar con su paso mínimo $11.30\, pm^3$. Si transformamos esta unidad a μL esto nos da como resultado $0.0113\, \mu L \approx 0.01 \mu L$. Con esto sabemos que el paso mínimo del actuador lineal tiene una precisión excelente tomando en cuenta que $1U = 10\mu L$ si utilizamos una insulina U-100 (Chait, 2014). Finalmente, si observamos el paso mínimo del motor paso a paso AM1020 observamos que esta no tiene especificaciones en su datasheet. Esto se debe a que la medida de paso mínimo la definimos a partir del movimiento del dispositivo. Ya que, el motor paso a paso tiene un movimiento rotativo no se establece un paso mínimo. El paso mínimo de este actuador dependerá del mecanismo de acople el cual se utiliza para dispensar insulina.

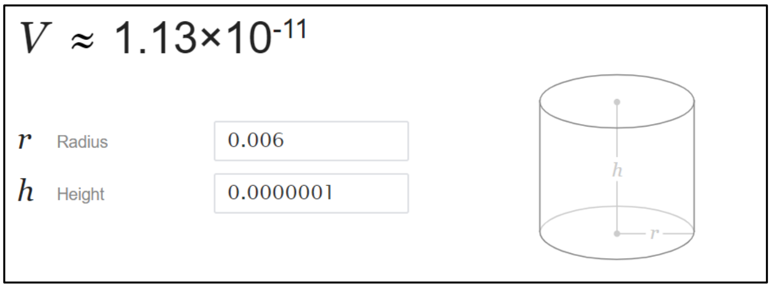


Figura. 40: Volumen mínimo alcanzado por el actuador

Como se estableció anteriormente todos los dispositivos propuestos presentan características distintas que son difíciles de comparar. Sin embargo, todas cumplen con la misma función de suministrar insulina a partir del sistema de dosificación diseñado. Si observamos la categoría de fuerza de la tabla 2, esta hace referencia a distintas características en cada dispositivo. Por ejemplo, la bomba en esta categoría tiene 5psi, los cuales corresponden a una unidad de presión. Por otro lado, el actuador lineal tiene 40N, los cuales en este caso si corresponden a una unidad de fuerza. Finalmente, el motor AM1020 tiene 0.2 mNm, los cuales corresponde a una unidad derivada de torque (Faulhaber, 2020). Todas estas características no son comparables como se indicó anteriormente, sin embargo, son las características con más similitud por eso se las agrupo en esta categoría.

Finalmente, tenemos la característica de adaptabilidad la cual se refiere a que tanto se puede modificar la forma de suministro. El actuador con la mayor modificabilidad es el motor paso a paso AM1020 ya que, este tiene opción de modificar el tornillo integrado al eje lo que reduciría el tamaño del dispositivo de suministro de manera considerable. El actuador M-232 tiene la opción de modificar la longitud de su pistón, pero este dispositivo es uno demasiado grande para ser adaptado a un sistema de suministro como el diseñado. Por último, la bomba Masterflex no posee ninguna característica de modificabilidad por la forma en la que esta funciona.

El actuador elegido fue el AM1020 en primer lugar porque es el más barato y adaptable a nuestro diseño del dispositivo de suministro de insulina. Este motor paso a paso se acopla de la mejor forma a los productos que ya existen en el mercado como el sistema embebido ESP32-S el cual se eligió como plataforma de desarrollo. Los otros dos dispositivos de suministro de insulina tienen características excelentes, sin embargo, son muy costosas y no tan amigables para el desarrollo del diseño presentado.

Figura. 41: Motor paso a paso AM1020

4.6. Diagramas electrónicos

A partir de la elección de la plataforma de desarrollo y del actuador de suministro se realiza el diseño del circuito necesario para llevar a cabo la construcción del dispositivo de suministro de insulina. En la figura 42 se observa el esquemático del circuito de todos los elementos necesarios para la construcción del sistema propuesto. En este esquemático se incluye la conexión del motor paso a paso, del módulo de reloj, de la pantalla oled, del driver del motor, de sistema embebido ESP32-S y de los botones que utiliza el usuario.

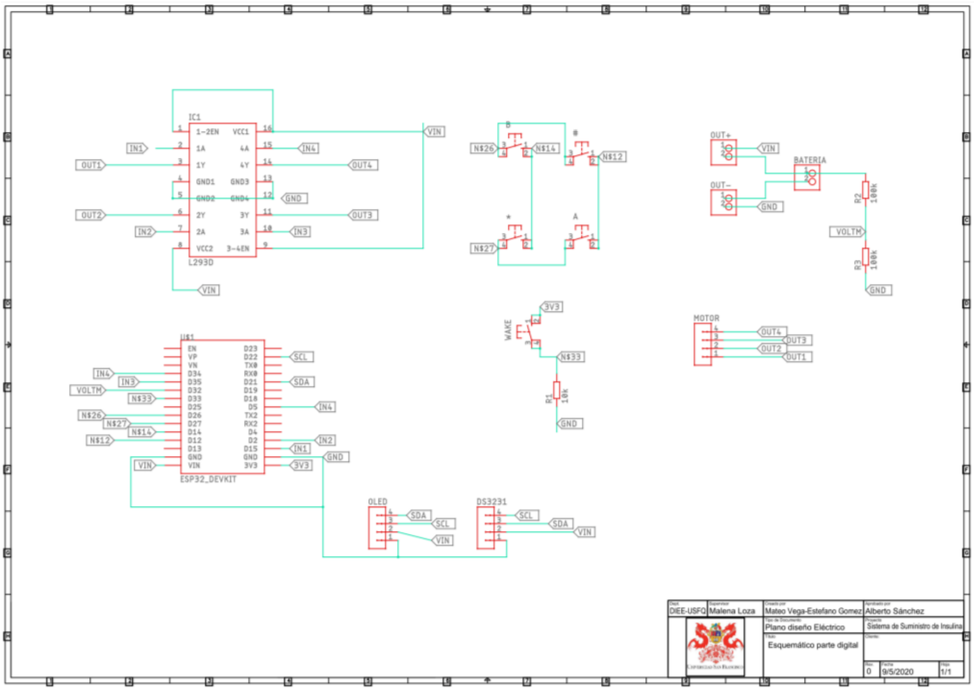


Figura. 42: Esquemático circuito principal del sistema de suministro

Por otro lado, en la figura 43 se observa el enrutamiento del circuito impreso. En este figura se observa como las conexiones previamente mencionadas se acoplan en un circuito real. De igual manera, se observa las dimensiones de esta circuito impreso las cuales son de gran ayuda para diseñar todo el diseño mecánico el cual se presenta en las siguiente secciones.

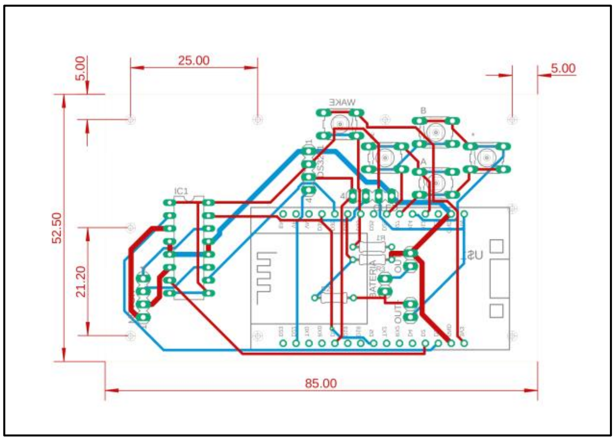


Figura. 43: Enrutamiento circuito impreso

4.7. Diagramas mecánicos

A partir del diseño del circuito impreso se diseñada todo el sistema mecánico del dispositivo de suministro de insulina. Este diseño mecánico engloba: una caja principal donde se almacenan todos los elementos necesarios para la construcción del dispositivo. La lista de todos los elementos necesarios para la construcción del diseño propuesto se los encuentra en el documento adjunto de requisición de materiales. Por otro lado, el diseño mecánico incluye el mecanismo el cual utiliza el motor paso a paso para suministrar insulina. También incluye la cubierta del circuito, la caja protectora del motor, el acople de eje del motor y el seguro del equipo de infusión.

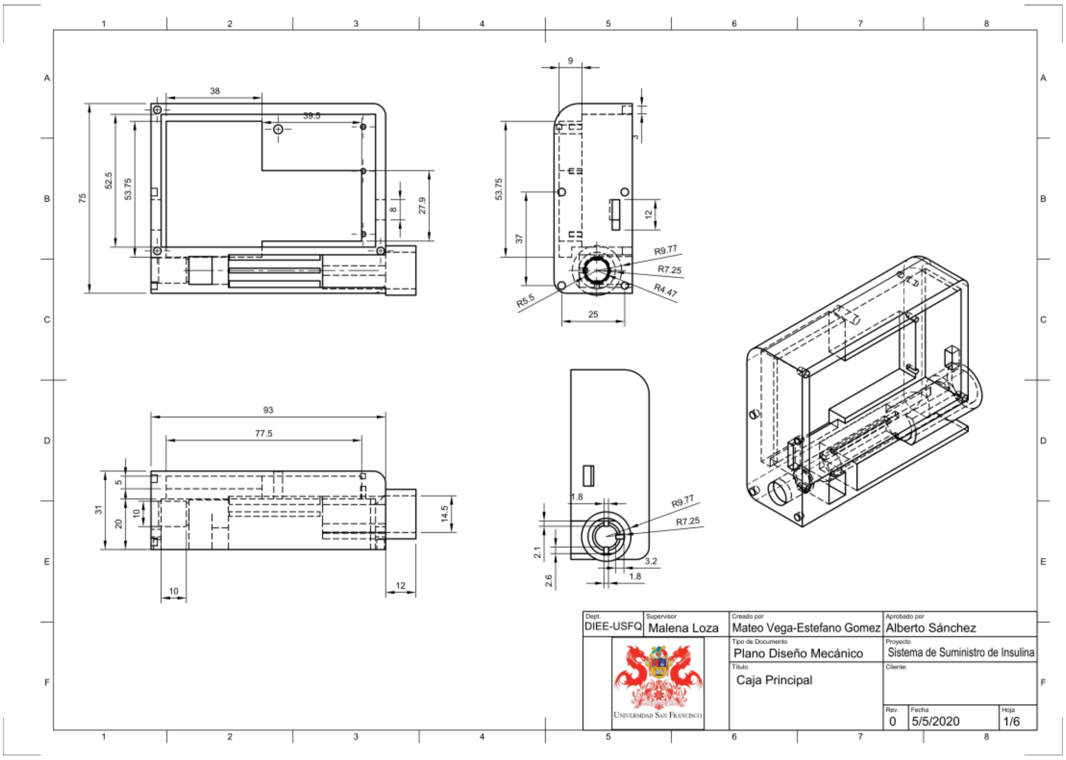


Figura. 44: Caja principal

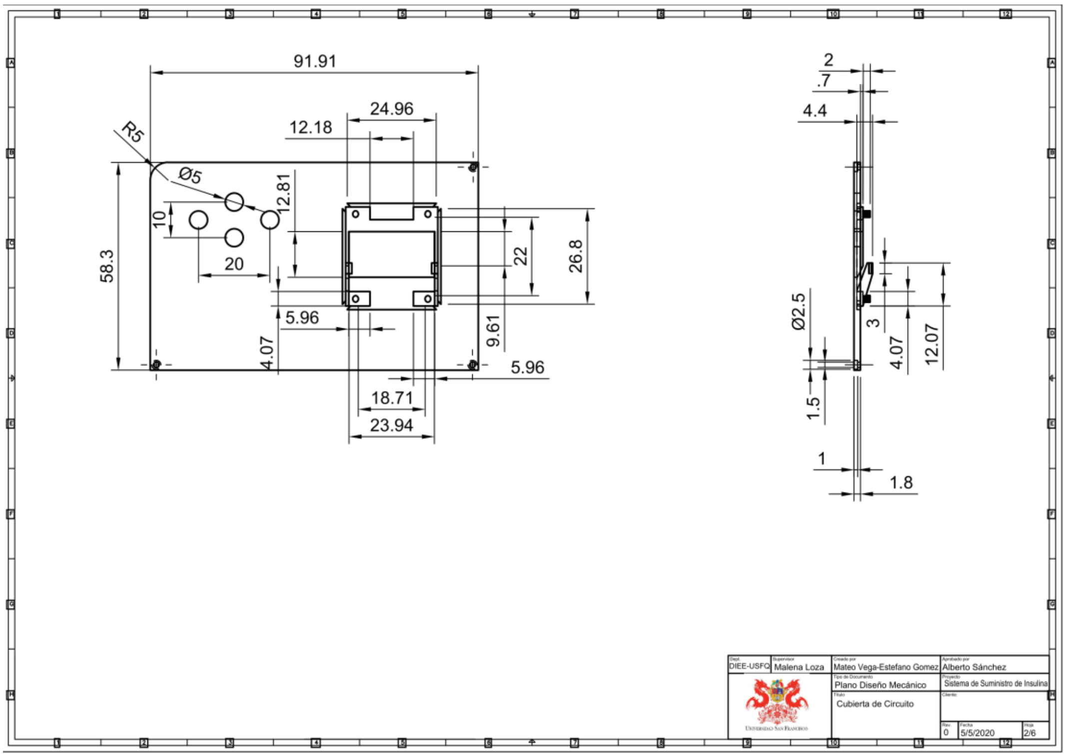


Figura. 45: Cubierta de circuito

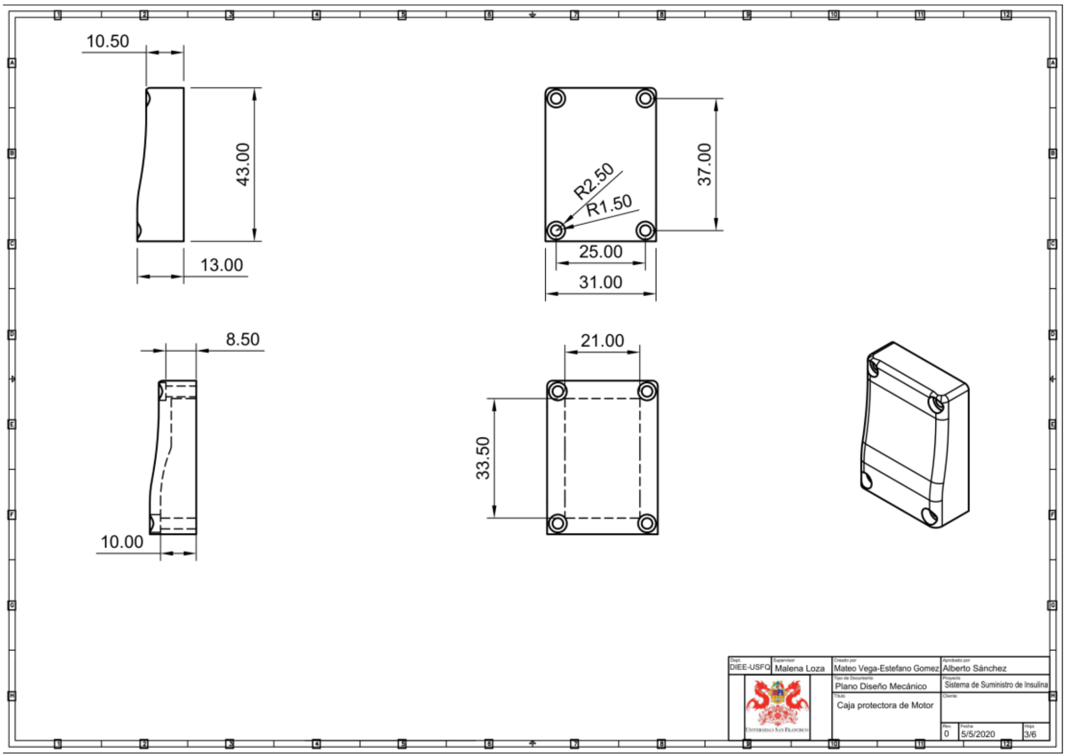


Figura. 46: Caja protectora de Motor

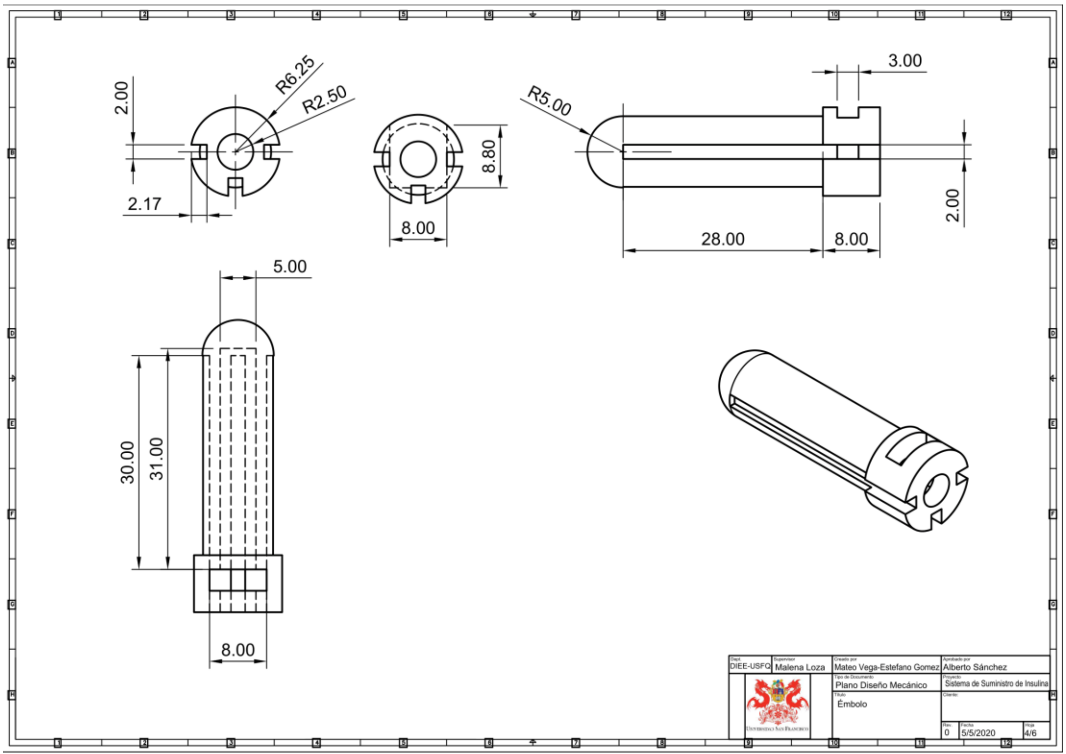


Figura. 47: Émbolo

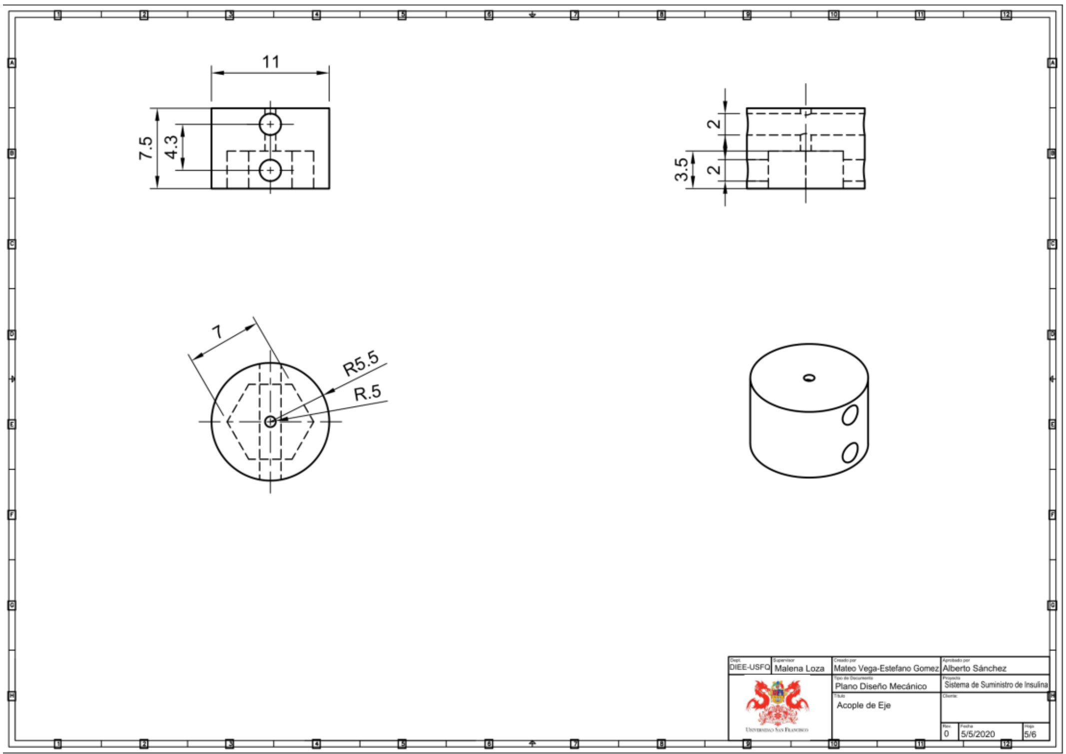


Figura. 48: Acople de eje

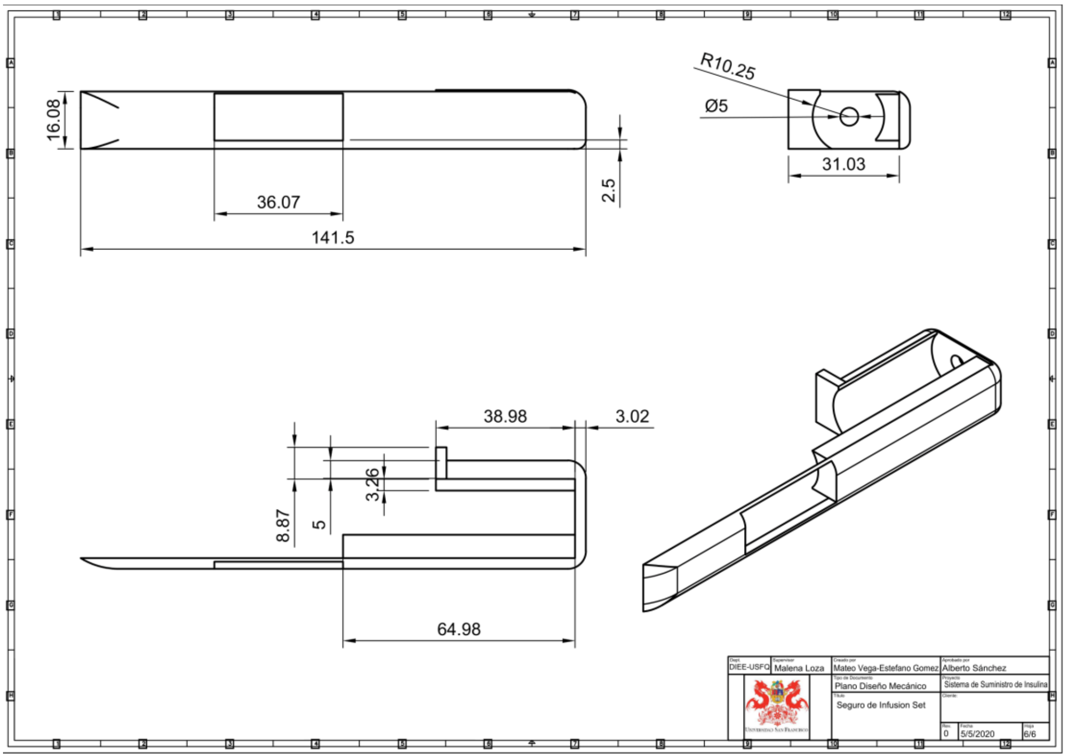


Figura. 49: Seguro de equipo de infusión

5. REFERENCIAS

Actuonix. (2020). *L16 Miniature Linear Servos for Rc & Arduino*. Recuperado de: https://www.actuonix.com/L16-R-Miniature-Linear-Servo-For-RC-p/l16-r.htm

Agencia Nacional de Regulación, Control y Vigilancia Sanitaria. (2019). *Inscripción de registro sanitario de Dispositivos Médicos de Fabricación Nacional*. Recuperado de: https://www.gob.ec/arcsa/tramites/inscripcion-registro-sanitario-dispositivos-medicos-fabricacion-nacional

American Diabetes Association. (2020). *Who should use a pump?* Recuperado de: https://www.diabetes.org/diabetes/device-technology/who-should-use-a-pump

Arduino. (2020). *Arduino Nano*. Recuperado de: https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf

Bamgbose, S. O., Li, X., & Qian, L. (2017). Closed loop control of blood glucose level with neural network predictor for diabetic patients. 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services

Bergman, R. N., Phillips, L. S., & Cobelli, C. (1981). Physiologic evaluation of factors controlling glucose tolerance in man: measurement of insulin sensitivity and beta-cell glucose sensitivity from the response to intravenous glucose. Journal of Clinical Investigation, 68(6), 1456–1467.

Chait, J. (2014). *Insulin Pumps*. Recuperado de: https://www.diabetesselfmanagement.com/diabetes-resources/tools-tech/insulin-pumps/

Cole-Palmer. (2020). *Masterflex Self-Priming Micro Pumps*. Recuperado de: https://www.coleparmer.com/p/masterflex-self-priming-micro-pumps/48758

Dinsmoor, R. (2006). *Basal Rate: Definition and Overview*. Recuperado de: https://www.diabetesselfmanagement.com/diabetes-resources/tools-tech/insulin-pumps/

El Hachimi, M., Ballouk, A., & Abdennaceur, B. (2018). Fuzzy Model Predictive Controller for Artificial Pancreas. *International Journal of Power Electronics and Drive Systems.*

Elleri, D., Dunger, D. B., & Hovorka, R. (2011). *Closed-loop insulin delivery for treatment of type 1 diabetes. BMC Medicine, 9(1).*

Faulhaber. (2020). *Stepper Motors Series AM1020.* Recuperado de: https://www.faulhaber.com/en/products/series/am1020/

FDA. (2019). *FDA authorizes first interoperable, automated insulin dosing controller designed to allow more choices for patients looking to customize their individual diabetes management device system.* Recuperado de: https://www.fda.gov/news-events/press-announcements/fda-authorizes-first-interoperable-automated-insulin-dosing-controller-designed-allow-more-choices

FDA. (2019). *Minimed 670G System – P160017/S031*. Recuperado de: https://www.fda.gov/medical-devices/recently-approved-devices/minimed-670g-system-p160017s031

FDA. (2020). *PMA Application Contents*. Recuperado de: https://www.fda.gov/medical-devices/premarket-approval-pma/pma-application-contents

FDA. (2020). *Classify Your Medical Device*. Recuperado de: https://www.fda.gov/medical-devices/overview-device-regulation/classify-your-medical-device

FDA. (2019). Code of Federal Regulations.*Food and Drug Administration Department of Health and Human Services: Medical Devices*. Recuperado de: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=862.1358

Freestyle libre. (2020). *Sensores.* Recuperado de: https://www.freestylelibre.es/libre/productos/sensores.html#product_1

Ghista, D. N., & Sarma, D. (2008). Glucose–Insulin Dynamics Modeling. In *Applied Biomedical Engineering Mechanics* (pp. 241-274)

Gillis, R., Palerm, C. C., Zisser, H., Jovanovic, L., Seborg, D. E., & Doyle, F. J. (2007). *Glucose Estimation and Prediction through Meal Responses Using Ambulatory Subject Data for Advisory Mode Model Predictive Control. Journal of Diabetes Science and Technology, 1(6), 825–833.*

Gomez, E. (2017*). Is My Blood Sugar Normal*. Recuperado de: https://www.diabetesdaily.com/learn-about-diabetes/understanding-blood-sugars/is-my-blood-sugar-normal/

Goodnow, T. (2004). *Glucose measuring module and insulin pump combination.* Recuperado de: https://patents.google.com/patent/US8460243B2/en

Hedengren, J. (2020). Blood Glucose Regulation in a Type-I Diabetic in Simulink, MATLAB Central File Exchange Recuperado de: https://www.mathworks.com/matlabcentral/fileexchange/48019-blood-glucose-regulation-in-a-type-i-diabetic-in-simulink

Heinemann, L., Fleming, A. (2015). *Insulin Pump Risks and Benefits*. Recuperado de: https://www.medscape.com/viewarticle/842369_2

Hovorka, R., Shojaee-Moradie, F., Carroll, P. V., Chassin, L. J., Gowrie, I. J., Jackson, N. C., … Jones, R. H. (2002). *Partitioning glucose distribution/transport, disposal, and endogenous production during IVGTT. American Journal of Physiology-Endocrinology and Metabolism, 282(5), E992–E1007*

Lynch, S. M., & Bequette, B. W. (2002). Model predictive control of blood glucose in type I diabetics using subcutaneous glucose measurements. Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301).

Mahmud, F., & Isse, N. H., Daud, N. A. M., & Morsin, M. (2017). Evaluation of PD/PID controller for insulin control on blood glucose regulation in a Type-I diabetes.

Maleki, A., & Geramipour, A. (2011). Continuous control of blood glucose in TIDM using fuzzy logic controller in insulin pump: A simulation study. The 2nd International Conference on Control, Instrumentation and Automation.

Medtronic. (2019). *Minimed reservoir's.* Recuperado de: https://www.medtronic-diabetes.co.uk/insulin-pump-therapy/insulin-delivery/reservoirs

Miao Miao. (2020). *MiaoMiao Smart reader.* Recuperado de: https://miaomiao.cool/

MikroElektronica. (2020). *MINI-32.* Recuperado de: https://download.mikroe.com/documents/starter-boards/mini/pic32/mini-32-manual-v101.pdf

Nalini, M., Balaji, V., & Gayathiri, R. (2016). Closed loop model for blood glucose regulation system using labview. *International Journal of Control Theory and Applications, 9*(28), 205–210.

National Library of Medicine. (2008). *Diabetes tipo 1.* Recuperado de: https://medlineplus.gov/spanish/diabetestype1.html

Palma, R. (2013). Estimation and Personalization of Clinical Insulin Therapy Parameters (Doctoral thesis, University of Texas, Austin, USA). Recuperado de: https://repositories.lib.utexas.edu/bitstream/handle/2152/21375/PALMA-DISSERTATION-2013.pdf?sequence=1

Parker, R. S., Doyle, J. F., Harting, J. E., & Peppas, N. A. (1996). Model predictive control for infusion pump insulin delivery. Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society

Pedersen, G., & Cobelli, C. (2014). Minimal Model. Modelling Methodology for Physiology and Medicine, 333–353.
Friis-Jensen, E. (2007). Modeling and Simulation of Glucose-Insulin Metabolism (University of Denmark, Copenhagen, Denmark). Recuperado de: http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/5312/pdf/imm5312.pdf

Physik Instruments. (2020). *M-232 High-Resolution Linear Actuator with DC Motor*. Recuperado de: https://www.physikinstrumente.com/en/products/linear-actuators/linear-actuators-with-stepper-dc-servo-motors/m-232-high-resolution-linear-actuator-with-dc-and-stepper-motor-703600/

Schlotthauer, G., Nicolini, G. A., Gamero, L. G., & Torres, M. E. (2002). Type I diabetes: Modeling, Identification and Non-Linear Model Predictive Control. Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society Engineering in Medicine and Biology

Sorensen, J. (1985). A physiologic model of glucose metabolism in man and its use to design and assess improved insulin therapies for diabetes (Doctoral Thesis, Massachusetts Institute of Technology, Cambridge, USA). Recuperado de: http://www.cs.cmu.edu/~./dmilam/files/sorensen_thesis.pdf

Sperling, M. (2014). *Pediatric Endocrinology: Fourth Edition*. (pp. 1–1064). Elsevier Inc.

Tandem Diabetes Care. (2020). *The t:slim X2 Insulin Pump*. Recuperado de: https://www.tandemdiabetes.com/providers/products/tslim-x2

Texas Instruments. (2016). *MSP-EXP430G2 LaunchPad Development Kit.* Recuperado de: http://www.ti.com/lit/ug/slau318g/slau318g.pdf?&ts=1589066668827

University of Iowa, (2020). *Insulin-to-carb ratios to calculate meal insulin doses with type 1 diabestes.* Recuperado de: https://uichildrens.org/health-library/insulin-carb-ratios-calculate-meal-insulin-doses-type-1-diabetes

Weissberg-Benchell, J. (2003). *Insulin pump therapy*. Recuperado de: https://care.diabetesjournals.org/content/26/4/1079.short

Zhang, B., Shankara, S., Ye, S., Zhang, H. (2019). Design and analysis of high-risk medical device clinical trials for diabetes monitoring and treatment a review. *Journal of Pancreatology.*

6. ANEXOS

Anexo A: Código de MATLAB paciente virtual

Anexo B: Código sistema de dosificación de insulina

Anexo C: Código aplicación de cálculo y de suministro de dosificaciones

Anexo D: Plano esquemático circuito principal del sistema de suministro

Anexo E: Plano caja principal

Anexo F: Plano cubierta de circuito

Anexo G: Plano caja protectora de motor

Anexo H: Plano émbolo

Anexo I: Plano acople de eje

Anexo J: Plano seguro de Infusion Set

Anexo K: Datasheets

REQUISICIÓN DE MATERIALES

UNIVERSIDAD SAN FRANCISCO	**EXHIBIT 1** REGISTRO 1	**DATE / Fecha:** **13-may-20**
	COVER SHEET PORTADA	**MR Nº** **USFQ-MR-001**

CLIENT / Cliente: **USFQ** **PROJECT / Proyecto:** **DISEÑO SISTEMA DE DOSIFICACION DE INSULINA** **Rev.:** **0**

INGENIERÍA

CONTRACTO No. **N/A**

REQUISICION DE MATERIAL

REQUISICION DE MATERIALES No: USFQ-MR-001

Rev No.	DATE	PREP. BY	LEADER	APP'D BY	REVISION DESCRIPTION
0	13/5/2020				PARA REVISIÓN

	EXHIBIT 1 REGISTRO 1	**DATE / Fecha:** **13-may-20**
	ATTACHMENT LIST FOR LISTA DE DOCUMENTACION ADJUNTA	**MR Nº** **USFQ-MR-001**

CLIENT / Cliente:	**USFQ**	**PROJECT / Proyecto:**	**DISEÑO SISTEMA DE DOSIFICACION DE INSULINA**	**Rev.:**	**0**

ITEM	**DESCRIPTION**	**REVISION**					
		A	**B**	**C**	**0**	**1**	**2**
1	REQUISICION DE MATERIAL PARA EL SISTEMA DE DOSIFICACION DE INSULINA				X		
2	LISTA DE DOCUMENTOS REQUERIDOS A LOS PROVEEDORES				X		

EXHIBIT 1

REGISTRO 1

MATERIAL REQUISITION FOR INSULIN DOSING SYSTEM

REQUISICIÓN DE MATERIALES SISTEMA DE DOSIFICACION DE INSULINA

DATE / Fecha: 13-may-20

MR Nº USFQ-MR-001

CLIENT / Cliente: USFQ

PROJECT / Proyecto: DISEÑO SISTEMA DE DOSIFICACION DE INSULINA

Rev.: 0

REV	ITEM	CODE	DESCRIPTION	QUANTITY										
				UNIT	A	B	C	D	E	F	G	H	I	TOTAL
0	1		SISTEMA EMBEBIDO PARA EL CONTROL DE TODAS LAS FUNCIONALIDADES DEL SISTEMA DE DOSIFICACION DE INSULINA ESP-32S 2.4GHz DUAL-MODE WIFI+BLUETOOTH O SIMILAR	U	1									1
0	2		BATERIA DE LITIO 3.7V/1000mah RECARGABLE PARA ALIMENTAR AL SISTEMA DE DOSIFICACION DE INSULINA MODELO: DTP603450(PHR) O SIMILAR	U	1									1
0	3		MOTOR PASO A PASO PARA SUMINSTRAR INSULINA A TRAVES DEL MECANISMO DE SUMINISTRO DEL SISTEMA DE DOSIFICACION MODELO: AM1020 FABRICANTE: FAULHABER O SIMILAR	U	1									1
0	4		DRIVER DE MOTOR PARA EL CONTROL DEL MOTOR PASO A PASO MODELO: L293D FABRICANTE: TEXAS INSTRUMENTS O SIMILAR	U	1									1
0	5		PANTALLA OLED DEL SISTEMA DE DOSIFICACION DE INSULINA CON CONTROLADOR INCLUIDO MODELO: SSD1306 O SIMILAR	U	1									1
0	6		MODULO DE CARGA Y DESCARGA PARA BATERIA DE LITIO MODELO: TP4056A (03962A) O SIMILAR	U	1									1
0	7		MODULO DE RELOJ RTC CON SOSTEN DE BATERIA CR2032 MODELO: DS3231 O SIMILAR	U	1									1
0	8		BATERIA PARA MODULO DE RELOJ MODELO: CR2032 FABRICANTE: ENERGIZER O SIMILAR	U	1									1
0	9		TIRA DE PINES MACHO DE 2.54mm TERMINACION CON ANGULO O SIMILAR	mm	1									1
0	10		TIRA DE PINES MACHO DE 2.54mm TERMINACION VERTICAL O SIMILAR	mm	1									1
0	11		SWITCH TACTIL B3F PARA MANEJO DEL SISTEMA DE DOSIFICACION DE INSULINA O SIMILAR	U	5									5
0	12		RESISTENCIA DE 10KΩ, 1/2 WATT, TOLERANCIA DE 0.05% O SIMILAR	U	1									1
0	13		'RESISTENCIA DE 100KΩ, 1/2 WATT, TOLERANCIA DE 0.05% O SIMILAR	U	2									2
0	14		TORNILLOS M1.6 DE 5mm ACERO INOXIDABLE O SIMILAR	U	7									7
0	15		TUERCA CUADRADA M4 4cm ACERO INOXIDABLE O SIMILAR	U	1									1
0	16		FILAMENTO OVERTURE PETG DE 1.75mm DE DIAMETRO 1KG O SIMILAR	U	1									1
0	17		RESERVORIO DE INSULINA DE 3ml FABRICANTE: MEDTRONIC O SIMILAR	ml	1									1
0	18		EQUIPO DE INFUSION DE INSULINA SURE-T CON AGUJA DE ACERO FABRICANTE: MEDTRONIC O SIMILAR	U	1									1
0	19		BAQUELITA DE DOBLE LADO 88.5 x 52.5 mm O SIMILAR	mm	1									1

UNIVERSIDAD SAN FRANCISCO	**EXHIBIT 1** REGISTRO 1	**DATE / Fecha:** **13-may-20**
	REQUIRED DOCUMENTS FROM VENDOR DOCUMENTOS REQUERIDOS AL PROVEEDOR	**MR Nº** **USFQ-MR-001**

CLIENT / Cliente:	**USFQ**	**PROJECT / Proyecto:**	**DISEÑO SISTEMA DE DOSIFICACION DE INSULINA**	**Rev.:**	**0**

ITEM		WITH THE BID	FOR APPROVAL		FINAL	
		QTY	QTY	DATE	QTY	DATE
1	COPIAS DE CATALOGO	2.0	2.0	1 SEMANA DESPUES DE LA ORDEN DE COMPRA	4.0	CON ENVIO

ANEXOS

ANEXO A: Código de MATLAB paciente virtual

```
function [sys,x0,str,ts,simStateCompliance] = diabetes(t,y,u,flag)

switch flag

    %%%%%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%%%%
    case 0
        [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes();

        %%%%%%%%%%%%%%%
        % Derivatives %
        %%%%%%%%%%%%%%%
    case 1
        sys=mdlDerivatives(t,y,u);

        %%%%%%%%%%
        % Update %
        %%%%%%%%%%
    case 2
        sys=mdlUpdate(t,y,u);

        %%%%%%%%%%%
        % Outputs %
        %%%%%%%%%%%
    case 3
        sys=mdlOutputs(t,y,u);

        %%%%%%%%%%%%%%%%%%%%%%%
        % GetTimeOfNextVarHit %
        %%%%%%%%%%%%%%%%%%%%%%%
    case 4
        sys=mdlGetTimeOfNextVarHit(t,y,u);

        %%%%%%%%%%%%%
        % Terminate %
        %%%%%%%%%%%%%
    case 9
        sys=mdlTerminate(t,y,u);

        %%%%%%%%%%%%%%%%%%%%
        % Unexpected flags %
        %%%%%%%%%%%%%%%%%%%%
    otherwise
        DAStudio.error('Simulink:blocks:unhandledFlag',
 num2str(flag));

end

% end sfuntmpl

%
```

```
%==============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-
function.
%==============================================================================
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes()

sizes = simsizes;

sizes.NumContStates  = 6;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);

x0 = [ 76.2159   33.3333   33.3333   16.6667   16.6667  250.0000]';

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts  = [0 0];


simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%=========================================================
% mdlDerivatives
% Return the derivatives for the continuous states.
%==============================================================================
%
function sys=mdlDerivatives(t,y,u)
%
% Model source:
% R. Palma and T.F. Edgar, Toward Patient Specific Insulin Therapy: A
 Novel
%    Insulin Bolus Calculator.  In Proceedings Texas Wisconsin
 California Control
%    Consortium, Austin, TX, Feb. 7-8, 2011.
%
% Expanded Bergman Minimal model to include meals and insulin
% Parameters for an insulin dependent type-I diabetic

% Inputs (2):
```

```
% Insulin infusion rate
ui = u(1);               % micro-U/min

% meal disturbance
d = u(2);

% States (6):
% In non-diabetic patients, the body maintains the blood glucose level
 at a
%   range between about 3.6 and 5.8 mmol/L (64.8 and 104.4 mg/dL).
g = y(1,1);               % blood glucose (mg/dl)
x = y(2,1);               % remote insulin (micro-u/ml)
i = y(3,1);               % insulin (micro-u/ml)
q1 = y(4,1);
q2 = y(5,1);
g_gut = y(6,1);           % gut blood glucose (mg/dl)

% Parameters:
gb    = 287;             % Basal Blood Glucose (mg/dL)
p1    = 3.17e-2;         % 1/min
p2    = 1.23e-2;         % 1/min
si    = 2.9e-2;          % 1/min * (mL/micro-U)
ke    = 9.0e-2;          % 1/min
kabs  = 1.2e-2;          % 1/min
kemp  = 1.8e-1;          % 1/min
f     = 8.00e-1;         % L
vi    = 12.0;            % L
vg    = 12.0;            % L
Ib    = 4e-2;
% Compute ydot:
sys(1,1) = -p1*(g-gb) - si*x*g + ...
    f*kabs/vg * g_gut;  % glucose dynamics
sys(2,1) =  p2*(i-x+Ib);            % remote insulin compartment
 dynamics
sys(3,1) = -ke*i + ui;           % insulin dynamics
sys(4,1) = d - kemp * q1;
sys(5,1) = -kemp*(q2-q1);
sys(6,1) = kemp*q2 - kabs*g_gut;

% convert from minutes to hours
sys = sys*60;
% end mdlDerivatives

%
%=============================================================================
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=============================================================================
%
function sys=mdlUpdate(t,y,u)

sys = [];
```

```
% end mdlUpdate

%
%=============================================================================
% mdlOutputs
% Return the block outputs.
%=============================================================================
%
function sys=mdlOutputs(t,y,u)

y1 = y(1);

sys = [y1];

% end mdlOutputs

%
%=============================================================================
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block.  Note that the
 result is
% absolute time.  Note that this function is only used when you
 specify a
% variable discrete-time sample time [-2 0] in the sample time array
 in
% mdlInitializeSizes.
%=============================================================================
%
function sys=mdlGetTimeOfNextVarHit(t,y,u)

sampleTime = 1;    %  Example, set the next hit to be one second
 later.
sys = t + sampleTime;

% end mdlGetTimeOfNextVarHit

%
%=============================================================================
% mdlTerminate
% Perform any end of simulation tasks.
%=============================================================================
%
function sys=mdlTerminate(t,y,u)

sys = [];

% end mdlTerminate

Not enough input arguments.

Error in diabetes (line 3)
switch flag
```

ANEXO B: Código sistema de dosificación de insulina

InsulSum01.ino

```
/*
* Archivo: InsulSum01.ino.ino
* Autor: Mateo Vega & Estefano Gomez
*
* Creado en 24 abril de 2020
*
* Este proyeto realiza el suministro de insulina segun lo indicado en el documento
* "DISEÑO DE UN CONTROL EN LAZO CERRADO DE INSULINA Y UN SISTEMA DE DOSIFICACIÓN DE INSULINA"
* (codigo:USFQ-DI-0001)
*
*
*
*/


#include <stdlib.h>
#include <Wire.h>
#include "BluetoothSerial.h"//Bluetooth
#include <Adafruit_GFX.h> //pantalla oled
#include <Adafruit_SSD1306.h>//pantalla oled
#include <Keypad.h> // para teclado
#include <Stepper.h>//libreria motor
#include <EEPROM.h>//memoria eeprom
#include "RTClib.h"//Real time clock
#include "Arduino.h"

volatile int interruptCounter;// interupciones del timer
int totalInterruptCounter; // cuantas interrupciones

hw_timer_t * timer = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED; //inicializadores de timer


#define uS_TO_S_FACTOR 1000000  //Conversor de micro a segundos*/
#define TIME_TO_SLEEP  5        // Tiempo que el esp va a estar dormido*/

RTC_DATA_ATTR int stepCount = 0;
RTC_DATA_ATTR int bootCount = 0;
RTC_DATA_ATTR int bcs = 0; // variables para guardar en memoria bcs es lo que despierta al ESP cada tiempo

#define OLED_RESET 4 // de la pantalla que aveces necesita
Adafruit_SSD1306 display(OLED_RESET); //inicializa la pantalla
BluetoothSerial ESP_BT;

const byte ROWS = 2; //four rows del teclado
const byte COLS = 2; //four columns

char hexaKeys[ROWS][COLS] = {
  {'A','B'},
  {'*','#'}//*=si  #=no //las teclas del teclado
};//define los symbolos del teclado
byte rowPins[ROWS] = {12,14}; //
byte colPins[COLS] = {27,26}; //
const int stepsPerRevolution = 20;// steps por revolucion del motor
int wake;

```

```
float res=0.16666;// resolucion en unidades dispensadas en unidades
int maxsteps=res*300*6;//hay 300 unidades en el reservorio y se aproxima un valor medio de 1/6 de unidad cada step
int pic = 0;// pantallas
int h1=0;
int h2=0;
int h11=0;
int h22=0;

int m1=0;
int m2=0;
int m11=0;
int m22=0;

int uxh1=0;
int du1xh=0;
int cu1xh=0;

int uxh2=0;
int du2xh=0;
int cu2xh=0;

int ubol=0;
int dbol=0;
int cbol=0;

float un1=0.00;
float de1=0.00;
float ce1=0.00;

float un2=0.00;
float de2=0.00;
float ce2=0.00;

float unbol=0.00;
float debol=0.00;
float cebol=0.00;

float rang1=0.00;
float rang2=0.00;
float rang3=0.00;

char hora1[2];
char min1[2];
char hora2[2];
char min2[2];

char buff1[2];
char buff2[2];
char buff3[2];

String ho1="";
String mi1="";
String ho2="";
String mi2=""; //string para hacer correcto display en pantalla, consumen memoria

float hour1;
```

```
float hour2;
float horas;
bool state = false;
String srang1="";
String srang2="";
String srang3=""; //string de los rangos de insulina de suministro
int h1u=0,h2u=0,h3u=0,h4u=0;
int blus1=0, blus2=0;
int segbol=0; int segfis=0;
int varArr[]={h1,m1,h11,m11,uxh1,du1xh,cu1xh,uxh2,du2xh,cu2xh,ubol,dbol,cbol};
int minPics_L1=0;
int maxPics_L1 = 4; //pics para primer nivel de opciones
int presionado1;
int presionado2;
int presionado3;
int presionado4;//variables para guardar que tecla se presiono
long lastmillis = 0;
long maxtime = 30000;//30 segundos maximo sin tocar, es lo que apaga la pantalla
bool cond=0; //condicion que prueba en que horario esta
int bolpas= 10;//10 pasos, hace un paso cada ciclo
int cont=0;//evalua el bolus
int minstp=1;//dejar en uno a menos que sea demasiado pequenno
int porcentaje=0; //mide nivel de bateria
int reciv[14];
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);//inicializa teclado
Stepper myStepper(stepsPerRevolution, 15, 2, 4, 5);//estps pines van seguidos , inicializa motor
RTC_DS3231 rtc; //inicializa reloj
String ReadBatt(){//mide la bateria, es necesario hacer un estudio de la curva de descarga de la bateria.
void IRAM_ATTR onTimer() {//Timer
void do_wakeup_reason(){
void reads(){ //lee los datos que se guardaron en el eeprom en caso de que se apague el esp
void calc(){//calcula los strings a partir de lo que se selecciona en las pantallas
void puts(){//guarda los datos en la memoria del eeprom
void blue(){//lee lo que hay en el BT
```

```

 void basal()

void bolus()

void bolusconfig(){

void basalconfig(){

void emergencia(){

void refresh()
void pres(){
void header() {

//^^Es para hacer display en las pantallas


bool isScheduledON(DateTime date)


void pist(){



void setup()   {


void loop() {
     DateTime now = rtc.now();
     pist();

  char customKey = customKeypad.getKey();
//  if (customKey){
  if (customKey != 'A'){
    presionado1 = 1;   //Variable del antirrebote que cambia cuando se presiona el pulsador
  }

  if (customKey == 'A' && presionado1 == 1){//menu 1 portada
    presionado1 = 0;  //Se reinicia la variable antirrebote
    lastmillis = millis();
    // pic++;       //Aumenta el contador
      if((pic>=0 && pic<=maxPics_L1) ){
        if (pic >= maxPics_L1){
          pic=1;
         }
        else if(pic <= maxPics_L1){
      pic++;
        }
      }
////////////////SUBMENU/////////////////
      if((pic>=10 && pic<=12) ){ //menu de 1era opcion 2 opciones
        if (pic >= 12){
          pic=11;
         }
        else if(pic <= 12){
      pic++;
```

```
        }
      }

      else if((pic>=20 && pic<=24) ){ //menu de 2da opcion, 4 opciones
        if (pic >= 24){
          pic=21;
         }
        else if(pic <= 24){
      pic++;
        }
      }

      else if((pic>=40 && pic<=42) ){ //menu de 3ra opcion suponiendo que hay 6 opciones
        if (pic >= 42){
          pic=41;
         }
        else if(pic <= 42){
      pic++;
        }
      }
      ////////////SUBSUBMENUS//////////
        if(pic>=110 && pic<120){ //menu de basal INFO y SET
    if (pic > 112){ // configurar el 1110 para mostrar el basal y el 1120 para cambiar las horas
      pic=111;
    }
    else if (pic < 113){
      pic++;
    }
  }
   // configurar las horas aqui
      else if(pic>=1120&& pic<=1144){ //este if configura la primera hora de comienzo
         if(pic >= 1144){
           pic=1121;}

         else if (pic <= 1144){
          pic++;
          h1=pic-1121;
         }
      }
      else if(pic>=1145 && pic<=1205){ //este if configura la primera minutos de comienzo
         if(pic >= 1205){
          pic=1146;
          //m1=0;
         }
         else if (pic <= 1205){
          pic++;
          m1=pic-1146;
         }
      }
      else if (pic>=1206 && pic <=1230){ //configura la 1era hora de final
        if(pic>=1230){
          pic=1207;
          h11=0;
        }
         else if (pic <= 1207){
          pic++;
```

```
        h11=pic-1206;
       }
    }
    else if (pic>=1231 && pic <=1291){ //configura la 1era minuto de final
      if(pic>=1291){
        pic=1232;
        m11=0;
      }
       else if (pic <= 1291){
        pic++;
        m11=m11-1231;
       }
    }
    else if(pic>=1292&& pic<1316){ //este if configura la segunda hora de comienzo
       if(pic >= 1316){
        pic=1293;
        h2=0;
       }
       else if (pic <= 1316){
        pic++;
        h2=h2-1293;
       }
    }
    else if(pic>=1317 && pic<=1377){ //este if configura la segunda minutos de comienzo
       if(pic >= 1377){
        pic=1318;
        m2=0;
       }
       else if (pic <= 1377){
        pic++;
        m2=m2-1318;
       }
    }
    else if (pic>=1378 && pic <=1402){ //configura la 2da hora de final
      if(pic>=1402){
        pic=1379;
        h22=0;
      }
       else if (pic <= 1402){
        pic++;
        h22=h22-1379;
       }
    }
    else if (pic>=1403 && pic <=1463){ //configura la 2da minuto de final
      if(pic>=1463){
        pic=1404;
        m22=0;
      }
       else if (pic <= 1463){
        pic++;
        m22=m22-1404;
       }
    }
    else if (pic>=1464 && pic <=1474){ //configura unidades de u1 por hor
      if(pic>=1474){
        pic=1465;

```

```
      }
       else if (pic <= 1474){
        pic++;
        uxh1=pic-1465;
       }
    }
    else if (pic>=1475 && pic <=1485){ //configura decimas de u1 xh
      if(pic>=1485){
        pic=1476;
        ;
      }
       else if (pic <= 1485){
        pic++;
        du1xh=pic-1476;
       }
    }
    else if (pic>=1486 && pic <=1488){ //configura centimas de u1 xh
      if(pic>=1488){
        pic=1487;
        cu1xh=0;
        ;
      }
       else if (pic <= 1488){
        pic++;
        cu1xh+=5;
       }
    }
    else if (pic>=1489 && pic <=1499){ //configura unidades de u2 por hor
          if(pic>=1499){
            pic=1490;

          }
           else if (pic <= 1499){
            pic++;
            uxh2=pic-1490;
           }
        }
        else if (pic>=1500 && pic <=1510){ //configura decimas de u2 xh
          if(pic>=1510){
            pic=1501;
            ;
          }
           else if (pic <= 1510){
            pic++;
            du2xh=pic-1501;
           }
        }
        else if (pic>=1511 && pic <=1513){ //configura centimas de u2 xh
          if(pic>=1521){
            pic=1512;
            cu2xh=0;
            ;
          }
           else if (pic <= 1513){
            pic++;
            cu2xh+=5;
           }
```

```
            }

        else if(pic >= 220 && pic <=222 ){//menu de preguntar suministro
              if(pic>=222){
                pic=221;
              }
               else if (pic <= 222){
                pic++;
               }
        }
        else if(pic >= 231 && pic <=232){//menu de preguntar modo fiesta
              if(pic>=232){
                pic=231;
              }
               else if (pic <= 232){
                pic++;
               }
            }

        else if (pic>=2410 && pic <=2420){ //configura unidades de u3 por hor
              if(pic>=2420){
                pic=2411;

              }
               else if (pic <= 2420){
                pic++;
                ubol=pic-2411;
               }
            }
            else if (pic>=2421 && pic <=2431){ //configura decimas de u3 xh
              if(pic>=2431){
                pic=2422;
                ;
              }
               else if (pic <= 2431){
                pic++;
                dbol=pic-2422;
               }
            }
            else if (pic>=2432 && pic <=2434){ //configura centimas de u3 xh
              if(pic>=2442){
                pic=2433;
                cbol=0;
                ;
              }
               else if (pic <= 2434){
                pic++;
                cbol+=5;
               }
            }



      } //aqui termina la deteccion de tecla para bajar

    //Condicionales para antirrebote y uso del pulsador disminuir
    if (customKey != 'B'){
```

```
    presionado2 = 1;    //Variable del antirrebote que cambia cuando se presiona el pulsador
  }

  if (customKey == 'B' && presionado2 == 1){
    presionado2 = 0;   //Se reinicia la variable antirrebote
    lastmillis = millis();
    if(pic>=0 && pic<=maxPics_L1){
            if (pic <= 1){
              pic=maxPics_L1;
             }
            else if(pic <= maxPics_L1){
          pic--;
            }
          }
else if(pic>=10 && pic<=12){//sube y baja en el menu BASAL
        if (pic <= 11){
          pic=12;
        }
        else if (pic <= 12){
          pic--;
        }
      }
else if(pic>=20 && pic <=24){//va del 4 al 1 en BOLUS
    if (pic<=21){
        pic=24;
    }
    else if(pic<=24){
        pic--;
    }
}
else if(pic>=40 && pic <=42){
    if (pic<=41){
        pic=42;
    }
    else if(pic<=42){
        pic--;
    }

}




//submenus DEL B
        else if(pic>=1120&& pic<=1144){ //este if configura la primera hora de comienzo
                 if(pic <= 1121){
                   pic=1144;}

                 else if (pic <= 1144){
                  pic--;
                  h1=pic-1121;
                 }
              }
        else if(pic>=1145 && pic<=1205){ //este if configura la primera minutos de comienzo
                 if(pic <= 1146){
                  pic=1205;
                  //m1=0;
```

```
                }
                else if (pic <= 1205){
                 pic--;
                 m1=pic-1146;
                }
             }
         else if (pic>=1206 && pic <=1230){ //configura la 1era hora de final
               if(pic<=1207){
                 pic=1230;
                 h11=23;
               }
                else if (pic <= 1207){
                 pic--;
                 h11=pic-1206;
                }
             }
         else if (pic>=1231 && pic <=1291){ //configura la 1era minuto de final
               if(pic<=1232){
                 pic=1291;
                 m11=59;
               }
                else if (pic <= 1291){
                 pic--;
                 m11=m11-1231;
                }
             }
         else if(pic>=1292&& pic<1316){ //este if configura la segunda hora de comienzo
                if(pic <= 1293){
                 pic=1316;
                 h2=23;
                }
                else if (pic <= 1316){
                 pic--;
                 h2=h2-1293;
                }
             }
         else if(pic>=1317 && pic<=1377){ //este if configura la segunda minutos de comienzo
                if(pic <= 1318){
                 pic=1377;
                 m2=59;
                }
                else if (pic <= 1377){
                 pic--;
                 m2=m2-1318;
                }
             }
         else if (pic>=1378 && pic <=1402){ //configura la 2da hora de final
               if(pic<=1379){
                 pic=1402;
                 h22=23;
               }
                else if (pic <= 1402){
                 pic--;
                 h22=h22-1379;
                }
             }
         else if (pic>=1403 && pic <=1463){ //configura la 2da minuto de final
```

```
                if(pic<=1404){
                  pic=14063;
                  m22=59;
                }
                 else if (pic <= 1463){
                  pic--;
                  m22=m22-1404;
                 }
              }
          else if (pic>=1464 && pic <=1474){ //configura unidades de u1 por hor
            if(pic<=1465){
              pic=1474;

            }
             else if (pic <= 1474){
              pic--;
              uxh1=pic-1465;
             }
          }
          else if (pic>=1475 && pic <=1485){ //configura decimas de u1 xh
            if(pic<=1476){
              pic=1485;
              ;
            }
             else if (pic <= 1485){
              pic--;
              du1xh=pic-1476;
             }
          }
          else if (pic>=1486 && pic <=1488){ //configura centimas de u1 xh
            if(pic<=1487){
              pic=1488;
              cu1xh=5;
              ;
            }
             else if (pic <= 1488){
              pic--;
              cu1xh-=5;
             }
          }
          else if (pic>=1489 && pic <=1499){ //configura unidades de u2 por hor
                if(pic<=1490){
                  pic=1499;

                }
                 else if (pic <= 1499){
                  pic--;
                  uxh2=pic-1490;
                 }
              }
              else if (pic>=1500 && pic <=1510){ //configura decimas de u2 xh
                if(pic<=1501){
                  pic=1510;
                  ;
                }
                 else if (pic <= 1510){
                  pic--;
```

```
                  du2xh=pic-1501;
                 }
              }
              else if (pic>=1511 && pic <=1513){ //configura centimas de u2 xh
                if(pic<=1512){
                  pic=1521;
                  cu2xh=5;
                  ;
                }
                 else if (pic <= 1513){
                  pic--;
                  cu2xh-=5;
                 }
              }
                if(pic<=2526){
                  pic=2549;
                }
                 else if (pic <= 2549){
                  pic--;
                  h4u=pic-2526;
                 }


  else if (pic>=2410 && pic <=2420){ //configura unidades de u3 por hor
                if(pic<=2411){
                  pic=2420;

                }
                 else if (pic <= 2420){
                  pic--;
                  ubol=pic-2411;
                 }
              }
              else if (pic>=2421 && pic <=2431){ //configura decimas de u3 xh
                if(pic<=2422){
                  pic=2431;
                }
                 else if (pic <= 2431){
                  pic--;
                  dbol=pic-2431;
                 }
              }
              else if (pic>=2432 && pic <=2434){ //configura centimas de u3 xh
                if(pic<=2433){
                  pic=2434;
                  cbol=5;
                  ;
                }
                 else if (pic <= 2434){
                  pic--;
                  cbol-=5;
                 }
              }
}

  if (customKey != '#'){
    presionado3 = 1;    //Variable del antirrebote que cambia cuando se presiona el pulsador
```

```
  }
  if(customKey =='#' && presionado3==1){ //regresar
  lastmillis = millis();
  presionado3=0;
  if (pic >= 11 && pic <100) pic=1; //del 0 al 10 es LV1, 10-100 LV2,100-1000 LV3
  if(pic>=110&&pic<200) pic=11;
  if (pic >= 210 && pic <300 ) pic=21;
  if (pic >= 310 && pic <400) pic=31;
  if (pic >= 410 && pic <500) pic=41;
  if (pic >= 510 && pic <600) pic=51;
  if (pic>=1120 && pic<1513) pic=121;//esto hace que pueda volver a la pantalla de SETBASAL cualquier rato
  if (pic >= 2410 && pic<2575) pic=241;//regresa de la config de hora al display de config
  }
  if (customKey != '*'){
    presionado4 = 1;    //Variable del antirrebote que cambia cuando se presiona el pulsador
  }
  if(customKey=='*' && presionado4==1){ //entrar
  lastmillis = millis();
  presionado4=0;
  if ((pic>0 && pic<11) || (pic>12 && pic <42) || (pic>45 && pic<220) ||(pic>300 && pic <1000) )
  {
    pic=pic*10+1;
  }

  else if(pic==12) {pic=1121;}// [ara llegar al menu de opciones de hora
  else if(pic==42){//aqui se selecciona el DEEPSLEEP
      esp_deep_sleep_start();

  }
  else if (pic>=1120 && pic<=1144){ //pasa la primera hora
    pic=1146;
    //Serial.println('a');
  }
  else if (pic>=1145 && pic<=1205){// pasan los primeros minutos
    pic=1207;
  }

  else if(pic >=1206 && pic <=1230){//psan las segundas horas
    pic=1232;
  }
  else if(pic >=1231 && pic <=1291){//pasan los segundos minutos //aqui para configurar para solo setear la hora de comienzo y final
    pic=1293;
  }
  else if (pic>=1292 && pic<=1316){ // tercera hora
    pic=1318;
  }
  else if (pic>=1317 && pic<=1377){//tercera minutos
    pic=1379;
  }

  else if(pic >=1378 && pic <=1402){//cuarta hora
    pic=1404;
  }
  else if(pic >=1403 && pic <=1463){//cuarto minuto
```

```
  pic=1465;
}

else if (pic>=1464 && pic <=1474){ //configura unidades de u1 por hor
    pic=1476;
}
else if (pic>=1475 && pic <=1485){ //configura decimas de u1 xh
    pic=1487;
}
else if (pic>=1486 && pic <=1488){ //configura centimas de u1 xh
    pic=1490;
}
else if (pic>=1489 && pic <=1499){ //configura unidades de u2 por hor
    pic=1501;
    }
else if (pic>=1500 && pic <=1510){ //configura decimas de u2 xh
        pic=1512;
    }
else if (pic>=1511 && pic <=1513){ //configura centimas de u2 xh
    pic=1476;
    }
else if(pic==221){//no suministrar bolus
    pic=22;
    segbol=0;
}
else if(pic==222){
    pic=22;
    cond=1;
}

else if(pic==231){//no party
    pic=23;
    segfis=0;
}
else if(pic==232){
    pic=23;
    segfis=1;
}

else if(pic==241){pic=2411;}

else if (pic>=2410 && pic <=2420){ //configura unidades de u3 por hor
      pic=2422;
    }
    else if (pic>=2421 && pic <=2431){ //configura decimas de u3 xh
      pic=2432;
    }
    else if (pic>=2431 && pic <=2434){ //configura centimas de u3 xh
      pic=2422;
    }

}//aqui termina *entrar

blue();

calc();

```

```
  puts();

  ReadBatt();

//JUMP TO DEFAULT IF NO CLICK IS DETECTED
if (millis() >= (lastmillis + maxtime))
{
    pic = 0;
    esp_deep_sleep_start();
}
if(pic>=1121 && pic<=1513){//aqui la programacion de la pantalla de configuracion de basal
basalconfig();
display.setCursor(0,11);  display.print (ho1);display.print (":");display.print (mi1);display.print ("-");display.print (ho2);display.print (":");display.print (mi2);
display.setCursor(0,20);  display.print (srang1);  display.print ("U x hora");
display.setCursor(0,29);  display.print (ho2);display.print (":");display.print (mi2);display.print ("-");display.print (ho1);display.print (":");display.print (mi1);
display.setCursor(0,47);  display.print (srang2);  display.print ("U x hora");
refresh();
}

else if(pic>=2411 && pic<=2513 ){//aqui la programacion de la pantalla de configuracion de bolus
bolusconfig();
display.setCursor(0,11);  display.print (srang3);
display.setCursor(0,20);  display.print ("UNIDADES");
refresh();
}
    switch (pic){

      case 0: //este es el caso de presentacion
pres();
      break;
    //LAYER 1////////////////////////////////////////////////////
      case 1:
        header();
        display.setCursor(0,11);  display.print (">BASAL");
        display.setCursor(0,20);  display.print (" BOLUS");
        display.setCursor(0,29);  display.print (" EMERGENCIA");
        display.setCursor(0,47);  display.print (" DEEP SLEEP");
        refresh();
      break;

      case 2:
        header();
        display.setCursor(0,11);  display.print (" BASAL");
        display.setCursor(0,20);  display.print (">BOLUS");
        display.setCursor(0,29);  display.print (" EMERGENCIA");
        display.setCursor(0,47);  display.print (" DEEP SLEEP");
        refresh();
      break;

      case 3:
        header();
        display.setCursor(0,11);  display.print (" BASAL");
        display.setCursor(0,20);  display.print (" BOLUS");
        display.setCursor(0,29);  display.print (">EMERGENCIA");
```

```
        display.setCursor(0,47);  display.print (" DEEP SLEEP");
        refresh();
      break;

      case 4:
        header();
        display.setCursor(0,11);  display.print (" BASAL");
        display.setCursor(0,20);  display.print (" BOLUS");
        display.setCursor(0,29);  display.print (" EMERGENCIA");
        display.setCursor(0,47);  display.print (">DEEP SLEEP");
        refresh();
      break;


      case 5:
        header();
        display.setCursor(0,11);  display.print (" BASAL");
        display.setCursor(0,20);  display.print (" BOLUS");
        display.setCursor(0,29);  display.print (" TIEMPO");
        display.setCursor(0,47);  display.print (" CALIBRE");
        display.setCursor(0,56);  display.print (">RESET");
        refresh();
      break;
    //LAYER 2////////////////////////////////////////////////////
      case 11:
        basal();
        display.setCursor(0,11);  display.print (">INFO");
        display.setCursor(0,20);  display.print (" CONFIG");
        refresh();
      break;

      case 12:
            basal();
            display.setCursor(0,11);  display.print (" INFO");
            display.setCursor(0,20);  display.print (">CONFIG");
            refresh();
      break;

      case 21:
          bolus();
          display.setCursor(0,11); display.print (">INFO");
          display.setCursor(0,20); display.print (" SUMINISTRO");
          display.setCursor(0,29); display.print (" FIESTA");
          display.setCursor(0,47); display.print (" CONFIG");
          refresh();
    break;
      case 22:
          bolus();
          display.setCursor(0,11); display.print ("INFO");
          display.setCursor(0,20); display.print ("> SUMINISTRO");
          display.setCursor(0,29); display.print (" FIESTA");
          display.setCursor(0,47); display.print (" CONFIG");
          refresh();
          break;
      case 23:
          bolus();
          display.setCursor(0,11); display.print ("INFO");
```

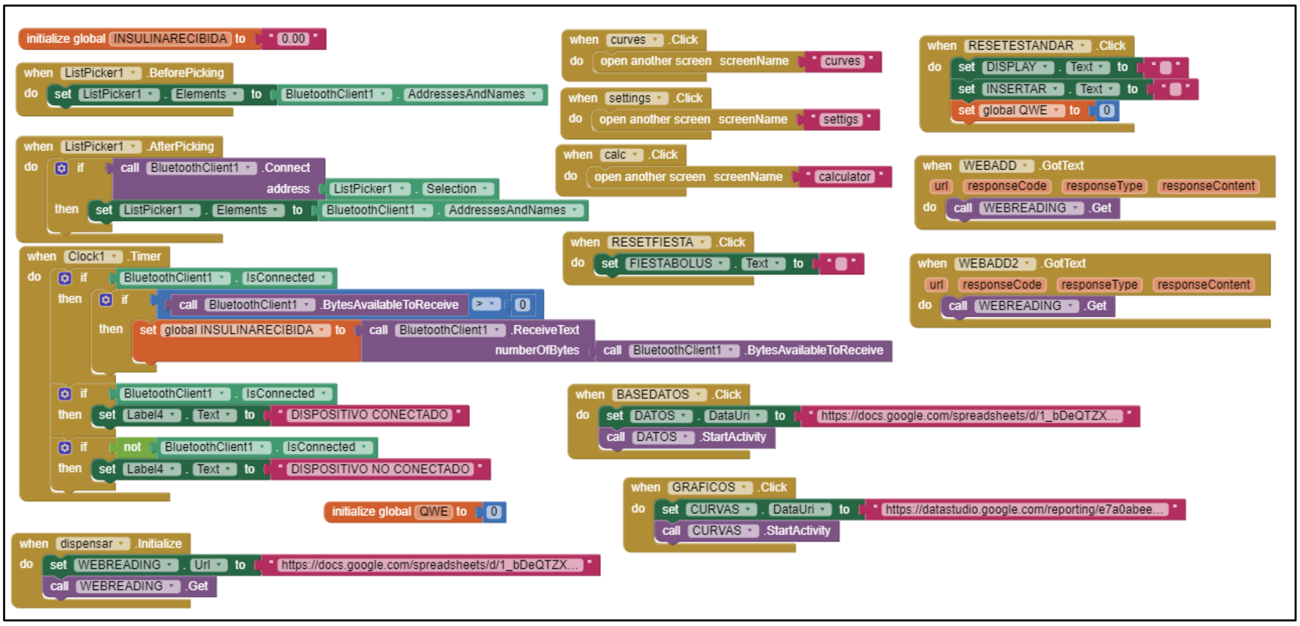
```
          display.setCursor(0,20); display.print (" SUMINISTRO");
          display.setCursor(0,29); display.print (">FIESTA");
          display.setCursor(0,47); display.print (" CONFIG");
          refresh();
          break;
      case 24:
          bolus();
          display.setCursor(0,11); display.print (" INFO");
          display.setCursor(0,20); display.print (" SUMINISTRO");
          display.setCursor(0,29); display.print (" FIESTA");
          display.setCursor(0,47); display.print (">CONFIG");
          refresh();
          break;
      case 31:
        emergencia();
        refresh();
      break;
      case 41:
          display.setTextSize(1);
          display.setTextColor(WHITE);
          display.setCursor(22,0);  display.print("DEEPSLEEP");
          display.drawLine (0,9,128,9, WHITE);
          display.setCursor(0,11);  display.print ("SEGURO?");
          display.setCursor(0,20);  display.print (">NO");
          display.setCursor(0,29);  display.print (" SI");
          refresh();
          break;
      case 42:
          display.setTextSize(1);
          display.setTextColor(WHITE);
          display.setCursor(22,0);  display.print("DEEPSLEEP");
          display.drawLine (0,9,128,9, WHITE);
          display.setCursor(0,11);  display.print ("SEGURO?");
          display.setCursor(0,20);  display.print (" NO");
          display.setCursor(0,29);  display.print (">SI");
          refresh();
          break;

////////////////////LAYER3////////////////
      case 111:
          display.setTextSize(1);
          display.setTextColor(WHITE);
          display.setCursor(22,0);  display.print("BASAL INFO");
          display.drawLine (0,9,128,9, WHITE);
          display.setCursor(0,11);  display.print (ho1);display.print (":");display.print (mi1);display.print ("-");display.print (ho2);display.print (":");display.print (mi2);
          display.setCursor(0,20);  display.print (srang1);  display.print ("U x hora");
          display.setCursor(0,29);  display.print (ho2);display.print (":");display.print (mi2);display.print ("-");display.print (ho1);display.print (":");display.print (mi1);
          display.setCursor(0,47);  display.print (srang2);  display.print ("U x hora");
          refresh();
          break;
      case 211:
          display.setTextSize(1);
          display.setTextColor(WHITE);
          display.setCursor(22,0);  display.print("BOLUS INFO");
          display.drawLine (0,9,128,9, WHITE);
```

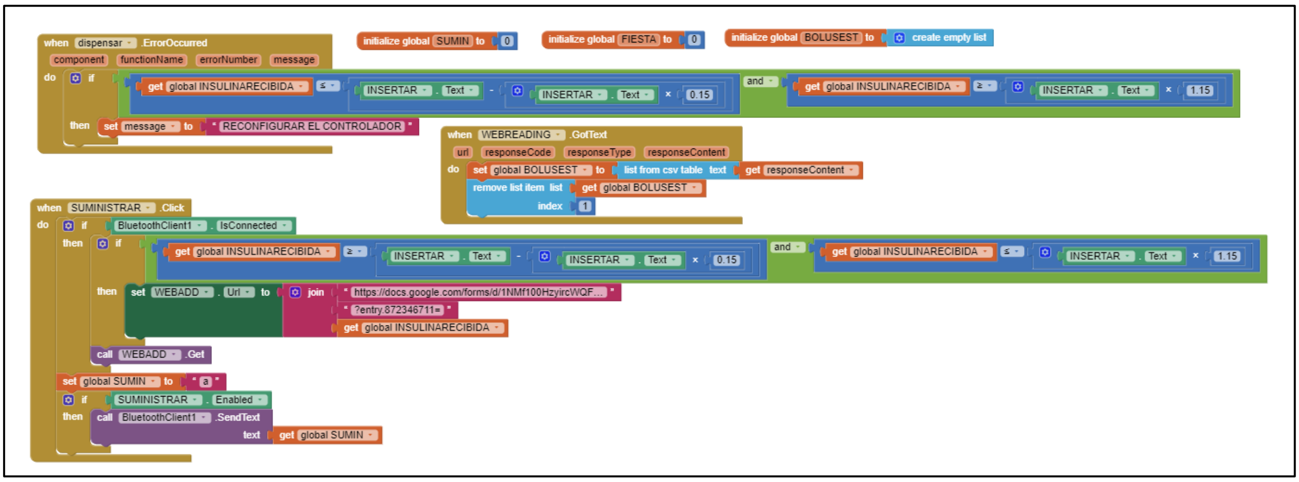
```
            display.setTextSize(2);
            display.setCursor(0,11);  display.print (srang3);
            display.setCursor(0,30);  display.print ("UNIDADES");
            refresh();
            break;

        case 221:
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(22,0);  display.print("SUMINISTRAR");
            display.drawLine (0,9,128,9, WHITE);
            display.setCursor(0,11);  display.print ("SEGURO?");
            display.setCursor(0,20);  display.print (">NO");
            display.setCursor(0,29);  display.print (" SI");
            refresh();
            break;

        case 222:
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(22,0);  display.print("SUMINISTRAR");
            display.drawLine (0,9,128,9, WHITE);
            display.setCursor(0,11);  display.print ("SEGURO?");
            display.setCursor(0,20);  display.print (" NO");
            display.setCursor(0,29);  display.print (">SI");
            refresh();
            break;
        case 231:
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(22,0);  display.print("FIESTA");
            display.drawLine (0,9,128,9, WHITE);
            display.setCursor(0,11);  display.print ("SEGURO?");
            display.setCursor(0,20);  display.print (">NO");
            display.setCursor(0,29);  display.print (" SI");
            refresh();
            break;

        case 232:
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(22,0);  display.print("FIESTA");
            display.drawLine (0,9,128,9, WHITE);
            display.setCursor(0,11);  display.print ("SEGURO?");
            display.setCursor(0,20);  display.print (" NO");
            display.setCursor(0,29);  display.print (">SI");
            refresh();
            break;


    }

//comparador de reloj
    if (isScheduledON(now))
       {
            state = true;
            bcs=(3600.0*res)/rang1; //esto cambia los segundos por bit
```

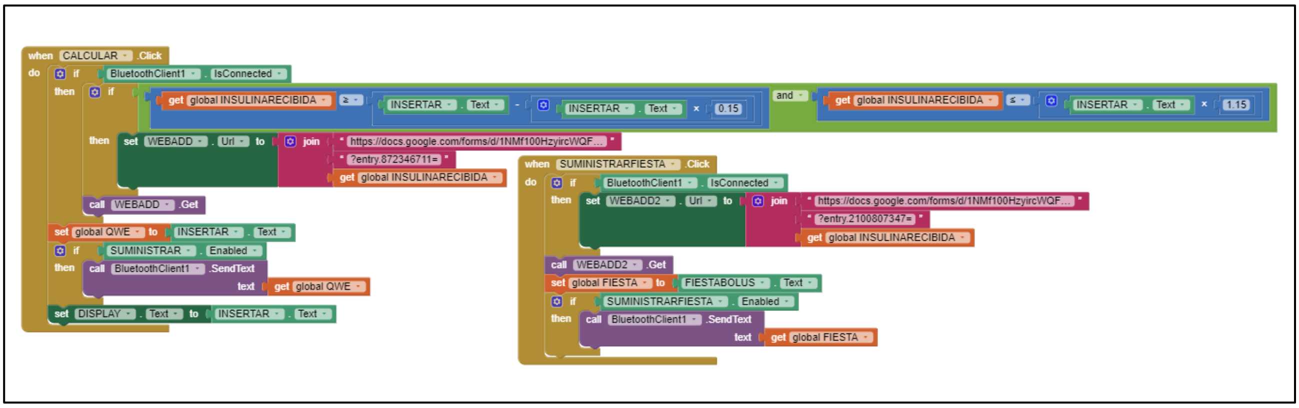
```
       }
       else if (!isScheduledON(now)){
          state = true;
          bcs=(3600.0*res)/rang2;
       }


do_wakeup_reason();



if(stepCount<maxsteps){

if(bcs>0){
    esp_sleep_enable_timer_wakeup(bcs * uS_TO_S_FACTOR);
}


     if(wake==3){
         if(state==true && h1u!=0 && h2u!=0){
             myStepper.step(minstp);
         }
        esp_deep_sleep_start();
     }


    if(cond==1){
        if(cont<bolpas){ //cambiar el bolpas para cambia el numero de steps que da el motor
            myStepper.step(minstp); //cada 15 segundos el stepper da 1 step
                        stepCount++;
                        cont++;
        }
        else if(cont==bolpas){
            cond=0;
            cont=0;
        }

    }

    }
else{
    myStepper.step(-maxsteps);
    stepCount=0;
}


}
/////////////////////////////////////////////////////////////////////////////////////////////

```

ANEXO C: Código aplicación de cálculo y de suministro de dosificaciones

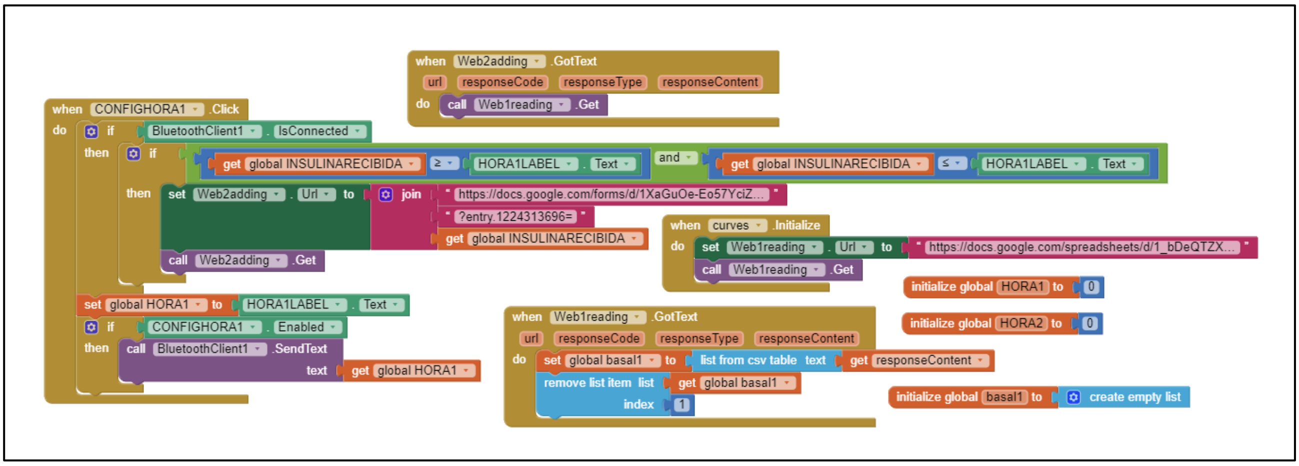
PROGRAMACIÓN POR BLOQUES APLICACIÓN DE CÁLCULO Y DE SUMINISTRO
PANTALLA BOLUS

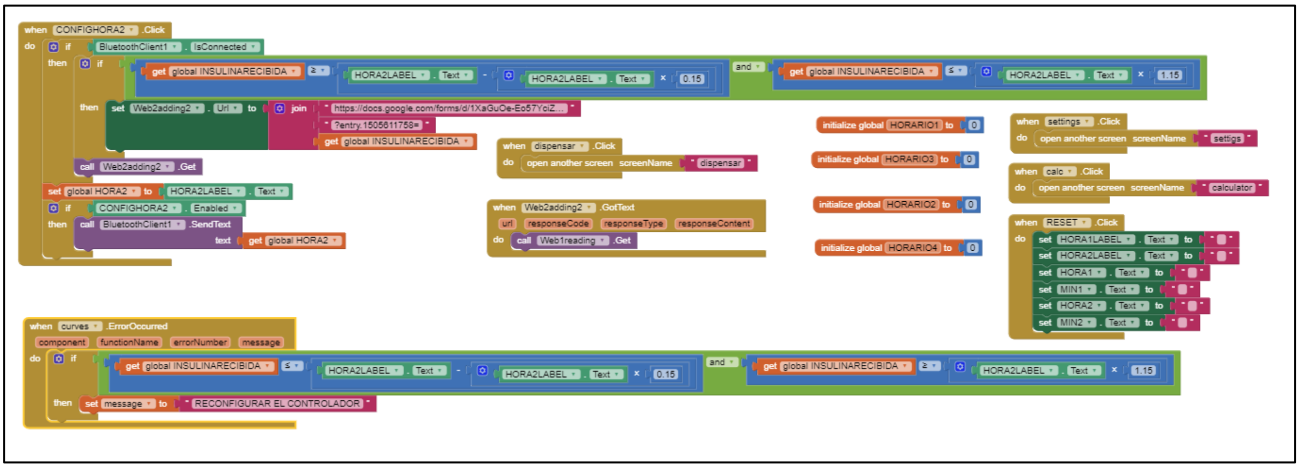


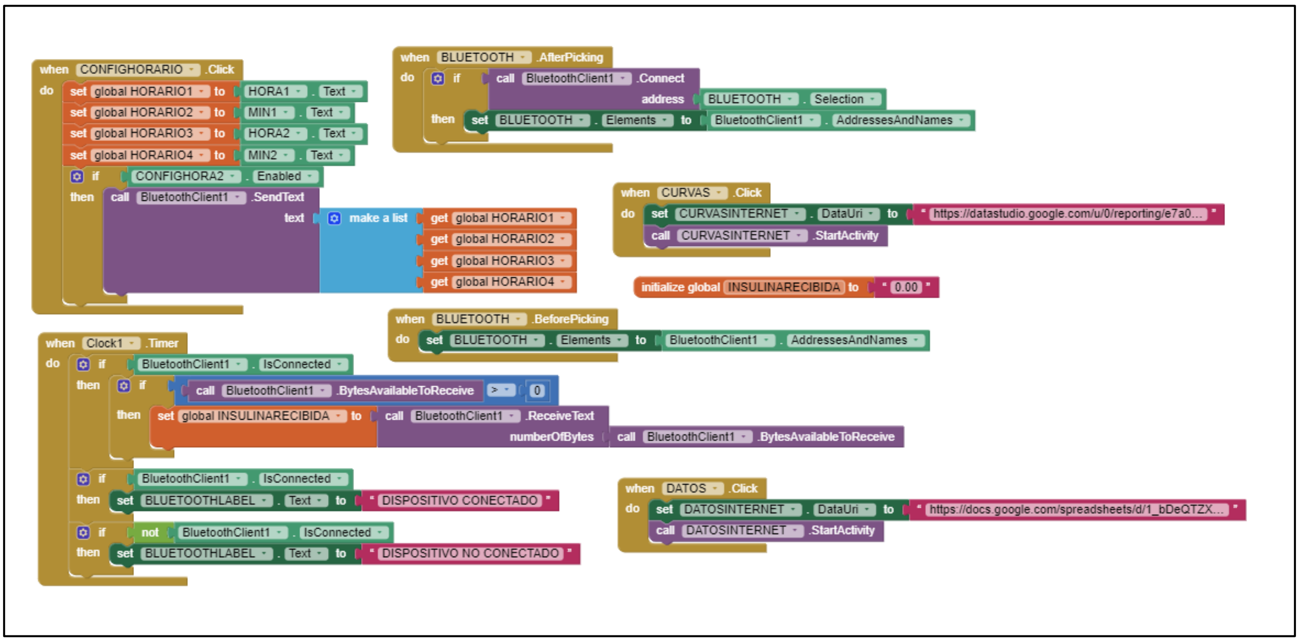




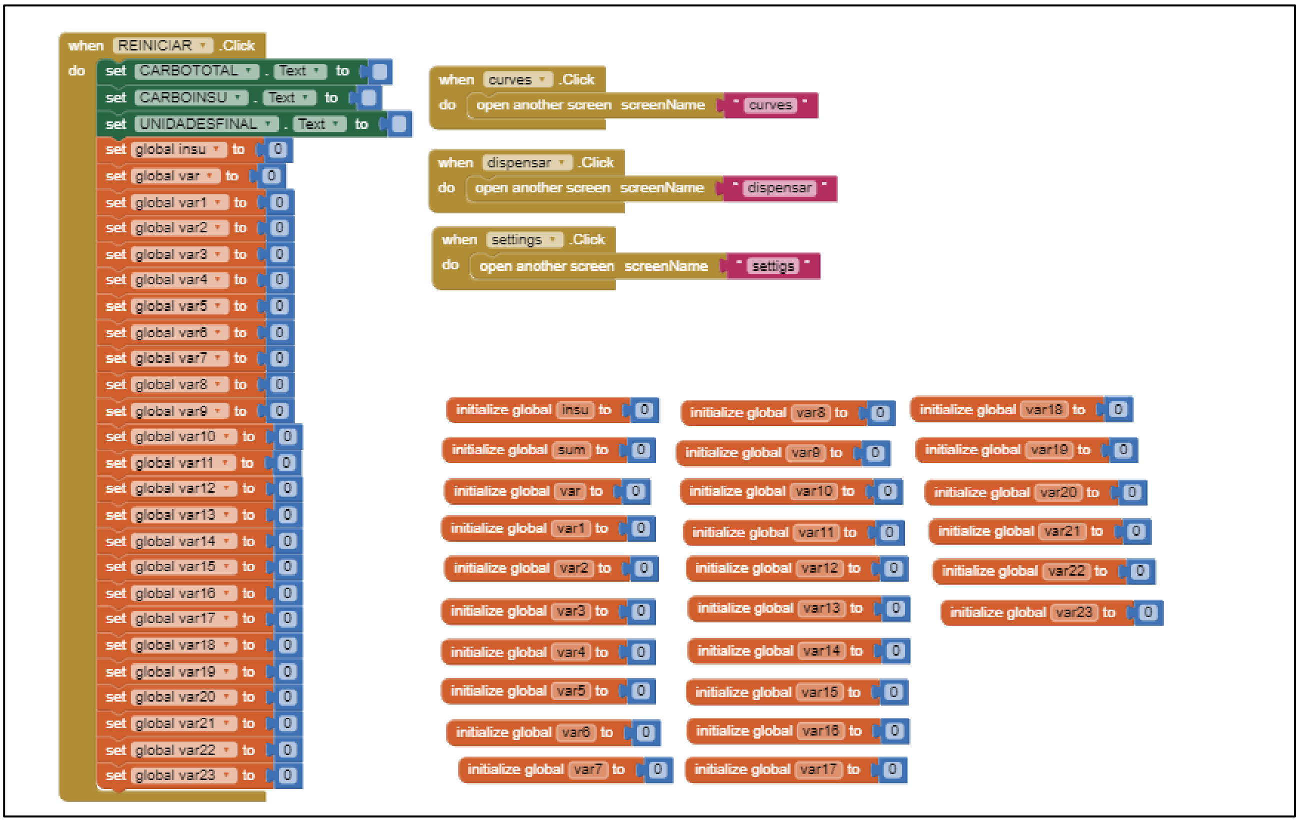
PANTALLA BASAL

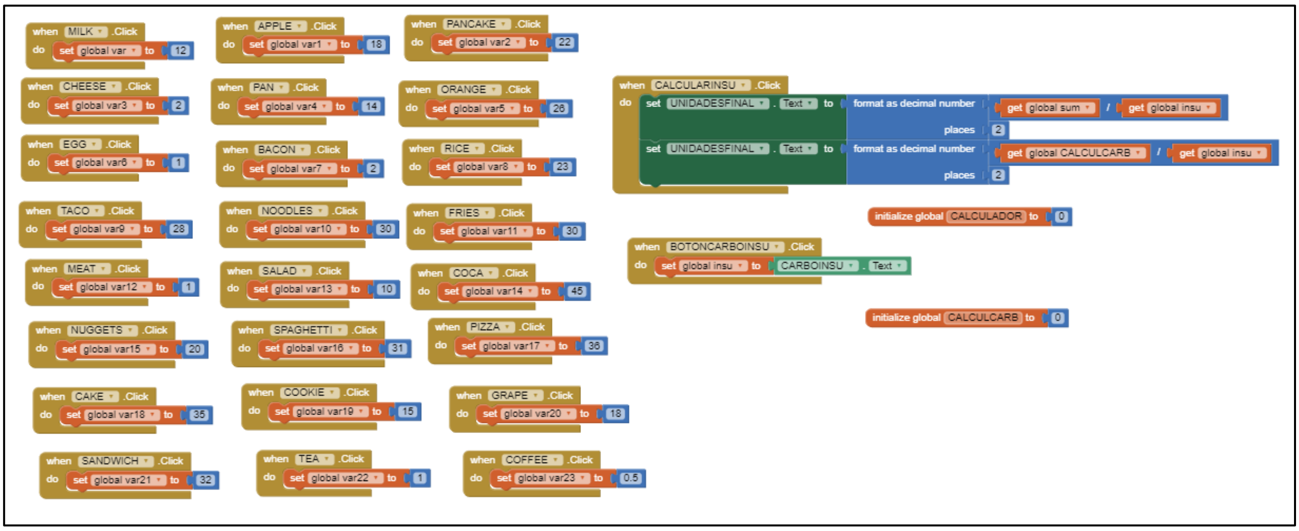


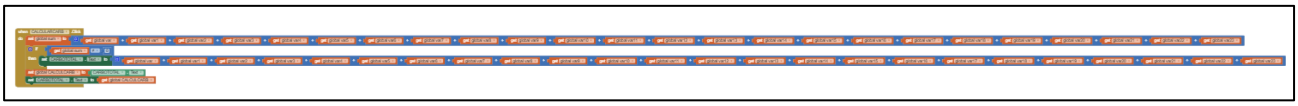




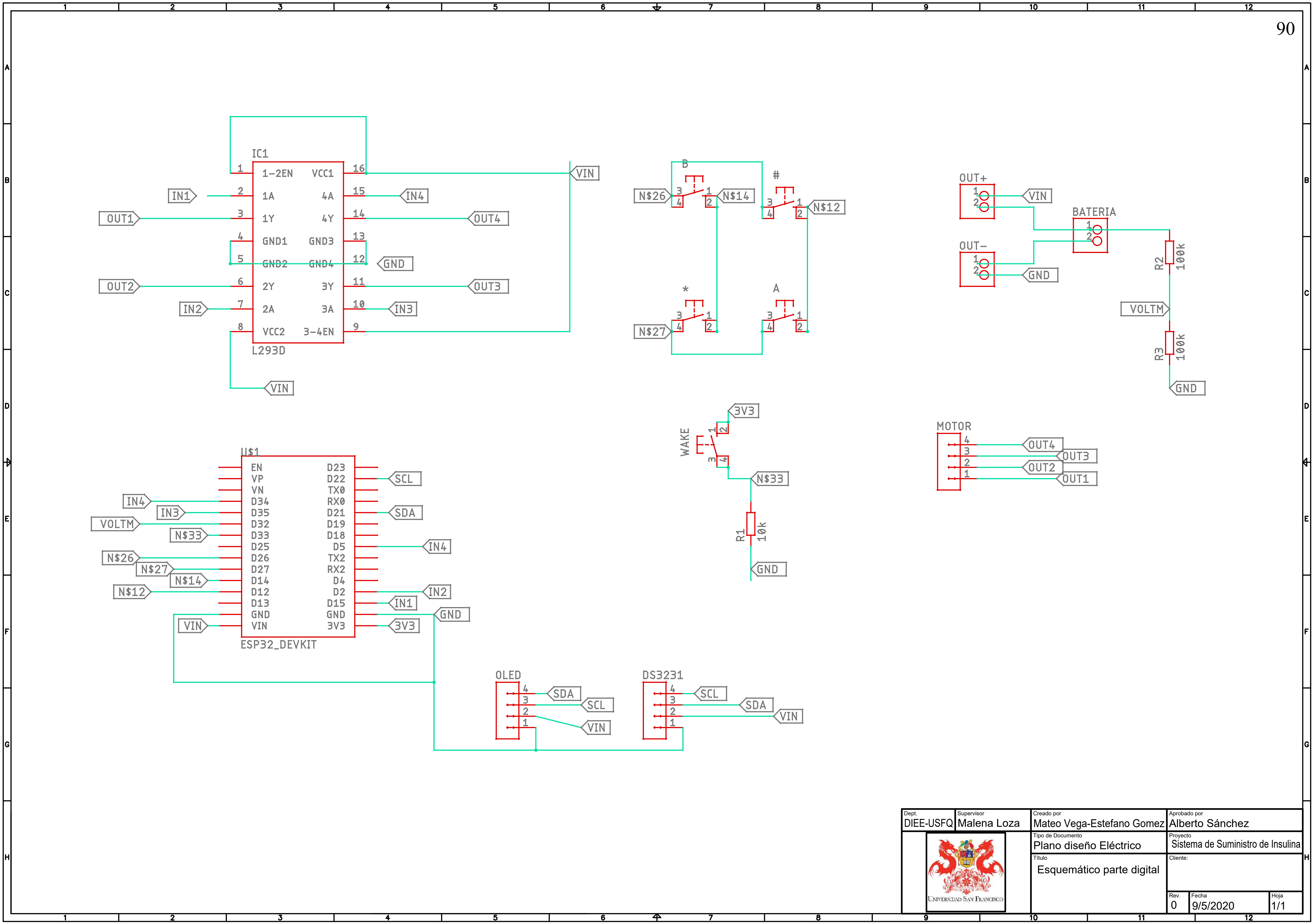
PANTALLA CALCULADORA



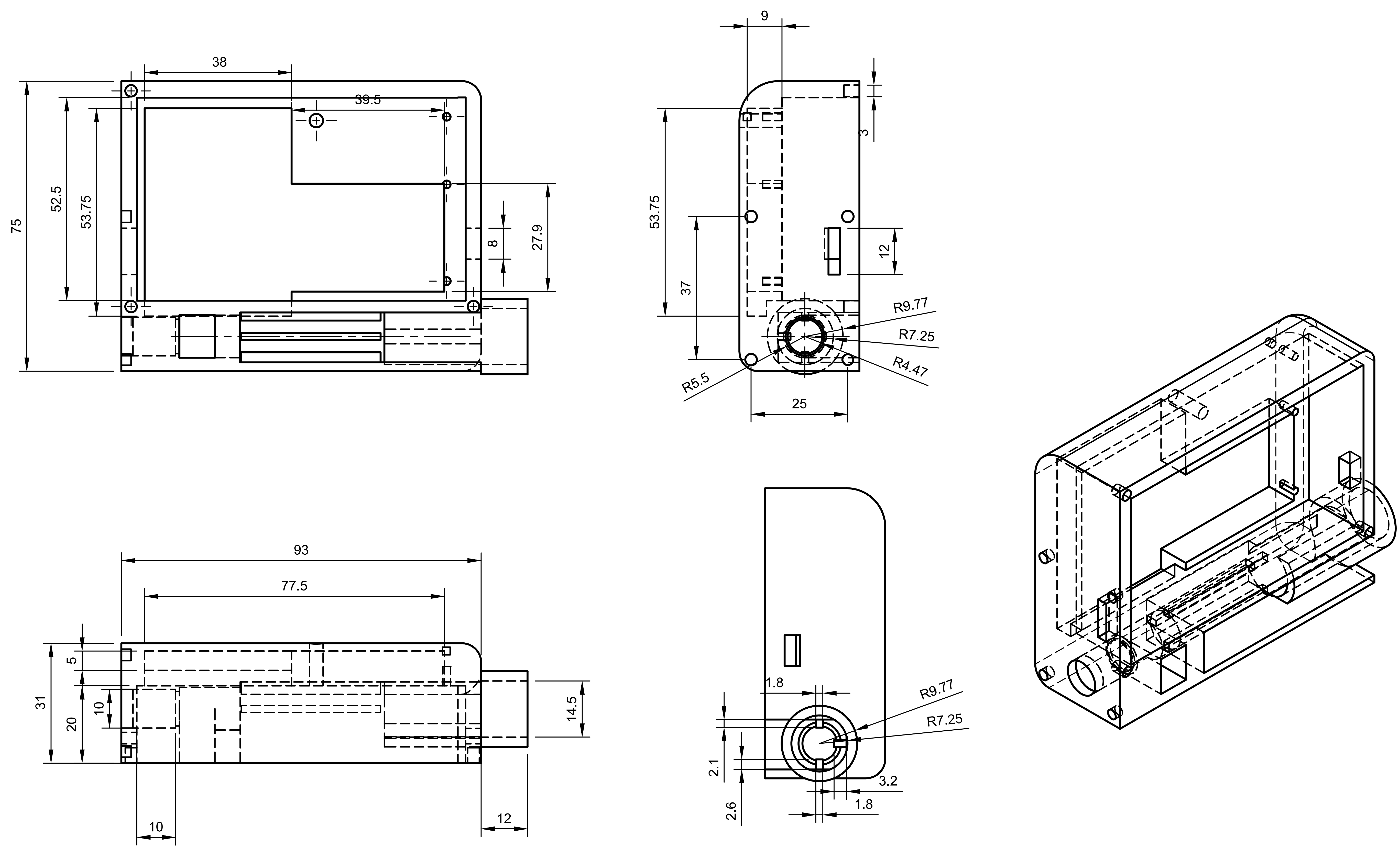




ANEXO D: Plano esquemático circuito principal del sistema de suministro

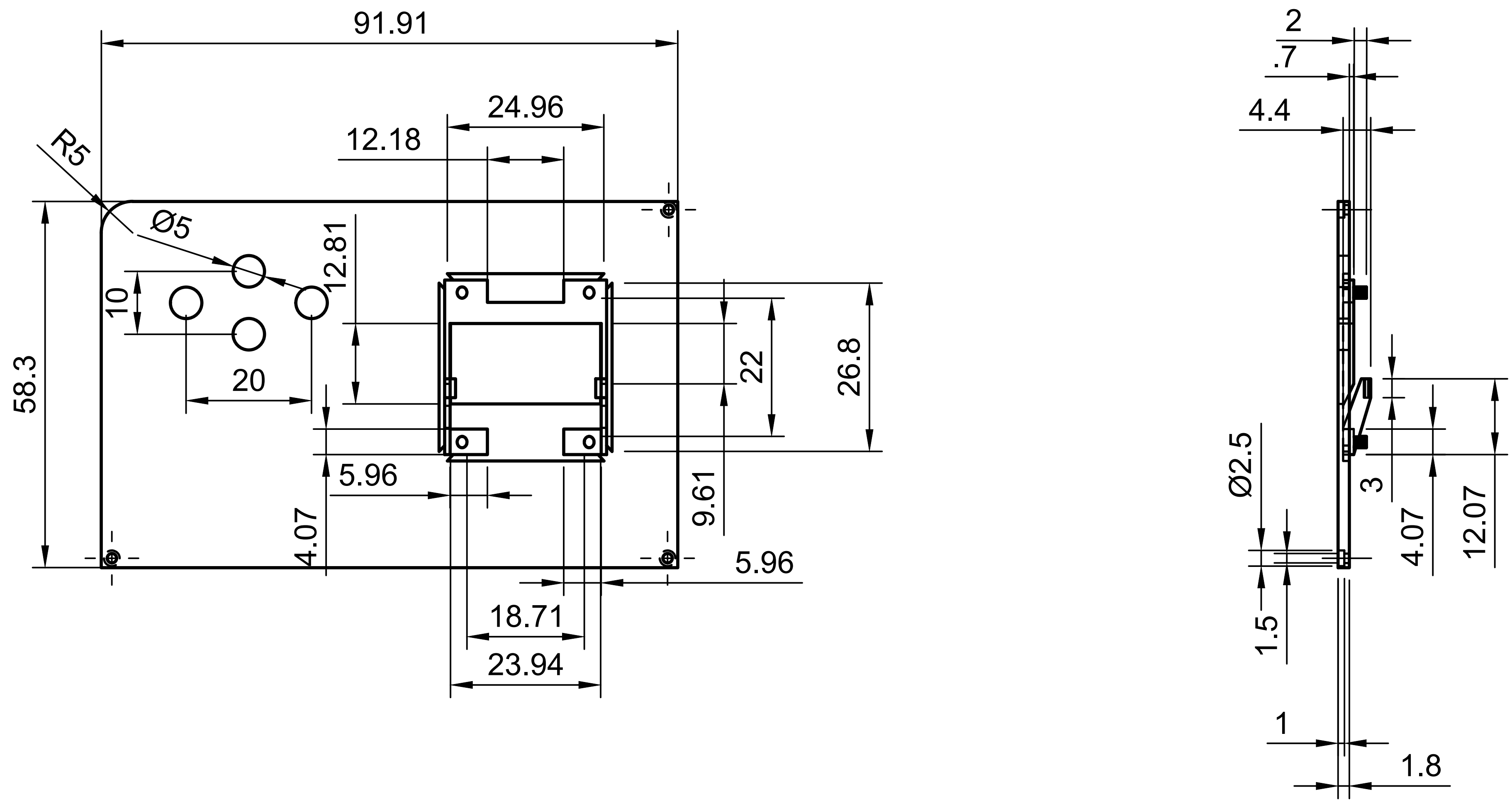

IC1
1 1-2EN
2 1A
3 1Y
4 GND1
5 GND2
6 2Y
7 2A
8 VCC2
16 VCC1
15 4A
14 4Y
13 GND3
12 GND4
11 3Y
10 3A
9 3-4EN
L293D
IN1
OUT1
OUT2
IN2
VIN
IN4
OUT4
GND
OUT3
IN3
VIN
B
#
*
A
N$26
N$14
N$12
N$27
OUT+
OUT-
VIN
GND
BATERIA
R2 100k
VOLTM
R3 100k
GND
3V3
WAKE
N$33
R1 10k
GND
MOTOR
OUT4
OUT3
OUT2
OUT1
U$1
EN
VP
VN
D34
D35
D32
D33
D25
D26
D27
D14
D12
D13
GND
VIN
D23
D22
TX0
RX0
D21
D19
D18
D5
TX2
RX2
D4
D2
D15
GND
3V3
ESP32_DEVKIT
IN4
IN3
VOLTM
N$33
N$26
N$27
N$14
N$12
VIN
SCL
SDA
IN4
IN2
IN1
GND
3V3
OLED
SDA
SCL
VIN
DS3231
SCL
SDA
VIN
Dept. DIEE-USFQ
Supervisor Malena Loza
Creado por Mateo Vega-Estefano Gomez
Aprobado por Alberto Sánchez
Tipo de Documento Plano diseño Eléctrico
Proyecto Sistema de Suministro de Insulina
Titulo Esquemático parte digital
Cliente:
UNIVERSIDAD SAN FRANCISCO
Rev. 0
Fecha 9/5/2020
Hoja 1/1

ANEXO E: Plano caja principal



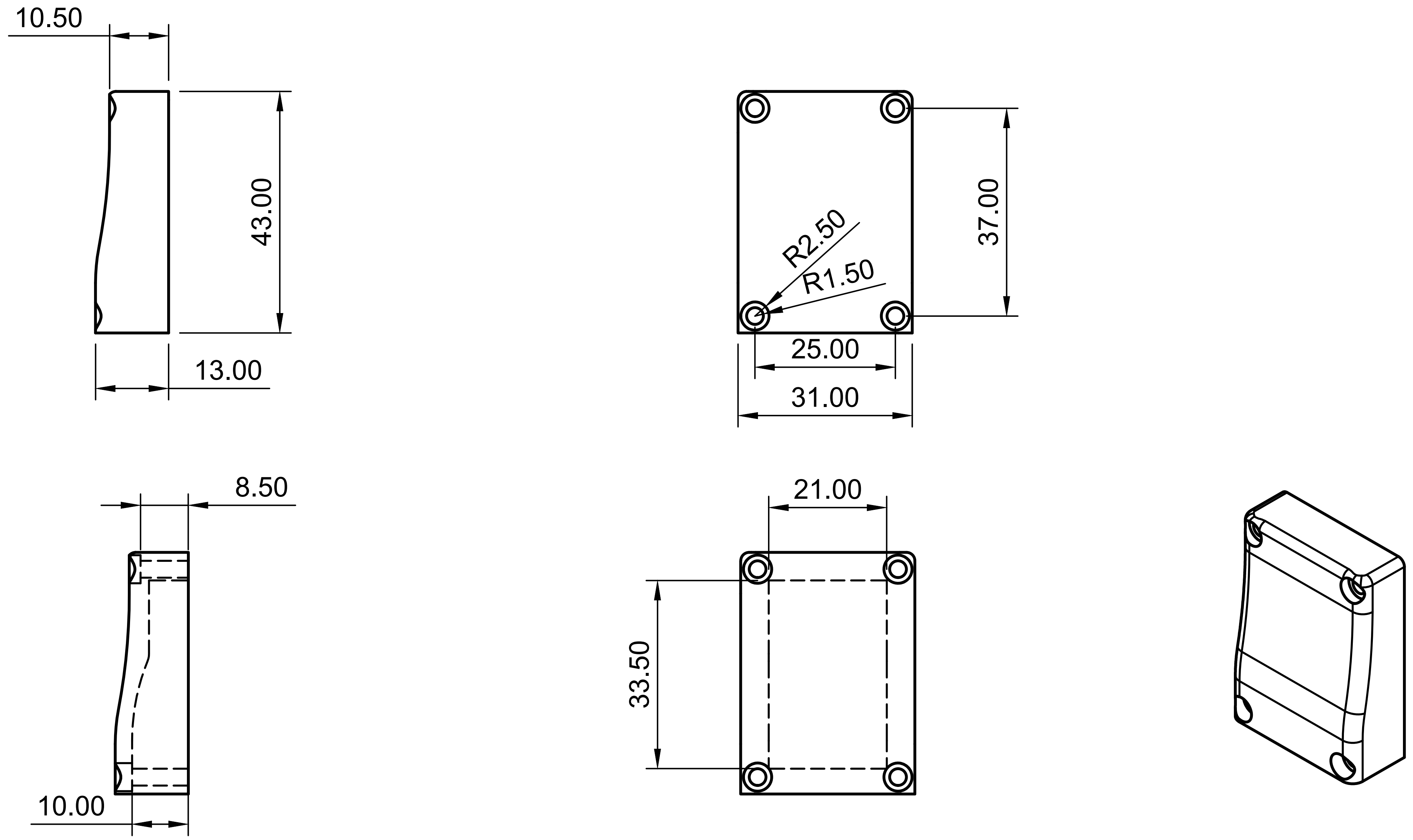
Dept.	Supervisor	Creado por	Aprobado por		
DIEE-USFQ	Malena Loza	Mateo Vega-Estefano Gomez	Alberto Sánchez		
Universidad San Francisco		Tipo de Documento: Plano Diseño Mecánico	Proyecto: Sistema de Suministro de Insulina		
		Título: Caja Principal	Cliente:		
			Rev.: 0	Fecha: 5/5/2020	Hoja: 1/6

ANEXO F: Plano cubierta de circuito



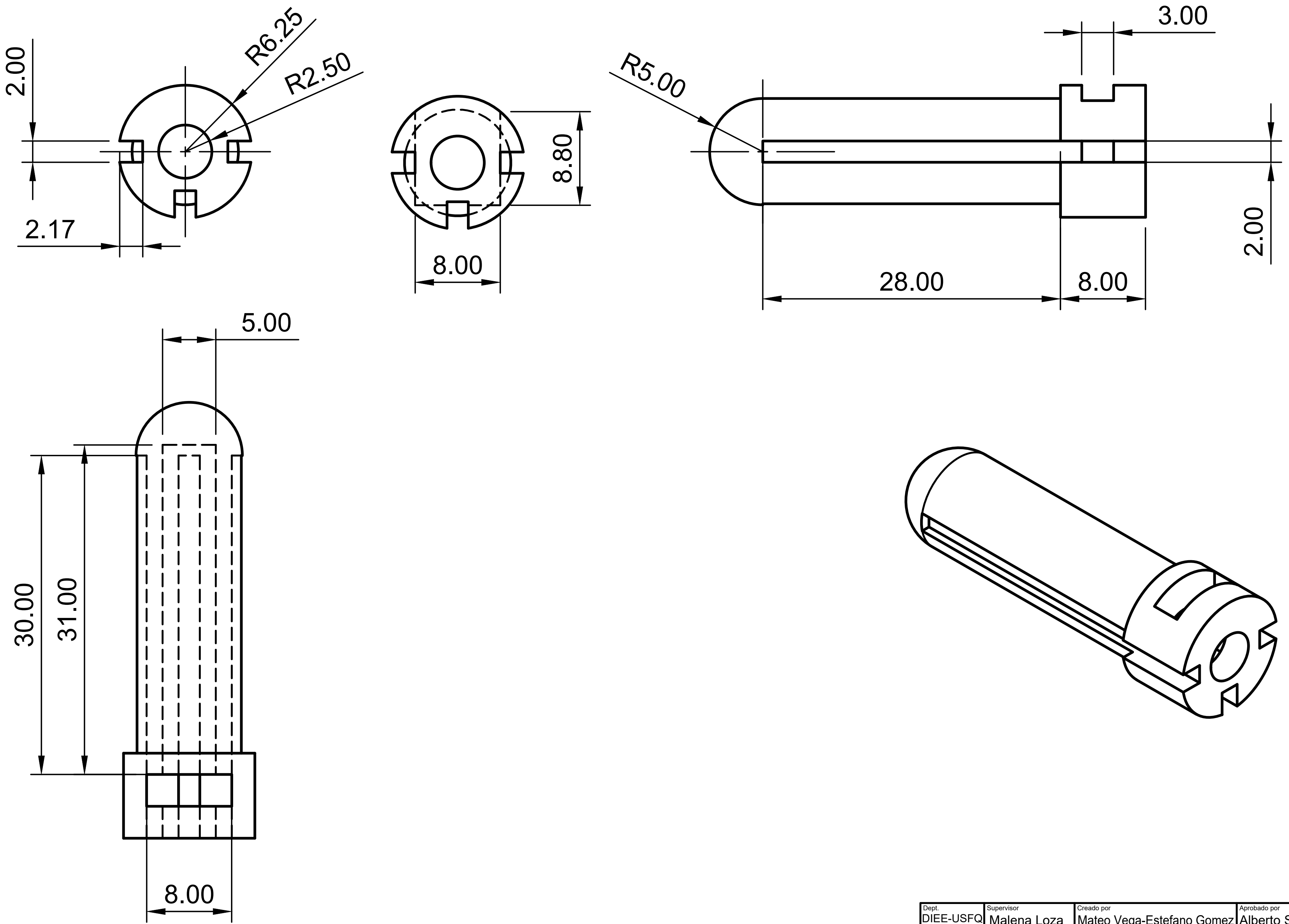
Dept.	Supervisor	Creado por	Aprobado por
DIEE-USFQ	Malena Loza	Mateo Vega-Estefano Gomez	Alberto Sánchez
Universidad San Francisco		Tipo de Documento: Plano Diseño Mecánico	Proyecto: Sistema de Suministro de Insulina
		Título: Cubierta de Circuito	Cliente:
			Rev.: 0 · Fecha: 5/5/2020 · Hoja: 2/6

ANEXO G: Plano caja protectora de motor



Dept. DIEE-USFQ	Supervisor Malena Loza	Creado por Mateo Vega-Estefano Gomez	Aprobado por Alberto Sánchez
Universidad San Francisco		Tipo de Documento Plano Diseño Mecánico	Proyecto Sistema de Suministro de Insulina
		Título Caja protectora de Motor	Cliente:
			Rev. 0 · Fecha 5/5/2020 · Hoja 3/6

ANEXO H: Plano émbolo



Dept. DIEE-USFQ

Supervisor: Malena Loza

Creado por: Mateo Vega-Estefano Gomez

Aprobado por: Alberto Sánchez

Universidad San Francisco

Tipo de Documento: Plano Diseño Mecánico

Proyecto: Sistema de Suministro de Insulina

Título: Émbolo

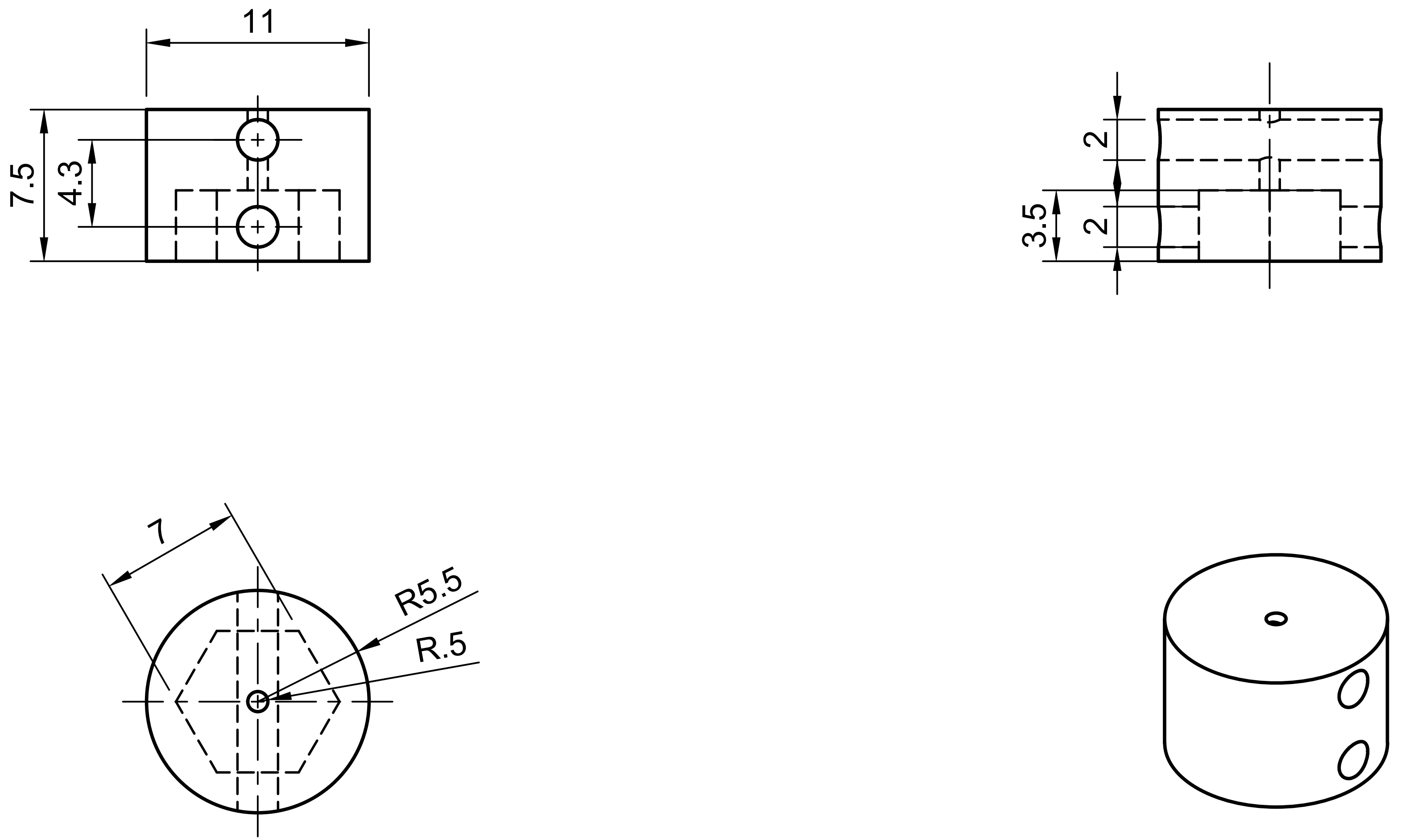
Cliente:

Rev. 0

Fecha: 5/5/2020

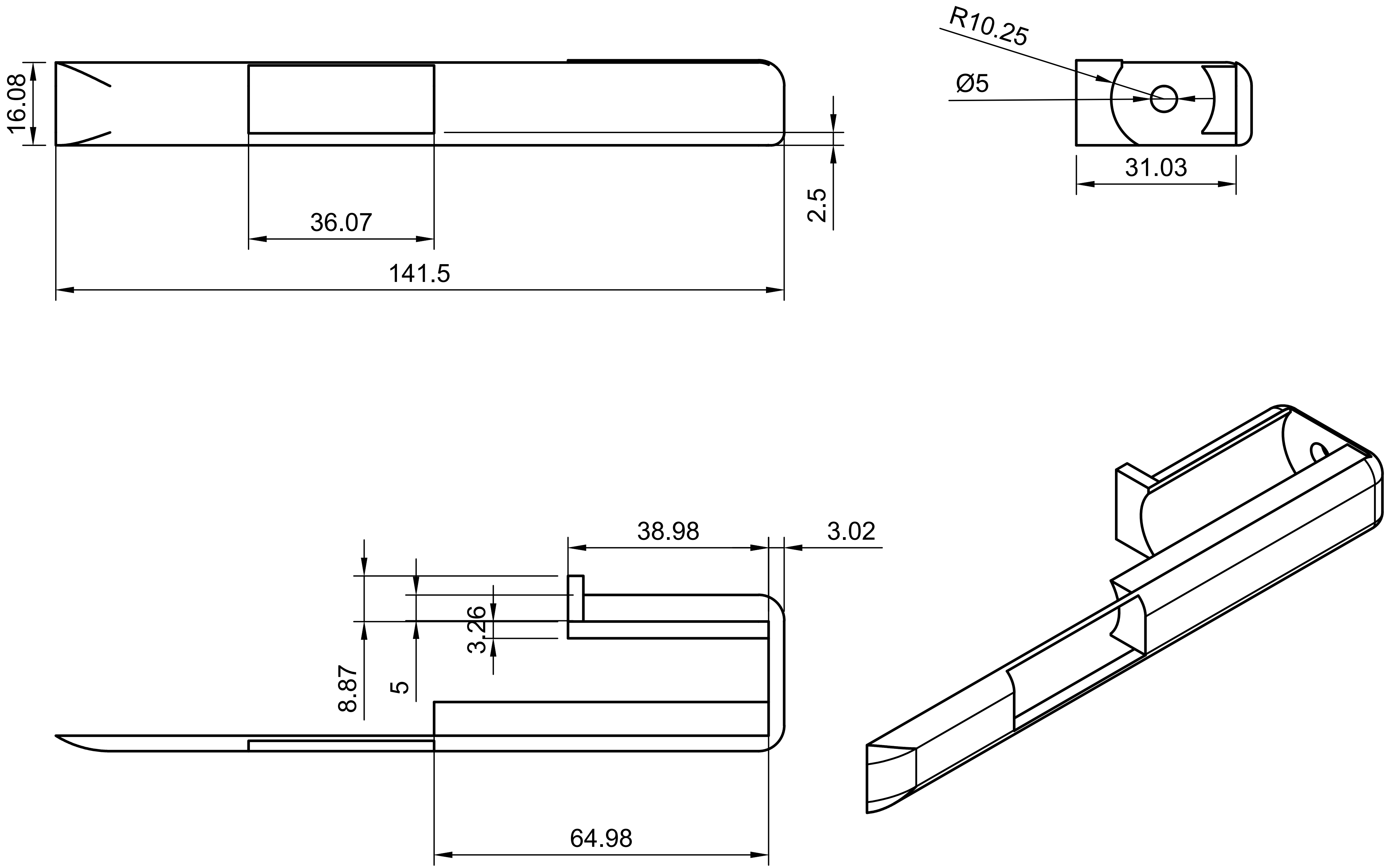
Hoja: 4/6

ANEXO I: Plano acople de eje



Dept.	Supervisor	Creado por	Aprobado por
DIEE-USFQ	Malena Loza	Mateo Vega-Estefano Gomez	Alberto Sánchez
Universidad San Francisco		Tipo de Documento: Plano Diseño Mecánico	Proyecto: Sistema de Suministro de Insulina
		Título: Acople de Eje	Cliente:
			Rev.: 0 · Fecha: 5/5/2020 · Hoja: 5/6

ANEXO J: Plano seguro de Infusion Set



Dept.	Supervisor	Creado por	Aprobado por
DIEE-USFQ	Malena Loza	Mateo Vega-Estefano Gomez	Alberto Sánchez
		Tipo de Documento: Plano Diseño Mecánico	Proyecto: Sistema de Suministro de Insulina
Universidad San Francisco		Título: Seguro de Infusion Set	Cliente:
			Rev.: 0 · Fecha: 5/5/2020 · Hoja: 6/6

ANEXO K: Datasheets

Stepper Motors

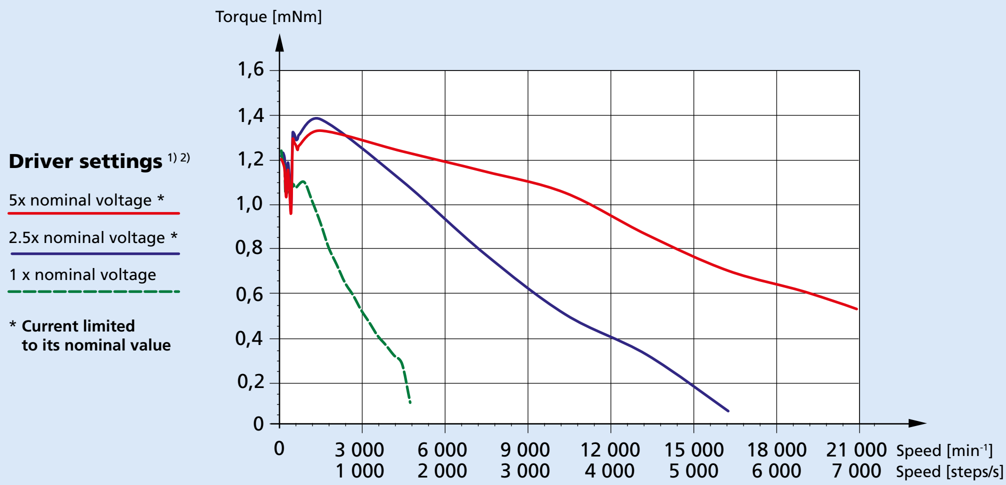
1,6 mNm

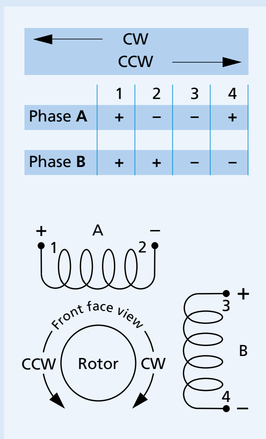
Two phase, 20 steps per revolution
PRECIstep® Technology

AM1020-ww-ee

	ww =	A-0,25-8		V-3-16		V-6-65		V-12-250		
		Current	Voltage	Current	Voltage	Current	Voltage	Current	Voltage	Drive mode
1	Nominal current per phase (both phases ON) [1]	0,25	–	0,18	–	0,09	–	0,045	–	A
2	Nominal voltage per phase (both phases ON) [1]	–	2	–	3	–	6	–	12	V DC
3	Phase resistance (at 20°C)	8		16		65		250		Ω
4	Phase inductance (1kHz)	2,4		5,2		21,4		80,1		mH
5	Back-EMF amplitude	1,8		2,6		5,3		10,5		V/k step/s
6	Holding torque (at nominal current in both phases)	1,6								mNm
7	Holding torque (at twice the nominal current)	2,4								mNm
8	Step angle (full step)	18								degree
9	Angular accuracy [1]	± 10								% of full step
10	Residual torque, max.	0,20								mNm
11	Rotor inertia	9								$\cdot 10^{-9}$ kgm²
12	Resonance frequency (at no load)	140								Hz
13	Electrical time constant	0,32								ms
14	Ambient temperature range	–35 ... +70								°C
15	Winding temperature tolerated, max.	130								°C
16	Thermal resistance R_{th1} / R_{th2}	3,9 / 53,8								°C/W
17	Thermal time constant τ_{w1} / τ_{w2}	4,5 / 200								s
18	Shaft bearings	sintered sleeve bearings (standard)				ball bearings, preloaded (optional)				
19	Shaft load, max.:									
	– radial (3 mm from bearing)	0,3				4,0				N
	– axial	0,3				3,0				N
20	Shaft play, max.:									
	– radial (0,2N)	15				12				µm
	– axial (0,2N)	150				~0				µm
21	Mass	5,5								g

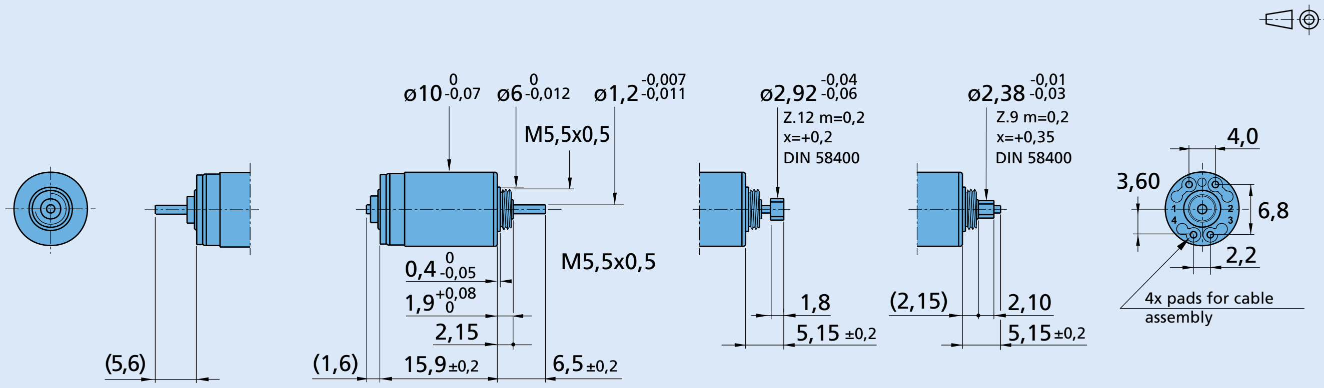
[1] Relevant for 2 phases ON only. On PWM drivers or chopper (current mode), the current is set to the nominal value and the supply voltage is typically 3 to 5x higher than the nominal voltage.
[2] Curves measured with a load inertia of 6 $\cdot 10^{-9}$ kgm², in half-step mode for the "1 x nominal voltage" curve, in 1/4 micro-stepping mode for the other curves.





FAULHABER

Dimensional drawing

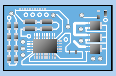
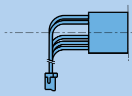
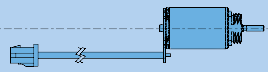
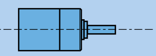


AM1020

for Gearhead 10/1

for Gearhead 12/3, 12/5

Combinations

Drive Electronics	Encoders	Cables	Gearheads / Lead screws
			
MCST3601	Available on request	List available on request	**10/1**
			12/3
			12/5*
			Lead screws **M1,2 M1,6**
			Lead screws **M2 - M3**

* Zero Backlash Gearheads

Ordering information

Example: **AM1020-2R-V-3-16-08**

Motor type	Bearings (rr)	Winding (ww)	Motor execution (ee)		
AM = Motor design 10 = Motor diameter (mm) 20 = Steps per revolution	Special lubricant options available		Only front output shaft	With double output shaft	Front output shaft
AM1020	**-** (sleeve bearings)	**-V-3-16**	**-01**	**-00**	Plain shaft
	-2R (2 ball bearings)	**-V-6-65**	**-08**	**-09**	Pinion 10/1
		-V-12-250	**-10**	**-11**	Pinion 12/5
		-A-0,25-8		**-12**	Plain shaft, Rear = 3,7mm for encoder
				-13	Pinion 10/1, Rear = 3,7mm for encoder
				-14	Pinion 12/5, Rear = 3,7mm for encoder
			-21	**-20**	Plain shaft for lead screw M1,2
			-23	**-22**	Plain shaft for lead screw M2 - M3
			-25	**-24**	Plain shaft for lead screw M1,6

Product Specifications

Type : Polymer Li-ion Recharged Battery

Model : DTP603450(PHR)

Specification : **3.7V/1000mAh**

Prepared By/Date	Checked By/Date	Approved By/Date
Yang qin 2016/09/29		

Customer confirmation:

Sign/Date:

Tel: +86-755 23460581 Fax: +86-755 23460503

Http://www.dtpbattery.com E-mail: info@dtpbattery.com

Revise the history

Revision Num	Date	Revise the items
01	2016-09-29	Publishes for the first time-----------

Contents

1. Scope

This specification shall be applied to the batteries from Data Power Technology Limited's product.

2. Product Type and Product Model

2.1 Type: Polymer Li-ion Recharged Battery

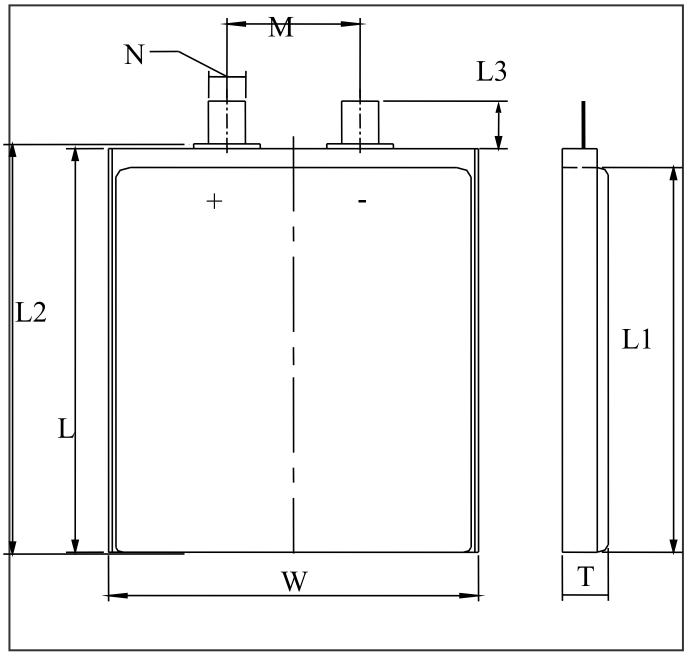
2.2 Model: DTP603450(PHR)

3. Product Basic Characteristics

No	Item	Characteristics
3.1	Rated Capacity	1000mAh
3.2	Minimum Capacity	1000mAh
3.3	Nominal Voltage	3.70V
3.4	Charge Limited Voltage	4.20V
3.5	Discharge Cut-off Voltage	2.40V
3.6	End-of-charge Current	0.01C
3.7	Standard Charge	Charge with 0.2C(200mA) up to Limited Voltage , Charge with limited Voltage up to end-of-charge current.
3.8	Standard Discharge	Using 3C(3000mA) constant current discharge to the Discharge Cut-off Voltage.
3.9	Maximum Continuous Charge Current	1C (1000mA)
3.10	Maximum Continuous Discharge Current	3C (3000mA)
3.11	Operating Temperature Range	Charge 0 ~ 45℃
		Discharge – 20 ~ 60℃
	Storage Temperature Range	-20 ~ 60℃
3.12	Operating And Storage Humidity Range	65±20% RH
3.13	Weight	Less than 20g

4. Cell Dimension

Item	Dimension（mm）
T	Max 6.0
W	Max 34
L	Max 50
L1	Max 46
L2	Max 50.3
L3	8.0±2.0
M	18.0±4.0
N	2.0±0.5



5.Appearance

It shall be free from any defects such as remarkable scratches, breaks, cracks, discoloration, leakage, or middle deformation

6. Basic Electrical Characteristics

No.	Items	Criteria	Test Method
6.1	Open Circuit Voltage	3.75V～3.95V	Measure with voltmeter.
6.2	Internal Impedance	≤150mΩ	Measure cells using an alternate current impedance meter at 1kHz .
6.3	Rated Capacity (0.2C_5A)	≥1000mAh	Discharged after the standard charged cells rest 10min at 23±2℃， Test can be discontinued when more than Rated capacity. Three cycles are permitted..
6.4	1C_5A.discharge capacity	≥1000×90%	Discharged after the standard charged cells rest 10min at 23±2℃ ， Test can be discontinued when more than 90%*rated capacity. Three cycles are permitted.
6.5	Temperature Characteristics	1. Appearance： No deformation、ruptures nor leakage。 2.Discharge Capacity： 55℃:≥85%×initial capacity; -10℃≥70%×initial capacity	Measured the 0.2C5A capacity at 23±2℃ as the initial capacity. Stored the rechargeable batteries for 16-20hrs at -10 ± 2℃; 2h for 55 ± 2℃, and then 0.2C5A discharged at this temperature, Checked the batteries' appearance after rest for 2 hrs at room temperature.
6.6	Storage Characteristics	Retention Capacity: ≥85% ×initial capacity	Measured the0.2C_5A capacity at (20±5)℃ as the initial capacity. Stored the recharged cells for 6 days at 20 ± 5℃ and then rest for 2 hrs at room temperature, 0.2C_5A discharged after checked the cells' appearance.
6.7	Cycle Life (20℃)	Capacity≥initial capacity× 80%	0.5C discharged after 0.5C_5A full charges at 20± 5℃.Carry out 300 cycles

Remark 1 Standard charge： 0.2C_5A charge up to charge limited voltage at (20±5)℃. Charge with limited voltage up to end of current. It is the same to the next content

7.Safety Characteristics

No.	Items	Criteria	Test Method
7.1	Overcharge Characteristics	Appearance: No rupture, fire, smoke, nor leakage.	When the battery is fully charged, go on loading for 8h with a twice rating voltage, 2.0C**5**A out put current, it starts the over charge protection function.

7.2	Over-discharge Characteristics	Appearance: No rupture, fire, smoke, nor leakage.	The battery is discharged at 0.2C5A in the constant current till it reaches over discharge protection voltage at（20±5）℃, connected with a 30Ω lead and discharged for 24h
7.3	Short-circuit Characteristics	OCV ≥3.6V; Appearance: No rupture, fire, smoke, nor leakage.	As the battery has completed charging, short circuit the positive and negative contacts with 0.1Ω resistor for 1h for appearance check, then disconnect the resistor between the contacts, the battery shall be charged at 1.0C5A mA in the constant current for 5S
7.4	Hot Oven Characteristics	Appearance:.No explode.No fire.	The battery is to be heated in a gravity convection or circulating air oven after standard charged at 23±2℃,The temperature of the oven is to be raised at a rate of 5±2℃ /min. The oven is to remain for 30 minutes at 400±2℃ before the test is discontinued.
7.5	Heavy Collision	Appearance:.No explode.No fire.	Putting the battery on the platform, using 10KG heavy hammer free drop from 1M height onto the fixed battery.

Remark 2 All safety characteristics are carried out by specialized personnel familiar with Li-ion knowledge or under instruction of our technical personnel after detailed consultation.

8.Reliability Characteristics

No.	Items	Criteria	Test Method
8.1	Static Humidity and Temperature Characteristics	Retention Capacity: ≥60%× initial capacity Appearance: No leakage, damage,smoke,ruputer.	Measured the 1C5A capacity at 23±2℃ as the initial capacity. Stored the rechargeable batteries for 2 days at 40 ± 2 ℃ and 90%-95%RH, then rest for 2 hrs at room temperature. 0.2C5A discharged after checked the batteries appearance. Measured recoverable 1C5A discharge capacity with 3 cycles..
8.2	Vibration Characteristics	OCV ≥3.6V; Appearance: No fire, leakage, explode, rupture	After fully charging, fixing the battery onto the vibration platform. with amplitude 0.38mm circularly scanning vibrating in the frequency of 10HZ-55HZ from three directions X、Y、Z for 30min respectively in its scanning frequency velocity 10CT/min.

8.3	Bump Characteristics	OCV ≥3.6V; Appearance: No fire, leakage, explode, rupture	After vibration testing, use a clip or directly fix the battery on to the platform in the direction of X、Y、Z vertical complementary axis, then adjust its acceleration and pulse duration as below to have a bump test. Pulse peak acceleration 100m/s2. Bumps per minute 40-80.Pulse duration 16ms. Bump times 1000±10.
8.4	Free Drop Characteristics	Retention Capacity: ≥85% ×nominal capacity. Appearance: No fire, leakage, explode, rupture	After bump testing, the battery shall be immediately dropped from the height of 1000mm （minimum height） onto a 18mm～20mm hard board on the cement floor. Free drop one time respectively from X、Y、Z positive and negative axis(six directions). After that, the battery is discharged at 1C**5**A to its final voltage.

9. Assembling Request

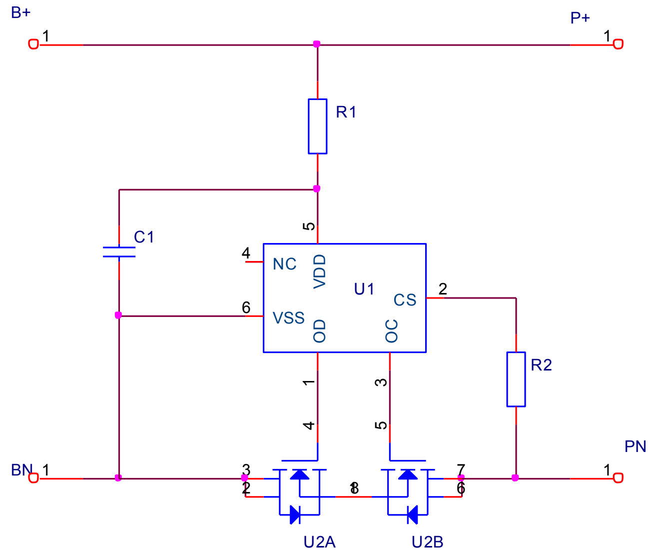
9.1 List of Parameter

Item	Symbol	Content	Criterion
Over charge Protection	V_{DET1}	Over charge detection voltage	4.200V±0.050V
	tV_{DET1}	Over charge detection delay time	80 ms
	V_{REL1}	Over charge release voltage	4.100±0.050V
Over discharge protection	V_{DET2}	Over discharge detection voltage	2.4V±0.100V
	tV_{DET2}	Over discharge detection delay time	20ms
	V_{REL2}	Over discharge release voltage	2.8V±0.100V
Over current protection	V_{DET3}	Over current detection voltage	0.150±0.030V
	I_{DP}	Over current detection current	2.5～4.5A
	tV_{DET3}	Detection delay time	10ms
		Release condition	Cut load
Short protection		Detection condition	Exterior short circuit
	T_{SHORT}	Detection delay time	≤5us
		Release condition	Cut short circuit
Interior resistance	R_{DS}	Main loop electrify resistance	VC=3.6V；RDS≤60mΩ

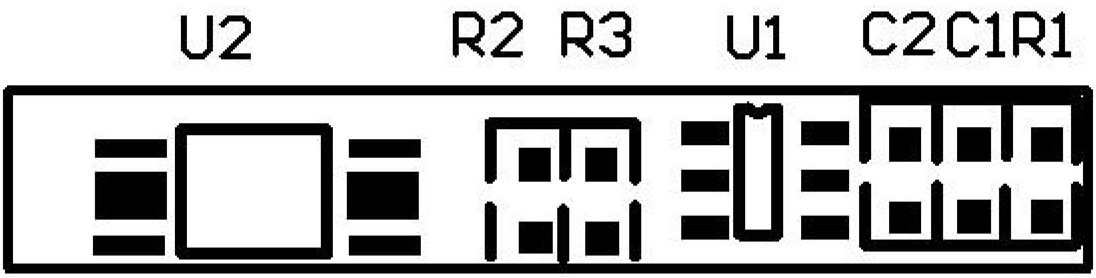
9.2 Parts list

NO.	Location	Part name	Specification	Pack type	Q’ ty	Maker/Remark
1	U1	Battery protection IC	DW01+	SOT23-6	1	Fortune
2	U2	Silicon MOSFET	8205	SOT-6	1	MT
3	R1	Resistance	SMD 100Ω ±5%	0603	1	YAGEO
4	R2	Resistance	SMD 1KΩ ±5%	0603	1	YAGEO

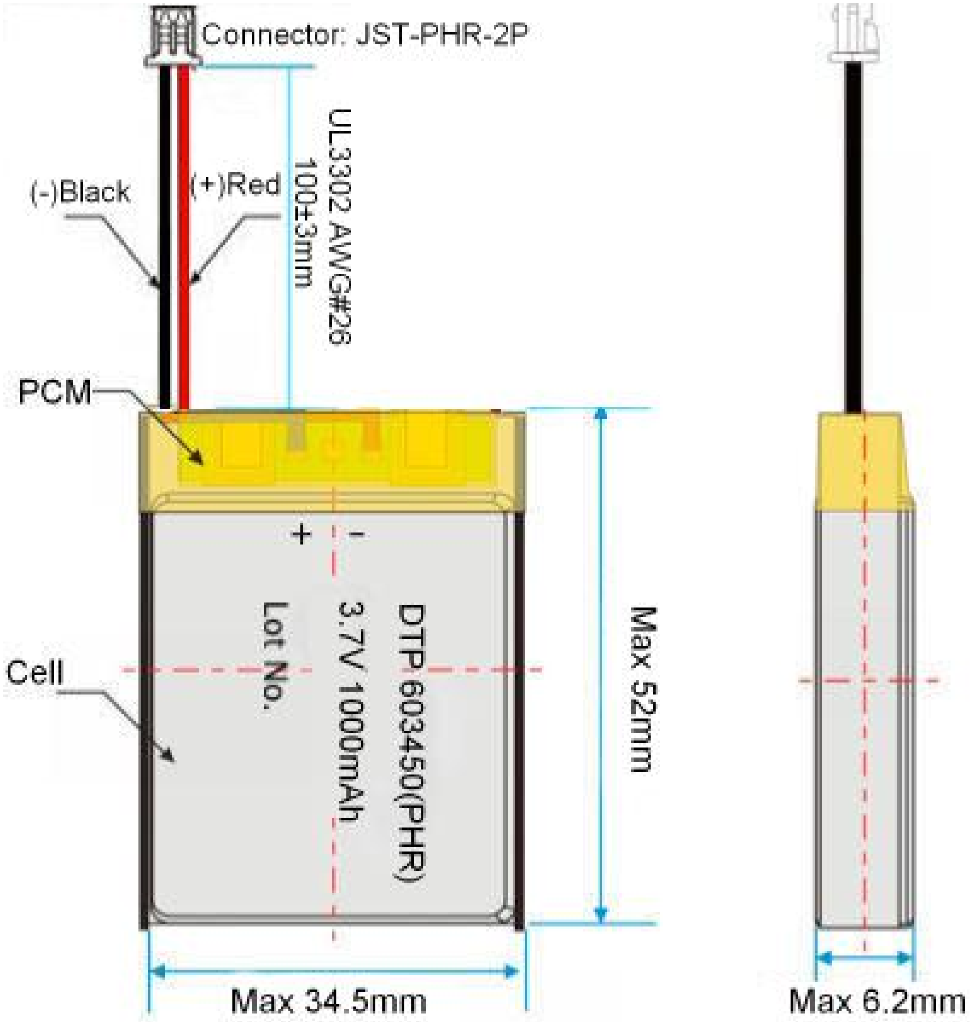
9.3 Application Circuit



9.4 Maps



9.5 External Dimension Drawing



10. Guarantee Period of Quality

Guarantee period of quality is 12 months after sold.

11. Matters needing attention

Strictly observes the following needing attention. Data Power will not be responsible for any accident occurred by handling outside of the precautions in this specification.

! Danger

- Strictly prohibits heat or throw cell into fire.
- Strictly prohibits throw and wet cell in liquid such as water、gasoline or drink etc.
- Strictly prohibits use leave cell close to fire or inside of a car where temperature may be above 60℃. Also do not charge / discharge in such conditions.
- Strictly prohibits put batteries in your pockets or a bag together with metal objects such as necklaces. Hairpins, coins, or screws. Do not store or transportation batteries with such objects.
- Strictly prohibits short circuit the (+) and (-) terminals with other metals.
- Do not place Cell in a device with the (+) and (-) in the wrong way around.
- Strictly prohibits pierce Cell with a sharp object such as a needle.
- Strictly prohibits disassemble or modify the cell.

- Strictly prohibits welding a cell directly.
- Do not use a Cell with serious scar or deformation.
- Thoroughly read the user's manual before use, inaccurate handling of lithium ion rechargeable cell may cause leakage, heat, smoke, an explosion, or fire, capacity decreasing.

! Warning

- Strictly prohibits put cell into a microware oven, dryer, or high-pressure container.
- Strictly prohibits use cell with dry cells and other primary batteries, or new and old battery or batteries of a different package, type, or brand.
- Stop charging the Cell if charging is not completed within the specified time.
- Stop using the Cell if abnormal heat, odor, discoloration, deformation or abnormal condition is detected during use, charge, or storage.
- Keep away from fire immediately when leakage or foul odor is detected.
- If liquid leaks onto your skin or clothes, wash well with fresh water immediately.
- If liquid leaking from the Cell gets into your eyes, do not rub your eyes. Wash them well with clean edible oil and go to see a doctor immediately.

! Caution

- Before using the Cell, be sure to read the user's manual and cautions on handling thoroughly.
- Charging with specific charger according to product specification. Charge with CC/CV method. Strictly prohibits revered charging. Connect cell reverse will not charge the cel. At the same time, it will reduce the charge-discharge characteristics and safety characteristics, this will lead to product heat and leakage.
- Store batteries out of reach of children so that they are not accidentally swallowed.
- If younger children use the Cell, their guardians should explain the proper handling.
- Before using the Cell, be sure to read the user's manual and cautions on handling thoroughly.
- Batteries have life cycles. If the time that the Cell powers equipment becomes much shorter than usual, the Cell life is at an end. Replace the Cell with a new same one.
- When not using Cell for an extended period, remove it from the equipment and store in a place with low humidity and low temperature.
- While the Cell pack is charged, used and stored, keep it away from objects or materials with static electric charges
- If the terminals of the Cell become dirty, wipe with a dry clothe before using the Cell.
- Storage the cells in storage temperature range as the specifications, Afer full discharged, we suggest that charging to 3.9~4.0V with no using for a long time.
- Do not exceed these ranges of the following temperature ranges.

Charge temperature range : 0 ℃ to 45 ℃ ; Discharge temperature range : -20 ℃ to 60 ℃ .(When using equipment)

11. Statement

If our specifications material, product process or product control system has changed, the information will be transmitted to consumer by way of written with quality and reliability data.

ESP32 Series
Datasheet

Including:

ESP32-D0WD-V3

ESP32-D0WDQ6-V3

ESP32-D0WD

ESP32-D0WDQ6

ESP32-D2WD

ESP32-S0WD

ESP32-U4WDH

Version 3.4
Espressif Systems

About This Guide

This document provides the specifications of ESP32 family of chips.

Document Updates

Please always refer to the latest version on https://www.espressif.com/en/support/download/documents.

Revision History

For any changes to this document over time, please refer to the last page.

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe at www.espressif.com/en/subscribe. Note that you need to update your subscription to receive notifications of new products you are not currently subscribed to.

Certification

Download certificates for Espressif products from www.espressif.com/en/certificates.

Disclaimer and Copyright Notice

Contents

List of Tables

List of Figures

1. Overview

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

The ESP32 series of chips includes ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32-D0WDQ6, ESP32-D2WD, ESP32-S0WD, and ESP32-U4WDH, among which, ESP32-D0WD-V3, ESP32-D0WDQ6-V3, and ESP32-U4WDH are based on ECO V3 wafer.

For details on part numbers and ordering information, please refer to Section 7.

For details on ECO V3 instructions, please refer to *ESP32 ECO V3 User Guide*.

1.1 Featured Solutions

1.1.1 Ultra-Low-Power Solution

ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

Note:
For more information, refer to Section 3.7 *RTC and Low-Power Management*.

1.1.2 Complete Integration Solution

ESP32 is a highly-integrated solution for Wi-Fi-and-Bluetooth IoT applications, with around 20 external components. ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area.

ESP32 uses CMOS for single-chip fully-integrated radio and baseband, while also integrating advanced calibration circuitries that allow the solution to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi testing equipment.

1.2 Wi-Fi Key Features

- 802.11 b/g/n
- 802.11 n (2.4 GHz), up to 150 Mbps
- WMM
- TX/RX A-MPDU, RX A-MSDU

- Immediate Block ACK
- Defragmentation
- Automatic Beacon monitoring (hardware TSF)
- 4 × virtual Wi-Fi interfaces
- Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes
  Note that when ESP32 is in Station mode, performing a scan, the SoftAP channel will be changed.
- Antenna diversity

Note:
For more information, please refer to Section 3.5 *Wi-Fi*.

1.3 BT Key Features

- Compliant with Bluetooth v4.2 BR/EDR and BLE specifications
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced Power Control
- +12 dBm transmitting power
- NZIF receiver with –94 dBm BLE sensitivity
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- Bluetooth 4.2 BR/EDR BLE dual mode controller
- Synchronous Connection-Oriented/Extended (SCO/eSCO)
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet
- Multi-connections in Classic BT and BLE
- Simultaneous advertising and scanning

1.4 MCU and Advanced Features

1.4.1 CPU and Memory

- Xtensa® single-/dual-core 32-bit LX6 microprocessor(s), up to 600 MIPS (200 MIPS for ESP32-S0WD/ESP32-U4WDH, 400 MIPS for ESP32-D2WD)
- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC
- QSPI supports multiple flash/SRAM chips

1.4.2 Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz ~ 60 MHz crystal oscillator (40 MHz only for Wi-Fi/BT functionality)
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 × 64-bit timers and 1 × main watchdog in each group
- One RTC timer
- RTC watchdog

1.4.3 Advanced Peripheral Interfaces

- 34 × programmable GPIOs
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DAC
- 10 × touch sensors
- 4 × SPI
- 2 × I²S
- 2 × I²C
- 3 × UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN 2.0
- IR (TX/RX)
- Motor PWM
- LED PWM up to 16 channels
- Hall sensor

1.4.4 Security

- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration:
  - AES
  - Hash (SHA-2)
  - RSA

- ECC
- Random Number Generator (RNG)

1.5 Applications (A Non-exhaustive List)

- Generic Low-power IoT Sensor Hub
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- Over-the-top (OTT) Devices
- Speech Recognition
- Image Recognition
- Mesh Network
- Home Automation
  - Light control
  - Smart plugs
  - Smart door locks
- Smart Building
  - Smart lighting
  - Energy monitoring
- Industrial Automation
  - Industrial wireless control
  - Industrial robotics
- Smart Agriculture
  - Smart greenhouses
  - Smart irrigation
  - Agriculture robotics
- Audio Applications
  - Internet music players
  - Live streaming devices
  - Internet radio players
  - Audio headsets
- Health Care Applications
  - Health monitoring
  - Baby monitors
- Wi-Fi-enabled Toys
  - Remote control toys
  - Proximity sensing toys
  - Educational toys
- Wearable Electronics
  - Smart watches
  - Smart bracelets
- Retail & Catering Applications
  - POS machines
  - Service robots

1.6 Block Diagram

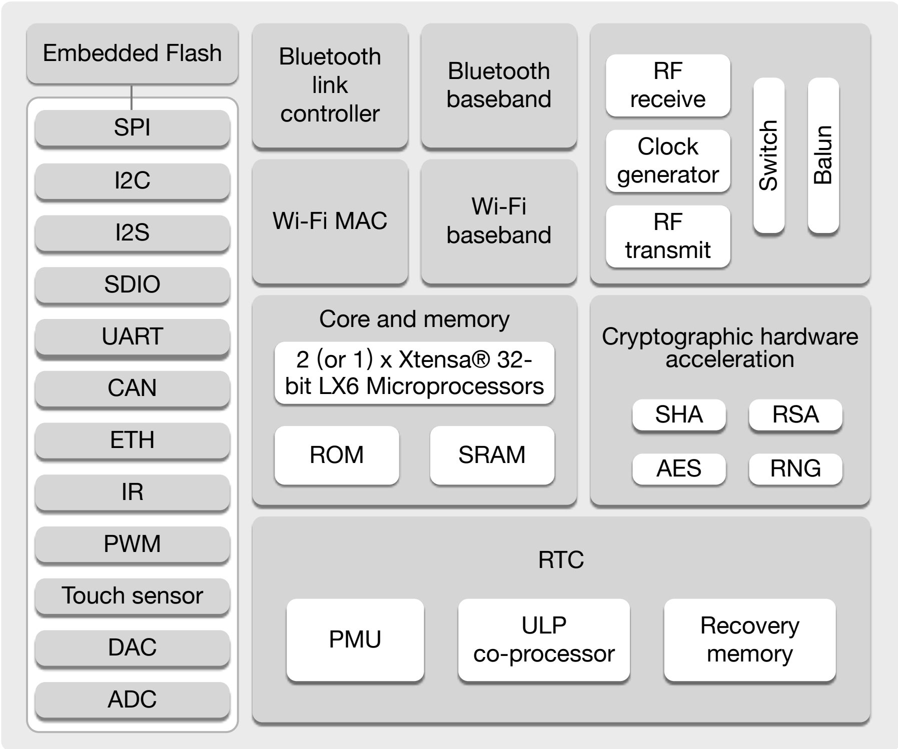


Figure 1: Functional Block Diagram

> **Note:**
> Products in the ESP32 series differ from each other in terms of their support for embedded flash and the number of CPUs they have. For details, please refer to Section 7 *Part Number and Ordering Information*.

2. Pin Definitions

2.1 Pin Layout

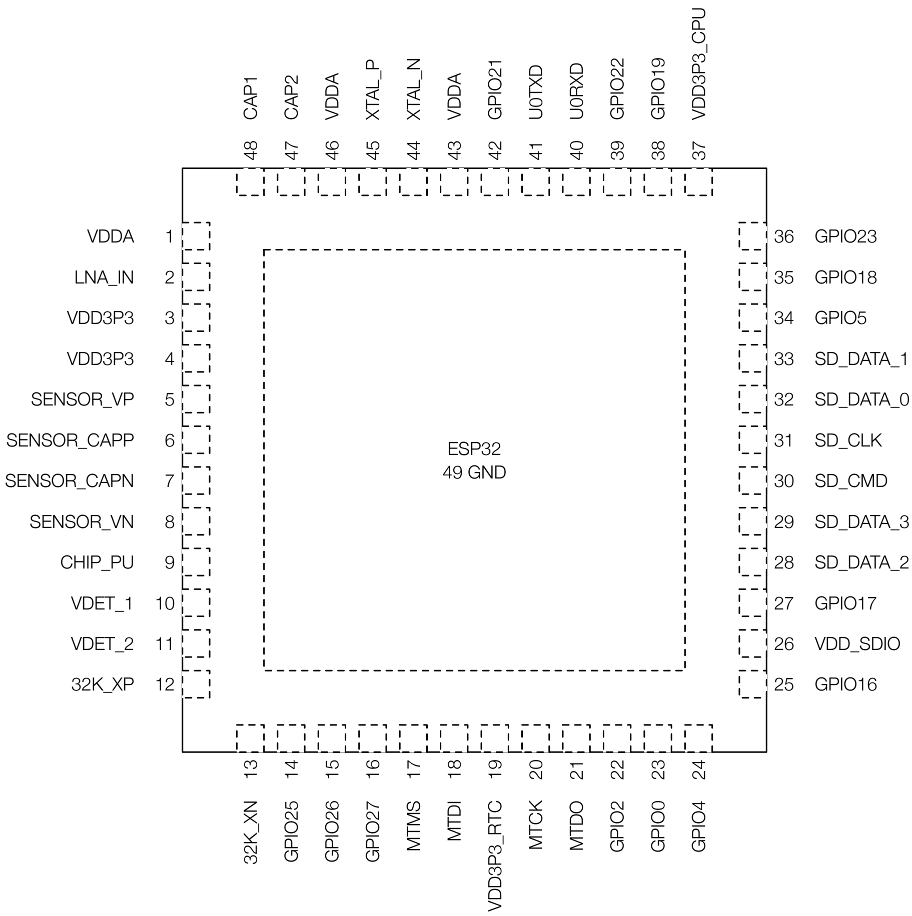


Figure 2: ESP32 Pin Layout (QFN 6*6, Top View)

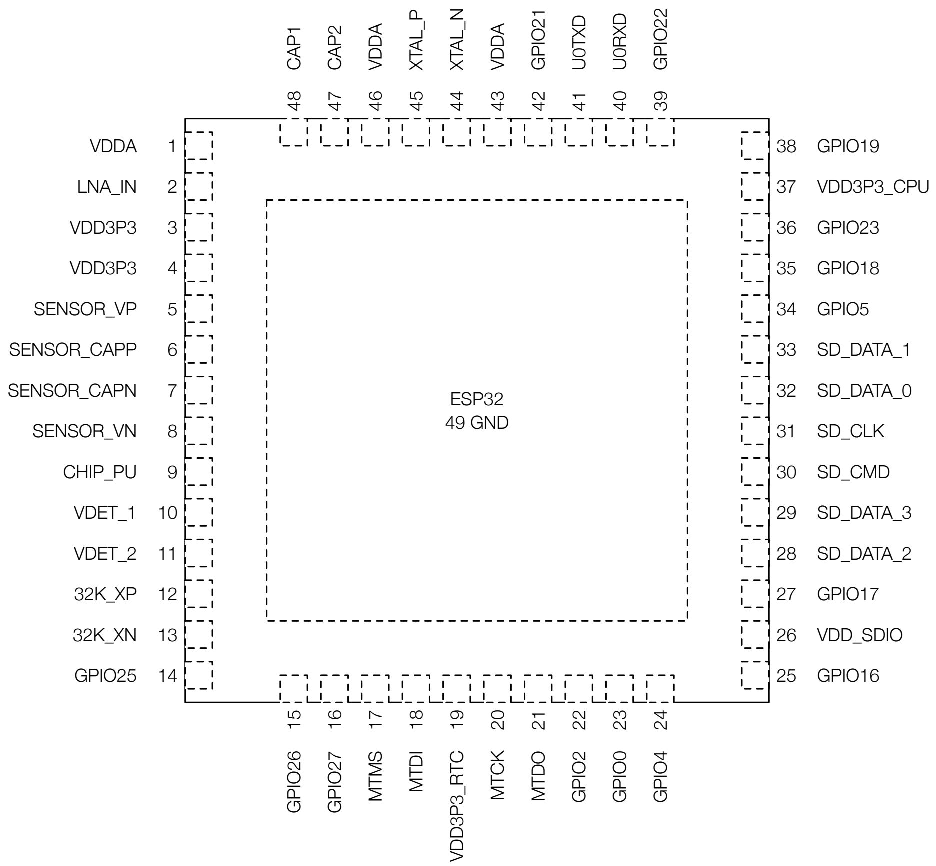


Figure 3: ESP32 Pin Layout (QFN 5*5, Top View)

Note:
For details on ESP32's part numbers and the corresponding packaging, please refer to Section 7 *Part Number and Ordering Information*.

2.2 Pin Description

Table 1: Pin Description

Name	No.	Type	Function
			Analog
VDDA	1	P	Analog power supply (2.3 V ~ 3.6 V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3	4	P	Analog power supply (2.3 V ~ 3.6 V)
			VDD3P3_RTC
SENSOR_VP	5	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_CAPP	6	I	GPIO37, ADC1_CH1, RTC_GPIO1
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, RTC_GPIO2
SENSOR_VN	8	I	GPIO39, ADC1_CH3, RTC_GPIO3
CHIP_PU	9	I	High: On; enables the chip Low: Off; the chip powers off Note: Do not leave the CHIP_PU pin floating.
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	I/O	GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	I/O	GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0
GPIO26	15	I/O	GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS
MTDI	18	I/O	GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIQ, HS2_DATA2, SD_DATA2, MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V ~ 3.6 V)
MTCK	20	I/O	GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK
MTDO	21	I/O	GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPICS0, HS2_CMD, SD_CMD, MTDO

Name	No.	Type	Function
GPIO2	22	I/O	GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, RTC_GPIO11, TOUCH1, EMAC_TX_CLK, CLK_OUT1,
GPIO4	24	I/O	GPIO4, ADC2_CH0, RTC_GPIO10, TOUCH0, EMAC_TX_ER, HSPIHD, HS2_DATA1, SD_DATA1
VDD_SDIO			
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	Output power supply: 1.8 V or the same voltage as VDD3P3_RTC
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPIHD
SD_DATA_3	29	I/O	GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPIWP
SD_CMD	30	I/O	GPIO11, HS1_CMD, U1RTS, SD_CMD, SPICS0
SD_CLK	31	I/O	GPIO6, HS1_CLK, U1CTS, SD_CLK, SPICLK
SD_DATA_0	32	I/O	GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPIQ
SD_DATA_1	33	I/O	GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPID
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, HS1_DATA6, VSPICS0, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, HS1_DATA7, VSPICLK
GPIO23	36	I/O	GPIO23, HS1_STROBE, VSPID
VDD3P3_CPU	37	P	Input power supply for CPU IO (1.8 V ~ 3.6 V)
GPIO19	38	I/O	GPIO19, U0CTS, VSPIQ, EMAC_TXD0
GPIO22	39	I/O	GPIO22, U0RTS, VSPIWP, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
Analog			
VDDA	43	P	Analog power supply (2.3 V ~ 3.6 V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Analog power supply (2.3 V ~ 3.6 V)
CAP2	47	I	Connects to a 3 nF capacitor and 20 kΩ resistor in parallel to CAP1

Name	No.	Type	Function
CAP1	48	I	Connects to a 10 nF series capacitor to ground
GND	49	P	Ground

Note:

- The pin-pin mapping between ESP32-D2WD/ESP32-U4WDH and the embedded flash is as follows: GPIO16 = CS#, GPIO17 = IO1/DO, SD_CMD = IO3/HOLD#, SD_CLK = CLK, SD_DATA_0 = IO2/WP#, SD_DATA_1 = IO0/DI. The pins used for embedded flash are not recommended for other uses.
- In most cases, the data port connection between ESP32 series of chips other than ESP32-D2WD/ESP32-U4WDH and external flash is as follows: SD_DATA0/SPIQ = IO1/DO, SD_DATA1/SPID = IO0/DI, SD_DATA2/SPIHD = IO3/HOLD#, SD_DATA3/SPIWP = IO2/WP#.
- For a quick reference guide to using the IO_MUX, Ethernet MAC, and GIPO Matrix pins of ESP32, please refer to Appendix *ESP32 Pin Lists*.

2.3 Power Scheme

ESP32's digital pins are divided into three different power domains:

- VDD3P3_RTC
- VDD3P3_CPU
- VDD_SDIO

VDD3P3_RTC is also the input power supply for RTC and CPU.

VDD3P3_CPU is also the input power supply for CPU.

VDD_SDIO connects to the output of an internal LDO whose input is VDD3P3_RTC. When VDD_SDIO is connected to the same PCB net together with VDD3P3_RTC, the internal LDO is disabled automatically. The power scheme diagram is shown below:

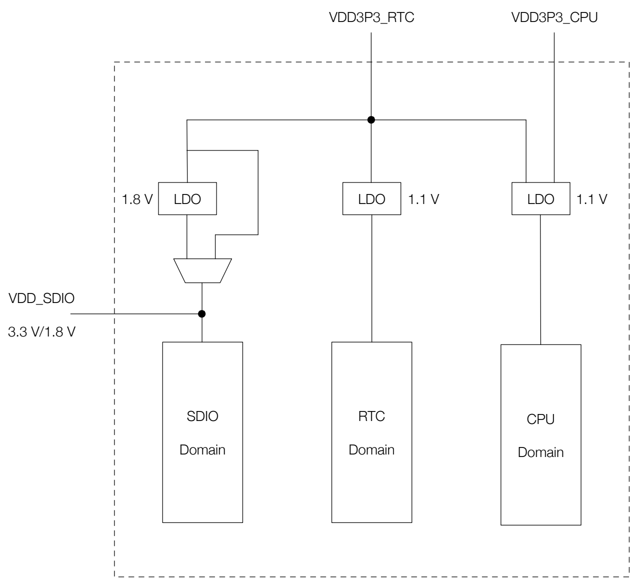


Figure 4: ESP32 Power Scheme

The internal LDO can be configured as having 1.8 V, or the same voltage as VDD3P3_RTC. It can be powered off via software to minimize the current of flash/SRAM during the Deep-sleep mode.

Notes on CHIP_PU:

- The illustration below shows the ESP32 power-up and reset timing. Details about the parameters are listed in Table 2.

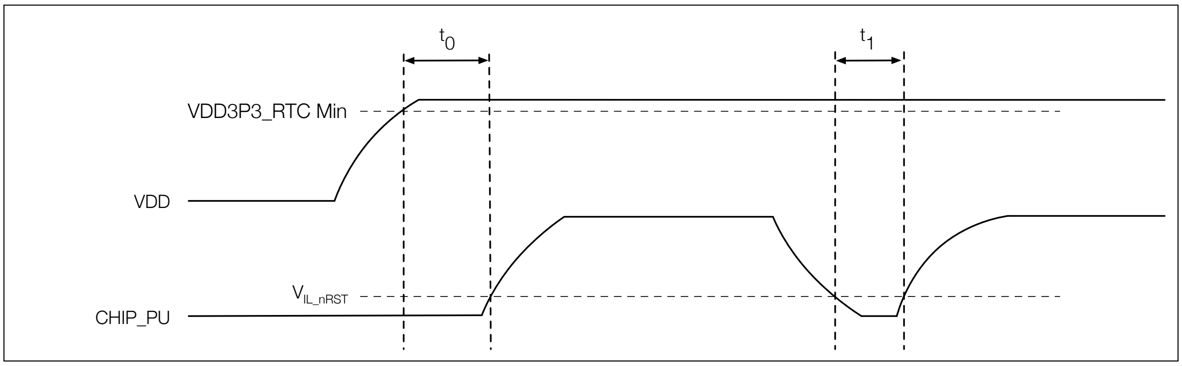


Figure 5: ESP32 Power-up and Reset Timing

Table 2: Description of ESP32 Power-up and Reset Timing Parameters

Parameters	Description	Min.	Unit
t_0	Time between the 3.3 V rails being brought up and CHIP_PU being activated	50	μs
t_1	Duration of CHIP_PU signal level < V_{IL_nRST} (refer to its value in Table 13 DC Characteristics) to reset the chip	50	μs

- In scenarios where ESP32 is powered on and off repeatedly by switching the power rails, while there is a large capacitor on the VDD33 rail and CHIP_PU and VDD33 are connected, simply switching off the CHIP_PU power rail and immediately switching it back on may cause an incomplete power discharge cycle and failure to reset the chip adequately.
  An additional discharge circuit may be required to accelerate the discharge of the large capacitor on rail VDD33, which will ensure proper power-on-reset when the ESP32 is powered up again. Please find the discharge circuit in Figure **ESP32-WROOM-32 Peripheral Schematics**, in *ESP32-WROOM-32 Datasheet*.
- When a battery is used as the power supply for the ESP32 series of chips and modules, a supply voltage supervisor is recommended, so that a boot failure due to low voltage is avoided. Users are recommended to pull CHIP_PU low if the power supply for ESP32 is below 2.3 V. For the reset circuit, please refer to Figure **ESP32-WROOM-32 Peripheral Schematics**, in *ESP32-WROOM-32 Datasheet*.

Notes on power supply:

- The operating voltage of ESP32 ranges from 2.3 V to 3.6 V. When using a single-power supply, the recommended voltage of the power supply is 3.3 V, and its recommended output current is 500 mA or more.
- When VDD_SDIO 1.8 V is used as the power supply for external flash/PSRAM, a 2-kohm grounding resistor should be added to VDD_SDIO. For the circuit design, please refer to Figure **ESP32-WROVER Schematics**, in *ESP32-WROVER Datasheet*.
- When the three digital power supplies are used to drive peripherals, e.g., 3.3 V flash, they should comply with the peripherals' specifications.

2.4 Strapping Pins

There are five strapping pins:

- MTDI
- GPIO0
- GPIO2
- MTDO
- GPIO5

Software can read the values of these five bits from register "GPIO_STRAPPING".

During the chip's system reset release (power-on-reset, RTC watchdog reset and brownout reset), the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device's boot mode, the operating voltage of VDD_SDIO and other initial system settings.

Each strapping pin is connected to its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or use the host MCU's GPIOs to control the voltage level of these pins when powering on the chip.

After reset release, the strapping pins work as normal-function pins.

Refer to Table 3 for a detailed boot-mode configuration by strapping pins.

Table 3: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Booting Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Booting					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

- FE: falling-edge, RE: rising-edge.
- Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave", after booting.
- For ESP32 chips that contain an embedded flash, users need to note the logic level of MTDI. For example, ESP32-D2WD contains an embedded flash that operates at 1.8 V, therefore, the MTDI should be pulled high. ESP32-U4WDH contains an embedded flash that operates at 3.3 V, therefore, the MTDI should be low.

The illustration below shows the setup and hold times for the strapping pin before and after the CHIP_PU signal goes high. Details about the parameters are listed in Table 4.

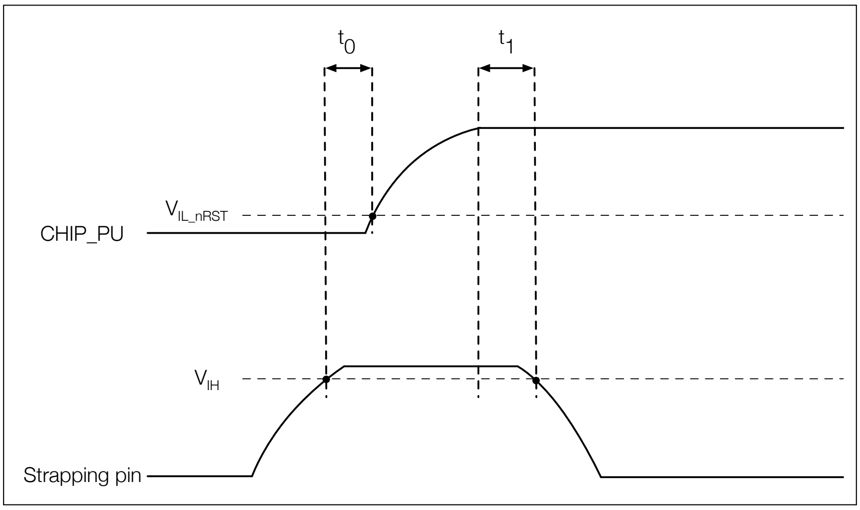


Figure 6: Setup and Hold Times for the Strapping Pin

Table 4: Parameter Descriptions of Setup and Hold Times for the Strapping Pin

Parameters	Description	Min.	Unit
t_0	Setup time before CHIP_PU goes from low to high	0	ms
t_1	Hold time after CHIP_PU goes high	1	ms

3. Functional Description

This chapter describes the functions integrated in ESP32.

3.1 CPU and Memory

3.1.1 CPU

ESP32 contains one or two low-power Xtensa® 32-bit LX6 microprocessor(s) with the following features:

- 7-stage pipeline to support the clock frequency of up to 240 MHz (160 MHz for ESP32-S0WD, ESP32-D2WD, and ESP32-U4WDH)
- 16/24-bit Instruction Set provides high code-density
- Support for Floating Point Unit
- Support for DSP instructions, such as a 32-bit multiplier, a 32-bit divider, and a 40-bit MAC
- Support for 32 interrupt vectors from about 70 interrupt sources

The single-/dual-CPU interfaces include:

- Xtensa RAM/ROM Interface for instructions and data
- Xtensa Local Memory Interface for fast peripheral register access
- External and internal interrupt sources
- JTAG for debugging

3.1.2 Internal Memory

ESP32's internal memory includes:

- 448 KB of ROM for booting and core functions
- 520 KB of on-chip SRAM for data and instructions
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.
- 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.
- Embedded flash

Note:
Products in the ESP32 series differ from each other, in terms of their support for embedded flash and the size of it. For details, please refer to Section 7 *Part Number and Ordering Information*.

3.1.3 External Flash and SRAM

ESP32 supports multiple external QSPI flash and SRAM chips. More details can be found in Chapter SPI in the *ESP32 Technical Reference Manual*. ESP32 also supports hardware encryption/decryption based on AES to protect developers' programs and data in flash.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously.
  - When external flash is mapped into CPU instruction memory space, up to 11 MB + 248 KB can be mapped at a time. Note that if more than 3 MB + 248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.
  - When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.
- External SRAM can be mapped into CPU data memory space. SRAM up to 8 MB is supported and up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

Note:
After ESP32 is initialized, firmware can customize the mapping of external SRAM or flash into the CPU address space.

3.1.4 Memory Map

The structure of address mapping is shown in Figure 7. The memory and peripheral mapping of ESP32 is shown in Table 5.

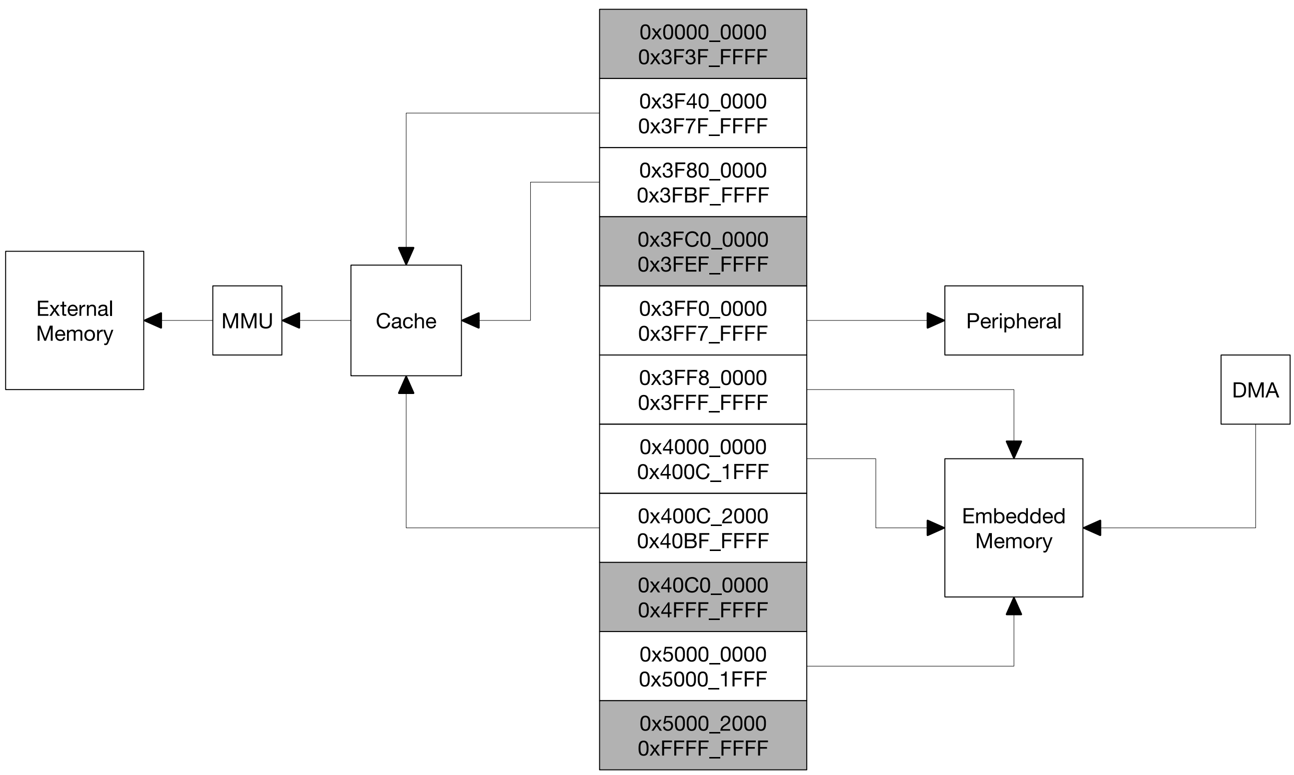


Figure 7: Address Mapping Structure

Table 5: Memory and Peripheral Mapping

Category	Target	Start Address	End Address	Size
Embedded Memory	Internal ROM 0	0x4000_0000	0x4005_FFFF	384 KB
	Internal ROM 1	0x3FF9_0000	0x3FF9_FFFF	64 KB
	Internal SRAM 0	0x4007_0000	0x4009_FFFF	192 KB
	Internal SRAM 1	0x3FFE_0000	0x3FFF_FFFF	128 KB
		0x400A_0000	0x400B_FFFF	
	Internal SRAM 2	0x3FFA_E000	0x3FFD_FFFF	200 KB
	RTC FAST Memory	0x3FF8_0000	0x3FF8_1FFF	8 KB
		0x400C_0000	0x400C_1FFF	
	RTC SLOW Memory	0x5000_0000	0x5000_1FFF	8 KB
External Memory	External Flash	0x3F40_0000	0x3F7F_FFFF	4 MB
		0x400C_2000	0x40BF_FFFF	11 MB+248 KB
	External RAM	0x3F80_0000	0x3FBF_FFFF	4 MB
Peripheral	DPort Register	0x3FF0_0000	0x3FF0_0FFF	4 KB
	AES Accelerator	0x3FF0_1000	0x3FF0_1FFF	4 KB
	RSA Accelerator	0x3FF0_2000	0x3FF0_2FFF	4 KB
	SHA Accelerator	0x3FF0_3000	0x3FF0_3FFF	4 KB
	Secure Boot	0x3FF0_4000	0x3FF0_4FFF	4 KB
	Cache MMU Table	0x3FF1_0000	0x3FF1_3FFF	16 KB
	PID Controller	0x3FF1_F000	0x3FF1_FFFF	4 KB
	UART0	0x3FF4_0000	0x3FF4_0FFF	4 KB
	SPI1	0x3FF4_2000	0x3FF4_2FFF	4 KB
	SPI0	0x3FF4_3000	0x3FF4_3FFF	4 KB
	GPIO	0x3FF4_4000	0x3FF4_4FFF	4 KB
	RTC	0x3FF4_8000	0x3FF4_8FFF	4 KB
	IO MUX	0x3FF4_9000	0x3FF4_9FFF	4 KB
	SDIO Slave	0x3FF4_B000	0x3FF4_BFFF	4 KB
	UDMA1	0x3FF4_C000	0x3FF4_CFFF	4 KB
	I2S0	0x3FF4_F000	0x3FF4_FFFF	4 KB
	UART1	0x3FF5_0000	0x3FF5_0FFF	4 KB
	I2C0	0x3FF5_3000	0x3FF5_3FFF	4 KB
	UDMA0	0x3FF5_4000	0x3FF5_4FFF	4 KB
	SDIO Slave	0x3FF5_5000	0x3FF5_5FFF	4 KB
	RMT	0x3FF5_6000	0x3FF5_6FFF	4 KB
	PCNT	0x3FF5_7000	0x3FF5_7FFF	4 KB
	SDIO Slave	0x3FF5_8000	0x3FF5_8FFF	4 KB
	LED PWM	0x3FF5_9000	0x3FF5_9FFF	4 KB
	Efuse Controller	0x3FF5_A000	0x3FF5_AFFF	4 KB
	Flash Encryption	0x3FF5_B000	0x3FF5_BFFF	4 KB
	PWM0	0x3FF5_E000	0x3FF5_EFFF	4 KB
	TIMG0	0x3FF5_F000	0x3FF5_FFFF	4 KB
	TIMG1	0x3FF6_0000	0x3FF6_0FFF	4 KB

Category	Target	Start Address	End Address	Size
Peripheral	SPI2	0x3FF6_4000	0x3FF6_4FFF	4 KB
	SPI3	0x3FF6_5000	0x3FF6_5FFF	4 KB
	SYSCON	0x3FF6_6000	0x3FF6_6FFF	4 KB
	I2C1	0x3FF6_7000	0x3FF6_7FFF	4 KB
	SDMMC	0x3FF6_8000	0x3FF6_8FFF	4 KB
	EMAC	0x3FF6_9000	0x3FF6_AFFF	8 KB
	PWM1	0x3FF6_C000	0x3FF6_CFFF	4 KB
	I2S1	0x3FF6_D000	0x3FF6_DFFF	4 KB
	UART2	0x3FF6_E000	0x3FF6_EFFF	4 KB
	PWM2	0x3FF6_F000	0x3FF6_FFFF	4 KB
	PWM3	0x3FF7_0000	0x3FF7_0FFF	4 KB
	RNG	0x3FF7_5000	0x3FF7_5FFF	4 KB

3.2 Timers and Watchdogs

3.2.1 64-bit Timers

There are four general-purpose timers embedded in the chip. They are all 64-bit generic timers which are based on 16-bit prescalers and 64-bit auto-reload-capable up/down-timers.

The timers feature:

- A 16-bit clock prescaler, from 2 to 65536
- A 64-bit timer
- Configurable up/down timer: incrementing or decrementing
- Halt and resume of time-base counter
- Auto-reload at alarming
- Software-controlled instant reload
- Level and edge interrupt generation

3.2.2 Watchdog Timers

The chip has three watchdog timers: one in each of the two timer modules (called the Main Watchdog Timer, or MWDT) and one in the RTC module (called the RTC Watchdog Timer, or RWDT). These watchdog timers are intended to recover from an unforeseen fault causing the application program to abandon its normal sequence. A watchdog timer has four stages. Each stage may trigger one of three or four possible actions upon the expiry of its programmed time period, unless the watchdog is fed or disabled. The actions are: interrupt, CPU reset, core reset, and system reset. Only the RWDT can trigger the system reset, and is able to reset the entire chip, including the RTC itself. A timeout value can be set for each stage individually.

During flash boot the RWDT and the first MWDT start automatically in order to detect, and recover from, booting problems.

The watchdogs have the following features:

- Four stages, each of which can be configured or disabled separately

- A programmable time period for each stage
- One of three or four possible actions (interrupt, CPU reset, core reset, and system reset) upon the expiry of each stage
- 32-bit expiry counter
- Write protection that prevents the RWDT and MWDT configuration from being inadvertently altered
- SPI flash boot protection
  If the boot process from an SPI flash does not complete within a predetermined time period, the watchdog will reboot the entire system.

3.3 System Clocks

3.3.1 CPU Clock

Upon reset, an external crystal clock source is selected as the default CPU clock. The external crystal clock source also connects to a PLL to generate a high-frequency clock (typically 160 MHz).

In addition, ESP32 has an internal 8 MHz oscillator. The application can select the clock source from the external crystal clock source, the PLL clock or the internal 8 MHz oscillator. The selected clock source drives the CPU clock directly, or after division, depending on the application.

3.3.2 RTC Clock

The RTC clock has five possible sources:

- external low-speed (32 kHz) crystal clock
- external crystal clock divided by 4
- internal RC oscillator (typically about 150 kHz, and adjustable)
- internal 8 MHz oscillator
- internal 31.25 kHz clock (derived from the internal 8 MHz oscillator divided by 256)

When the chip is in the normal power mode and needs faster CPU accessing, the application can choose the external high-speed crystal clock divided by 4 or the internal 8 MHz oscillator. When the chip operates in the low-power mode, the application chooses the external low-speed (32 kHz) crystal clock, the internal RC clock or the internal 31.25 kHz clock.

3.3.3 Audio PLL Clock

The audio clock is generated by the ultra-low-noise fractional-N PLL. More details can be found in Chapter Reset and Clock in the *ESP32 Technical Reference Manual*.

3.4 Radio

The radio module consists of the following blocks:

- 2.4 GHz receiver
- 2.4 GHz transmitter

- bias and regulators
- balun and transmit-receive switch
- clock generator

3.4.1 2.4 GHz Receiver

The 2.4 GHz receiver demodulates the 2.4 GHz RF signal to quadrature baseband signals and converts them to the digital domain with two high-resolution, high-speed ADCs. To adapt to varying signal channel conditions, RF filters, Automatic Gain Control (AGC), DC offset cancelation circuits and baseband filters are integrated in the chip.

3.4.2 2.4 GHz Transmitter

The 2.4 GHz transmitter modulates the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high-powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling state-of-the-art performance in delivering up to +20.5 dBm of power for an 802.11b transmission and +18 dBm for an 802.11n transmission.

Additional calibrations are integrated to cancel any radio imperfections, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities
- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the amount of time required for product testing, and render the testing equipment unnecessary.

3.4.3 Clock Generator

The clock generator produces quadrature clock signals of 2.4 GHz for both the receiver and the transmitter. All components of the clock generator are integrated into the chip, including all inductors, varactors, filters, regulators and dividers.

The clock generator has built-in calibration and self-test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms which ensure the best performance of the receiver and the transmitter.

3.5 Wi-Fi

ESP32 implements a TCP/IP and full 802.11 b/g/n Wi-Fi MAC protocol. It supports the Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF). Power management is handled with minimal host interaction to minimize the active-duty period.

3.5.1 Wi-Fi Radio and Baseband

The ESP32 Wi-Fi Radio and Baseband support the following features:

- 802.11b/g/n
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11n MCS32 (RX)
- 802.11n 0.4 μs guard-interval
- up to 150 Mbps of data rate
- Receiving STBC 2×1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity
  ESP32 supports antenna diversity with an external RF switch. One or more GPIOs control the RF switch and selects the best antenna to minimize the effects of channel fading.

3.5.2 Wi-Fi MAC

The ESP32 Wi-Fi MAC applies low-level protocol functions automatically. They are as follows:

- 4 × virtual Wi-Fi interfaces
- Simultaneous Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- RTS protection, CTS protection, Immediate Block ACK
- Defragmentation
- TX/RX A-MPDU, RX A-MSDU
- TXOP
- WMM
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4) and CRC
- Automatic beacon monitoring (hardware TSF)

3.6 Bluetooth

The chip integrates a Bluetooth link controller and Bluetooth baseband, which carry out the baseband protocols and other low-level link routines, such as modulation/demodulation, packet processing, bit stream processing, frequency hopping, etc.

3.6.1 Bluetooth Radio and Baseband

The Bluetooth Radio and Baseband support the following features:

- Class-1, class-2 and class-3 transmit output powers, and a dynamic control range of up to 24 dB
- $\pi/4$ DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with over 94 dBm of dynamic range

- Class-1 operation without external PA
- Internal SRAM allows full-speed data-transfer, mixed voice and data, and full piconet operation
- Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO and AFH
- A-law, μ-law and CVSD digital audio CODEC in PCM interface
- SBC audio CODEC
- Power management for low-power applications
- SMP with 128-bit AES

3.6.2 Bluetooth Interface

- Provides UART HCI interface, up to 4 Mbps
- Provides SDIO / SPI HCI interface
- Provides PCM / I^{2}S audio interface

3.6.3 Bluetooth Stack

The Bluetooth stack of the chip is compliant with the Bluetooth v4.2 BR/EDR and Bluetooth LE specifications.

3.6.4 Bluetooth Link Controller

The link controller operates in three major states: standby, connection and sniff. It enables multiple connections, and other operations, such as inquiry, page, and secure simple-pairing, and therefore enables Piconet and Scatternet. Below are the features:

- Classic Bluetooth
  - Device Discovery (inquiry, and inquiry scan)
  - Connection establishment (page, and page scan)
  - Multi-connections
  - Asynchronous data reception and transmission
  - Synchronous links (SCO/eSCO)
  - Master/Slave Switch
  - Adaptive Frequency Hopping and Channel assessment
  - Broadcast encryption
  - Authentication and encryption
  - Secure Simple-Pairing
  - Multi-point and scatternet management
  - Sniff mode
  - Connectionless Slave Broadcast (transmitter and receiver)

- Enhanced power control
- Ping

- Bluetooth Low Energy
  - Advertising
  - Scanning
  - Simultaneous advertising and scanning
  - Multiple connections
  - Asynchronous data reception and transmission
  - Adaptive Frequency Hopping and Channel assessment
  - Connection parameter update
  - Data Length Extension
  - Link Layer Encryption
  - LE Ping

3.7 RTC and Low-Power Management

With the use of advanced power-management technologies, ESP32 can switch between different power modes.

- Power modes
  - **Active mode**: The chip radio is powered on. The chip can receive, transmit, or listen.
  - **Modem-sleep mode**: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.
  - **Light-sleep mode**: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP co-processor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
  - **Deep-sleep mode**: Only the RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP co-processor is functional.
  - **Hibernation mode**: The internal 8-MHz oscillator and ULP co-processor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and certain RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

Table 6: Power Consumption by Power Modes

Power mode	Description			Power consumption
Active (RF working)	Wi-Fi Tx packet			Please refer to Table 15 for details.
	Wi-Fi/BT Tx packet			
	Wi-Fi/BT Rx and listening			
Modem-sleep	The CPU is powered on.	240 MHz *	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz *	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA

Power mode	Description			Power consumption
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	The ULP co-processor is powered on.			150 μA
	ULP sensor-monitored pattern			100 μA @1% duty
	RTC timer + RTC memory			10 μA
Hibernation	RTC timer only			5 μA
Power off	CHIP_PU is set to low level, the chip is powered off.			1 μA

Note:

- * Among the ESP32 series of SoCs, ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, and ESP32-D0WDQ6 have a maximum CPU frequency of 240 MHz, ESP32-D2WD, ESP32-S0WD, and ESP32-U4WDH have a maximum CPU frequency of 160 MHz.
- When Wi-Fi is enabled, the chip switches between Active and Modem-sleep modes. Therefore, power consumption changes accordingly.
- In Modem-sleep mode, the CPU frequency changes automatically. The frequency depends on the CPU load and the peripherals used.
- During Deep-sleep, when the ULP co-processor is powered on, peripherals such as GPIO and I²C are able to operate.
- When the system works in the ULP sensor-monitored pattern, the ULP co-processor works with the ULP sensor periodically and the ADC works with a duty cycle of 1%, so the power consumption is 100 μA.

4. Peripherals and Sensors

4.1 Descriptions of Peripherals and Sensors

4.1.1 General Purpose Input / Output Interface (GPIO)

ESP32 has 34 GPIO pins which can be assigned various functions by programming the appropriate registers. There are several kinds of GPIOs: digital-only, analog-enabled, capacitive-touch-enabled, etc. Analog-enabled GPIOs and Capacitive-touch-enabled GPIOs can be configured as digital GPIOs.

Most of the digital GPIOs can be configured as internal pull-up or pull-down, or set to high impedance. When configured as an input, the input value can be read through the register. The input can also be set to edge-trigger or level-trigger to generate CPU interrupts. Most of the digital IO pins are bi-directional, non-inverting and tristate, including input and output buffers with tristate control. These pins can be multiplexed with other functions, such as the SDIO, UART, SPI, etc. (More details can be found in the Appendix, Table IO_MUX.) For low-power operations, the GPIOs can be set to hold their states.

4.1.2 Analog-to-Digital Converter (ADC)

ESP32 integrates 12-bit SAR ADCs and supports measurements on 18 channels (analog-enabled pins). The ULP-coprocessor in ESP32 is also designed to measure voltage, while operating in the sleep mode, which enables low-power consumption. The CPU can be woken up by a threshold setting and/or via other triggers.

With appropriate settings, the ADCs can be configured to measure voltage on 18 pins maximum.

Table 7 describes the ADC characteristics.

Table 7: ADC Characteristics

Parameter	Description	Min	Max	Unit
DNL (Differential nonlinearity)	RTC controller; ADC connected to an external 100 nF capacitor; DC signal input; ambient temperature at 25 °C; Wi-Fi&BT off	–7	7	LSB
INL (Integral nonlinearity)		–12	12	LSB
Sampling rate	RTC controller	-	200	ksps
	DIG controller	-	2	Msps

Notes:

- When atten=3 and the measurement result is above 3000 (voltage at approx. 2450 mV), the ADC accuracy will be worse than described in the table above.
- To get better DNL results, users can take multiple sampling tests with a filter, or calculate the average value.
- The input voltage range of GPIO pins within VDD3P3_RTC domain should strictly follow the DC characteristics provided in Table 13. Otherwise, measurement errors may be introduced, and chip performance may be affected.

By default, there are ±6% differences in measured results between chips. ESP-IDF provides couple of calibration methods for ADC1. Results after calibration using eFuse Vref value are shown in Table 8. For higher accuracy, users may apply other calibration methods provided in ESP-IDF, or implement their own.

Table 8: ADC Calibration Results

Parameter	Description	Min	Max	Unit
Total error	Atten=0, effective measurement range of 100 ∼ 950 mV	–23	23	mV
	Atten=1, effective measurement range of 100 ∼ 1250 mV	–30	30	mV
	Atten=2, effective measurement range of 150 ∼ 1750 mV	–40	40	mV
	Atten=3, effective measurement range of 150 ∼ 2450 mV	–60	60	mV

4.1.3 Hall Sensor

ESP32 integrates a Hall sensor based on an N-carrier resistor. When the chip is in the magnetic field, the Hall sensor develops a small voltage laterally on the resistor, which can be directly measured by the ADC.

4.1.4 Digital-to-Analog Converter (DAC)

Two 8-bit DAC channels can be used to convert two digital signals into two analog voltage signal outputs. The design structure is composed of integrated resistor strings and a buffer. This dual DAC supports power supply as input voltage reference. The two DAC channels can also support independent conversions.

4.1.5 Touch Sensor

ESP32 has 10 capacitive-sensing GPIOs, which detect variations induced by touching or approaching the GPIOs with a finger or other objects. The low-noise nature of the design and the high sensitivity of the circuit allow relatively small pads to be used. Arrays of pads can also be used, so that a larger area or more points can be detected. The 10 capacitive-sensing GPIOs are listed in Table 9.

Table 9: Capacitive-Sensing GPIOs Available on ESP32

Capacitive-sensing signal name	Pin name
T0	GPIO4
T1	GPIO0
T2	GPIO2
T3	MTDO
T4	MTCK
T5	MTDI
T6	MTMS
T7	GPIO27
T8	32K_XN
T9	32K_XP

4.1.6 Ultra-Low-Power Co-processor

The ULP processor and RTC memory remain powered on during the Deep-sleep mode. Hence, the developer can store a program for the ULP processor in the RTC slow memory to access the peripheral devices, internal timers and internal sensors during the Deep-sleep mode. This is useful for designing applications where the CPU needs to be woken up by an external event, or a timer, or a combination of the two, while maintaining minimal power consumption.

4.1.7 Ethernet MAC Interface

An IEEE-802.3-2008-compliant Media Access Controller (MAC) is provided for Ethernet LAN communications. ESP32 requires an external physical interface device (PHY) to connect to the physical LAN bus (twisted-pair, fiber, etc.). The PHY is connected to ESP32 through 17 signals of MII or nine signals of RMII. The following features are supported on the Ethernet MAC (EMAC) interface:

- 10 Mbps and 100 Mbps rates
- Dedicated DMA controller allowing high-speed transfer between the dedicated SRAM and Ethernet MAC
- Tagged MAC frame (VLAN support)
- Half-duplex (CSMA/CD) and full-duplex operation
- MAC control sublayer (control frames)
- 32-bit CRC generation and removal
- Several address-filtering modes for physical and multicast address (multicast and group addresses)
- 32-bit status code for each transmitted or received frame
- Internal FIFOs to buffer transmit and receive frames. The transmit FIFO and the receive FIFO are both 512 words (32-bit)
- Hardware PTP (Precision Time Protocol) in accordance with IEEE 1588 2008 (PTP V2)
- 25 MHz/50 MHz clock output

4.1.8 SD/SDIO/MMC Host Controller

An SD/SDIO/MMC host controller is available on ESP32, which supports the following features:

- Secure Digital memory (SD mem Version 3.0 and Version 3.01)
- Secure Digital I/O (SDIO Version 3.0)
- Consumer Electronics Advanced Transport Architecture (CE-ATA Version 1.1)
- Multimedia Cards (MMC Version 4.41, eMMC Version 4.5 and Version 4.51)

The controller allows up to 80 MHz clock output in three different data-bus modes: 1-bit, 4-bit and 8-bit. It supports two SD/SDIO/MMC4.41 cards in a 4-bit data-bus mode. It also supports one SD card operating at 1.8 V.

4.1.9 SDIO/SPI Slave Controller

ESP32 integrates an SD device interface that conforms to the industry-standard SDIO Card Specification Version 2.0, and allows a host controller to access the SoC, using the SDIO bus interface and protocol. ESP32 acts as the slave on the SDIO bus. The host can access the SDIO-interface registers directly and can access shared memory via a DMA engine, thus maximizing performance without engaging the processor cores.

The SDIO/SPI slave controller supports the following features:

- SPI, 1-bit SDIO, and 4-bit SDIO transfer modes over the full clock range from 0 to 50 MHz
- Configurable sampling and driving clock edge
- Special registers for direct access by host
- Interrupts to host for initiating data transfer

- Automatic loading of SDIO bus data and automatic discarding of padding data
- Block size of up to 512 bytes
- Interrupt vectors between the host and the slave, allowing both to interrupt each other
- Supports DMA for data transfer

4.1.10 Universal Asynchronous Receiver Transmitter (UART)

ESP32 has three UART interfaces, i.e., UART0, UART1 and UART2, which provide asynchronous communication (RS232 and RS485) and IrDA support, communicating at a speed of up to 5 Mbps. UART provides hardware management of the CTS and RTS signals and software flow control (XON and XOFF). All of the interfaces can be accessed by the DMA controller or directly by the CPU.

4.1.11 I²C Interface

ESP32 has two I²C bus interfaces which can serve as I²C master or slave, depending on the user's configuration. The I²C interfaces support:

- Standard mode (100 Kbit/s)
- Fast mode (400 Kbit/s)
- Up to 5 MHz, yet constrained by SDA pull-up strength
- 7-bit/10-bit addressing mode
- Dual addressing mode

Users can program command registers to control I²C interfaces, so that they have more flexibility.

4.1.12 I²S Interface

Two standard I²S interfaces are available in ESP32. They can be operated in master or slave mode, in full duplex and half-duplex communication modes, and can be configured to operate with an 8-/16-/32-/48-/64-bit resolution as input or output channels. BCK clock frequency, from 10 kHz up to 40 MHz, is supported. When one or both of the I²S interfaces are configured in the master mode, the master clock can be output to the external DAC/CODEC.

Both of the I²S interfaces have dedicated DMA controllers. PDM and BT PCM interfaces are supported.

4.1.13 Infrared Remote Controller

The infrared remote controller supports eight channels of infrared remote transmission and receiving. By programming the pulse waveform, it supports various infrared protocols. Eight channels share a 512 x 32-bit block of memory to store the transmitting or receiving waveform.

4.1.14 Pulse Counter

The pulse counter captures pulse and counts pulse edges through seven modes. It has eight channels, each of which captures four signals at a time. The four input signals include two pulse signals and two control signals. When the counter reaches a defined threshold, an interrupt is generated.

4.1.15 Pulse Width Modulation (PWM)

The Pulse Width Modulation (PWM) controller can be used for driving digital motors and smart lights. The controller consists of PWM timers, the PWM operator and a dedicated capture sub-module. Each timer provides timing in synchronous or independent form, and each PWM operator generates a waveform for one PWM channel. The dedicated capture sub-module can accurately capture events with external timing.

4.1.16 LED PWM

The LED PWM controller can generate 16 independent channels of digital waveforms with configurable periods and duties.

The 16 channels of digital waveforms operate with an APB clock of 80 MHz. Eight of these channels have the option of using the 8 MHz oscillator clock. Each channel can select a 20-bit timer with configurable counting range, while its accuracy of duty can be up to 16 bits within a 1 ms period.

The software can change the duty immediately. Moreover, each channel automatically supports step-by-step duty increase or decrease, which is useful for the LED RGB color-gradient generator.

4.1.17 Serial Peripheral Interface (SPI)

ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes. These SPIs also support the following general-purpose SPI features:

- Four modes of SPI transfer format, which depend on the polarity (CPOL) and the phase (CPHA) of the SPI clock
- Up to 80 MHz (The actual speed it can reach depends on the selected pads, PCB tracing, peripheral characteristics, etc.)
- up to 64-byte FIFO

All SPIs can also be connected to the external flash/SRAM and LCD. Each SPI can be served by DMA controllers.

4.1.18 Accelerator

ESP32 is equipped with hardware accelerators of general algorithms, such as AES (FIPS PUB 197), SHA (FIPS PUB 180-4), RSA, and ECC, which support independent arithmetic, such as Big Integer Multiplication and Big Integer Modular Multiplication. The maximum operation length for RSA, ECC, Big Integer Multiply and Big Integer Modular Multiplication is 4096 bits.

The hardware accelerators greatly improve operation speed and reduce software complexity. They also support code encryption and dynamic decryption, which ensures that code in the flash will not be hacked.

4.2 Peripheral Pin Configurations

Table 10: Peripheral Pin Configurations

Interface	Signal	Pin	Function
ADC	ADC1_CH0	SENSOR_VP	Two 12-bit SAR ADCs
	ADC1_CH1	SENSOR_CAPP	
	ADC1_CH2	SENSOR_CAPN	
	ADC1_CH3	SENSOR_VN	
	ADC1_CH4	32K_XP	
	ADC1_CH5	32K_XN	
	ADC1_CH6	VDET_1	
	ADC1_CH7	VDET_2	
	ADC2_CH0	GPIO4	
	ADC2_CH1	GPIO0	
	ADC2_CH2	GPIO2	
	ADC2_CH3	MTDO	
	ADC2_CH4	MTCK	
	ADC2_CH5	MTDI	
	ADC2_CH6	MTMS	
	ADC2_CH7	GPIO27	
	ADC2_CH8	GPIO25	
	ADC2_CH9	GPIO26	
DAC	DAC_1	GPIO25	Two 8-bit DACs
	DAC_2	GPIO26	
Touch Sensor	TOUCH0	GPIO4	Capacitive touch sensors
	TOUCH1	GPIO0	
	TOUCH2	GPIO2	
	TOUCH3	MTDO	
	TOUCH4	MTCK	
	TOUCH5	MTDI	
	TOUCH6	MTMS	
	TOUCH7	GPIO27	
	TOUCH8	32K_XN	
	TOUCH9	32K_XP	
JTAG	MTDI	MTDI	JTAG for software debugging
	MTCK	MTCK	
	MTMS	MTMS	
	MTDO	MTDO	

Interface	Signal	Pin	Function
SD/SDIO/MMC Host Controller	HS2_CLK	MTMS	Supports SD memory card V3.01 standard
	HS2_CMD	MTDO	
	HS2_DATA0	GPIO2	
	HS2_DATA1	GPIO4	
	HS2_DATA2	MTDI	
	HS2_DATA3	MTCK	
Motor PWM	PWM0_OUT0~2	Any GPIO Pins	Three channels of 16-bit timers generate PWM waveforms. Each channel has a pair of output signals, three fault detection signals, three event-capture signals, and three sync signals.
	PWM1_OUT_IN0~2		
	PWM0_FLT_IN0~2		
	PWM1_FLT_IN0~2		
	PWM0_CAP_IN0~2		
	PWM1_CAP_IN0~2		
	PWM0_SYNC_IN0~2		
	PWM1_SYNC_IN0~2		
SDIO/SPI Slave Controller	SD_CLK	MTMS	SDIO interface that conforms to the industry standard SDIO 2.0 card specification
	SD_CMD	MTDO	
	SD_DATA0	GPIO2	
	SD_DATA1	GPIO4	
	SD_DATA2	MTDI	
	SD_DATA3	MTCK	
UART	U0RXD_in	Any GPIO Pins	Two UART devices with hardware flow-control and DMA
	U0CTS_in		
	U0DSR_in		
	U0TXD_out		
	U0RTS_out		
	U0DTR_out		
	U1RXD_in		
	U1CTS_in		
	U1TXD_out		
	U1RTS_out		
	U2RXD_in		
	U2CTS_in		
	U2TXD_out		
	U2RTS_out		
I^{2}C	I2CEXT0_SCL_in	Any GPIO Pins	Two I^{2}C devices in slave or master mode
	I2CEXT0_SDA_in		
	I2CEXT1_SCL_in		
	I2CEXT1_SDA_in		
	I2CEXT0_SCL_out		
	I2CEXT0_SDA_out		
	I2CEXT1_SCL_out		
	I2CEXT1_SDA_out		

Interface	Signal	Pin	Function
LED PWM	ledc_hs_sig_out0~7	Any GPIO Pins	16 independent channels @80 MHz clock/RTC CLK. Duty accuracy: 16 bits.
	ledc_ls_sig_out0~7		
I2S	I2S0I_DATA_in0~15	Any GPIO Pins	Stereo input and output from/to the audio codec; parallel LCD data output; parallel camera data input
	I2S0O_BCK_in		
	I2S0O_WS_in		
	I2S0I_BCK_in		
	I2S0I_WS_in		
	I2S0I_H_SYNC		
	I2S0I_V_SYNC		
	I2S0I_H_ENABLE		
	I2S0O_BCK_out		
	I2S0O_WS_out		
	I2S0I_BCK_out		
	I2S0I_WS_out		
	I2S0O_DATA_out0~23		
	I2S1I_DATA_in0~15		
	I2S1O_BCK_in		
	I2S1O_WS_in		
	I2S1I_BCK_in		
	I2S1I_WS_in		
	I2S1I_H_SYNC		
	I2S1I_V_SYNC		
	I2S1I_H_ENABLE		
	I2S1O_BCK_out		
	I2S1O_WS_out		
	I2S1I_BCK_out		
	I2S1I_WS_out		
	I2S1O_DATA_out0~23		
Infrared Remote Controller	RMT_SIG_IN0~7	Any GPIO Pins	Eight channels for an IR transmitter and receiver of various waveforms
	RMT_SIG_OUT0~7		
General Purpose SPI	HSPIQ_in/_out	Any GPIO Pins	Standard SPI consists of clock, chip-select, MOSI and MISO. These SPIs can be connected to LCD and other external devices. They support the following features: • Both master and slave modes; • Four sub-modes of the SPI transfer format; • Configurable SPI frequency; • Up to 64 bytes of FIFO and DMA.
	HSPID_in/_out		
	HSPICLK_in/_out		
	HSPI_CS0_in/_out		
	HSPI_CS1_out		
	HSPI_CS2_out		
	VSPIQ_in/_out		
	VSPID_in/_out		
	VSPICLK_in/_out		
	VSPI_CS0_in/_out		
	VSPI_CS1_out		
	VSPI_CS2_out		

Interface	Signal	Pin	Function
Parallel QSPI	SPIHD	SD_DATA_2	Supports Standard SPI, Dual SPI, and Quad SPI that can be connected to the external flash and SRAM
	SPIWP	SD_DATA_3	
	SPICS0	SD_CMD	
	SPICLK	SD_CLK	
	SPIQ	SD_DATA_0	
	SPID	SD_DATA_1	
	HSPICLK	MTMS	
	HSPICS0	MTDO	
	HSPIQ	MTDI	
	HSPID	MTCK	
	HSPIHD	GPIO4	
	HSPIWP	GPIO2	
	VSPICLK	GPIO18	
	VSPICS0	GPIO5	
	VSPIQ	GPIO19	
	VSPID	GPIO23	
	VSPIHD	GPIO21	
	VSPIWP	GPIO22	
EMAC	EMAC_TX_CLK	GPIO0	Ethernet MAC with MII/RMII interface
	EMAC_RX_CLK	GPIO5	
	EMAC_TX_EN	GPIO21	
	EMAC_TXD0	GPIO19	
	EMAC_TXD1	GPIO22	
	EMAC_TXD2	MTMS	
	EMAC_TXD3	MTDI	
	EMAC_RX_ER	MTCK	
	EMAC_RX_DV	GPIO27	
	EMAC_RXD0	GPIO25	
	EMAC_RXD1	GPIO26	
	EMAC_RXD2	U0TXD	
	EMAC_RXD3	MTDO	
	EMAC_CLK_OUT	GPIO16	
	EMAC_CLK_OUT_180	GPIO17	
	EMAC_TX_ER	GPIO4	
	EMAC_MDC_out	Any GPIO Pins	
	EMAC_MDI_in	Any GPIO Pins	
	EMAC_MDO_out	Any GPIO Pins	
	EMAC_CRS_out	Any GPIO Pins	
	EMAC_COL_out	Any GPIO Pins	

Interface	Signal	Pin	Function
Pulse Counter	pcnt_sig_ch0_in0	Any GPIO Pins	Operating in seven different modes, the pulse counter captures pulse and counts pulse edges.
	pcnt_sig_ch1_in0		
	pcnt_ctrl_ch0_in0		
	pcnt_ctrl_ch1_in0		
	pcnt_sig_ch0_in1		
	pcnt_sig_ch1_in1		
	pcnt_ctrl_ch0_in1		
	pcnt_ctrl_ch1_in1		
	pcnt_sig_ch0_in2		
	pcnt_sig_ch1_in2		
	pcnt_ctrl_ch0_in2		
	pcnt_ctrl_ch1_in2		
	pcnt_sig_ch0_in3		
	pcnt_sig_ch1_in3		
	pcnt_ctrl_ch0_in3		
	pcnt_ctrl_ch1_in3		
	pcnt_sig_ch0_in4		
	pcnt_sig_ch1_in4		
	pcnt_ctrl_ch0_in4		
	pcnt_ctrl_ch1_in4		
	pcnt_sig_ch0_in5		
	pcnt_sig_ch1_in5		
	pcnt_ctrl_ch0_in5		
	pcnt_ctrl_ch1_in5		
	pcnt_sig_ch0_in6		
	pcnt_sig_ch1_in6		
	pcnt_ctrl_ch0_in6		
	pcnt_ctrl_ch1_in6		
	pcnt_sig_ch0_in7		
	pcnt_sig_ch1_in7		
	pcnt_ctrl_ch0_in7		
	pcnt_ctrl_ch1_in7		

5. Electrical Characteristics

5.1 Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in the table below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the recommended operating conditions.

Table 11: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO	Voltage applied to power supply pins per power domain	–0.3	3.6	V
I_{output} *	Cumulative IO output current	-	1200	mA
T_{store}	Storage temperature	–40	150	°C

* The chip worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO) output high logic level to ground.

5.2 Recommended Operating Conditions

Table 12: Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VDDA, VDD3P3_RTC [1] VDD3P3, VDD_SDIO (3.3 V mode) [2]	Voltage applied to power supply pins per power domain	2.3	3.3	3.6	V
VDD3P3_CPU	Voltage applied to power supply pin	1.8	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	-	-	A
T [3]	Operating temperature	–40	-	125	°C

1. When writing eFuse, VDD3P3_RTC should be at least 3.3 V.
2. • VDD_SDIO works as the power supply for the related IO, and also for an external device. Please refer to the Appendix IO_MUX of this datasheet for more details.
   • VDD_SDIO can be sourced internally by the ESP32 from the VDD3P3_RTC power domain:
     – When VDD_SDIO operates at 3.3 V, it is driven directly by VDD3P3_RTC through a 6 Ω resistor, therefore, there will be some voltage drop from VDD3P3_RTC.
     – When VDD_SDIO operates at 1.8 V, it can be generated from ESP32's internal LDO. The maximum current this LDO can offer is 40 mA, and the output voltage range is 1.65 V ~ 2.0 V.
   • VDD_SDIO can also be driven by an external power supply.
   • Please refer to Power Scheme, section 2.3, for more information.
3. The operating temperature of ESP32-D2WD and ESP32-U4WDH ranges from –40 °C to 105 °C, due to the flash embedded in them. The other chips in this series have no embedded flash, so their range of operating temperatures is –40 °C ~ 125 °C.

5.3 DC Characteristics (3.3 V, 25 °C)

Table 13: DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter		Min	Typ	Max	Unit
C_{IN}	Pin capacitance		-	2	-	pF
V_{IH}	High-level input voltage		0.75×VDD[1]	-	VDD[1]+0.3	V
V_{IL}	Low-level input voltage		–0.3	-	0.25×VDD[1]	V
I_{IH}	High-level input current		-	-	50	nA
I_{IL}	Low-level input current		-	-	50	nA
V_{OH}	High-level output voltage		0.8×VDD[1]	-	-	V
V_{OL}	Low-level output voltage		-	-	0.1×VDD[1]	V
I_{OH}	High-level source current (VDD[1] = 3.3 V, V_{OH} >= 2.64 V, output drive strength set to the maximum)	VDD3P3_CPU power domain [1, 2]	-	40	-	mA
		VDD3P3_RTC power domain [1, 2]	-	40	-	mA
		VDD_SDIO power domain [1, 3]	-	20	-	mA
I_{OL}	Low-level sink current (VDD[1] = 3.3 V, V_{OL} = 0.495 V, output drive strength set to the maximum)		-	28	-	mA
R_{PU}	Resistance of internal pull-up resistor		-	45	-	kΩ
R_{PD}	Resistance of internal pull-down resistor		-	45	-	kΩ
V_{IL_nRST}	Low-level input voltage of CHIP_PU to power off the chip		-	-	0.6	V

Notes:

1. Please see Table IO_MUX for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2. For VDD3P3_CPU and VDD3P3_RTC power domain, per-pin current sourced in the same domain is gradually reduced from around 40 mA to around 29 mA, V_{OH}>=2.64 V, as the number of current-source pins increases.
3. For VDD_SDIO power domain, per-pin current sourced in the same domain is gradually reduced from around 30 mA to around 10 mA, V_{OH}>=2.64 V, as the number of current-source pins increases.

5.4 Reliability Qualifications

Table 14: Reliability Qualifications

Reliability tests	Standards	Test conditions	Result
Electro-Static Discharge Sensitivity (ESD), Charge Device Mode (CDM) [1]	JEDEC EIA/JESD22-C101	±500 V, all pins	Pass
Electro-Static Discharge Sensitivity (ESD), Human Body Mode (HBM) [2]	JEDEC EIA/JESD22-A114	±1500 V, all pins	Pass
Latch-up (Over-current test)	JEDEC STANDARD NO.78	±50 mA ~ ±200 mA, room temperature, test for IO	Pass
Latch-up (Over-voltage test)	JEDEC STANDARD NO.78	1.5 × Vmax, room temperature, test for V_{supply}	Pass

Reliability tests	Standards	Test conditions	Result
Moisture Sensitivity Level (MSL)	J-STD-020, MSL 3	30 °C, 60% RH, 192 hours, IR × 3 @260 °C	Pass

1. JEDEC document JEP157 states that 250 V CDM allows safe manufacturing with a standard ESD control process.
2. JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process.

5.5 RF Power-Consumption Specifications

The power consumption measurements are taken with a 3.3 V supply at 25 °C of ambient temperature at the RF port. All transmitters' measurements are based on a 50% duty cycle.

Table 15: RF Power-Consumption Specifications

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11n, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

5.6 Wi-Fi Radio

Table 16: Wi-Fi Radio Characteristics

Parameter	Condition	Min	Typical	Max	Unit
Operating frequency range ^{note1}	-	2412	-	2484	MHz
Output impedance ^{note2}	-	-	*note 2*	-	Ω
TX power ^{note3}	11n, MCS7	12	13	14	dBm
	11b mode	18.5	19.5	20.5	dBm
Sensitivity	11b, 1 Mbps	-	–98	-	dBm
	11b, 11 Mbps	-	–88	-	dBm
	11g, 6 Mbps	-	–93	-	dBm
	11g, 54 Mbps	-	–75	-	dBm
	11n, HT20, MCS0	-	–93	-	dBm
	11n, HT20, MCS7	-	–73	-	dBm
	11n, HT40, MCS0	-	–90	-	dBm
	11n, HT40, MCS7	-	–70	-	dBm
Adjacent channel rejection	11g, 6 Mbps	-	27	-	dB
	11g, 54 Mbps	-	13	-	dB
	11n, HT20, MCS0	-	27	-	dB
	11n, HT20, MCS7	-	12	-	dB

1. Device should operate in the frequency range allocated by regional regulatory authorities. Target operating frequency range is configurable by software.
2. The typical value of ESP32's Wi-Fi radio output impedance is different between chips in different QFN packages. For ESP32 chips with a QFN 6×6 package, the value is 30+j10 Ω. For ESP32 chips with a QFN 5×5 package, the value is 35+j10 Ω.
3. Target TX power is configurable based on device or certification requirements.

5.7 Bluetooth Radio

5.7.1 Receiver – Basic Data Rate

Table 17: Receiver Characteristics – Basic Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @0.1% BER	-	–90	–89	–88	dBm
Maximum received signal @0.1% BER	-	0	-	-	dBm
Co-channel C/I	-	-	+7	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	-	–6	dB
	F = F0 – 1 MHz	-	-	–6	dB
	F = F0 + 2 MHz	-	-	–25	dB
	F = F0 – 2 MHz	-	-	–33	dB
	F = F0 + 3 MHz	-	-	–25	dB
	F = F0 – 3 MHz	-	-	–45	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	–10	-	-	dBm
	2000 MHz ~ 2400 MHz	–27	-	-	dBm
	2500 MHz ~ 3000 MHz	–27	-	-	dBm
	3000 MHz ~ 12.5 GHz	–10	-	-	dBm
Intermodulation	-	–36	-	-	dBm

5.7.2 Transmitter – Basic Data Rate

Table 18: Transmitter Characteristics – Basic Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 18)	-	-	0	-	dBm
Gain control step	-	-	3	-	dB
RF power control range	-	–12	-	+9	dBm
+20 dB bandwidth	-	-	0.9	-	MHz
Adjacent channel transmit power	F = F0 ± 2 MHz	-	–47	-	dBm
	F = F0 ± 3 MHz	-	–55	-	dBm
	F = F0 ± > 3 MHz	-	–60	-	dBm
$\Delta f1_{avg}$	-	-	-	155	kHz
$\Delta f2_{max}$	-	133.7	-	-	kHz
$\Delta f2_{avg}/\Delta f1_{avg}$	-	-	0.92	-	-
ICFT	-	-	–7	-	kHz

Parameter	Conditions	Min	Typ	Max	Unit
Drift rate	-	-	0.7	-	kHz/50 μs
Drift (DH1)	-	-	6	-	kHz
Drift (DH5)	-	-	6	-	kHz

Note:
There are a total of eight power levels from 0 to 7, and the transmit power ranges from –12 dBm to 9 dBm. When the power level rises by 1, the transmit power increases by 3 dB. Power level 4 is used by default and the corresponding transmit power is 0 dBm.

5.7.3 Receiver – Enhanced Data Rate

Table 19: Receiver Characteristics – Enhanced Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
π/4 DQPSK					
Sensitivity @0.01% BER	-	–90	–89	–88	dBm
Maximum received signal @0.01% BER	-	-	0	-	dBm
Co-channel C/I	-	-	11	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	–7	-	dB
	F = F0 – 1 MHz	-	–7	-	dB
	F = F0 + 2 MHz	-	–25	-	dB
	F = F0 – 2 MHz	-	–35	-	dB
	F = F0 + 3 MHz	-	–25	-	dB
	F = F0 – 3 MHz	-	–45	-	dB
8DPSK					
Sensitivity @0.01% BER	-	–84	–83	–82	dBm
Maximum received signal @0.01% BER	-	-	–5	-	dBm
C/I c-channel	-	-	18	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	2	-	dB
	F = F0 – 1 MHz	-	2	-	dB
	F = F0 + 2 MHz	-	–25	-	dB
	F = F0 – 2 MHz	-	–25	-	dB
	F = F0 + 3 MHz	-	–25	-	dB
	F = F0 – 3 MHz	-	–38	-	dB

5.7.4 Transmitter – Enhanced Data Rate

Table 20: Transmitter Characteristics – Enhanced Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 18)	-	-	0	-	dBm
Gain control step	-	-	3	-	dB
RF power control range	-	–12	-	+9	dBm
π/4 DQPSK max w0	-	-	–0.72	-	kHz

Parameter	Conditions	Min	Typ	Max	Unit
π/4 DQPSK max wi	-	-	–6	-	kHz
π/4 DQPSK max |wi + w0|	-	-	–7.42	-	kHz
8DPSK max w0	-	-	0.7	-	kHz
8DPSK max wi	-	-	–9.6	-	kHz
8DPSK max |wi + w0|	-	-	–10	-	kHz
π/4 DQPSK modulation accuracy	RMS DEVM	-	4.28	-	%
	99% DEVM	-	100	-	%
	Peak DEVM	-	13.3	-	%
8 DPSK modulation accuracy	RMS DEVM	-	5.8	-	%
	99% DEVM	-	100	-	%
	Peak DEVM	-	14	-	%
In-band spurious emissions	F = F0 ± 1 MHz	-	–46	-	dBm
	F = F0 ± 2 MHz	-	–40	-	dBm
	F = F0 ± 3 MHz	-	–46	-	dBm
	F = F0 +/– > 3 MHz	-	-	–53	dBm
EDR differential phase coding	-	-	100	-	%

5.8 Bluetooth LE Radio

5.8.1 Receiver

Table 21: Receiver Characteristics – BLE

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	-	–94	–93	–92	dBm
Maximum received signal @30.8% PER	-	0	-	-	dBm
Co-channel C/I	-	-	+10	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	–5	-	dB
	F = F0 – 1 MHz	-	–5	-	dB
	F = F0 + 2 MHz	-	–25	-	dB
	F = F0 – 2 MHz	-	–35	-	dB
	F = F0 + 3 MHz	-	–25	-	dB
	F = F0 – 3 MHz	-	–45	-	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	–10	-	-	dBm
	2000 MHz ~ 2400 MHz	–27	-	-	dBm
	2500 MHz ~ 3000 MHz	–27	-	-	dBm
	3000 MHz ~ 12.5 GHz	–10	-	-	dBm
Intermodulation	-	–36	-	-	dBm

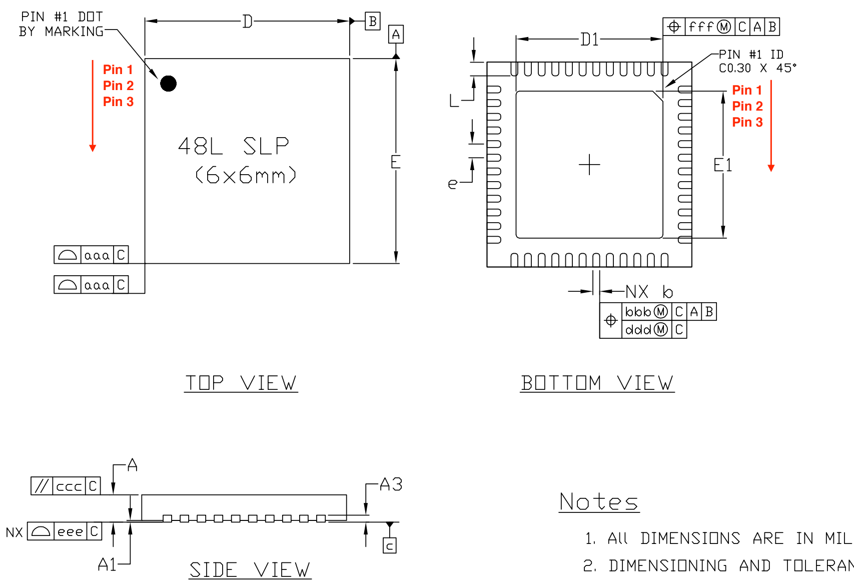
5.8.2 Transmitter

Table 22: Transmitter Characteristics – BLE

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 18)	-	-	0	-	dBm

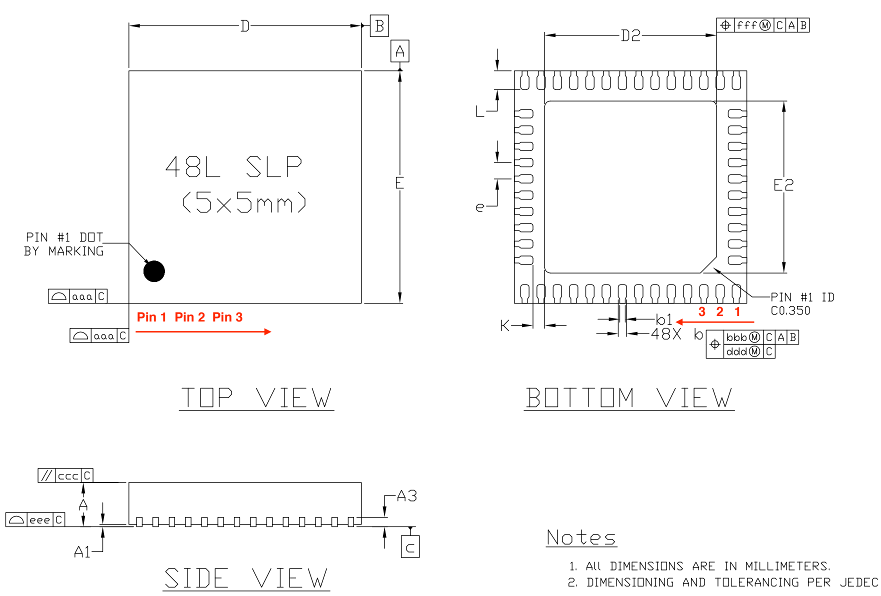
Parameter	Conditions	Min	Typ	Max	Unit
Gain control step	-	-	3	-	dB
RF power control range	-	–12	-	+9	dBm
Adjacent channel transmit power	F = F0 ± 2 MHz	-	–52	-	dBm
	F = F0 ± 3 MHz	-	–58	-	dBm
	F = F0 ± > 3 MHz	-	–60	-	dBm
$\Delta f1_{avg}$	-	-	-	265	kHz
$\Delta f2_{max}$	-	247	-	-	kHz
$\Delta f2_{avg}/\Delta f1_{avg}$	-	-	0.92	-	-
ICFT	-	-	–10	-	kHz
Drift rate	-	-	0.7	-	kHz/50 μs
Drift	-	-	2	-	kHz

6. Package Information



Dimensional Ref.			
REF.	Min.	Nom.	Max.
A	0.800	0.850	0.900
A1	0.000	---	0.050
A3	0.203 Ref.		
D	5.950	6.000	6.050
E	5.950	6.000	6.050
D1	4.250	4.300	4.350
E1	4.250	4.300	4.350
b	0.150	0.200	0.250
L	0.350	0.400	0.450
e	0.400 BSC		
Tol. of Form&Position			
aaa	0.10		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Figure 8: QFN48 (6x6 mm) Package



Dimensional Ref.			
REF.	Min.	Nom.	Max.
A	0.800	0.850	0.900
A1	0.000	--	0.050
A3	0.203 Ref.		
D	4.950	5.000	5.050
E	4.950	5.000	5.050
D2	3.650	3.700	3.750
E2	3.650	3.700	3.750
b	0.130	0.180	0.230
b1	0.070	0.120	0.170
L	0.350	0.400	0.450
e	0.350 BSC		
k	0.250 Ref		
Tol. of Form&Position			
aaa	0.10		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Figure 9: QFN48 (5x5 mm) Package

Note:
The pins of the chip are numbered in an anti-clockwise direction from Pin 1 in the top view.

7. Part Number and Ordering Information

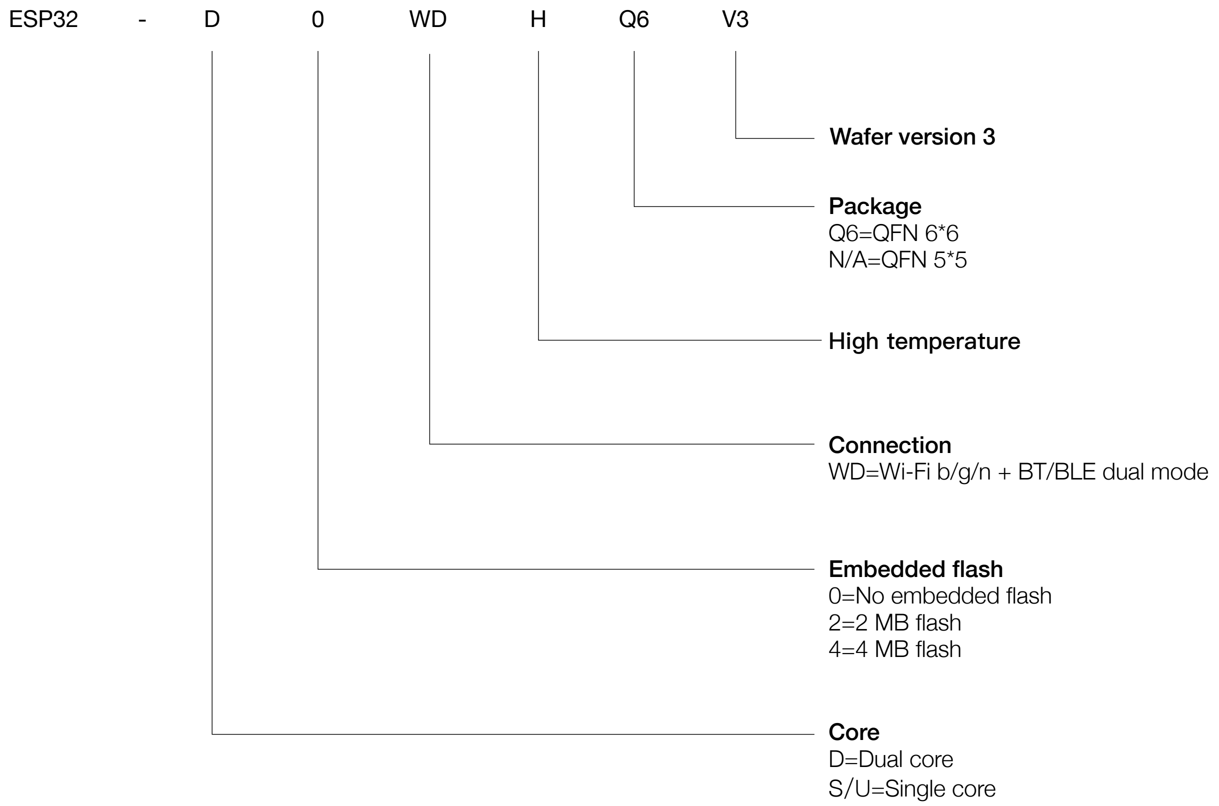


Figure 10: ESP32 Part Number

The table below provides the ordering information of the ESP32 series of chips.

Table 23: ESP32 Ordering Information

Ordering code	Core	Embedded flash	Package
ESP32-D0WD-V3	Dual core	No embedded flash	QFN 5*5
ESP32-D0WDQ6-V3	Dual core	No embedded flash	QFN 6*6
ESP32-D0WD	Dual core	No embedded flash	QFN 5*5
ESP32-D0WDQ6	Dual core	No embedded flash	QFN 6*6
ESP32-D2WD	Dual core	2 MB embedded flash (40 MHz)	QFN 5*5
ESP32-S0WD	Single core	No embedded flash	QFN 5*5
ESP32-U4WDH	Single core	4 MB embedded flash (80 MHz)	QFN 5*5
Note: All above chips support Wi-Fi b/g/n + BT/BLE Dual Mode connection.			

8. Learning Resources

8.1 Must-Read Documents

Click on the following links to access documents related to ESP32.

- *ESP32 ECO V3 User Guide*
  This document describes differences between V3 and previous ESP32 silicon wafer revisions.
- *ECO and Workarounds for Bugs in ESP32*
  This document details hardware errata and workarounds in the ESP32.
- *ESP-IDF Programming Guide*
  It hosts extensive documentation for ESP-IDF, ranging from hardware guides to API reference.
- *ESP32 Technical Reference Manual*
  The manual provides detailed information on how to use the ESP32 memory and peripherals.
- ESP32 Hardware Resources
  The zip files include schematics, PCB layout, Gerber and BOM list.
- *ESP32 Hardware Design Guidelines*
  The guidelines provide recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including the ESP32 chip, the ESP32 modules and development boards.
- *ESP32 AT Instruction Set and Examples*
  This document introduces the ESP32 AT commands, explains how to use them, and provides examples of several common AT commands.
- *Espressif Products Ordering Information*

8.2 Must-Have Resources

Here are the ESP32-related must-have resources.

- ESP32 BBS
  This is an Engineer-to-Engineer (E2E) Community for ESP32, where you can post questions, share knowledge, explore ideas, and solve problems together with fellow engineers.
- ESP32 GitHub
  ESP32 development projects are freely distributed under Espressif's MIT license on GitHub. This channel of communication has been established to help developers get started with ESP32 and encourage them to share their knowledge of ESP32-related hardware and software.
- ESP32 Tools
  This is a webpage where users can download ESP32 Flash Download Tools and the zip file "ESP32 Certification and Test".
- ESP-IDF
  This webpage links users to the official IoT development framework for ESP32.
- ESP32 Resources
  This webpage provides the links to all available ESP32 documents, SDK and tools.

Appendix A – ESP32 Pin Lists

A.1. Notes on ESP32 Pin Lists

Table 24: Notes on ESP32 Pin Lists

No.	Description
1	In Table IO_MUX, the boxes highlighted in yellow indicate the GPIO pins that are input-only. Please see the following note for further details.
2	GPIO pins 34-39 are input-only. These pins do not feature an output driver or internal pull-up/pull-down circuitry. The pin names are: SENSOR_VP (GPIO36), SENSOR_CAPP (GPIO37), SENSOR_CAPN (GPIO38), SENSOR_VN (GPIO39), VDET_1 (GPIO34), VDET_2 (GPIO35).
3	The pins are grouped into four power domains: VDDA (analog power supply), VDD3P3_RTC (RTC power supply), VDD3P3_CPU (power supply of digital IOs and CPU cores), VDD_SDIO (power supply of SDIO IOs). VDD_SDIO is the output of the internal SDIO-LDO. The voltage of SDIO-LDO can be configured at 1.8 V or be the same as that of VDD3P3_RTC. The strapping pin and eFuse bits determine the default voltage of the SDIO-LDO. Software can change the voltage of the SDIO-LDO by configuring register bits. For details, please see the column "Power Domain" in Table IO_MUX.
4	The functional pins in the VDD3P3_RTC domain are those with analog functions, including the 32 kHz crystal oscillator, ADC, DAC, and the capacitive touch sensor. Please see columns "Analog Function 1~3" in Table IO_MUX.
5	These VDD3P3_RTC pins support the RTC function, and can work during Deep-sleep. For example, an RTC-GPIO can be used for waking up the chip from Deep-sleep.
6	The GPIO pins support up to six digital functions, as shown in columns "Function 1~6" In Table IO_MUX. The function selection registers will be set as "*N*-1", where *N* is the function number. Below are some definitions: • SD_* is for signals of the SDIO slave. • HS1_* is for Port 1 signals of the SDIO host. • HS2_* is for Port 2 signals of the SDIO host. • MT* is for signals of the JTAG. • U0* is for signals of the UART0 module. • U1* is for signals of the UART1 module. • U2* is for signals of the UART2 module. • SPI* is for signals of the SPI01 module. • HSPI* is for signals of the SPI2 module. • VSPI* is for signals of the SPI3 module.

No.	Description
7	Each column about digital "Function" is accompanied by a column about "Type". Please see the following explanations for the meanings of "type" with respect to each "function" they are associated with. For each "Function-*N*", "type" signifies: • I: input only. If a function other than "Function-*N*" is assigned, the input signal of "Function-*N*" is still from this pin. • I1: input only. If a function other than "Function-*N*" is assigned, the input signal of "Function-*N*" is always "1". • I0: input only. If a function other than "Function-*N*" is assigned, the input signal of "Function-*N*" is always "0". • O: output only. • T: high-impedance. • I/O/T: combinations of input, output, and high-impedance according to the function signal. • I1/O/T: combinations of input, output, and high-impedance, according to the function signal. If a function is not selected, the input signal of the function is "1". For example, pin 30 can function as HS1_CMD or SD_CMD, where HS1_CMD is of an "I1/O/T" type. If pin 30 is selected as HS1_CMD, this pin's input and output are controlled by the SDIO host. If pin 30 is not selected as HS1_CMD, the input signal of the SDIO host is always "1".
8	Each digital output pin is associated with its configurable drive strength. Column "Drive Strength" in Table IO_MUX lists the default values. The drive strength of the digital output pins can be configured into one of the following four options: • 0: ~5 mA • 1: ~10 mA • 2: ~20 mA • 3: ~40 mA The default value is 2. The drive strength of the internal pull-up (wpu) and pull-down (wpd) is ~75 μA.
9	Column "At Reset" in Table IO_MUX lists the status of each pin during reset, including input-enable (ie=1), internal pull-up (wpu) and internal pull-down (wpd). During reset, all pins are output-disabled.
10	Column "After Reset" in Table IO_MUX lists the status of each pin immediately after reset, including input-enable (ie=1), internal pull-up (wpu) and internal pull-down (wpd). After reset, each pin is set to "Function 1". The output-enable is controlled by digital Function 1.
11	Table Ethernet_MAC is about the signal mapping inside Ethernet MAC. The Ethernet MAC supports MII and RMII interfaces, and supports both the internal PLL clock and the external clock source. For the MII interface, the Ethernet MAC is with/without the TX_ERR signal. MDC, MDIO, CRS and COL are slow signals, and can be mapped onto any GPIO pin through the GPIO-Matrix.
12	Table GPIO Matrix is for the GPIO-Matrix. The signals of the on-chip functional modules can be mapped onto any GPIO pin. Some signals can be mapped onto a pin by both IO-MUX and GPIO-Matrix, as shown in the column tagged as "Same input signal from IO_MUX core" in Table GPIO Matrix.

No.	Description
13	*In Table GPIO_Matrix the column "Default Value if unassigned" records the default value of the an input signal if no GPIO is assigned to it. The actual value is determined by register GPIO_FUNC*m*_IN_INV_SEL and GPIO_FUNC*m*_IN_SEL. (The value of *m* ranges from 1 to 255.)

A.2. GPIO_Matrix

Table 25: GPIO_Matrix

Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
0	SPICLK_in	0	yes	SPICLK_out	SPICLK_oe
1	SPIQ_in	0	yes	SPIQ_out	SPIQ_oe
2	SPID_in	0	yes	SPID_out	SPID_oe
3	SPIHD_in	0	yes	SPIHD_out	SPIHD_oe
4	SPIWP_in	0	yes	SPIWP_out	SPIWP_oe
5	SPICS0_in	0	yes	SPICS0_out	SPICS0_oe
6	SPICS1_in	0	no	SPICS1_out	SPICS1_oe
7	SPICS2_in	0	no	SPICS2_out	SPICS2_oe
8	HSPICLK_in	0	yes	HSPICLK_out	HSPICLK_oe
9	HSPIQ_in	0	yes	HSPIQ_out	HSPIQ_oe
10	HSPID_in	0	yes	HSPID_out	HSPID_oe
11	HSPICS0_in	0	yes	HSPICS0_out	HSPICS0_oe
12	HSPIHD_in	0	yes	HSPIHD_out	HSPIHD_oe
13	HSPIWP_in	0	yes	HSPIWP_out	HSPIWP_oe
14	U0RXD_in	0	yes	U0TXD_out	1'd1
15	U0CTS_in	0	yes	U0RTS_out	1'd1
16	U0DSR_in	0	no	U0DTR_out	1'd1
17	U1RXD_in	0	yes	U1TXD_out	1'd1
18	U1CTS_in	0	yes	U1RTS_out	1'd1
23	I2S0O_BCK_in	0	no	I2S0O_BCK_out	1'd1
24	I2S1O_BCK_in	0	no	I2S1O_BCK_out	1'd1
25	I2S0O_WS_in	0	no	I2S0O_WS_out	1'd1
26	I2S1O_WS_in	0	no	I2S1O_WS_out	1'd1
27	I2S0I_BCK_in	0	no	I2S0I_BCK_out	1'd1
28	I2S0I_WS_in	0	no	I2S0I_WS_out	1'd1
29	I2CEXT0_SCL_in	1	no	I2CEXT0_SCL_out	1'd1
30	I2CEXT0_SDA_in	1	no	I2CEXT0_SDA_out	1'd1
31	pwm0_sync0_in	0	no	sdio_tohost_int_out	1'd1
32	pwm0_sync1_in	0	no	pwm0_out0a	1'd1
33	pwm0_sync2_in	0	no	pwm0_out0b	1'd1
34	pwm0_f0_in	0	no	pwm0_out1a	1'd1

Signal No.	Input signals	Default value if unassigned	Same input signal from IO_MUX core	Output signals	Output enable of output signals
35	pwm0_f1_in	0	no	pwm0_out1b	1'd1
36	pwm0_f2_in	0	no	pwm0_out2a	1'd1
37	-	0	no	pwm0_out2b	1'd1
39	pcnt_sig_ch0_in0	0	no	-	1'd1
40	pcnt_sig_ch1_in0	0	no	-	1'd1
41	pcnt_ctrl_ch0_in0	0	no	-	1'd1
42	pcnt_ctrl_ch1_in0	0	no	-	1'd1
43	pcnt_sig_ch0_in1	0	no	-	1'd1
44	pcnt_sig_ch1_in1	0	no	-	1'd1
45	pcnt_ctrl_ch0_in1	0	no	-	1'd1
46	pcnt_ctrl_ch1_in1	0	no	-	1'd1
47	pcnt_sig_ch0_in2	0	no	-	1'd1
48	pcnt_sig_ch1_in2	0	no	-	1'd1
49	pcnt_ctrl_ch0_in2	0	no	-	1'd1
50	pcnt_ctrl_ch1_in2	0	no	-	1'd1
51	pcnt_sig_ch0_in3	0	no	-	1'd1
52	pcnt_sig_ch1_in3	0	no	-	1'd1
53	pcnt_ctrl_ch0_in3	0	no	-	1'd1
54	pcnt_ctrl_ch1_in3	0	no	-	1'd1
55	pcnt_sig_ch0_in4	0	no	-	1'd1
56	pcnt_sig_ch1_in4	0	no	-	1'd1
57	pcnt_ctrl_ch0_in4	0	no	-	1'd1
58	pcnt_ctrl_ch1_in4	0	no	-	1'd1
61	HSPICS1_in	0	no	HSPICS1_out	HSPICS1_oe
62	HSPICS2_in	0	no	HSPICS2_out	HSPICS2_oe
63	VSPICLK_in	0	yes	VSPICLK_out_mux	VSPICLK_oe
64	VSPIQ_in	0	yes	VSPIQ_out	VSPIQ_oe
65	VSPID_in	0	yes	VSPID_out	VSPID_oe
66	VSPIHD_in	0	yes	VSPIHD_out	VSPIHD_oe
67	VSPIWP_in	0	yes	VSPIWP_out	VSPIWP_oe
68	VSPICS0_in	0	yes	VSPICS0_out	VSPICS0_oe
69	VSPICS1_in	0	no	VSPICS1_out	VSPICS1_oe
70	VSPICS2_in	0	no	VSPICS2_out	VSPICS2_oe
71	pcnt_sig_ch0_in5	0	no	ledc_hs_sig_out0	1'd1
72	pcnt_sig_ch1_in5	0	no	ledc_hs_sig_out1	1'd1
73	pcnt_ctrl_ch0_in5	0	no	ledc_hs_sig_out2	1'd1
74	pcnt_ctrl_ch1_in5	0	no	ledc_hs_sig_out3	1'd1
75	pcnt_sig_ch0_in6	0	no	ledc_hs_sig_out4	1'd1
76	pcnt_sig_ch1_in6	0	no	ledc_hs_sig_out5	1'd1
77	pcnt_ctrl_ch0_in6	0	no	ledc_hs_sig_out6	1'd1

Signal No.	Input signals	Default value if unassigned	Same input signal from IO_MUX core	Output signals	Output enable of output signals
78	pcnt_ctrl_ch1_in6	0	no	ledc_hs_sig_out7	1'd1
79	pcnt_sig_ch0_in7	0	no	ledc_ls_sig_out0	1'd1
80	pcnt_sig_ch1_in7	0	no	ledc_ls_sig_out1	1'd1
81	pcnt_ctrl_ch0_in7	0	no	ledc_ls_sig_out2	1'd1
82	pcnt_ctrl_ch1_in7	0	no	ledc_ls_sig_out3	1'd1
83	rmt_sig_in0	0	no	ledc_ls_sig_out4	1'd1
84	rmt_sig_in1	0	no	ledc_ls_sig_out5	1'd1
85	rmt_sig_in2	0	no	ledc_ls_sig_out6	1'd1
86	rmt_sig_in3	0	no	ledc_ls_sig_out7	1'd1
87	rmt_sig_in4	0	no	rmt_sig_out0	1'd1
88	rmt_sig_in5	0	no	rmt_sig_out1	1'd1
89	rmt_sig_in6	0	no	rmt_sig_out2	1'd1
90	rmt_sig_in7	0	no	rmt_sig_out3	1'd1
91	-	-	-	rmt_sig_out4	1'd1
92	-	-	-	rmt_sig_out6	1'd1
94	-	-	-	rmt_sig_out7	1'd1
95	I2CEXT1_SCL_in	1	no	I2CEXT1_SCL_out	1'd1
96	I2CEXT1_SDA_in	1	no	I2CEXT1_SDA_out	1'd1
97	host_card_detect_n_1	0	no	host_ccmd_od_pullup_en_n	1'd1
98	host_card_detect_n_2	0	no	host_rst_n_1	1'd1
99	host_card_write_prt_1	0	no	host_rst_n_2	1'd1
100	host_card_write_prt_2	0	no	gpio_sd0_out	1'd1
101	host_card_int_n_1	0	no	gpio_sd1_out	1'd1
102	host_card_int_n_2	0	no	gpio_sd2_out	1'd1
103	pwm1_sync0_in	0	no	gpio_sd3_out	1'd1
104	pwm1_sync1_in	0	no	gpio_sd4_out	1'd1
105	pwm1_sync2_in	0	no	gpio_sd5_out	1'd1
106	pwm1_f0_in	0	no	gpio_sd6_out	1'd1
107	pwm1_f1_in	0	no	gpio_sd7_out	1'd1
108	pwm1_f2_in	0	no	pwm1_out0a	1'd1
109	pwm0_cap0_in	0	no	pwm1_out0b	1'd1
110	pwm0_cap1_in	0	no	pwm1_out1a	1'd1
111	pwm0_cap2_in	0	no	pwm1_out1b	1'd1
112	pwm1_cap0_in	0	no	pwm1_out2a	1'd1
113	pwm1_cap1_in	0	no	pwm1_out2b	1'd1
114	pwm1_cap2_in	0	no	pwm2_out1h	1'd1
115	pwm2_flta	1	no	pwm2_out1l	1'd1
116	pwm2_fltb	1	no	pwm2_out2h	1'd1
117	pwm2_cap1_in	0	no	pwm2_out2l	1'd1
118	pwm2_cap2_in	0	no	pwm2_out3h	1'd1

Signal No.	Input signals	Default value if unassigned	Same input signal from IO_MUX core	Output signals	Output enable of output signals
119	pwm2_cap3_in	0	no	pwm2_out3l	1'd1
120	pwm3_flta	1	no	pwm2_out4h	1'd1
121	pwm3_fltb	1	no	pwm2_out4l	1'd1
122	pwm3_cap1_in	0	no	-	1'd1
123	pwm3_cap2_in	0	no	-	1'd1
124	pwm3_cap3_in	0	no	-	1'd1
140	I2S0I_DATA_in0	0	no	I2S0O_DATA_out0	1'd1
141	I2S0I_DATA_in1	0	no	I2S0O_DATA_out1	1'd1
142	I2S0I_DATA_in2	0	no	I2S0O_DATA_out2	1'd1
143	I2S0I_DATA_in3	0	no	I2S0O_DATA_out3	1'd1
144	I2S0I_DATA_in4	0	no	I2S0O_DATA_out4	1'd1
145	I2S0I_DATA_in5	0	no	I2S0O_DATA_out5	1'd1
146	I2S0I_DATA_in6	0	no	I2S0O_DATA_out6	1'd1
147	I2S0I_DATA_in7	0	no	I2S0O_DATA_out7	1'd1
148	I2S0I_DATA_in8	0	no	I2S0O_DATA_out8	1'd1
149	I2S0I_DATA_in9	0	no	I2S0O_DATA_out9	1'd1
150	I2S0I_DATA_in10	0	no	I2S0O_DATA_out10	1'd1
151	I2S0I_DATA_in11	0	no	I2S0O_DATA_out11	1'd1
152	I2S0I_DATA_in12	0	no	I2S0O_DATA_out12	1'd1
153	I2S0I_DATA_in13	0	no	I2S0O_DATA_out13	1'd1
154	I2S0I_DATA_in14	0	no	I2S0O_DATA_out14	1'd1
155	I2S0I_DATA_in15	0	no	I2S0O_DATA_out15	1'd1
156	-	-	-	I2S0O_DATA_out16	1'd1
157	-	-	-	I2S0O_DATA_out17	1'd1
158	-	-	-	I2S0O_DATA_out18	1'd1
159	-	-	-	I2S0O_DATA_out19	1'd1
160	-	-	-	I2S0O_DATA_out20	1'd1
161	-	-	-	I2S0O_DATA_out21	1'd1
162	-	-	-	I2S0O_DATA_out22	1'd1
163	-	-	-	I2S0O_DATA_out23	1'd1
164	I2S1I_BCK_in	0	no	I2S1I_BCK_out	1'd1
165	I2S1I_WS_in	0	no	I2S1I_WS_out	1'd1
166	I2S1I_DATA_in0	0	no	I2S1O_DATA_out0	1'd1
167	I2S1I_DATA_in1	0	no	I2S1O_DATA_out1	1'd1
168	I2S1I_DATA_in2	0	no	I2S1O_DATA_out2	1'd1
169	I2S1I_DATA_in3	0	no	I2S1O_DATA_out3	1'd1
170	I2S1I_DATA_in4	0	no	I2S1O_DATA_out4	1'd1
171	I2S1I_DATA_in5	0	no	I2S1O_DATA_out5	1'd1
172	I2S1I_DATA_in6	0	no	I2S1O_DATA_out6	1'd1
173	I2S1I_DATA_in7	0	no	I2S1O_DATA_out7	1'd1

Signal No.	Input signals	Default value if unassigned	Same input signal from IO_MUX core	Output signals	Output enable of output signals
174	I2S1I_DATA_in8	0	no	I2S1O_DATA_out8	1'd1
175	I2S1I_DATA_in9	0	no	I2S1O_DATA_out9	1'd1
176	I2S1I_DATA_in10	0	no	I2S1O_DATA_out10	1'd1
177	I2S1I_DATA_in11	0	no	I2S1O_DATA_out11	1'd1
178	I2S1I_DATA_in12	0	no	I2S1O_DATA_out12	1'd1
179	I2S1I_DATA_in13	0	no	I2S1O_DATA_out13	1'd1
180	I2S1I_DATA_in14	0	no	I2S1O_DATA_out14	1'd1
181	I2S1I_DATA_in15	0	no	I2S1O_DATA_out15	1'd1
182	-	-	-	I2S1O_DATA_out16	1'd1
183	-	-	-	I2S1O_DATA_out17	1'd1
184	-	-	-	I2S1O_DATA_out18	1'd1
185	-	-	-	I2S1O_DATA_out19	1'd1
186	-	-	-	I2S1O_DATA_out20	1'd1
187	-	-	-	I2S1O_DATA_out21	1'd1
188	-	-	-	I2S1O_DATA_out22	1'd1
189	-	-	-	I2S1O_DATA_out23	1'd1
190	I2S0I_H_SYNC	0	no	pwm3_out1h	1'd1
191	I2S0I_V_SYNC	0	no	pwm3_out1l	1'd1
192	I2S0I_H_ENABLE	0	no	pwm3_out2h	1'd1
193	I2S1I_H_SYNC	0	no	pwm3_out2l	1'd1
194	I2S1I_V_SYNC	0	no	pwm3_out3h	1'd1
195	I2S1I_H_ENABLE	0	no	pwm3_out3l	1'd1
196	-	-	-	pwm3_out4h	1'd1
197	-	-	-	pwm3_out4l	1'd1
198	U2RXD_in	0	yes	U2TXD_out	1'd1
199	U2CTS_in	0	yes	U2RTS_out	1'd1
200	emac_mdc_i	0	no	emac_mdc_o	emac_mdc_oe
201	emac_mdi_i	0	no	emac_mdo_o	emac_mdo_o_e
202	emac_crs_i	0	no	emac_crs_o	emac_crs_oe
203	emac_col_i	0	no	emac_col_o	emac_col_oe
204	pcmfsync_in	0	no	bt_audio0_irq	1'd1
205	pcmclk_in	0	no	bt_audio1_irq	1'd1
206	pcmdin	0	no	bt_audio2_irq	1'd1
207	-	-	-	ble_audio0_irq	1'd1
208	-	-	-	ble_audio1_irq	1'd1
209	-	-	-	ble_audio2_irq	1'd1
210	-	-	-	pcmfsync_out	pcmfsync_en
211	-	-	-	pcmclk_out	pcmclk_en
212	-	-	-	pcmdout	pcmdout_en
213	-	-	-	ble_audio_sync0_p	1'd1

Signal No.	Input signals	Default value if unassigned	Same input signal from IO_MUX core	Output signals	Output enable of output signals
214	-	-	-	ble_audio_sync1_p	1'd1
215	-	-	-	ble_audio_sync2_p	1'd1
224	-	-	-	sig_in_func224	1'd1
225	-	-	-	sig_in_func225	1'd1
226	-	-	-	sig_in_func226	1'd1
227	-	-	-	sig_in_func227	1'd1
228	-	-	-	sig_in_func228	1'd1

A.3. Ethernet_MAC

Table 26: Ethernet_MAC

PIN Name	Function6	MII (int_osc)	MII (ext_osc)	RMII (int_osc)	RMII (ext_osc)
GPIO0	EMAC_TX_CLK	TX_CLK (I)	TX_CLK (I)	CLK_OUT(O)	EXT_OSC_CLK(I)
GPIO5	EMAC_RX_CLK	RX_CLK (I)	RX_CLK (I)	-	-
GPIO21	EMAC_TX_EN	TX_EN(O)	TX_EN(O)	TX_EN(O)	TX_EN(O)
GPIO19	EMAC_TXD0	TXD[0](O)	TXD[0](O)	TXD[0](O)	TXD[0](O)
GPIO22	EMAC_TXD1	TXD[1](O)	TXD[1](O)	TXD[1](O)	TXD[1](O)
MTMS	EMAC_TXD2	TXD[2](O)	TXD[2](O)	-	-
MTDI	EMAC_TXD3	TXD[3](O)	TXD[3](O)	-	-
MTCK	EMAC_RX_ER	RX_ER(I)	RX_ER(I)	-	-
GPIO27	EMAC_RX_DV	RX_DV(I)	RX_DV(I)	CRS_DV(I)	CRS_DV(I)
GPIO25	EMAC_RXD0	RXD[0](I)	RXD[0](I)	RXD[0](I)	RXD[0](I)
GPIO26	EMAC_RXD1	RXD[1](I)	RXD[1](I)	RXD[1](I)	RXD[1](I)
U0TXD	EMAC_RXD2	RXD[2](I)	RXD[2](I)	-	-
MTDO	EMAC_RXD3	RXD[3](I)	RXD[3](I)	-	-
GPIO16	EMAC_CLK_OUT	CLK_OUT(O)	-	CLK_OUT(O)	-
GPIO17	EMAC_CLK_OUT_180	CLK_OUT_180(O)	-	CLK_OUT_180(O)	-
GPIO4	EMAC_TX_ER	TX_ERR(O)*	TX_ERR(O)*	-	-
In GPIO Matrix*	-	MDC(O)	MDC(O)	MDC(O)	MDC(O)
In GPIO Matrix*	-	MDIO(IO)	MDIO(IO)	MDIO(IO)	MDIO(IO)
In GPIO Matrix*	-	CRS(I)	CRS(I)	-	-
In GPIO Matrix*	-	COL(I)	COL(I)	-	-
*Notes: 1. The GPIO Matrix can be any GPIO. 2. The TX_ERR (O) is optional.					

A.4. IO_MUX

For the list of IO_MUX pins, please see the next page.

IO_MUX

Pin No.	Power Supply Pin	Analog Pin	Digital Pin	Power Domain	Analog Function1	Analog Function2	Analog Function3	RTC Function1	RTC Function2	Function1	Type	Function2	Type	Function3	Type	Function4	Type	Function5	Type	Function6	Type	Drive Strength (2'd2: 20 mA)	At Reset	After Reset
1	VDDA			VDDA supply in																				
2		LNA_IN		VDD3P3																				
3	VDD3P3			VDD3P3 supply in																				
4	VDD3P3			VDD3P3 supply in																				
5		SENSOR_VP		VDD3P3_RTC	ADC_H	ADC1_CH0		RTC_GPIO0		GPIO36	I			GPIO36	I								oe=0, ie=0	oe=0, ie=0
6		SENSOR_CAPP		VDD3P3_RTC	ADC_H	ADC1_CH1		RTC_GPIO1		GPIO37	I			GPIO37	I								oe=0, ie=0	oe=0, ie=0
7		SENSOR_CAPN		VDD3P3_RTC	ADC_H	ADC1_CH2		RTC_GPIO2		GPIO38	I			GPIO38	I								oe=0, ie=0	oe=0, ie=0
8		SENSOR_VN		VDD3P3_RTC	ADC_H	ADC1_CH3		RTC_GPIO3		GPIO39	I			GPIO39	I								oe=0, ie=0	oe=0, ie=0
9		CHIP_PU		VDD3P3_RTC																				
10		VDET_1		VDD3P3_RTC		ADC1_CH6		RTC_GPIO4		GPIO34	I			GPIO34	I								oe=0, ie=0	oe=0, ie=0
11		VDET_2		VDD3P3_RTC		ADC1_CH7		RTC_GPIO5		GPIO35	I			GPIO35	I								oe=0, ie=0	oe=0, ie=0
12		32K_XP		VDD3P3_RTC	XTAL_32K_P	ADC1_CH4	TOUCH9	RTC_GPIO9		GPIO32	I/O/T			GPIO32	I/O/T							2'd2	oe=0, ie=0	oe=0, ie=0
13		32K_XN		VDD3P3_RTC	XTAL_32K_N	ADC1_CH5	TOUCH8	RTC_GPIO8		GPIO33	I/O/T			GPIO33	I/O/T							2'd2	oe=0, ie=0	oe=0, ie=0
14			GPIO25	VDD3P3_RTC	DAC_1	ADC2_CH8		RTC_GPIO6		GPIO25	I/O/T			GPIO25	I/O/T					EMAC_RXD0	I	2'd2	oe=0, ie=0	oe=0, ie=0
15			GPIO26	VDD3P3_RTC	DAC_2	ADC2_CH9		RTC_GPIO7		GPIO26	I/O/T			GPIO26	I/O/T					EMAC_RXD1	I	2'd2	oe=0, ie=0	oe=0, ie=0
16			GPIO27	VDD3P3_RTC		ADC2_CH7	TOUCH7	RTC_GPIO17		GPIO27	I/O/T			GPIO27	I/O/T					EMAC_RX_DV	I	2'd2	oe=0, ie=0	oe=0, ie=1
17			MTMS	VDD3P3_RTC		ADC2_CH6	TOUCH6	RTC_GPIO16		MTMS	I0	HSPICLK	I/O/T	GPIO14	I/O/T	HS2_CLK	O	SD_CLK	I0	EMAC_TXD2	O	2'd2	oe=0, ie=0	oe=0, ie=1
18			MTDI	VDD3P3_RTC		ADC2_CH5	TOUCH5	RTC_GPIO15		MTDI	I1	HSPIQ	I/O/T	GPIO12	I/O/T	HS2_DATA2	I1/O/T	SD_DATA2	I1/O/T	EMAC_TXD3	O	2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd
19	VDD3P3_RTC			VDD3P3_RTC supply in																				
20			MTCK	VDD3P3_RTC		ADC2_CH4	TOUCH4	RTC_GPIO14		MTCK	I1	HSPID	I/O/T	GPIO13	I/O/T	HS2_DATA3	I1/O/T	SD_DATA3	I1/O/T	EMAC_RX_ER	I	2'd2	oe=0, ie=0	oe=0, ie=1
21			MTDO	VDD3P3_RTC		ADC2_CH3	TOUCH3	RTC_GPIO13	I2C_SDA	MTDO	O/T	HSPICS0	I/O/T	GPIO15	I/O/T	HS2_CMD	I1/O/T	SD_CMD	I1/O/T	EMAC_RXD3	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
22			GPIO2	VDD3P3_RTC		ADC2_CH2	TOUCH2	RTC_GPIO12	I2C_SCL	GPIO2	I/O/T	HSPIWP	I/O/T	GPIO2	I/O/T	HS2_DATA0	I1/O/T	SD_DATA0	I1/O/T			2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd
23			GPIO0	VDD3P3_RTC		ADC2_CH1	TOUCH1	RTC_GPIO11	I2C_SDA	GPIO0	I/O/T	CLK_OUT1	O	GPIO0	I/O/T					EMAC_TX_CLK	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
24			GPIO4	VDD3P3_RTC		ADC2_CH0	TOUCH0	RTC_GPIO10	I2C_SCL	GPIO4	I/O/T	HSPIHD	I/O/T	GPIO4	I/O/T	HS2_DATA1	I1/O/T	SD_DATA1	I1/O/T	EMAC_TX_ER	O	2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd
25			GPIO16	VDD_SDIO						GPIO16	I/O/T			GPIO16	I/O/T	HS1_DATA4	I1/O/T	U2RXD	I1	EMAC_CLK_OUT	O	2'd2	oe=0, ie=0	oe=0, ie=1
26	VDD_SDIO			VDD_SDIO supply out/in																				
27			GPIO17	VDD_SDIO						GPIO17	I/O/T			GPIO17	I/O/T	HS1_DATA5	I1/O/T	U2TXD	O	EMAC_CLK_OUT_180	O	2'd2	oe=0, ie=0	oe=0, ie=1
28			SD_DATA_2	VDD_SDIO						SD_DATA2	I1/O/T	SPIHD	I/O/T	GPIO9	I/O/T	HS1_DATA2	I1/O/T	U1RXD	I1			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
29			SD_DATA_3	VDD_SDIO						SD_DATA3	I0/O/T	SPIWP	I/O/T	GPIO10	I/O/T	HS1_DATA3	I1/O/T	U1TXD	O			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
30			SD_CMD	VDD_SDIO						SD_CMD	I1/O/T	SPICS0	I/O/T	GPIO11	I/O/T	HS1_CMD	I1/O/T	U1RTS	O			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
31			SD_CLK	VDD_SDIO						SD_CLK	I0	SPICLK	I/O/T	GPIO6	I/O/T	HS1_CLK	O	U1CTS	I1			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
32			SD_DATA_0	VDD_SDIO						SD_DATA0	I1/O/T	SPIQ	I/O/T	GPIO7	I/O/T	HS1_DATA0	I1/O/T	U2RTS	O			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
33			SD_DATA_1	VDD_SDIO						SD_DATA1	I1/O/T	SPID	I/O/T	GPIO8	I/O/T	HS1_DATA1	I1/O/T	U2CTS	I1			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
34			GPIO5	VDD3P3_CPU						GPIO5	I/O/T	VSPICS0	I/O/T	GPIO5	I/O/T	HS1_DATA6	I1/O/T			EMAC_RX_CLK	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
35			GPIO18	VDD3P3_CPU						GPIO18	I/O/T	VSPICLK	I/O/T	GPIO18	I/O/T	HS1_DATA7	I1/O/T					2'd2	oe=0, ie=0	oe=0, ie=1
36			GPIO23	VDD3P3_CPU						GPIO23	I/O/T	VSPID	I/O/T	GPIO23	I/O/T	HS1_STROBE	I0					2'd2	oe=0, ie=0	oe=0, ie=1
37	VDD3P3_CPU			VDD3P3_CPU supply in																				
38			GPIO19	VDD3P3_CPU						GPIO19	I/O/T	VSPIQ	I/O/T	GPIO19	I/O/T	U0CTS	I1			EMAC_TXD0	O	2'd2	oe=0, ie=0	oe=0, ie=1
39			GPIO22	VDD3P3_CPU						GPIO22	I/O/T	VSPIWP	I/O/T	GPIO22	I/O/T	U0RTS	O			EMAC_TXD1	O	2'd2	oe=0, ie=0	oe=0, ie=1
40			U0RXD	VDD3P3_CPU						U0RXD	I1	CLK_OUT2	O	GPIO3	I/O/T							2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
41			U0TXD	VDD3P3_CPU						U0TXD	O	CLK_OUT3	O	GPIO1	I/O/T					EMAC_RXD2	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
42			GPIO21	VDD3P3_CPU						GPIO21	I/O/T	VSPIHD	I/O/T	GPIO21	I/O/T					EMAC_TX_EN	O	2'd2	oe=0, ie=0	oe=0, ie=1
43	VDDA			VDDA supply in																				
44		XTAL_N		VDDA																				
45		XTAL_P		VDDA																				
46	VDDA			VDDA supply in																				
47		CAP2		VDDA																				
48		CAP1		VDDA																				
Total Number	8	14	26																					

Notes:

- wpu: weak pull-up;
- wpd: weak pull-down;
- ie: input enable;
- oe: output enable;
- Please see *Table: Notes on ESP32 Pin Lists* for more information.（请参考表：管脚清单说明。）

Revision History

Date	Version	Release notes
2020-04-27	V3.4	Added one chip variant: ESP32-U4WDH Updated some figures in Table 6, 16, 17, 19, 21, 22 Added a note under Table 18
2020.01	V3.3	Added two chip variants: ESP32-D0WD-V3 and ESP32-D0WDQ6-V3. Added a note under Table 7.
2019.10	V3.2	Updated Figure 5: *ESP32 Power-up and Reset Timing*.
2019.07	V3.1	Added pin-pin mapping between ESP32-D2WD and the embedded flash under Table 1 *Pin Description*; Updated Figure 10 *ESP32 Part Number*.
2019.04	V3.0	Added information about the setup and hold times for the strapping pins in Section 2.4: Strapping Pins.
2019.02	V2.9	Applied new formatting to Table 1: Pin Description; Fixed typos with respect to the ADC1 channel mappings in Table 10: Peripheral Pin Configurations.
2019.01	V2.8	Changed the RF power control range in Table 18, Table 20 and Table 22 from –12 ~ +12 to –12 ~ +9 dBm; Small text changes.
2018.11	V2.7	Updated Section 1.5; Updated pin statuses at reset and after reset in Table IO_MUX.
2018.10	V2.6	Updated QFN package drawings in Chapter 6: Package Information.
2018.08	V2.5	• Added "Cumulative IO output current" entry to Table 11: Absolute Maximum Ratings; • Added more parameters to Table 13: DC Characteristics; • Changed the power domain names in Table IO_MUX to be consistent with the pin names.
2018.07	V2.4	• Deleted information on Packet Traffic Arbitration (PTA); • Added Figure 5: ESP32 Power-up and Reset Timing in Section 2.3: Power Scheme; • Added the power consumption of dual-core SoCs in Table 6: Power Consumption by Power Modes; • Updated section 4.1.2: Analog-to-Digital Converter (ADC).
2018.06	V2.3	Added the power consumption at CPU frequency of 160 MHz in Table 6: Power Consumption by Power Modes.

Date	Version	Release notes
2018.05	V2.2	• Changed the voltage range of VDD3P3_RTC from 1.8-3.6V to 2.3-3.6V in Table 1: Pin Description; • Updated Section 2.3: Power Scheme; • Updated Section 3.1.3: External Flash and SRAM; • Updated Table 6: Power Consumption by Power Modes; • Deleted content about temperature sensor; Changes to electrical characteristics: • Updated Table 11: Absolute Maximum Ratings; • Added Table 12: Recommended Operating Conditions; • Added Table 13: DC Characteristics; • Added Table 14: Reliability Qualifications; • Updated the values of "Gain control step" and "Adjacent channel transmit power" in Table 18: Transmitter Characteristics - Basic Data Rate; • Updated the values of "Gain control step", "$\pi/4$ DQPSK modulation accuracy", "8 DPSK modulation accuracy" and "In-band spurious emissions" in Table 20: Transmitter Characteristics – Enhanced Data Rate; • Updated the values of "Gain control step", "Adjacent channel transmit power" in Table 22: Transmitter Characteristics - BLE.
2018.01	V2.1	• Deleted software-specific features; • Deleted information on LNA pre-amplifier; • Specified the CPU speed and flash speed of ESP32-D2WD; • Added notes to Section 2.3: Power Scheme.
2017.12	V2.0	Added a note on the sequence of pin number in Chapter 6.
2017.10	V1.9	• Updated the description of the pin CHIP_PU in Table 1; • Added a note to Section 2.3: Power Scheme; • Updated the description of the chip's system reset in Section 2.4: Strapping Pins; • Added a description of antenna diversity and selection to Section 3.5.1; • Deleted "Association sleep pattern" in Table 6 and added notes to Active sleep and Modem-sleep.
2017.08	V1.8	• Added Table 4.2 in Section 4; • Corrected a typo in Figure 1.

Date	Version	Release notes
2017.08	V1.7	• Changed the transmitting power to +12 dBm; the sensitivity of NZIF receiver to -97 dBm in Section 1.3; • Added a note to Table 1 Pin Description; • Added 160 MHz clock frequency in section 3.1.1; • Changed the transmitting power from 21 dBm to 20.5 dBm in Section 3.5.1; • Changed the dynamic control range of class-1, class-2 and class-3 transmit output powers to "up to 24 dBm"; and changed the dynamic range of NZIF receiver sensitivity to "over 97 dB" in Section 3.6.1; • Updated Table 6: Power Consumption by Power Modes, and added two notes to it; • Updated sections 4.1.1, 4.1.9; • Updated Table 11: Absolute Maximum Ratings; • Updated Table 15: RF Power Consumption Specifications, and changed the duty cycle on which the transmitters' measurements are based by 50%. • Updated Table 16: Wi-Fi Radio Characteristics and added a note on "Output impedance" to it; • Updated parameter "Sensitivity" in Table 17, 19, 21; • Updated parameters "RF transmit power" and "RF power control range", and added parameter "Gain control step" in Table 18, 20, 22; • Deleted Chapters: "Touch Sensor" and "Code Examples"; • Added a link to certification download.
2017.06	V1.6	Corrected two typos: • Changed the number of external components to 20 in Section 1.1.2; • Changed the number of GPIO pins to 34 in Section 4.1.1.
2017.06	V1.5	• Changed the power supply range in Section: 1.4.1 CPU and Memory; • Updated the note in Section 2.3: Power Scheme; • Updated Table 11: Absolute Maximum Ratings; • Changed the drive strength values of the digital output pins in Note 8, in Table 24: Notes on ESP32 Pin Lists; • Added the option to subscribe for notifications of documentation changes.
2017.05	V1.4	• Added a note to the frequency of the external crystal oscillator in Section 1.4.2: Clocks and Timers; • Added a note to Section 2.4: Strapping Pins; • Updated Section 3.7: RTC and Low-Power Management; • Changed the maximum driving capability from 12 mA to 80 mA, in Table 11: Absolulte Maximum Ratings; • Changed the input impedance value of 50Ω, in Table 16: Wi-Fi Radio Characteristics, to output impedance value of 30+j10 Ω; • Added a note to No.8 in Table 24: Notes on ESP32 Pin Lists; • Deleted GPIO20 in Table IO_MUX.
2017.04	V1.3	• Added Appendix: ESP32 Pin Lists; • Updated Table: Wi-Fi Radio Characteristics; • Updated Figure: ESP32 Pin Layout (for QFN 5*5).

Date	Version	Release notes
2017.03	V1.2	• Added a note to Table: Pin Description; • Updated the note in Section: Internal Memory.
2017.02	V1.1	• Added Chapter: Part Number and Ordering Information; • Updated Section: MCU and Advanced Features; • Updated Section: Block Diagram; • Updated Chapter: Pin Definitions; • Updated Section: CPU and Memory; • Updated Section: Audio PLL Clock; • Updated Section: Absolute Maximum Ratings; • Updated Chapter: Package Information; • Updated Chapter: Learning Resources.
2016.08	V1.0	First release.

L293x Quadruple Half-H Drivers

1 Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

2 Applications

- Stepper Motor Drivers
- DC Motor Drivers
- Latching Relay Drivers

3 Description

The L293 and L293D devices are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.

The L293 and L293D are characterized for operation from 0°C to 70°C.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
L293NE	PDIP (16)	19.80 mm × 6.35 mm
L293DNE	PDIP (16)	19.80 mm × 6.35 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Logic Diagram

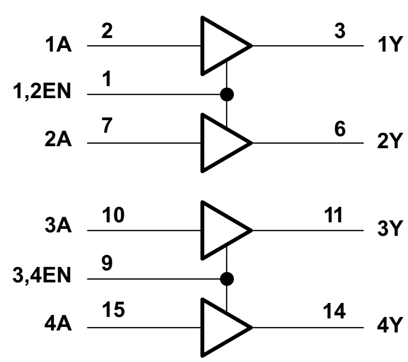


Table of Contents

4 Revision History

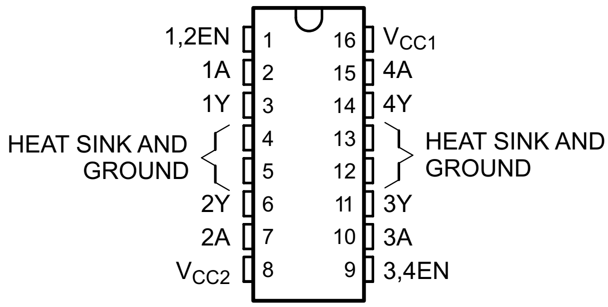
NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision C (November 2004) to Revision D **Page**

- Removed *Ordering Information* table 1
- Added *ESD Ratings* and *Thermal Information* tables, *Feature Description* section, *Device Functional Modes*, *Application and Implementation* section, *Power Supply Recommendations* section, *Layout* section, *Device and Documentation Support* section, and *Mechanical, Packaging, and Orderable Information* section. 1

5 Pin Configuration and Functions

NE Package
16-Pin PDIP
Top View



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V_{CC1}	16	—	5-V supply for internal logic translation
V_{CC2}	8	—	Power VCC for drivers 4.5 V to 36 V

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)

	MIN	MAX	UNIT
Supply voltage, V_{CC1} (2)		36	V
Output supply voltage, V_{CC2}		36	V
Input voltage, V_I		7	V
Output voltage, V_O	–3	V_{CC2} + 3	V
Peak output current, I_O (nonrepetitive, t ≤ 5 ms): L293	–2	2	A
Peak output current, I_O (nonrepetitive, t ≤ 100 µs): L293D	–1.2	1.2	A
Continuous output current, I_O: L293	–1	1	A
Continuous output current, I_O: L293D	–600	600	mA
Maximum junction temperature, T_J		150	°C
Storage temperature, T_{stg}	–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values are with respect to the network ground terminal.

6.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001(1)	±2000	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101(2)	±1000	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

			MIN	NOM	MAX	UNIT
	Supply voltage	V_{CC1}	4.5		7	V
		V_{CC2}	V_{CC1}		36	
V_{IH}	High-level input voltage	V_{CC1} ≤ 7 V	2.3		V_{CC1}	V
		V_{CC1} ≥ 7 V	2.3		7	V
V_{IL}	Low-level output voltage		–0.3(1)		1.5	V
T_A	Operating free-air temperature		0		70	°C

(1) The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

6.4 Thermal Information

THERMAL METRIC(1)		L293, L293D NE (PDIP) 16 PINS	UNIT
$R_{\theta JA}$	Junction-to-ambient thermal resistance (2)	36.4	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	22.5	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	16.5	°C/W
ψ_{JT}	Junction-to-top characterization parameter	7.1	°C/W
ψ_{JB}	Junction-to-board characterization parameter	16.3	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.
(2) The package thermal impedance is calculated in accordance with JESD 51-7.

6.5 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		L293: I_{OH} = −1 A		V_{CC2} – 1.8	V_{CC2} – 1.4		V
			L293D: I_{OH} = − 0.6 A					
V_{OL}	Low-level output voltage		L293: I_{OL} = 1 A			1.2	1.8	V
			L293D: I_{OL} = 0.6 A					
V_{OKH}	High-level output clamp voltage		L293D: I_{OK} = –0.6 A			V_{CC2} + 1.3		V
V_{OKL}	Low-level output clamp voltage		L293D: I_{OK} = 0.6 A			1.3		V
I_{IH}	High-level input current	A	V_I = 7 V			0.2	100	µA
		EN				0.2	10	
I_{IL}	Low-level input current	A	V_I = 0			–3	–10	µA
		EN				–2	–100	
I_{CC1}	Logic supply current		I_O = 0	All outputs at high level		13	22	mA
				All outputs at low level		35	60	
				All outputs at high impedance		8	24	
I_{CC2}	Output supply current		I_O = 0	All outputs at high level		14	24	mA
				All outputs at low level		2	6	
				All outputs at high impedance		2	4	

6.6 Switching Characteristics

over operating free-air temperature range (unless otherwise noted) V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output from A input	L293NE, L293DNE	C_L = 30 pF, See Figure 2		800		ns
		L293DWP, L293N L293DN			750		
t_{PHL}	Propagation delay time, high-to-low-level output from A input	L293NE, L293DNE			400		ns
		L293DWP, L293N L293DN			200		
t_{TLH}	Transition time, low-to-high-level output	L293NE, L293DNE			300		ns
		L293DWP, L293N L293DN			100		
t_{THL}	Transition time, high-to-low-level output	L293NE, L293DNE			300		ns
		L293DWP, L293N L293DN			350		

6.7 Typical Characteristics

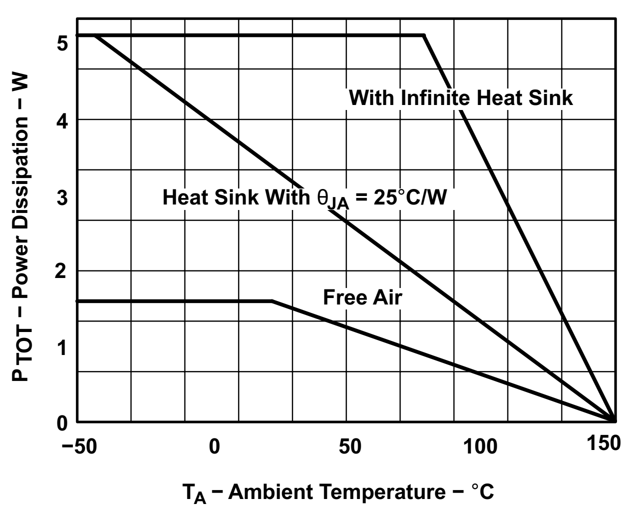
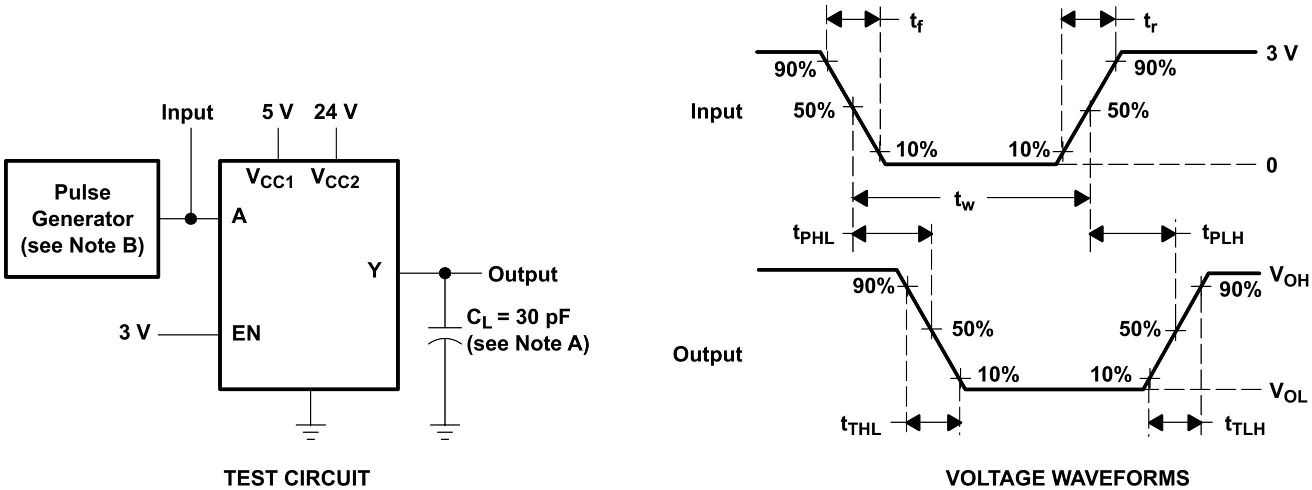


Figure 1. Maximum Power Dissipation vs Ambient Temperature

7 Parameter Measurement Information



NOTES: A. C_L includes probe and jig capacitance.

B. The pulse generator has the following characteristics: $t_r \leq 10$ ns, $t_f \leq 10$ ns, $t_w = 10$ μs, PRR = 5 kHz, $Z_O = 50\ \Omega$.

Figure 2. Test Circuit and Voltage Waveforms

8 Detailed Description

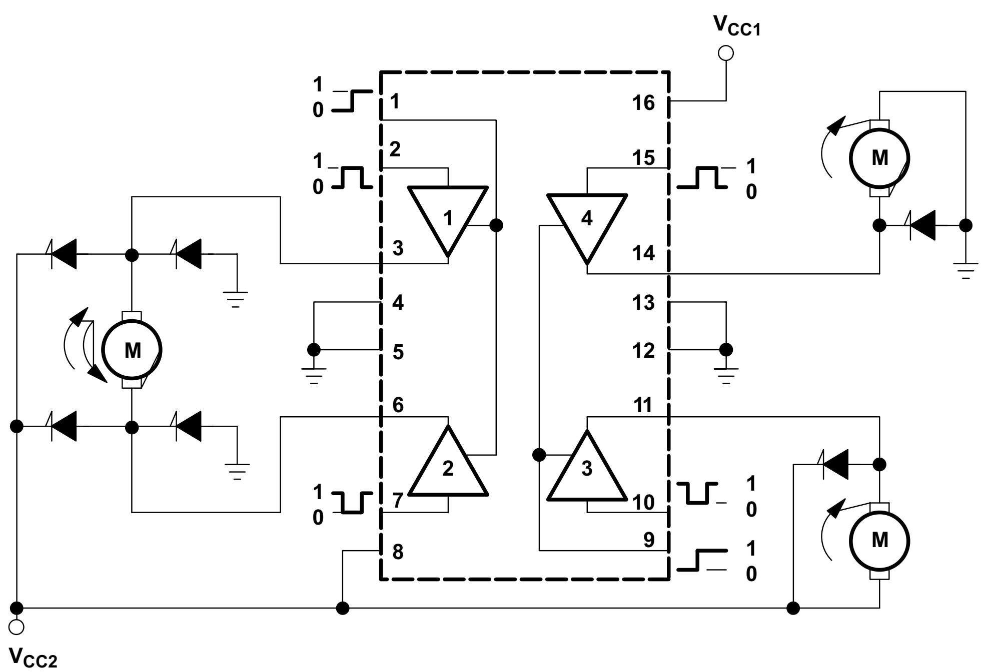
8.1 Overview

The L293 and L293D are quadruple high-current half-H drivers. These devices are designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current and high-voltage loads. All inputs are TTL compatible and tolerant up to 7 V.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. On the L293D, these diodes are integrated to reduce system complexity and overall system size. A V_{CC1} terminal, separate from V_{CC2}, is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0°C to 70°C.

8.2 Functional Block Diagram



Output diodes are internal in L293D.

8.3 Feature Description

The L293x has TTL-compatible inputs and high voltage outputs for inductive load driving. Current outputs can get up to 2 A using the L293.

8.4 Device Functional Modes

Table 1 lists the fuctional modes of the L293x.

Table 1. Function Table (Each Driver)(1)

INPUTS (2)		OUTPUT (Y)
A	EN	
H	H	H
L	H	L
X	L	Z

(1) H = high level, L = low level, X = irrelevant, Z = high impedance (off)
(2) In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

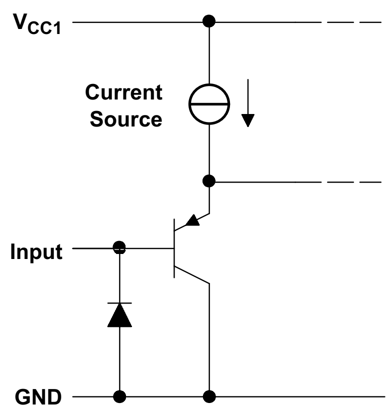


Figure 3. Schematic of Inputs for the L293x

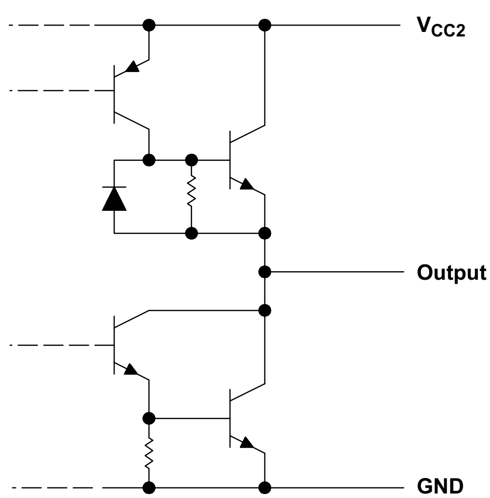


Figure 4. Schematic of Outputs for the L293

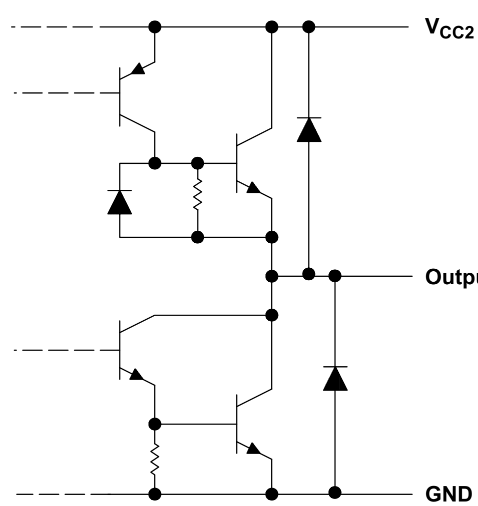


Figure 5. Schematic of Outputs for the L293D

9 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

9.1 Application Information

A typical application for the L293 device is driving a two-phase motor. Below is an example schematic displaying how to properly connect a two-phase motor to the L293 device.

Provide a 5-V supply to V_{CC1} and valid logic input levels to data and enable inputs. V_{CC2} must be connected to a power supply capable of supplying the needed current and voltage demand for the loads connected to the outputs.

9.2 Typical Application

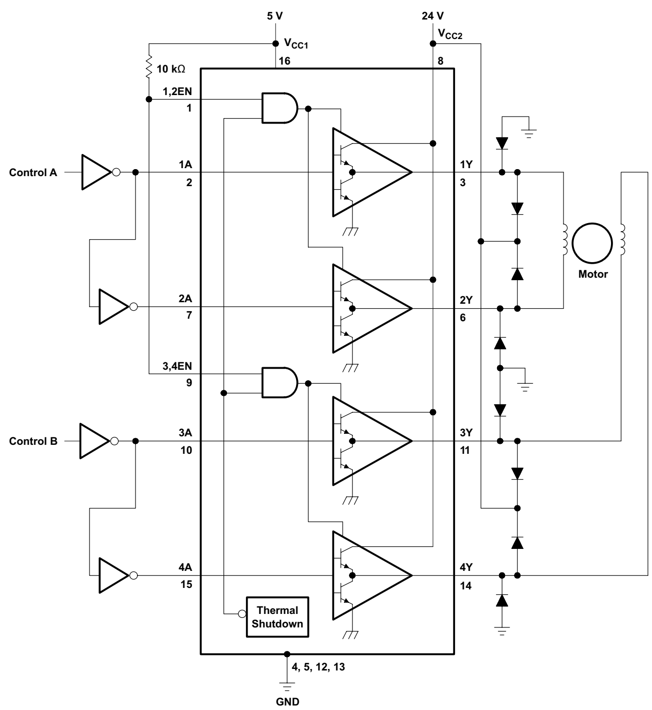


Figure 6. Two-Phase Motor Driver (L293)

9.2.1 Design Requirements

The design techniques in the application above as well as the applications below should fall within the following design requirements.

1. V_{CC1} should fall within the limits described in the *Recommended Operating Conditions*.
2. V_{CC2} should fall within the limits described in the *Recommended Operating Conditions*.
3. The current per channel should not exceed 1 A for the L293 (600mA for the L293D).

9.2.2 Detailed Design Procedure

When designing with the L293 or L293D, careful consideration should be made to ensure the device does not exceed the operating temperature of the device. Proper heatsinking will allow for operation over a larger range of current per channel. Refer to the *Power Supply Recommendations* as well as the *Layout Example*.

Typical Application (continued)

9.2.3 Application Curve

Refer to *Power Supply Recommendations* for additional information with regards to appropriate power dissipation. Figure 7 describes thermal dissipation based on Figure 14.

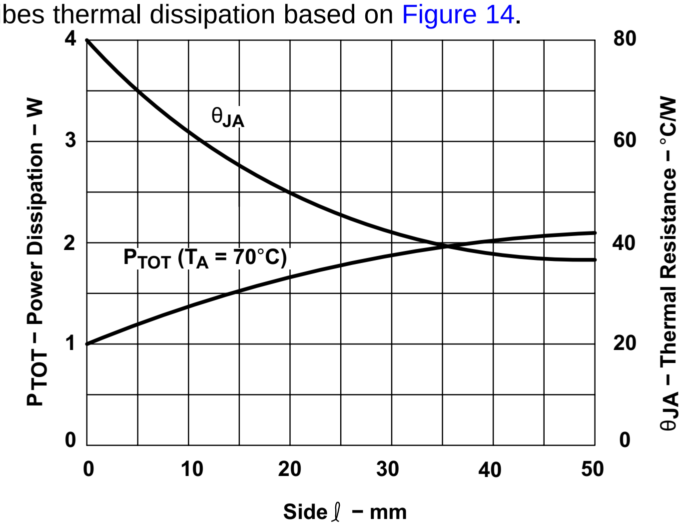


Figure 7. Maximum Power and Junction vs Thermal Resistance

9.3 System Examples

9.3.1 L293D as a Two-Phase Motor Driver

Figure 8 below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.

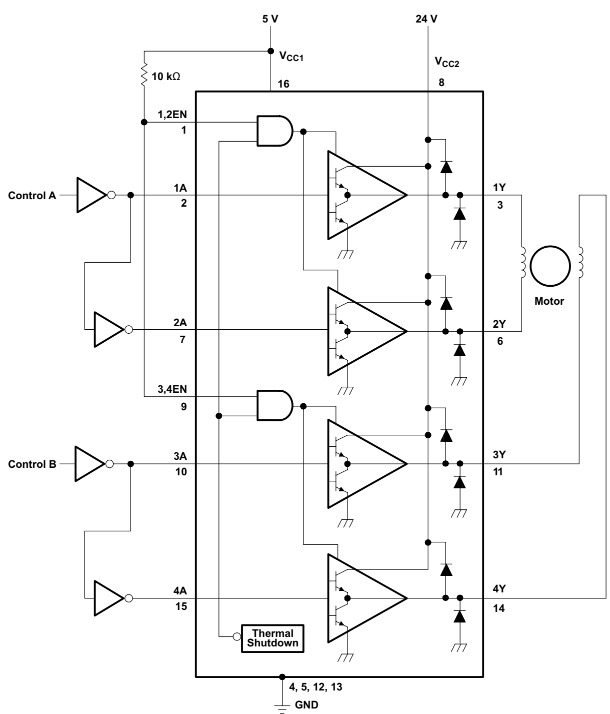
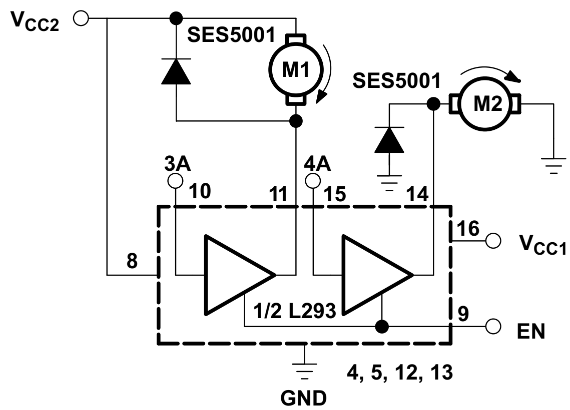


Figure 8. Two-Phase Motor Driver (L293D)

System Examples (continued)

9.3.2 DC Motor Controls

Figure 9 and Figure 10 below depict a typical setup for using the L293 device as a controller for DC motors. Note that the L293 device can be used as a simple driver for a motor to turn on and off in one direction, and can also be used to drive a motor in both directions. Refer to the function tables below to understand unidirectional vs bidirectional motor control. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.



Connections to ground and to supply voltage

Figure 9. DC Motor Controls

Table 2. Unidirectional DC Motor Control

EN	3A	M1 (1)	4A	M2
H	H	Fast motor stop	H	Run
H	L	run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

(1) L = low, H = high, X = don't care

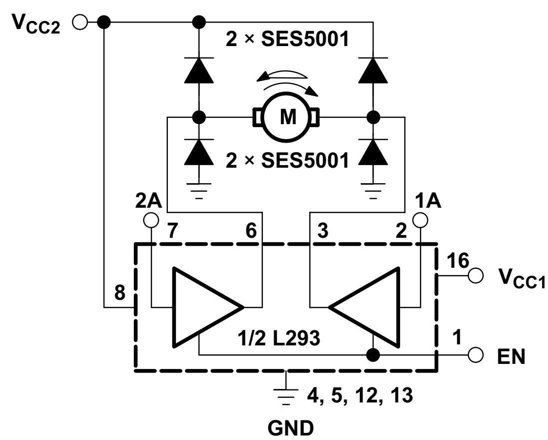


Figure 10. Bidirectional DC Motor Control

Table 3. Bidrectional DC Motor Control

EN	1A	2A	FUNCTION (1)
H	L	H	Turn right
H	H	L	Turn left

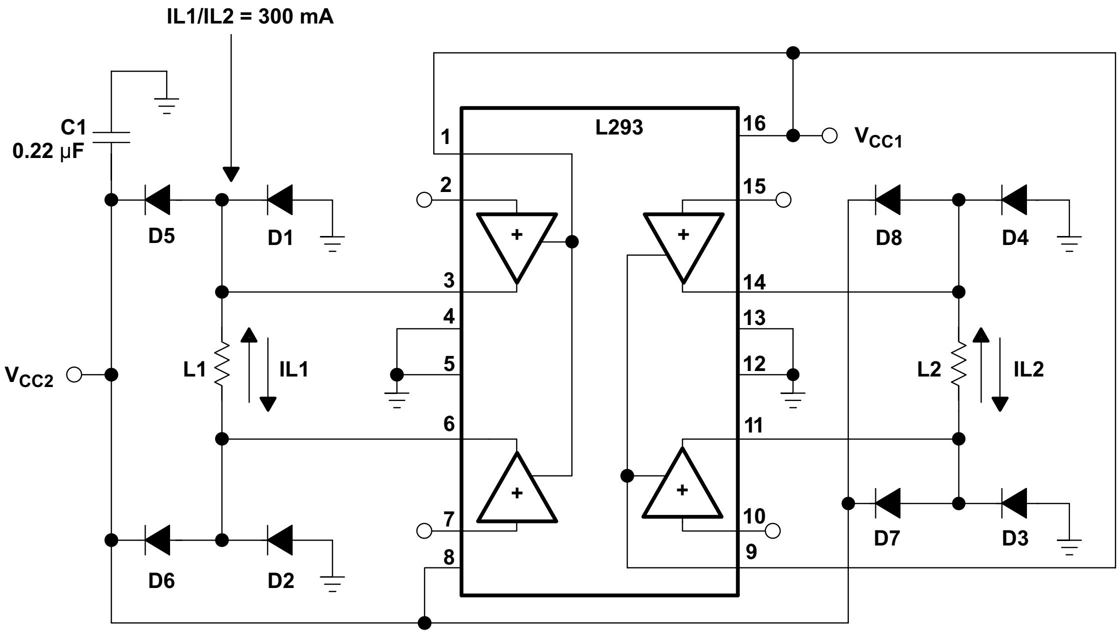
(1) L = low, H = high, X = don't care

Table 3. Bidirectional DC Motor Control (continued)

EN	1A	2A	FUNCTION (1)
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Free-running motor stop

9.3.3 Bipolar Stepping-Motor Control

Figure 11 below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.



D1−D8 = SES5001

Figure 11. Bipolar Stepping-Motor Control

10 Power Supply Recommendations

V_{CC1} is 5 V ± 0.5 V and V_{CC2} can be same supply as V_{CC1} or a higher voltage supply with peak voltage up to 36 V. Bypass capacitors of 0.1 uF or greater should be used at V_{CC1} and V_{CC2} pins. There are no power up or power down supply sequence order requirements.

Properly heatsinking the L293 when driving high-current is critical to design. The Rthj-amp of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heat sink.

Figure 14 shows the maximum package power PTOT and the θJA as a function of the side of two equal square copper areas having a thickness of 35 μm (see Figure 14). In addition, an external heat sink can be used (see Figure 12).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not exceed 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.

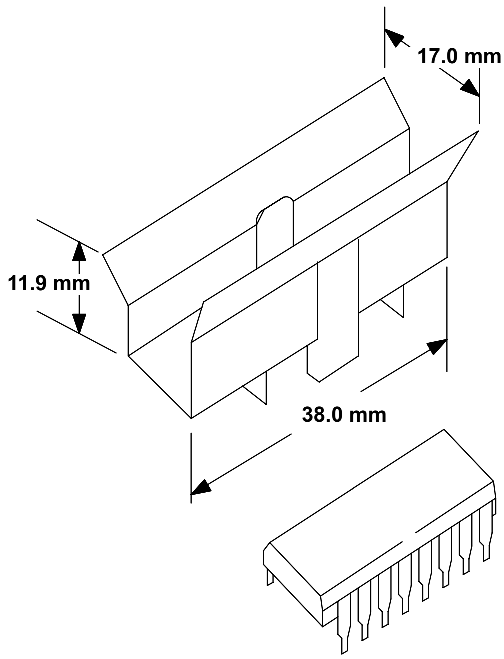


Figure 12. External Heat Sink Mounting Example (θ_{JA} = 25°C/W)

11 Layout

11.1 Layout Guidelines

Place the device near the load to keep output traces short to reduce EMI. Use solid vias to transfer heat from ground pins to ground plane of the printed-circuit-board.

11.2 Layout Example

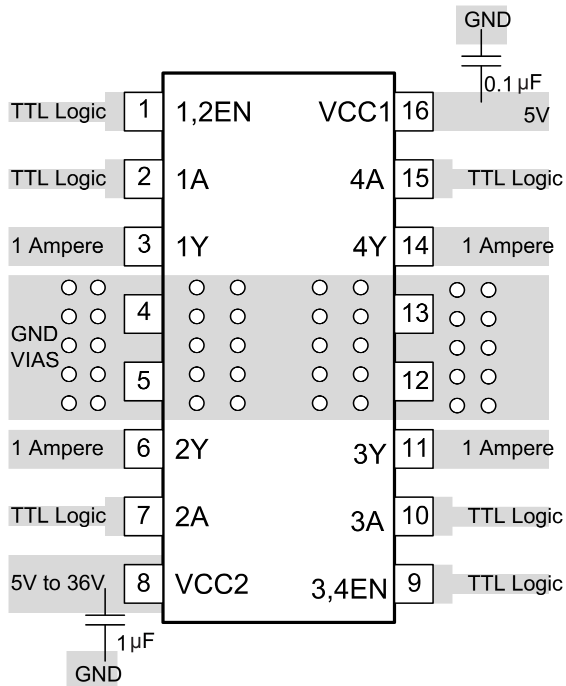


Figure 13. Layout Diagram

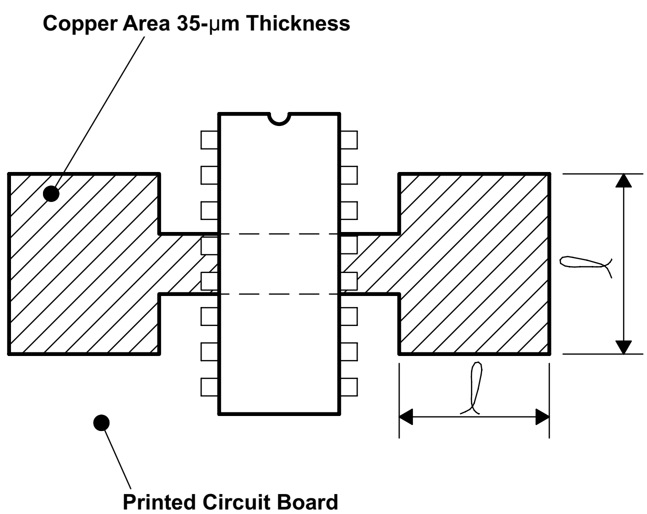


Figure 14. Example of Printed-Circuit-Board Copper Area (Used as Heat Sink)

12 Device and Documentation Support

12.1 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to sample or buy.

Table 4. Related Links

PARTS	PRODUCT FOLDER	SAMPLE & BUY	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
L293	Click here	Click here	Click here	Click here	Click here
L293D	Click here	Click here	Click here	Click here	Click here

12.2 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI E2E™ Online Community ***TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

Design Support ***TI's Design Support*** Quickly find helpful E2E forums along with design support tools and contact information for technical support.

12.3 Trademarks

E2E is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

12.4 Electrostatic Discharge Caution

These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

12.5 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
L293DNE	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	NIPDAU	N / A for Pkg Type	0 to 70	L293DNE	Samples
L293DNEE4	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	NIPDAU	N / A for Pkg Type	0 to 70	L293DNE	Samples
L293NE	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	NIPDAU	N / A for Pkg Type	0 to 70	L293NE	Samples
L293NEE4	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	NIPDAU	N / A for Pkg Type	0 to 70	L293NE	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

IMPORTANT NOTICE AND DISCLAIMER

SOLOMON SYSTECH
SEMICONDUCTOR TECHNICAL DATA

SSD1306

Advance Information

128 x 64 Dot Matrix
OLED/PLED Segment/Common Driver with Controller

CONTENTS

TABLES

FIGURES

1 GENERAL DESCRIPTION

SSD1306 is a single-chip CMOS OLED/PLED driver with controller for organic / polymer light emitting diode dot-matrix graphic display system. It consists of 128 segments and 64commons. This IC is designed for Common Cathode type OLED panel.

The SSD1306 embeds with contrast control, display RAM and oscillator, which reduces the number of external components and power consumption. It has 256-step brightness control. Data/Commands are sent from general MCU through the hardware selectable 6800/8000 series compatible Parallel Interface, I^2C interface or Serial Peripheral Interface. It is suitable for many compact portable applications, such as mobile phone sub-display, MP3 player and calculator, etc.

2 FEATURES

- Resolution: 128 x 64 dot matrix panel
- Power supply
    - V_{DD} = 1.65V to 3.3V for IC logic
    - V_{CC} = 7V to 15V for Panel driving
- For matrix display
    - OLED driving output voltage, 15V maximum
    - Segment maximum source current: 100uA
    - Common maximum sink current: 15mA
    - 256 step contrast brightness current control
- Embedded 128 x 64 bit SRAM display buffer
- Pin selectable MCU Interfaces:
    - 8-bit 6800/8080-series parallel interface
    - 3 /4 wire Serial Peripheral Interface
    - I^2C Interface
- Screen saving continuous scrolling function in both horizontal and vertical direction
- RAM write synchronization signal
- Programmable Frame Rate and Multiplexing Ratio
- Row Re-mapping and Column Re-mapping
- On-Chip Oscillator
- Chip layout for COG & COF
- Wide range of operating temperature: -40°C to 85°C

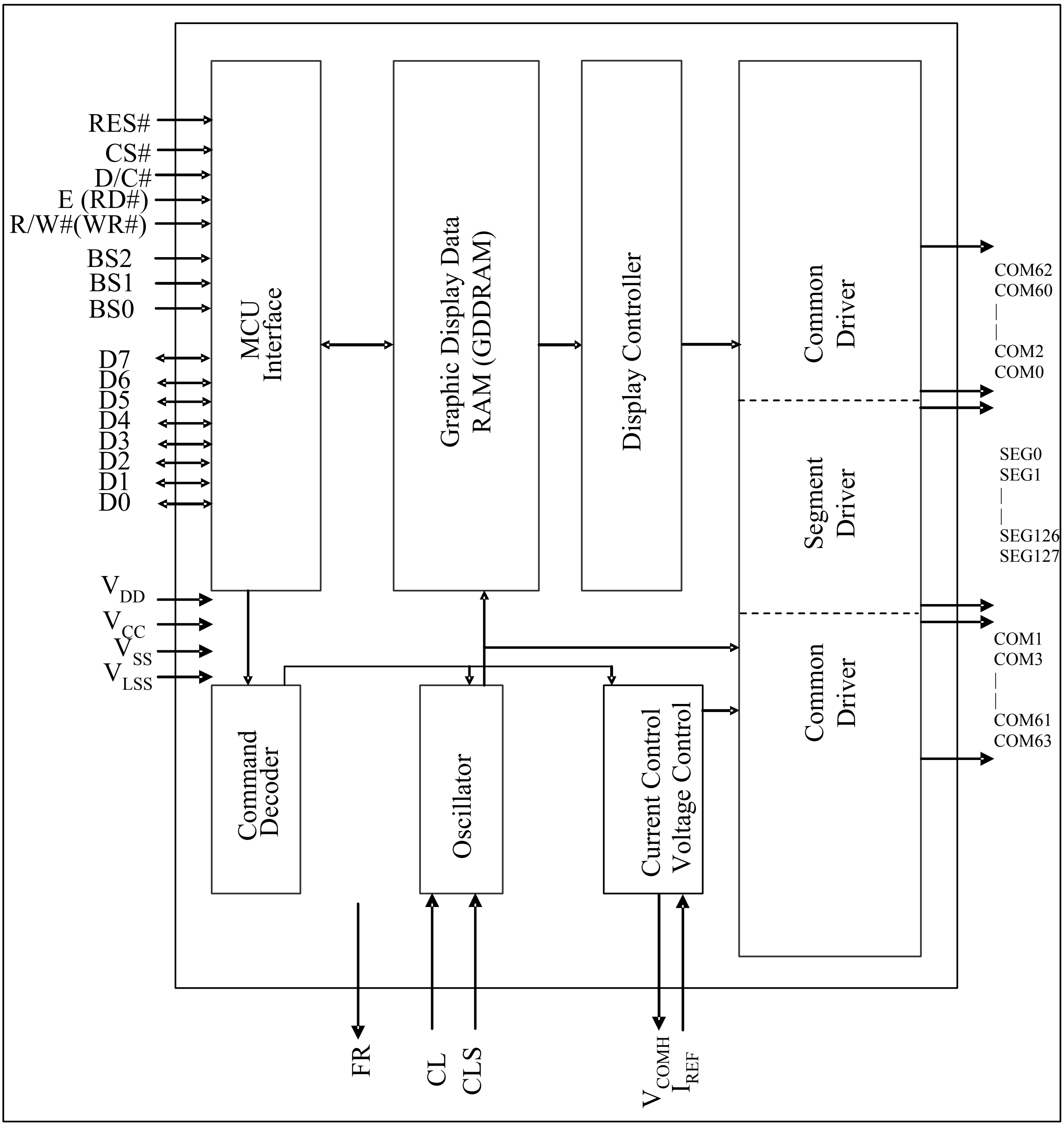
3 ORDERING INFORMATION

Table 3-1: Ordering Information

Ordering Part Number	SEG	COM	Package Form	Reference	Remark
SSD1306Z	128	64	COG	8	o Min SEG pad pitch : 47um o Min COM pad pitch : 40um o Die thickness: 300 +/- 25um
SSD1306TR1	104	48	TAB	11, 56	o 35mm film, 4 sprocket hole, Folding TAB o 8-bit 80 / 8-bit 68 / SPI / I^2C interface o SEG, COM lead pitch 0.1mm x 0.997 =0.0997mm o Die thickness: 457 +/- 25um

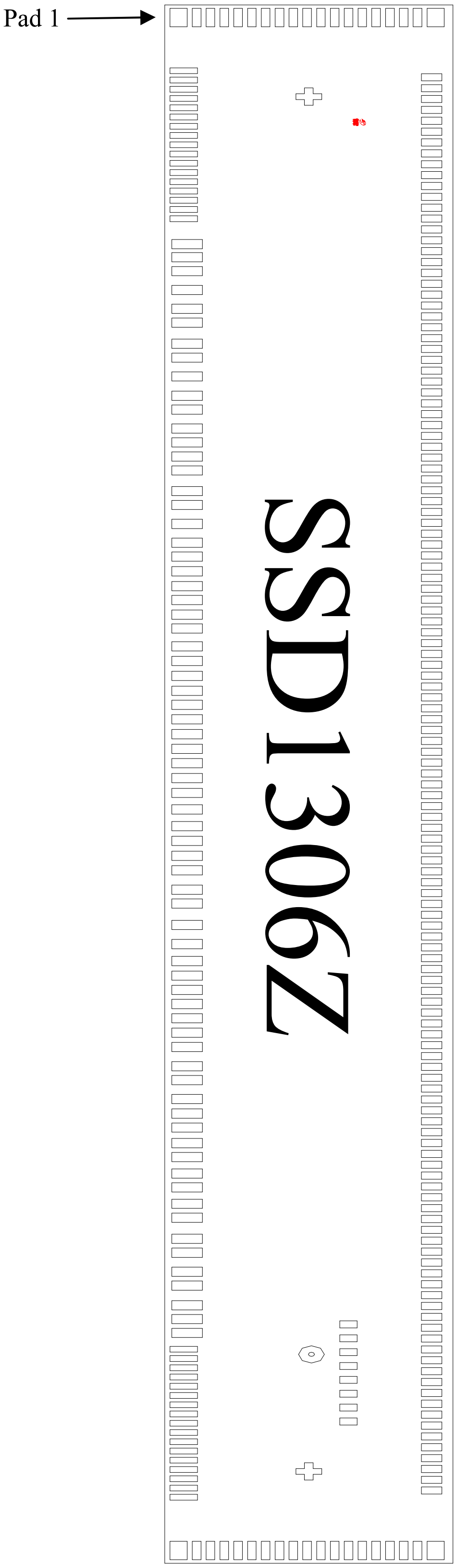
4 BLOCK DIAGRAM

Figure 4-1 SSD1306 Block Diagram



5 DIE PAD FLOOR PLAN

Figure 5-1 : SSD1306Z Die Drawing



Die size	6.76mm x 0.86mm
Die thickness	300 +/- 25um
Min I/O pad pitch	60um
Min SEG pad pitch	47um
Min COM pad pitch	40um
Bump height	Nominal 15um

Bump size	
Pad 1, 106, 124, 256	80um x 50um
Pad 2-18, 89-105, 107-123, 257-273	25ium x 80um
Pad 19-88	40um x 89um
Pad 125-255	31um x 59um
Pad 274-281 (TR pads)	30um x 50um

Alignment mark	Position	Size
+ shape	(-2973, 0)	75um x 75um
+ shape	(2973, 0)	75um x 75um
Circle	(2466.665, 7.575)	R37.5um, inner 18um
SSL Logo	(-2862.35, 144.82)	-

(For details dimension please see p.9)

Note
(1) Diagram showing the Gold bumps face up.
(2) Coordinates are referenced to center of the chip.
(3) Coordinate units and size of all alignment marks are in um.
(4) All alignment keys do not contain gold

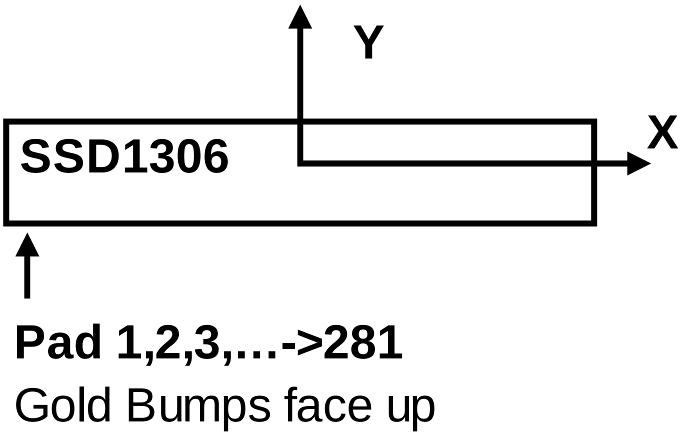
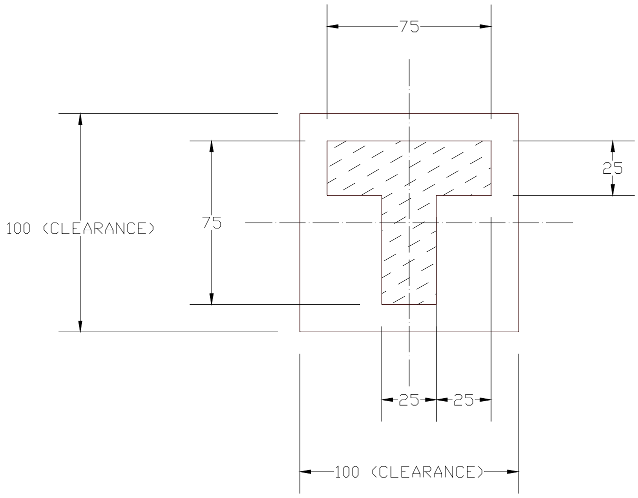
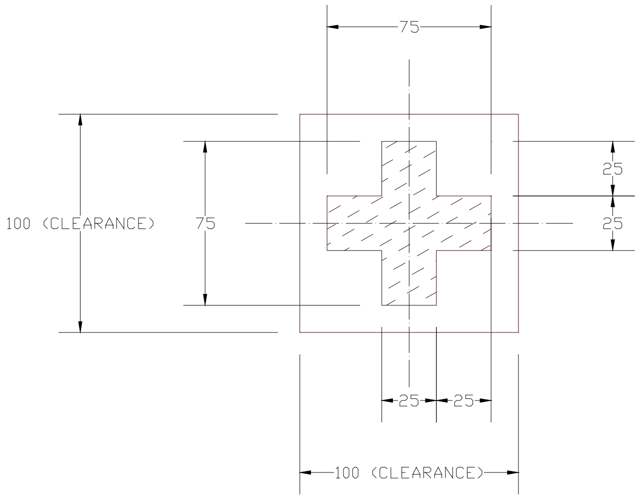


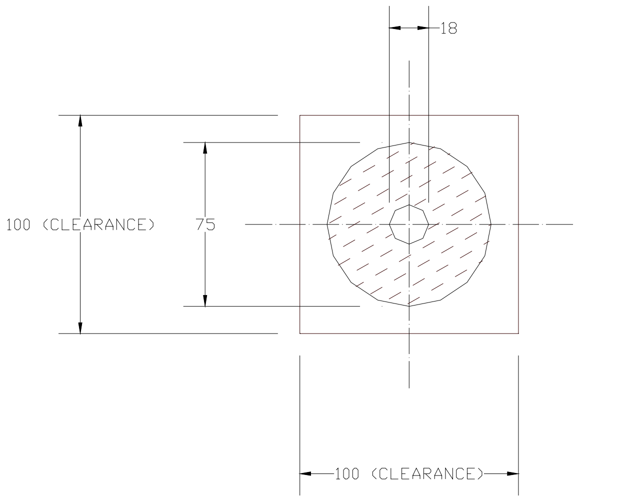
Figure 5-2 : SSD1306Z alignment mark dimensions



T shape



+ shape



Circle

*All units are in um

Table 5-1 : SSD1306Z Bump Die Pad Coordinates

Pad no.	Pad Name	X-pos	Y-pos
1	NC	-3315	-377.5
2	VSS	-3084.77	-362.5
3	COM49	-3044.77	-362.5
4	COM50	-3004.77	-362.5
5	COM51	-2964.77	-362.5
6	COM52	-2924.77	-362.5
7	COM53	-2884.77	-362.5
8	COM54	-2844.77	-362.5
9	COM55	-2804.77	-362.5
10	COM56	-2764.77	-362.5
11	COM57	-2724.77	-362.5
12	COM58	-2684.77	-362.5
13	COM59	-2644.77	-362.5
14	COM60	-2604.77	-362.5
15	COM61	-2564.77	-362.5
16	COM62	-2524.77	-362.5
17	COM63	-2484.77	-362.5
18	VCOMH	-2444.77	-362.5
19	NC	-2334.965	-352.83
20	C2P	-2278.265	-352.83
21	C2P	-2218.265	-352.83
22	C2N	-2136.715	-352.83
23	C2N	-2055.465	-352.83
24	C1P	-1995.465	-352.83
25	C1P	-1904.115	-352.83
26	C1N	-1844.115	-352.83
27	C1N	-1762.865	-352.83
28	VBAT	-1679.31	-352.83
29	VBAT	-1619.31	-352.83
30	VBREF	-1537.51	-352.83
31	BGGND	-1477.51	-352.83
32	VCC	-1416.01	-352.83
33	VCC	-1356.01	-352.83
34	VCOMH	-1266.955	-352.83
35	VCOMH	-1206.955	-352.83
36	VLSS	-1125.155	-352.83
37	VLSS	-1043.355	-352.83
38	VLSS	-983.355	-352.83
39	VSS	-920	-352.83
40	VSS	-856	-352.83
41	VSS	-796	-352.83
42	VDD	-732.645	-352.83
43	VDD	-672.645	-352.83
44	BS0	-595.655	-352.83
45	VSS	-531.955	-352.83
46	BS1	-467.655	-352.83
47	VDD	-403.155	-352.83
48	VDD	-342.555	-352.83
49	BS2	-279.705	-352.83
50	VSS	-215.705	-352.83
51	FR	-151.955	-352.83
52	CL	-89.815	-352.83
53	VSS	-25.665	-352.83
54	CS#	38.635	-352.83
55	RES#	109.835	-352.83
56	D/C#	182.425	-352.83
57	VSS	246.125	-352.83
58	R/W#	310.425	-352.83
59	E	373.125	-352.83
60	VDD	457.175	-352.83
61	VDD	517.175	-352.83
62	D0	609.275	-352.83
63	D1	692.475	-352.83
64	D2	765.675	-352.83
65	D3	828.875	-352.83
66	VSS	890.325	-352.83
67	D4	951.275	-352.83
68	D5	1013.315	-352.83
69	D6	1075.355	-352.83
70	D7	1137.395	-352.83
71	VSS	1220.735	-352.83
72	VSS	1280.735	-352.83
73	CLS	1362.585	-352.83
74	VDD	1425.285	-352.83
75	VDD	1485.885	-352.83
76	VDD	1553.185	-352.83
77	VDD	1613.185	-352.83
78	IREF	1684.585	-352.83
79	IREF	1744.585	-352.83
80	VCOMH	1815.585	-352.83

Pad no.	Pad Name	X-pos	Y-pos
81	VCOMH	1875.585	-352.83
82	VCC	1967.185	-352.83
83	VCC	2027.185	-352.83
84	VLSS	2109.185	-352.83
85	VLSS	2169.185	-352.83
86	VLSS	2254.185	-352.83
87	NC	2314.185	-352.83
88	NC	2374.185	-352.83
89	VSS	2444.77	-362.5
90	COM31	2484.77	-362.5
91	COM30	2524.77	-362.5
92	COM29	2564.77	-362.5
93	COM28	2604.77	-362.5
94	COM27	2644.77	-362.5
95	COM26	2684.77	-362.5
96	COM25	2724.77	-362.5
97	COM24	2764.77	-362.5
98	COM23	2804.77	-362.5
99	COM22	2844.77	-362.5
100	COM21	2884.77	-362.5
101	COM20	2924.77	-362.5
102	COM19	2964.77	-362.5
103	COM18	3004.77	-362.5
104	COM17	3044.77	-362.5
105	VSS	3084.77	-362.5
106	NC	3315	-377.5
107	COM16	3315	-325
108	COM15	3315	-285
109	COM14	3315	-245
110	COM13	3315	-205
111	COM12	3315	-165
112	COM11	3315	-125
113	COM10	3315	-85
114	COM9	3315	-45
115	COM8	3315	-5
116	COM7	3315	35
117	COM6	3315	75
118	COM5	3315	115
119	COM4	3315	155
120	COM3	3315	195
121	COM2	3315	235
122	COM1	3315	275
123	COM0	3315	315
124	NC	3315	367.5
125	NC	3055.5	356
126	SEG0	3009.5	356
127	SEG1	2962.5	356
128	SEG2	2915.5	356
129	SEG3	2868.5	356
130	SEG4	2821.5	356
131	SEG5	2774.5	356
132	SEG6	2727.5	356
133	SEG7	2680.5	356
134	SEG8	2633.5	356
135	SEG9	2586.5	356
136	SEG10	2539.5	356
137	SEG11	2492.5	356
138	SEG12	2445.5	356
139	SEG13	2398.5	356
140	SEG14	2351.5	356
141	SEG15	2304.5	356
142	SEG16	2257.5	356
143	SEG17	2210.5	356
144	SEG18	2163.5	356
145	SEG19	2116.5	356
146	SEG20	2069.5	356
147	SEG21	2022.5	356
148	SEG22	1975.5	356
149	SEG23	1928.5	356
150	SEG24	1881.5	356
151	SEG25	1834.5	356
152	SEG26	1787.5	356
153	SEG27	1740.5	356
154	SEG28	1693.5	356
155	SEG29	1646.5	356
156	SEG30	1599.5	356
157	SEG31	1552.5	356
158	SEG32	1505.5	356
159	SEG33	1458.5	356
160	SEG34	1411.5	356

Pad no.	Pad Name	X-pos	Y-pos
161	SEG35	1364.5	356
162	SEG36	1317.5	356
163	SEG37	1270.5	356
164	SEG38	1223.5	356
165	SEG39	1176.5	356
166	SEG40	1129.5	356
167	SEG41	1082.5	356
168	SEG42	1035.5	356
169	SEG43	988.5	356
170	SEG44	941.5	356
171	SEG45	894.5	356
172	SEG46	847.5	356
173	SEG47	800.5	356
174	SEG48	753.5	356
175	SEG49	706.5	356
176	SEG50	659.5	356
177	SEG51	612.5	356
178	SEG52	565.5	356
179	SEG53	518.5	356
180	SEG54	471.5	356
181	SEG55	424.5	356
182	SEG56	377.5	356
183	SEG57	330.5	356
184	SEG58	283.5	356
185	SEG59	236.5	356
186	SEG60	189.5	356
187	SEG61	142.5	356
188	SEG62	95.5	356
189	SEG63	48.5	356
190	SEG64	1.5	356
191	SEG65	-45.5	356
192	SEG66	-92.5	356
193	SEG67	-139.5	356
194	SEG68	-186.5	356
195	SEG69	-233.5	356
196	SEG70	-280.5	356
197	SEG71	-327.5	356
198	SEG72	-374.5	356
199	SEG73	-421.5	356
200	SEG74	-468.5	356
201	SEG75	-515.5	356
202	SEG76	-562.5	356
203	SEG77	-609.5	356
204	SEG78	-656.5	356
205	SEG79	-703.5	356
206	SEG80	-750.5	356
207	SEG81	-797.5	356
208	SEG82	-844.5	356
209	SEG83	-891.5	356
210	NC	-940	356
211	SEG84	-988.5	356
212	SEG85	-1035.5	356
213	SEG86	-1082.5	356
214	SEG87	-1129.5	356
215	SEG88	-1176.5	356
216	SEG89	-1223.5	356
217	SEG90	-1270.5	356
218	SEG91	-1317.5	356
219	SEG92	-1364.5	356
220	SEG93	-1411.5	356
221	SEG94	-1458.5	356
222	SEG95	-1505.5	356
223	SEG96	-1552.5	356
224	SEG97	-1599.5	356
225	SEG98	-1646.5	356
226	SEG99	-1693.5	356
227	SEG100	-1740.5	356
228	SEG101	-1787.5	356
229	SEG102	-1834.5	356
230	SEG103	-1881.5	356
231	SEG104	-1928.5	356
232	SEG105	-1975.5	356
233	SEG106	-2022.5	356
234	SEG107	-2069.5	356
235	SEG108	-2116.5	356
236	SEG109	-2163.5	356
237	SEG110	-2210.5	356
238	SEG111	-2257.5	356
239	SEG112	-2304.5	356
240	SEG113	-2351.5	356

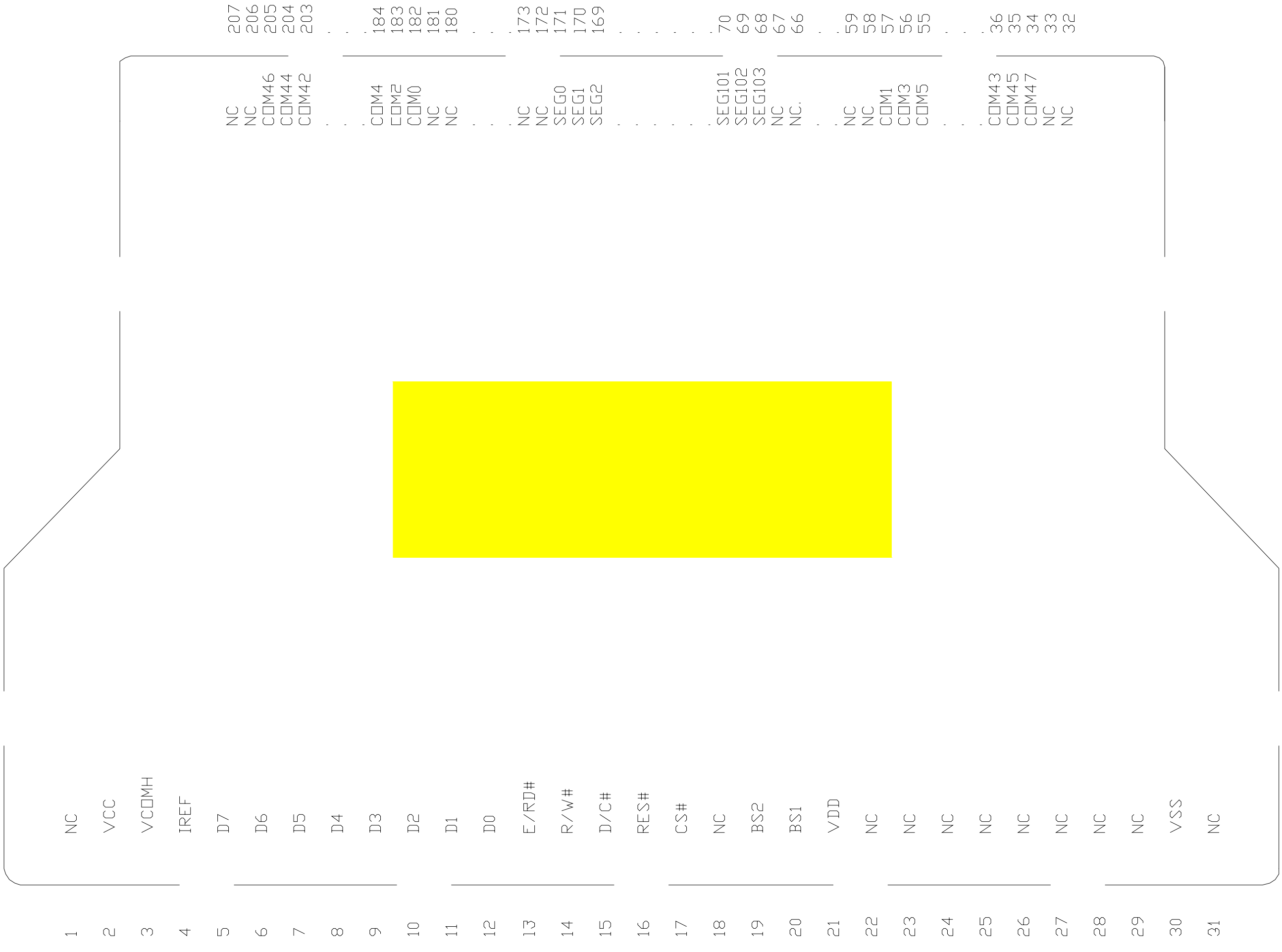
Pad no.	Pad Name	X-pos	Y-pos
241	SEG114	-2398.5	356
242	SEG115	-2445.5	356
243	SEG116	-2492.5	356
244	SEG117	-2539.5	356
245	SEG118	-2586.5	356
246	SEG119	-2633.5	356
247	SEG120	-2680.5	356
248	SEG121	-2727.5	356
249	SEG122	-2774.5	356
250	SEG123	-2821.5	356
251	SEG124	-2868.5	356
252	SEG125	-2915.5	356
253	SEG126	-2962.5	356
254	SEG127	-3009.5	356
255	NC	-3056.5	356
256	NC	-3315	367.5
257	COM32	-3315	315
258	COM33	-3315	275
259	COM34	-3315	235
260	COM35	-3315	195
261	COM36	-3315	155
262	COM37	-3315	115
263	COM38	-3315	75
264	COM39	-3315	35
265	COM40	-3315	-5
266	COM41	-3315	-45
267	COM42	-3315	-85
268	COM43	-3315	-125
269	COM44	-3315	-165
270	COM45	-3315	-205
271	COM46	-3315	-245
272	COM47	-3315	-285
273	COM48	-3315	-325

Pad no.	Pad Name	X-pos	Y-pos
Pin#	Pin name	X-dir	Y-dir
274	TR0	2757.05	114.8
275	TR1	2697.05	114.8
276	TR2	2637.05	114.8
277	TR3	2577.05	114.8
278	VSS	2517.05	114.8
279	TR4	2457.05	114.8
280	TR5	2397.05	114.8
281	TR6	2337.05	114.8

6 PIN ARRANGEMENT

6.1 SSD1306TR1 pin assignment

Figure 6-1 : SSD1306TR1 Pin Assignment



Note:
(1) COM sequence (Split) is under command setting: DAh, 12h

Table 6-1 : SSD1306TR1 Pin Assignment Table

Pin no.	Pin Name	Pin no.	Pin Name	Pin no.	Pin Name
1	NC	81	SEG90	161	SEG10
2	VCC	82	SEG89	162	SEG9
3	VCOMH	83	SEG88	163	SEG8
4	IREF	84	SEG87	164	SEG7
5	D7	85	SEG86	165	SEG6
6	D6	86	SEG85	166	SEG5
7	D5	87	SEG84	167	SEG4
8	D4	88	SEG83	168	SEG3
9	D3	89	SEG82	169	SEG2
10	D2	90	SEG81	170	SEG1
11	D1	91	SEG80	171	SEG0
12	D0	92	SEG79	172	NC
13	E/RD#	93	SEG78	173	NC
14	R/W#	94	SEG77	174	NC
15	D/C#	95	SEG76	175	NC
16	RES#	96	SEG75	176	NC
17	CS#	97	SEG74	177	NC
18	NC	98	SEG73	178	NC
19	BS2	99	SEG72	179	NC
20	BS1	100	SEG71	180	NC
21	VDD	101	SEG70	181	NC
22	NC	102	SEG69	182	COM0
23	NC	103	SEG68	183	COM2
24	NC	104	SEG67	184	COM4
25	NC	105	SEG66	185	COM6
26	NC	106	SEG65	186	COM8
27	NC	107	SEG64	187	COM10
28	NC	108	SEG63	188	COM12
29	NC	109	SEG62	189	COM14
30	VSS	110	SEG61	190	COM16
31	NC	111	SEG60	191	COM18
32	NC	112	SEG59	192	COM20
33	NC	113	SEG58	193	COM22
34	COM47	114	SEG57	194	COM24
35	COM45	115	SEG56	195	COM26
36	COM43	116	SEG55	196	COM28
37	COM41	117	SEG54	197	COM30
38	COM39	118	SEG53	198	COM32
39	COM37	119	SEG52	199	COM34
40	COM35	120	SEG51	200	COM36
41	COM33	121	SEG50	201	COM38
42	COM31	122	SEG49	202	COM40
43	COM29	123	SEG48	203	COM42
44	COM27	124	SEG47	204	COM44
45	COM25	125	SEG46	205	COM46
46	COM23	126	SEG45	206	NC
47	COM21	127	SEG44	207	NC
48	COM19	128	SEG43		
49	COM17	129	SEG42		
50	COM15	130	SEG41		
51	COM13	131	SEG40		
52	COM11	132	SEG39		
53	COM9	133	SEG38		
54	COM7	134	SEG37		
55	COM5	135	SEG36		
56	COM3	136	SEG35		
57	COM1	137	SEG34		
58	NC	138	SEG33		
59	NC	139	SEG32		
60	NC	140	SEG31		
61	NC	141	SEG30		
62	NC	142	SEG29		
63	NC	143	SEG28		
64	NC	144	SEG27		
65	NC	145	SEG26		
66	NC	146	SEG25		
67	NC	147	SEG24		
68	SEG103	148	SEG23		
69	SEG102	149	SEG22		
70	SEG101	150	SEG21		
71	SEG100	151	SEG20		
72	SEG99	152	SEG19		
73	SEG98	153	SEG18		
74	SEG97	154	SEG17		
75	SEG96	155	SEG16		
76	SEG95	156	SEG15		
77	SEG94	157	SEG14		
78	SEG93	158	SEG13		
79	SEG92	159	SEG12		
80	SEG91	160	SEG11		

7 PIN DESCRIPTION

Key:

I = Input	NC = Not Connected
O =Output	Pull LOW= connect to Ground
I/O = Bi-directional (input/output)	Pull HIGH= connect to V_{DD}
P = Power pin	

Figure 7-1 Pin Description

Pin Name	**Type**	**Description**
V_{DD}	P	Power supply pin for core logic operation.
V_{CC}	P	Power supply for panel driving voltage. This is also the most positive power voltage supply pin.
V_{SS}	P	This is a ground pin.
V_{LSS}	P	This is an analog ground pin. It should be connected to V_{SS} externally.
V_{COMH}	O	The pin for COM signal deselected voltage level. A capacitor should be connected between this pin and V_{SS}.
V_{BAT}	P	Reserved pin. It should be connected to $V_{DD.}$
BGGND	P	Reserved pin. It should be connected to ground.
C1P/C1N C2P/C2N	I	Reserved pin. It should be kept NC.
V_{BREF}	P	Reserved pin. It should be kept NC.
BS[2:0]	I	MCU bus interface selection pins. Please refer to **Table 7-1** for the details of setting.
I_{REF}	I	This is segment output current reference pin. A resistor should be connected between this pin and V_{SS} to maintain the I_{REF} current at 12.5 uA. Please refer to **Figure 8-15** for the details of resistor value.
FR	O	This pin outputs RAM write synchronization signal. Proper timing between MCU data writing and frame display timing can be achieved to prevent tearing effect. It should be kept NC if it is not used. Please refer to Section **8.4** for details usage.
CL	I	This is external clock input pin. When internal clock is enabled (i.e. HIGH in CLS pin), this pin is not used and should be connected to V_{SS}. When internal clock is disabled (i.e. LOW in CLS pin), this pin is the external clock source input pin.
CLS	I	This is internal clock enable pin. When it is pulled HIGH (i.e. connect to V_{DD}), internal clock is enabled. When it is pulled LOW, the internal clock is disabled; an external clock source must be connected to the CL pin for normal operation.
RES#	I	This pin is reset signal input. When the pin is pulled LOW, initialization of the chip is executed. Keep this pin HIGH (i.e. connect to V_{DD}) during normal operation.
CS#	I	This pin is the chip select input. (active LOW).

Pin Name	Type	Description
D/C#	I	This is Data/Command control pin. When it is pulled HIGH (i.e. connect to V_{DD}), the data at D[7:0] is treated as data. When it is pulled LOW, the data at D[7:0] will be transferred to the command register. In I^2C mode, this pin acts as SA0 for slave address selection. When 3-wire serial interface is selected, this pin must be connected to V_{SS}. For detail relationship to MCU interface signals, please refer to the Timing Characteristics Diagrams: **Figure 13-1** to **Figure 13-5**.
E (RD#)	I	When interfacing to a 6800-series microprocessor, this pin will be used as the Enable (E) signal. Read/write operation is initiated when this pin is pulled HIGH (i.e. connect to V_{DD}) and the chip is selected. When connecting to an 8080-series microprocessor, this pin receives the Read (RD#) signal. Read operation is initiated when this pin is pulled LOW and the chip is selected. When serial or I^2C interface is selected, this pin must be connected to V_{SS}.
R/W#(WR#)	I	This is read / write control input pin connecting to the MCU interface. When interfacing to a 6800-series microprocessor, this pin will be used as Read/Write (R/W#) selection input. Read mode will be carried out when this pin is pulled HIGH (i.e. connect to V_{DD}) and write mode when LOW. When 8080 interface mode is selected, this pin will be the Write (WR#) input. Data write operation is initiated when this pin is pulled LOW and the chip is selected. When serial or I^2C interface is selected, this pin must be connected to V_{SS}.
D[7:0]	IO	These are 8-bit bi-directional data bus to be connected to the microprocessor's data bus. When serial interface mode is selected, D0 will be the serial clock input: SCLK; D1 will be the serial data input: SDIN and D2 should be kept NC. When I^2C mode is selected, D2, D1 should be tied together and serve as SDA_{out}, SDA_{in} in application and D0 is the serial clock input, SCL.
TR0-TR6	-	Testing reserved pins. It should be kept NC.
SEG0 ~ SEG127	O	These pins provide Segment switch signals to OLED panel. These pins are V_{SS} state when display is OFF.
COM0 ~ COM63	O	These pins provide Common switch signals to OLED panel. They are in high impedance state when display is OFF.
NC	-	This is dummy pin. Do not group or short NC pins together.

Table 7-1 : MCU Bus Interface Pin Selection

SSD1306 Pin Name	I^2C Interface	6800-parallel interface (8 bit)	8080-parallel interface (8 bit)	4-wire Serial interface	3-wire Serial interface
BS0	0	0	0	0	1
BS1	1	0	1	0	0
BS2	0	1	1	0	0

Note
(1) 0 is connected to V_{SS}
(2) 1 is connected to V_{DD}

8 FUNCTIONAL BLOCK DESCRIPTIONS

8.1 MCU Interface selection

SSD1306 MCU interface consist of 8 data pins and 5 control pins. The pin assignment at different interface mode is summarized in Table 8-1. Different MCU mode can be set by hardware selection on BS[2:0] pins (please refer to Table 7-1 for BS[2:0] setting).

Table 8-1 : MCU interface assignment under different bus interface mode

Pin Name / Bus Interface	Data/Command Interface								Control Signal				
	D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**E**	**R/W#**	**CS#**	**D/C#**	**RES#**
8-bit 8080	D[7:0]								RD#	WR#	CS#	D/C#	RES#
8-bit 6800	D[7:0]								E	R/W#	CS#	D/C#	RES#
3-wire SPI	Tie LOW					NC	SDIN	SCLK	Tie LOW		CS#	Tie LOW	RES#
4-wire SPI	Tie LOW					NC	SDIN	SCLK	Tie LOW		CS#	D/C#	RES#
I^2C	Tie LOW					SDA_{OUT}	SDA_{IN}	SCL	Tie LOW			SA0	RES#

8.1.1 MCU Parallel 6800-series Interface

The parallel interface consists of 8 bi-directional data pins (D[7:0]), R/W#, D/C#, E and CS#.

A LOW in R/W# indicates WRITE operation and HIGH in R/W# indicates READ operation.
A LOW in D/C# indicates COMMAND read/write and HIGH in D/C# indicates DATA read/write.
The E input serves as data latch signal while CS# is LOW. Data is latched at the falling edge of E signal.

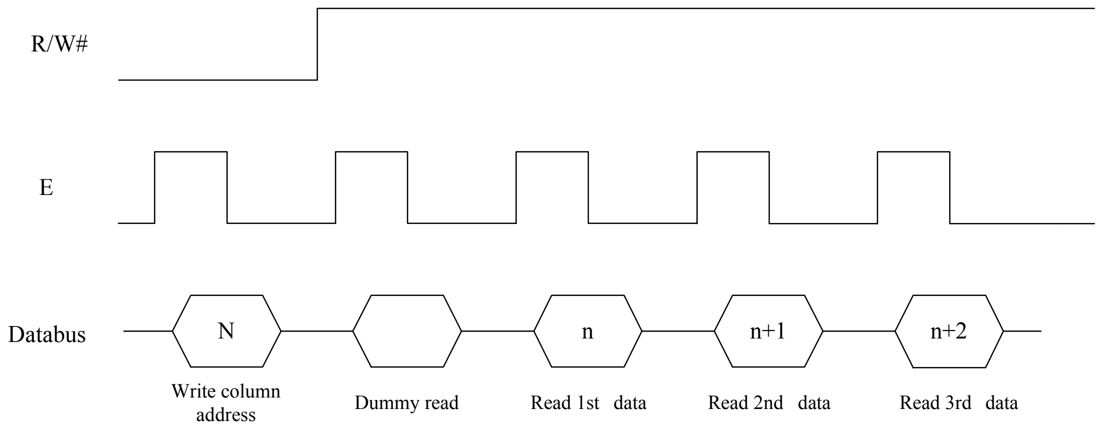
Table 8-2 : Control pins of 6800 interface

Function	E	R/W#	CS#	D/C#
Write command	↓	L	L	L
Read status	↓	H	L	L
Write data	↓	L	L	H
Read data	↓	H	L	H

Note
(1) ↓ stands for falling edge of signal
H stands for HIGH in signal
L stands for LOW in signal

In order to match the operating frequency of display RAM with that of the microprocessor, some pipeline processing is internally performed which requires the insertion of a dummy read before the first actual display data read. This is shown in Figure 8-1.

Figure 8-1 : Data read back procedure - insertion of dummy read



8.1.2 MCU Parallel 8080-series Interface

The parallel interface consists of 8 bi-directional data pins (D[7:0]), RD#, WR#, D/C# and CS#.

A LOW in D/C# indicates COMMAND read/write and HIGH in D/C# indicates DATA read/write.
A rising edge of RD# input serves as a data READ latch signal while CS# is kept LOW.
A rising edge of WR# input serves as a data/command WRITE latch signal while CS# is kept LOW.

Figure 8-2 : Example of Write procedure in 8080 parallel interface mode

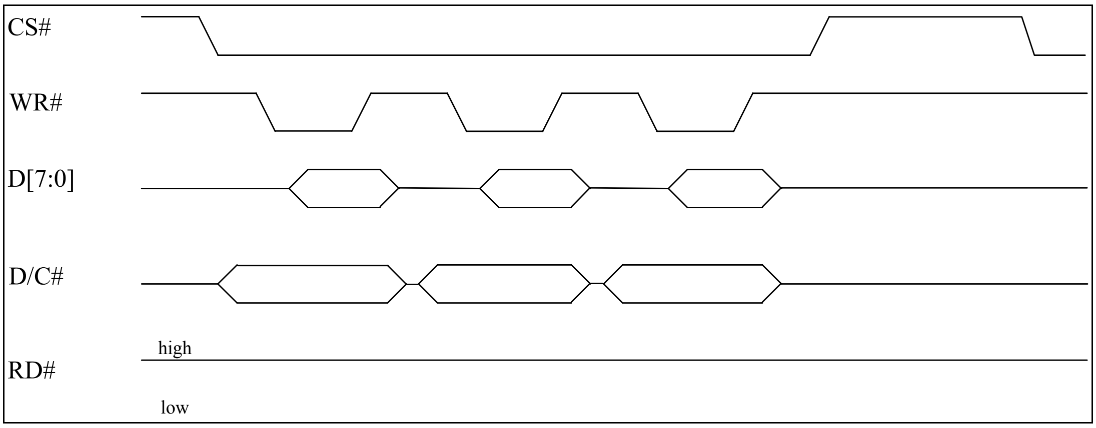


Figure 8-3 : Example of Read procedure in 8080 parallel interface mode

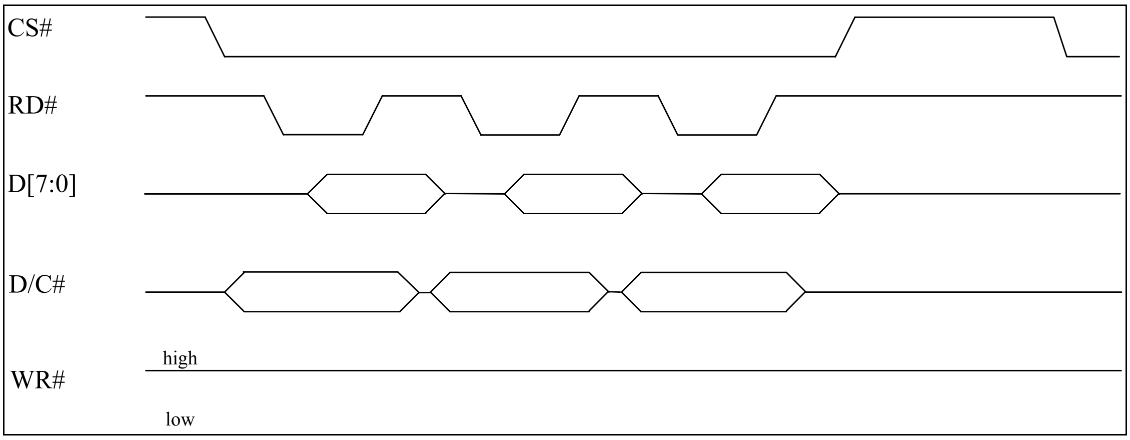


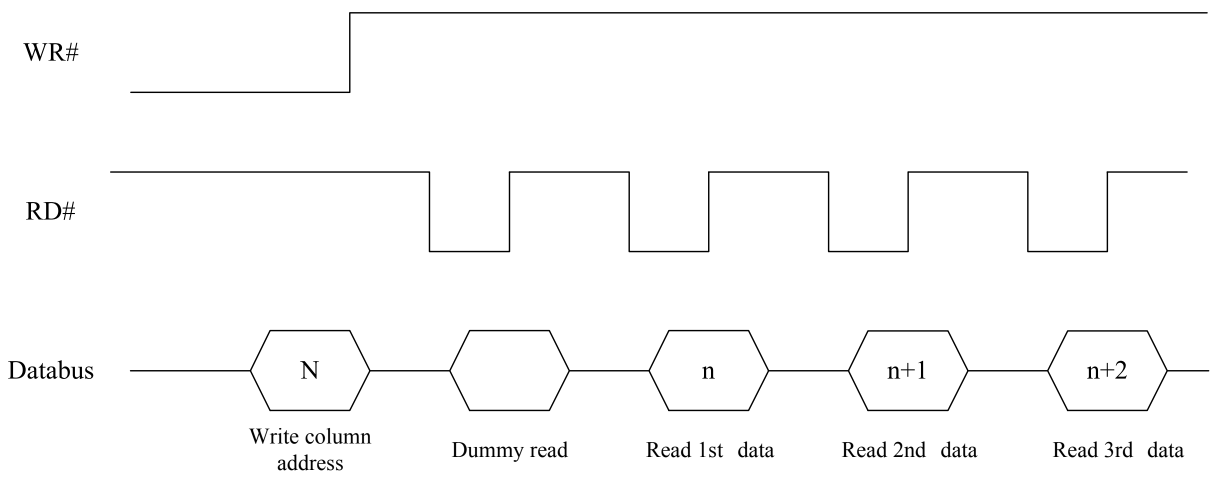
Table 8-3 : Control pins of 8080 interface

Function	RD#	WR#	CS#	D/C#
Write command	H	↑	L	L
Read status	↑	H	L	L
Write data	H	↑	L	H
Read data	↑	H	L	H

Note
(1) ↑ stands for rising edge of signal
(2) H stands for HIGH in signal
(3) L stands for LOW in signal

In order to match the operating frequency of display RAM with that of the microprocessor, some pipeline processing is internally performed which requires the insertion of a dummy read before the first actual display data read. This is shown in Figure 8-4.

Figure 8-4 : Display data read back procedure - insertion of dummy read



8.1.3 MCU Serial Interface (4-wire SPI)

The 4-wire serial interface consists of serial clock: SCLK, serial data: SDIN, D/C#, CS#. In 4-wire SPI mode, D0 acts as SCLK, D1 acts as SDIN. For the unused data pins, D2 should be left open. The pins from D3 to D7, E and R/W# (WR#)# can be connected to an external ground.

Table 8-4 : Control pins of 4-wire Serial interface

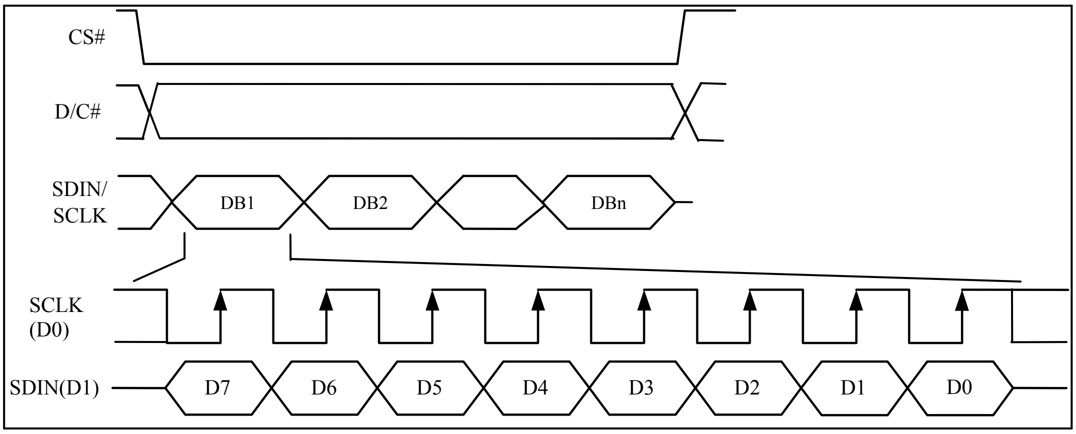
Function	E(RD#)	R/W#(WR#)	CS#	D/C#	D0
Write command	Tie LOW	Tie LOW	L	L	↑
Write data	Tie LOW	Tie LOW	L	H	↑

Note
(1) H stands for HIGH in signal
(2) L stands for LOW in signal

SDIN is shifted into an 8-bit shift register on every rising edge of SCLK in the order of D7, D6, ... D0. D/C# is sampled on every eighth clock and the data byte in the shift register is written to the Graphic Display Data RAM (GDDRAM) or command register in the same clock.

Under serial mode, only write operations are allowed.

Figure 8-5 : Write procedure in 4-wire Serial interface mode



8.1.4 MCU Serial Interface (3-wire SPI)

The 3-wire serial interface consists of serial clock SCLK, serial data SDIN and CS#.
In 3-wire SPI mode, D0 acts as SCLK, D1 acts as SDIN. For the unused data pins, D2 should be left open. The pins from D3 to D7, R/W# (WR#)#, E and D/C# can be connected to an external ground.

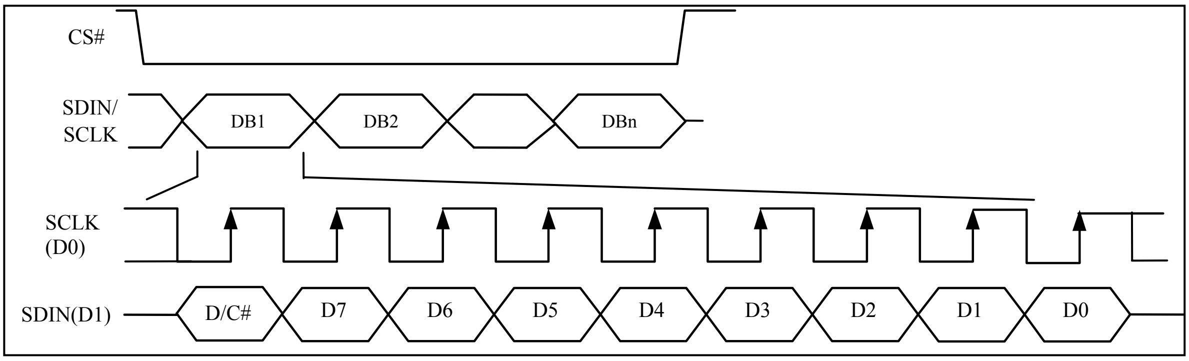
The operation is similar to 4-wire serial interface while D/C# pin is not used. There are altogether 9-bits will be shifted into the shift register on every ninth clock in sequence: D/C# bit, D7 to D0 bit. The D/C# bit (first bit of the sequential data) will determine the following data byte in the shift register is written to the Display Data RAM (D/C# bit = 1) or the command register (D/C# bit = 0). Under serial mode, only write operations are allowed.

Table 8-5 : Control pins of 3-wire Serial interface

Function	**E(RD#)**	**R/W#(WR#)**	**CS#**	**D/C#**	**D0**
Write command	Tie LOW	Tie LOW	L	Tie LOW	↑
Write data	Tie LOW	Tie LOW	L	Tie LOW	↑

Note
(1) L stands for LOW in signal

Figure 8-6 : Write procedure in 3-wire Serial interface mode



8.1.5 MCU I²C Interface

The I²C communication interface consists of slave address bit SA0, I²C-bus data signal SDA (SDA_{OUT}/D_2 for output and SDA_{IN}/D_1 for input) and I²C-bus clock signal SCL (D_0). Both the data and clock signals must be connected to pull-up resistors. RES# is used for the initialization of device.

a) Slave address bit (SA0)
   SSD1306 has to recognize the slave address before transmitting or receiving any information by the I²C-bus. The device will respond to the slave address following by the slave address bit ("SA0" bit) and the read/write select bit ("R/W#" bit) with the following byte format,
   b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
   0 1 1 1 1 0 SA0 R/W#
   "SA0" bit provides an extension bit for the slave address. Either "0111100" or "0111101", can be selected as the slave address of SSD1306. D/C# pin acts as SA0 for slave address selection.
   "R/W#" bit is used to determine the operation mode of the I²C-bus interface. R/W#=1, it is in read mode. R/W#=0, it is in write mode.

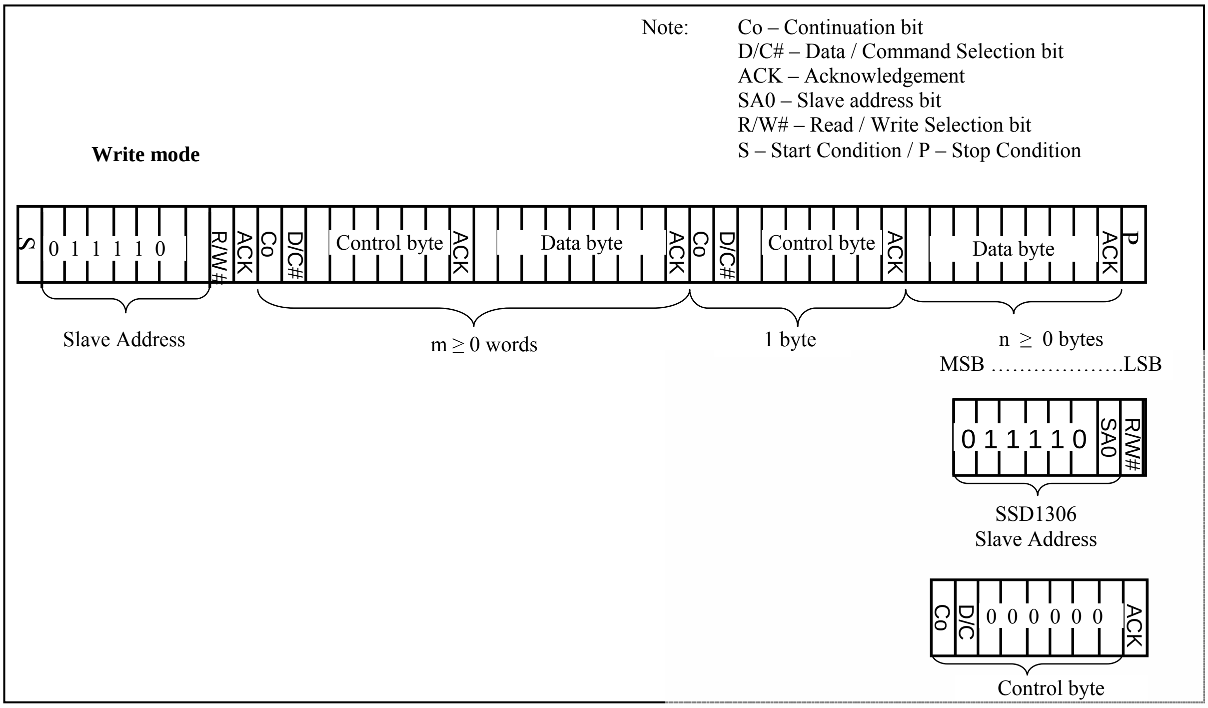
b) I²C-bus data signal (SDA)
   SDA acts as a communication channel between the transmitter and the receiver. The data and the acknowledgement are sent through the SDA.
   It should be noticed that the ITO track resistance and the pulled-up resistance at "SDA" pin becomes a voltage potential divider. As a result, the acknowledgement would not be possible to attain a valid logic 0 level in "SDA".
   "SDA_{IN}" and "SDA_{OUT}" are tied together and serve as SDA. The "SDA_{IN}" pin must be connected to act as SDA. The "SDA_{OUT}" pin may be disconnected. When "SDA_{OUT}" pin is disconnected, the acknowledgement signal will be ignored in the I²C-bus.

c) I²C-bus clock signal (SCL)
   The transmission of information in the I²C-bus is following a clock signal, SCL. Each transmission of data bit is taken place during a single clock period of SCL.

8.1.5.1 I²C-bus Write data

The I²C-bus interface gives access to write data and command into the device. Please refer to Figure 8-7 for the write mode of I²C-bus in chronological order.

Figure 8-7 : I²C-bus data format



8.1.5.2 Write mode for I²C

1) The master device initiates the data communication by a start condition. The definition of the start condition is shown in Figure 8-8. The start condition is established by pulling the SDA from HIGH to LOW while the SCL stays HIGH.
2) The slave address is following the start condition for recognition use. For the SSD1306, the slave address is either "b0111100" or "b0111101" by changing the SA0 to LOW or HIGH (D/C pin acts as SA0).
3) The write mode is established by setting the R/W# bit to logic "0".
4) An acknowledgement signal will be generated after receiving one byte of data, including the slave address and the R/W# bit. Please refer to the Figure 8-9 for the graphical representation of the acknowledge signal. The acknowledge bit is defined as the SDA line is pulled down during the HIGH period of the acknowledgement related clock pulse.
5) After the transmission of the slave address, either the control byte or the data byte may be sent across the SDA. A control byte mainly consists of Co and D/C# bits following by six "0" 's.
   a. If the Co bit is set as logic "0", the transmission of the following information will contain data bytes only.
   b. The D/C# bit determines the next data byte is acted as a command or a data. If the D/C# bit is set to logic "0", it defines the following data byte as a command. If the D/C# bit is set to logic "1", it defines the following data byte as a data which will be stored at the GDDRAM. The GDDRAM column address pointer will be increased by one automatically after each data write.
6) Acknowledge bit will be generated after receiving each control byte or data byte.
7) The write mode will be finished when a stop condition is applied. The stop condition is also defined in Figure 8-8. The stop condition is established by pulling the "SDA in" from LOW to HIGH while the "SCL" stays HIGH.

Figure 8-8 : Definition of the Start and Stop Condition

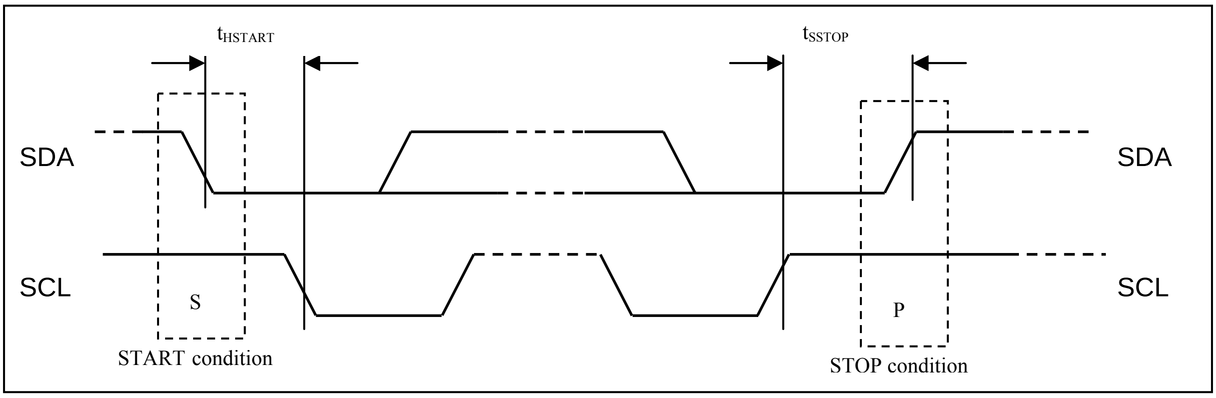
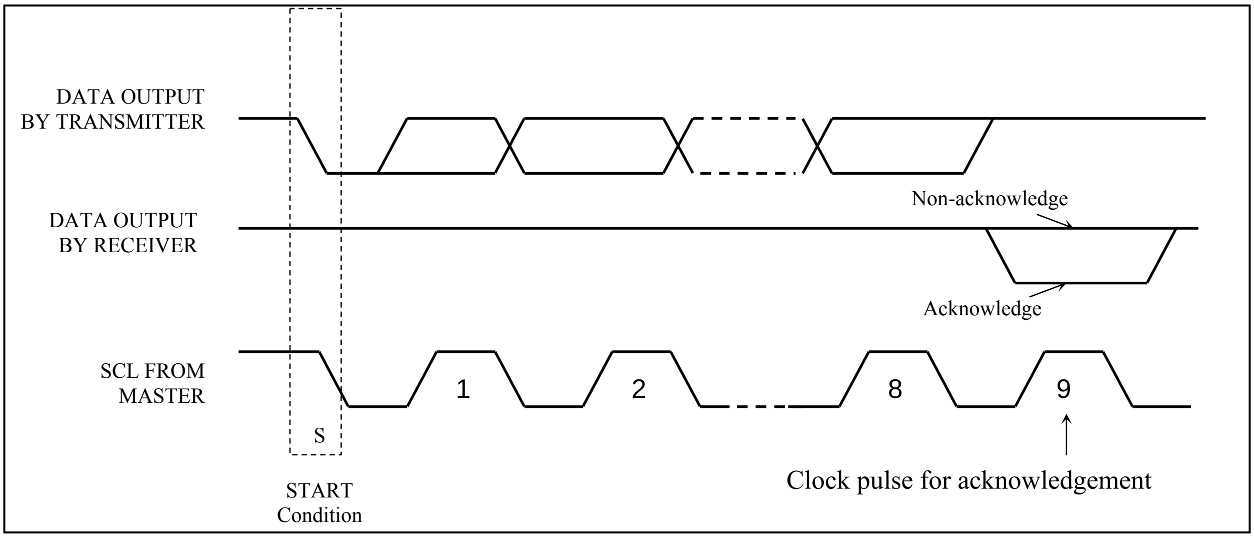


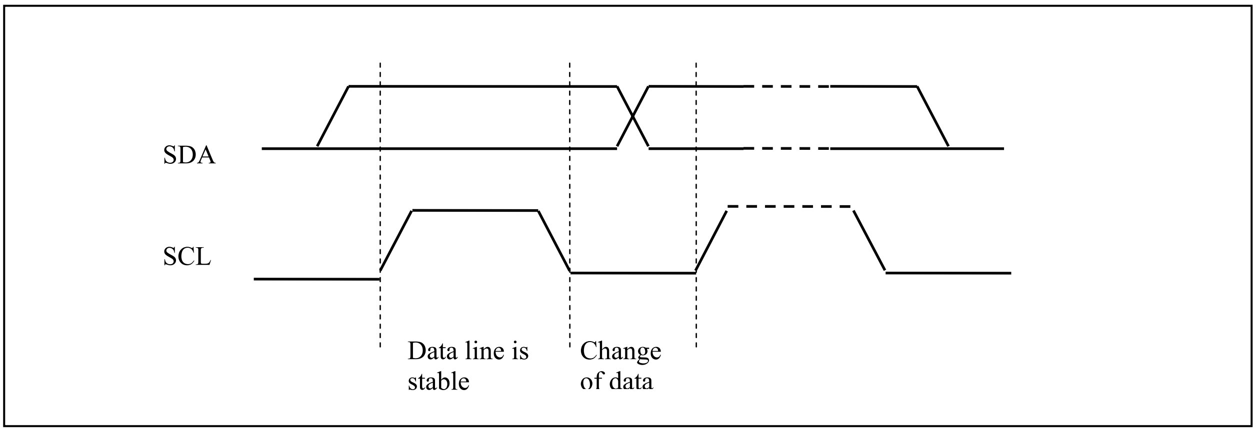
Figure 8-9 : Definition of the acknowledgement condition



Please be noted that the transmission of the data bit has some limitations.

1. The data bit, which is transmitted during each SCL pulse, must keep at a stable state within the "HIGH" period of the clock pulse. Please refer to the Figure 8-10 for graphical representations. Except in start or stop conditions, the data line can be switched only when the SCL is LOW.
2. Both the data line (SDA) and the clock line (SCL) should be pulled up by external resistors.

Figure 8-10 : Definition of the data transfer condition



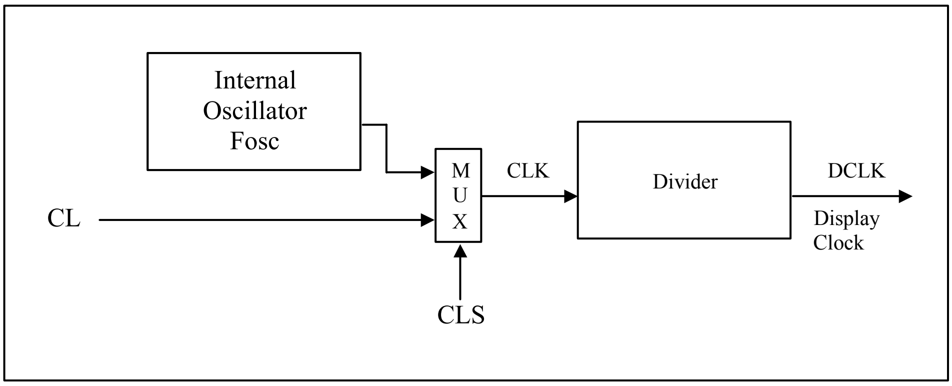
8.2 Command Decoder

This module determines whether the input data is interpreted as data or command. Data is interpreted based upon the input of the D/C# pin.

If D/C# pin is HIGH, D[7:0] is interpreted as display data written to Graphic Display Data RAM (GDDRAM). If it is LOW, the input at D[7:0] is interpreted as a command. Then data input will be decoded and written to the corresponding command register.

8.3 Oscillator Circuit and Display Time Generator

Figure 8-11 : Oscillator Circuit and Display Time Generator



This module is an on-chip LOW power RC oscillator circuitry. The operation clock (CLK) can be generated either from internal oscillator or external source CL pin. This selection is done by CLS pin. If CLS pin is pulled HIGH, internal oscillator is chosen and CL should be left open. Pulling CLS pin LOW disables internal oscillator and external clock must be connected to CL pins for proper operation. When the internal oscillator is selected, its output frequency Fosc can be changed by command D5h A[7:4].

The display clock (DCLK) for the Display Timing Generator is derived from CLK. The division factor "D" can be programmed from 1 to 16 by command D5h

$$DCLK = F_{OSC} / D$$

The frame frequency of display is determined by the following formula.

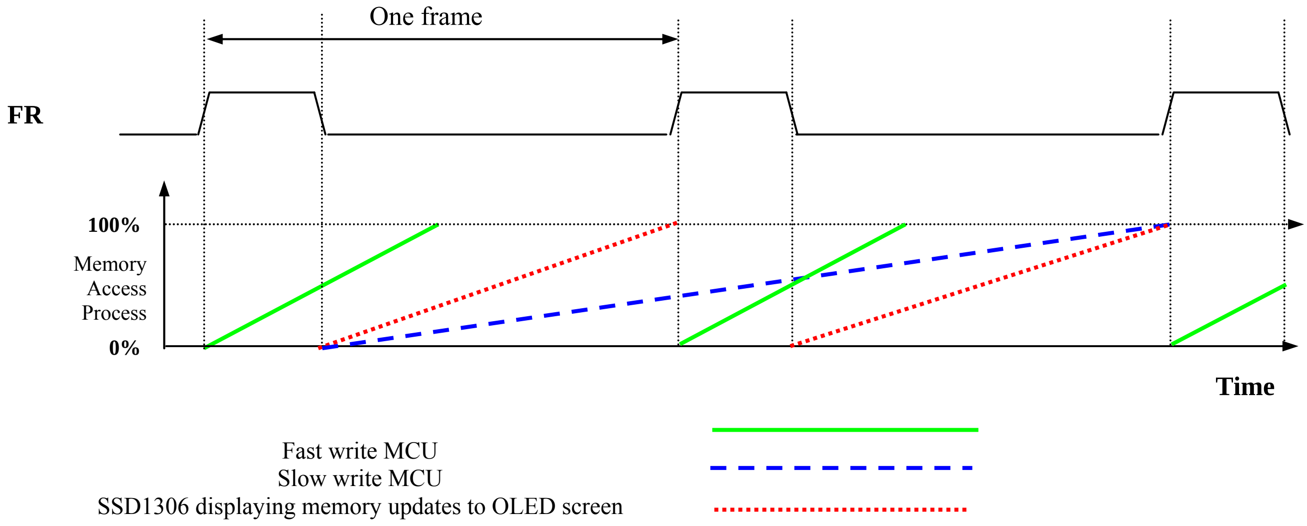
$$F_{FRM} = \frac{F_{osc}}{D \times K \times \text{No. of Mux}}$$

where

- D stands for clock divide ratio. It is set by command D5h A[3:0]. The divide ratio has the range from 1 to 16.
- K is the number of display clocks per row. The value is derived by
  K = Phase 1 period + Phase 2 period + BANK0 pulse width
  = 2 + 2 + 50 = 54 at power on reset
  (Please refer to Section 8.6 "Segment Drivers / Common Drivers" for the details of the "Phase")
- Number of multiplex ratio is set by command A8h. The power on reset value is 63 (i.e. 64MUX).
- F_{OSC} is the oscillator frequency. It can be changed by command D5h A[7:4]. The higher the register setting results in higher frequency.

8.4 FR synchronization

FR synchronization signal can be used to prevent tearing effect.



The starting time to write a new image to OLED driver is depended on the MCU writing speed. If MCU can finish writing a frame image within one frame period, it is classified as fast write MCU. For MCU needs longer writing time to complete (more than one frame but within two frames), it is a slow write one.

For fast write MCU: MCU should start to write new frame of ram data just after rising edge of FR pulse and should be finished well before the rising edge of the next FR pulse.

For slow write MCU: MCU should start to write new frame ram data after the falling edge of the 1st FR pulse and must be finished before the rising edge of the 3rd FR pulse.

8.5 Reset Circuit

When RES# input is LOW, the chip is initialized with the following status:

1. Display is OFF
2. 128 x 64 Display Mode
3. Normal segment and display data column address and row address mapping (SEG0 mapped to address 00h and COM0 mapped to address 00h)
4. Shift register data clear in serial interface
5. Display start line is set at display RAM address 0
6. Column address counter is set at 0
7. Normal scan direction of the COM outputs
8. Contrast control register is set at 7Fh
9. Normal display mode (Equivalent to A4h command)

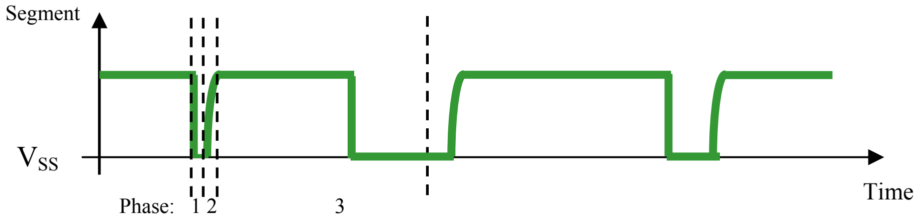
8.6 Segment Drivers / Common Drivers

Segment drivers deliver 128 current sources to drive the OLED panel. The driving current can be adjusted from 0 to 100uA with 256 steps. Common drivers generate voltage-scanning pulses.

The segment driving waveform is divided into three phases:

1. In phase 1, the OLED pixel charges of previous image are discharged in order to prepare for next image content display.
2. In phase 2, the OLED pixel is driven to the targeted voltage. The pixel is driven to attain the corresponding voltage level from V_{SS}. The period of phase 2 can be programmed in length from 1 to 15 DCLKs. If the capacitance value of the pixel of OLED panel is larger, a longer period is required to charge up the capacitor to reach the desired voltage.
3. In phase 3, the OLED driver switches to use current source to drive the OLED pixels and this is the current drive stage.

Figure 8-12 : Segment Output Waveform in three phases



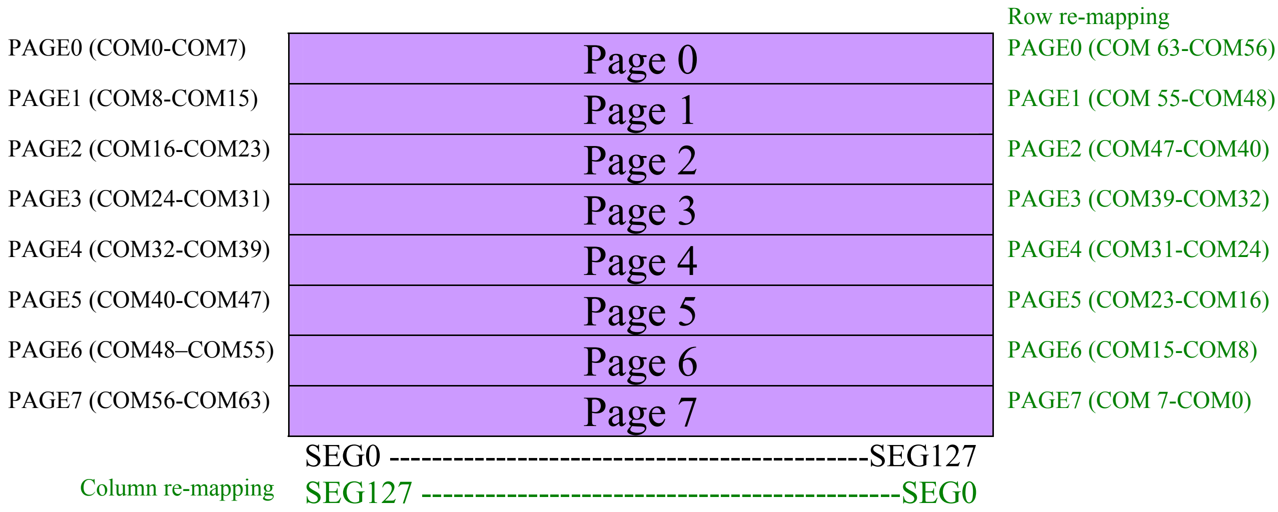
After finishing phase 3, the driver IC will go back to phase 1 to display the next row image data. This three-step cycle is run continuously to refresh image display on OLED panel.

In phase 3, if the length of current drive pulse width is set to 50, after finishing 50 DCLKs in current drive phase, the driver IC will go back to phase 1 for next row display.

8.7 Graphic Display Data RAM (GDDRAM)

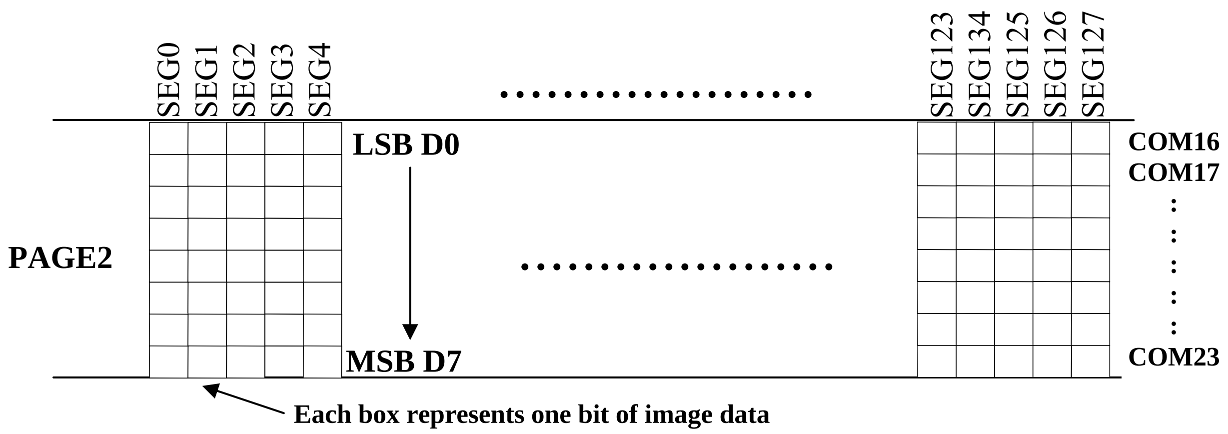
The GDDRAM is a bit mapped static RAM holding the bit pattern to be displayed. The size of the RAM is 128 x 64 bits and the RAM is divided into eight pages, from PAGE0 to PAGE7, which are used for monochrome 128x64 dot matrix display, as shown in Figure 8-13.

Figure 8-13 : GDDRAM pages structure of SSD1306



When one data byte is written into GDDRAM, all the rows image data of the same page of the current column are filled (i.e. the whole column (8 bits) pointed by the column address pointer is filled.). Data bit D0 is written into the top row, while data bit D7 is written into bottom row as shown in Figure 8-14.

Figure 8-14 : Enlargement of GDDRAM (No row re-mapping and column-remapping)



For mechanical flexibility, re-mapping on both Segment and Common outputs can be selected by software as shown in Figure 8-13.

For vertical shifting of the display, an internal register storing the display start line can be set to control the portion of the RAM data to be mapped to the display (command D3h).

8.8 SEG/COM Driving block

This block is used to derive the incoming power sources into the different levels of internal use voltage and current.

- V_{CC} is the most positive voltage supply.
- V_{COMH} is the Common deselected level. It is internally regulated.
- V_{LSS} is the ground path of the analog and panel current.
- I_{REF} is a reference current source for segment current drivers I_{SEG}. The relationship between reference current and segment current of a color is:

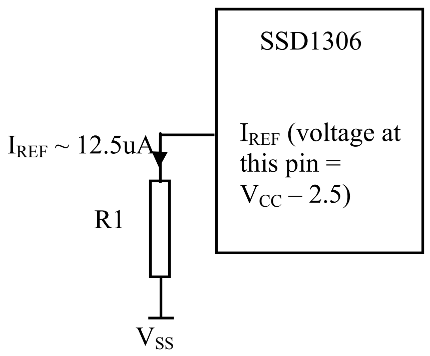
I_{SEG} = Contrast / 256 x I_{REF} x scale factor

in which

the contrast (0~255) is set by Set Contrast command 81h; and
the scale factor is 8 by default.

The magnitude of I_{REF} is controlled by the value of resistor, which is connected between I_{REF} pin and V_{SS} as shown in Figure 8-15. It is recommended to set I_{REF} to 12.5 ± 2uA so as to achieve I_{SEG} = 100uA at maximum contrast 255.

Figure 8-15 : I_{REF} Current Setting by Resistor Value



Since the voltage at I_{REF} pin is V_{CC} – 2.5V, the value of resistor R1 can be found as below:

For I_{REF} = 12.5uA, V_{CC} =12V:

R1 = (Voltage at I_{REF} – V_{SS}) / I_{REF}
= (12 – 2.5) / 12.5uA
= 760KΩ

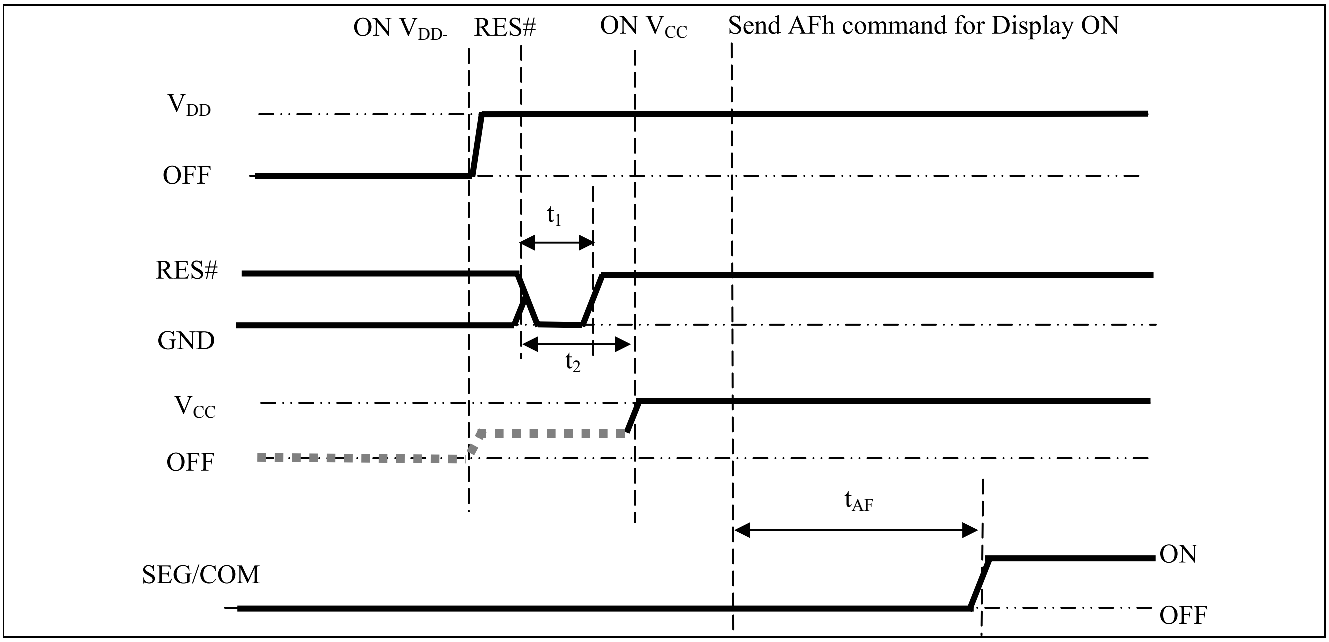
8.9 Power ON and OFF sequence

The following figures illustrate the recommended power ON and power OFF sequence of SSD1306

Power ON sequence:

1. Power ON V_{DD}
2. After V_{DD} become stable, set RES# pin LOW (logic low) for at least 3us (t_1) (4) and then HIGH (logic high).
3. After set RES# pin LOW (logic low), wait for at least 3us (t_2). Then Power ON V_{CC}. (1)
4. After V_{CC} become stable, send command AFh for display ON. SEG/COM will be ON after 100ms (t_{AF}).

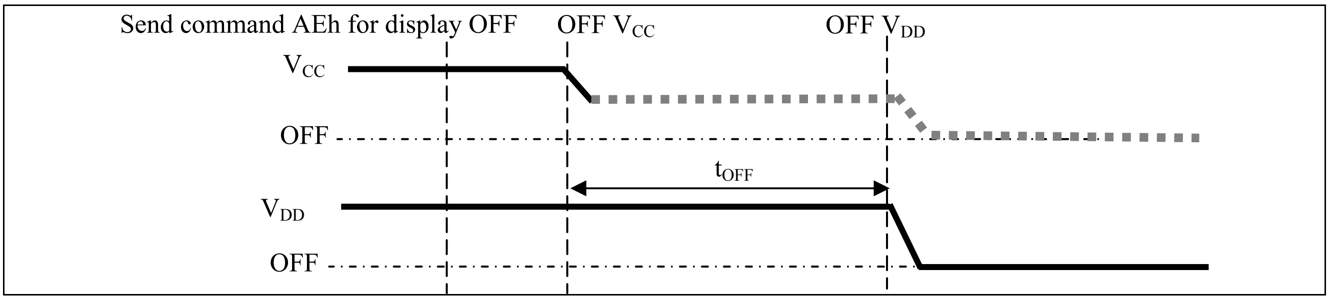
Figure 8-16 : The Power ON sequence



Power OFF sequence:

1. Send command AEh for display OFF.
2. Power OFF V_{CC}. (1), (2), (3)
3. Power OFF V_{DD} after t_{OFF}. (5) (Typical t_{OFF}=100ms)

Figure 8-17 : The Power OFF sequence



Note:

(1) Since an ESD protection circuit is connected between V_{DD} and V_{CC}, V_{CC} becomes lower than V_{DD} whenever V_{DD} is ON and V_{CC} is OFF as shown in the dotted line of V_{CC} in Figure 8-16 and Figure 8-17.
(2) V_{CC} should be kept float (i.e. disable) when it is OFF.
(3) Power Pins (V_{DD} , V_{CC}) can never be pulled to ground under any circumstance.
(4) The register values are reset after t_1.
(5) V_{DD} should not be Power OFF before V_{CC} Power OFF.

9 COMMAND TABLE

Table 9-1: Command Table

(D/C#=0, R/W#(WR#) = 0, E(RD#=1) unless specific setting is stated)

1. Fundamental Command Table

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0 0	81 A[7:0]	1 A_7	0 A_6	0 A_5	0 A_4	0 A_3	0 A_2	0 A_1	1 A_0	Set Contrast Control	Double byte command to select 1 out of 256 contrast steps. Contrast increases as the value increases. (RESET = 7Fh)
0	A4/A5	1	0	1	0	0	1	0	X_0	Entire Display ON	A4h, X_0=0b: Resume to RAM content display (RESET) Output follows RAM content A5h, X_0=1b: Entire display ON Output ignores RAM content
0	A6/A7	1	0	1	0	0	1	1	X_0	Set Normal/Inverse Display	A6h, X[0]=0b: Normal display (RESET) 0 in RAM: OFF in display panel 1 in RAM: ON in display panel A7h, X[0]=1b: Inverse display 0 in RAM: ON in display panel 1 in RAM: OFF in display panel
0	AE AF	1	0	1	0	1	1	1	X_0	Set Display ON/OFF	AEh, X[0]=0b:Display OFF (sleep mode) (RESET) AFh X[0]=1b:Display ON in normal mode

2. Scrolling Command Table

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0 0 0 0 0 0 0	26/27 A[7:0] B[2:0] C[2:0] D[2:0] E[7:0] F[7:0]	0 0 * * * 0 1	0 0 * * * 0 1	1 0 * * * 0 1	0 0 * * * 0 1	0 0 * * * 0 1	1 0 B_2 C_2 D_2 0 1	1 0 B_1 C_1 D_1 0 1	X_0 0 B_0 C_0 D_0 0 1	Continuous Horizontal Scroll Setup	26h, X[0]=0, Right Horizontal Scroll 27h, X[0]=1, Left Horizontal Scroll (Horizontal scroll by 1 column) A[7:0] : Dummy byte (Set as 00h) B[2:0] : Define start page address 000b – PAGE0 | 011b – PAGE3 | 110b – PAGE6 001b – PAGE1 | 100b – PAGE4 | 111b – PAGE7 010b – PAGE2 | 101b – PAGE5 C[2:0] : Set time interval between each scroll step in terms of frame frequency 000b – 5 frames | 100b – 3 frames 001b – 64 frames | 101b – 4 frames 010b – 128 frames | 110b – 25 frame 011b – 256 frames | 111b – 2 frame D[2:0] : Define end page address 000b – PAGE0 | 011b – PAGE3 | 110b – PAGE6 001b – PAGE1 | 100b – PAGE4 | 111b – PAGE7 010b – PAGE2 | 101b – PAGE5 The value of D[2:0] must be larger or equal to B[2:0] E[7:0] : Dummy byte (Set as 00h) F[7:0] : Dummy byte (Set as FFh)

2. Scrolling Command Table

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0	29/2A	0	0	1	0	1	0	X_1	X_0	Continuous Vertical and Horizontal Scroll Setup	29h, X_1X_0=01b : Vertical and Right Horizontal Scroll 2Ah, X_1X_0=10b : Vertical and Left Horizontal Scroll (Horizontal scroll by 1 column) A[7:0] : Dummy byte B[2:0] : Define start page address 000b – PAGE0 / 011b – PAGE3 / 110b – PAGE6 001b – PAGE1 / 100b – PAGE4 / 111b – PAGE7 010b – PAGE2 / 101b – PAGE5 C[2:0] : Set time interval between each scroll step in terms of frame frequency 000b – 5 frames / 100b – 3 frames 001b – 64 frames / 101b – 4 frames 010b – 128 frames / 110b – 25 frame 011b – 256 frames / 111b – 2 frame D[2:0] : Define end page address 000b – PAGE0 / 011b – PAGE3 / 110b – PAGE6 001b – PAGE1 / 100b – PAGE4 / 111b – PAGE7 010b – PAGE2 / 101b – PAGE5 The value of D[2:0] must be larger or equal to B[2:0] E[5:0] : Vertical scrolling offset e.g. E[5:0]= 01h refer to offset =1 row E[5:0] =3Fh refer to offset =63 rows **Note** (1) No continuous vertical scrolling is available.
0	A[2:0]	0	0	0	0	0	0	0	0		
0	B[2:0]	*	*	*	*	*	B_2	B_1	B_0		
0	C[2:0]	*	*	*	*	*	C_2	C_1	C_0		
0	D[2:0]	*	*	*	*	*	D_2	D_1	D_0		
0	E[5:0]	*	*	E_5	E_4	E_3	E_2	E_1	E_0		
0	2E	0	0	1	0	1	1	1	0	Deactivate scroll	Stop scrolling that is configured by command 26h/27h/29h/2Ah. **Note** (1) After sending 2Eh command to deactivate the scrolling action, the ram data needs to be rewritten.
0	2F	0	0	1	0	1	1	1	1	Activate scroll	Start scrolling that is configured by the scrolling setup commands :26h/27h/29h/2Ah with the following valid sequences: Valid command sequence 1: 26h ;2Fh. Valid command sequence 2: 27h ;2Fh. Valid command sequence 3: 29h ;2Fh. Valid command sequence 4: 2Ah ;2Fh. For example, if “26h; 2Ah; 2Fh.” commands are issued, the setting in the last scrolling setup command, i.e. 2Ah in this case, will be executed. In other words, setting in the last scrolling setup command overwrites the setting in the previous scrolling setup commands.

2. Scrolling Command Table											
D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0 0 0	A3 A[5:0] B[6:0]	1 * *	0 * B_6	1 A_5 B_5	0 A_4 B_4	0 A_3 B_3	0 A_2 B_2	1 A_1 B_1	1 A_0 B_0	Set Vertical Scroll Area	A[5:0] : Set No. of rows in top fixed area. The No. of rows in top fixed area is referenced to the top of the GDDRAM (i.e. row 0).[RESET = 0] B[6:0] : Set No. of rows in scroll area. This is the number of rows to be used for vertical scrolling. The scroll area starts in the first row below the top fixed area. [RESET = 64] **Note** (1) A[5:0]+B[6:0] <= MUX ratio (2) B[6:0] <= MUX ratio (3a) Vertical scrolling offset (E[5:0] in 29h/2Ah) < B[6:0] (3b) Set Display Start Line ($X_5X_4X_3X_2X_1X_0$ of 40h~7Fh) < B[6:0] (4) The last row of the scroll area shifts to the first row of the scroll area. (5) For 64d MUX display A[5:0] = 0, B[6:0]=64 : whole area scrolls A[5:0]= 0, B[6:0] < 64 : top area scrolls A[5:0] + B[6:0] < 64 : central area scrolls A[5:0] + B[6:0] = 64 : bottom area scrolls

3. Addressing Setting Command Table											
D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0	00~0F	0	0	0	0	X_3	X_2	X_1	X_0	Set Lower Column Start Address for Page Addressing Mode	Set the lower nibble of the column start address register for Page Addressing Mode using X[3:0] as data bits. The initial display line register is reset to 0000b after RESET. **Note** (1) This command is only for page addressing mode
0	10~1F	0	0	0	1	X_3	X_2	X_1	X_0	Set Higher Column Start Address for Page Addressing Mode	Set the higher nibble of the column start address register for Page Addressing Mode using X[3:0] as data bits. The initial display line register is reset to 0000b after RESET. **Note** (1) This command is only for page addressing mode
0 0	20 A[1:0]	0 *	0 *	1 *	0 *	0 *	0 *	0 A_1	0 A_0	Set Memory Addressing Mode	A[1:0] = 00b, Horizontal Addressing Mode A[1:0] = 01b, Vertical Addressing Mode A[1:0] = 10b, Page Addressing Mode (RESET) A[1:0] = 11b, Invalid
0 0 0	21 A[6:0] B[6:0]	0 * *	0 A_6 B_6	1 A_5 B_5	0 A_4 B_4	0 A_3 B_3	0 A_2 B_2	0 A_1 B_1	1 A_0 B_0	Set Column Address	Setup column start and end address A[6:0] : Column start address, range : 0-127d, (RESET=0d) B[6:0]: Column end address, range : 0-127d, (RESET =127d) **Note** (1) This command is only for horizontal or vertical addressing mode.

3. Addressing Setting Command Table											
D/C#	**Hex**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Command**	**Description**
0 0 0	22 A[2:0] B[2:0]	0 * *	0 * *	1 * *	0 * *	0 * *	0 A_2 B_2	1 A_1 B_1	0 A_0 B_0	Set Page Address	Setup page start and end address A[2:0] : Page start Address, range : 0-7d, (RESET = 0d) B[2:0] : Page end Address, range : 0-7d, (RESET = 7d) **Note** (1) This command is only for horizontal or vertical addressing mode.
0	B0~B7	1	0	1	1	0	X_2	X_1	X_0	Set Page Start Address for Page Addressing Mode	Set GDDRAM Page Start Address (PAGE0~PAGE7) for Page Addressing Mode using X[2:0]. **Note** (1) This command is only for page addressing mode

4. Hardware Configuration (Panel resolution & layout related) Command Table											
D/C#	**Hex**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Command**	**Description**
0	40~7F	0	1	X_5	X_4	X_3	X_2	X_1	X_0	Set Display Start Line	Set display RAM display start line register from 0-63 using $X_5X_3X_2X_1X_0$. Display start line register is reset to 000000b during RESET.
0	A0/A1	1	0	1	0	0	0	0	X_0	Set Segment Re-map	A0h, X[0]=0b: column address 0 is mapped to SEG0 (RESET) A1h, X[0]=1b: column address 127 is mapped to SEG0
0 0	A8 A[5:0]	1 *	0 *	1 A_5	0 A_4	1 A_3	0 A_2	0 A_1	0 A_0	Set Multiplex Ratio	Set MUX ratio to N+1 MUX N=A[5:0] : from 16MUX to 64MUX, RESET= 111111b (i.e. 63d, 64MUX) A[5:0] from 0 to 14 are invalid entry.
0	C0/C8	1	1	0	0	X_3	0	0	0	Set COM Output Scan Direction	C0h, X[3]=0b: normal mode (RESET) Scan from COM0 to COM[N –1] C8h, X[3]=1b: remapped mode. Scan from COM[N-1] to COM0 Where N is the Multiplex ratio.
0 0	D3 A[5:0]	1 *	1 *	0 A_5	1 A_4	0 A_3	0 A_2	1 A_1	1 A_0	Set Display Offset	Set vertical shift by COM from 0d~63d The value is reset to 00h after RESET.
0 0	DA A[5:4]	1 0	1 0	0 A_5	1 A_4	1 0	0 0	1 1	0 0	Set COM Pins Hardware Configuration	A[4]=0b, Sequential COM pin configuration A[4]=1b(RESET), Alternative COM pin configuration A[5]=0b(RESET), Disable COM Left/Right remap A[5]=1b, Enable COM Left/Right remap

<table>
<tr><th colspan="12">5. Timing & Driving Scheme Setting Command Table</th></tr>
<tr><th>D/C#</th><th>Hex</th><th>D7</th><th>D6</th><th>D5</th><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th><th>Command</th><th>Description</th></tr>
<tr><td>0
0</td><td>D5
A[7:0]</td><td>1
A_7</td><td>1
A_6</td><td>0
A_5</td><td>1
A_4</td><td>0
A_3</td><td>1
A_2</td><td>0
A_1</td><td>1
A_0</td><td>Set Display Clock Divide Ratio/Oscillator Frequency</td><td>A[3:0] : Define the divide ratio (D) of the display clocks (DCLK):
Divide ratio= A[3:0] + 1, RESET is 0000b (divide ratio = 1)

A[7:4] : Set the Oscillator Frequency, F_{OSC}.
Oscillator Frequency increases with the value of A[7:4] and vice versa.
RESET is 1000b
Range:0000b~1111b
Frequency increases as setting value increases.</td></tr>
<tr><td>0
0</td><td>D9
A[7:0]</td><td>1
A_7</td><td>1
A_6</td><td>0
A_5</td><td>1
A_4</td><td>1
A_3</td><td>0
A_2</td><td>0
A_1</td><td>1
A_0</td><td>Set Pre-charge Period</td><td>A[3:0] : Phase 1 period of up to 15 DCLK clocks 0 is invalid entry (RESET=2h)

A[7:4] : Phase 2 period of up to 15 DCLK clocks 0 is invalid entry (RESET=2h)</td></tr>
<tr><td>0
0</td><td>DB
A[6:4]</td><td>1
0</td><td>1
A_6</td><td>0
A_5</td><td>1
A_4</td><td>1
0</td><td>0
0</td><td>1
0</td><td>1
0</td><td>Set V_{COMH} Deselect Level</td><td><table><tr><th>A[6:4]</th><th>Hex code</th><th>V_{COMH} deselect level</th></tr><tr><td>000b</td><td>00h</td><td>~ 0.65 x V_{CC}</td></tr><tr><td>010b</td><td>20h</td><td>~ 0.77 x V_{CC} (RESET)</td></tr><tr><td>011b</td><td>30h</td><td>~ 0.83 x V_{CC}</td></tr></table></td></tr>
<tr><td>0</td><td>E3</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>NOP</td><td>Command for no operation</td></tr>
</table>

Note

(1) "*" stands for "Don't care".

Table 9-2 : Read Command Table

Bit Pattern	Command	Description
$D_7D_6D_5D_4D_3D_2D_1D_0$	Status Register Read	D[7] : Reserved D[6] : "1" for display OFF / "0" for display ON D[5] : Reserved D[4] : Reserved D[3] : Reserved D[2] : Reserved D[1] : Reserved D[0] : Reserved

Note
(1) Patterns other than those given in the Command Table are prohibited to enter the chip as a command; as unexpected results can occur.

9.1 Data Read / Write

To read data from the GDDRAM, select HIGH for both the R/W# (WR#) pin and the D/C# pin for 6800-series parallel mode and select LOW for the E (RD#) pin and HIGH for the D/C# pin for 8080-series parallel mode. No data read is provided in serial mode operation.
In normal data read mode the GDDRAM column address pointer will be increased automatically by one after each data read.
Also, a dummy read is required before the first data read.
To write data to the GDDRAM, select LOW for the R/W# (WR#) pin and HIGH for the D/C# pin for both 6800-series parallel mode and 8080-series parallel mode. The serial interface mode is always in write mode. The GDDRAM column address pointer will be increased automatically by one after each data write.

Table 9-3 : Address increment table (Automatic)

D/C#	R/W# (WR#)	Comment	Address Increment
0	0	Write Command	No
0	1	Read Status	No
1	0	Write Data	Yes
1	1	Read Data	Yes

10 COMMAND DESCRIPTIONS

10.1 Fundamental Command

10.1.1 Set Lower Column Start Address for Page Addressing Mode (00h~0Fh)

This command specifies the lower nibble of the 8-bit column start address for the display data RAM under Page Addressing Mode. The column address will be incremented by each data access. Please refer to Section Table 9-1 and Section 10.1.3 for details.

10.1.2 Set Higher Column Start Address for Page Addressing Mode (10h~1Fh)

This command specifies the higher nibble of the 8-bit column start address for the display data RAM under Page Addressing Mode. The column address will be incremented by each data access. Please refer to Section Table 9-1 and Section 10.1.3 for details.

10.1.3 Set Memory Addressing Mode (20h)

There are 3 different memory addressing mode in SSD1306: page addressing mode, horizontal addressing mode and vertical addressing mode. This command sets the way of memory addressing into one of the above three modes. In there, "COL" means the graphic display data RAM column.

Page addressing mode (A[1:0]=10xb)

In page addressing mode, after the display RAM is read/written, the column address pointer is increased automatically by 1. If the column address pointer reaches column end address, the column address pointer is reset to column start address and page address pointer is not changed. Users have to set the new page and column addresses in order to access the next page RAM content. The sequence of movement of the PAGE and column address point for page addressing mode is shown in Figure 10-1.

Figure 10-1 : Address Pointer Movement of Page addressing mode

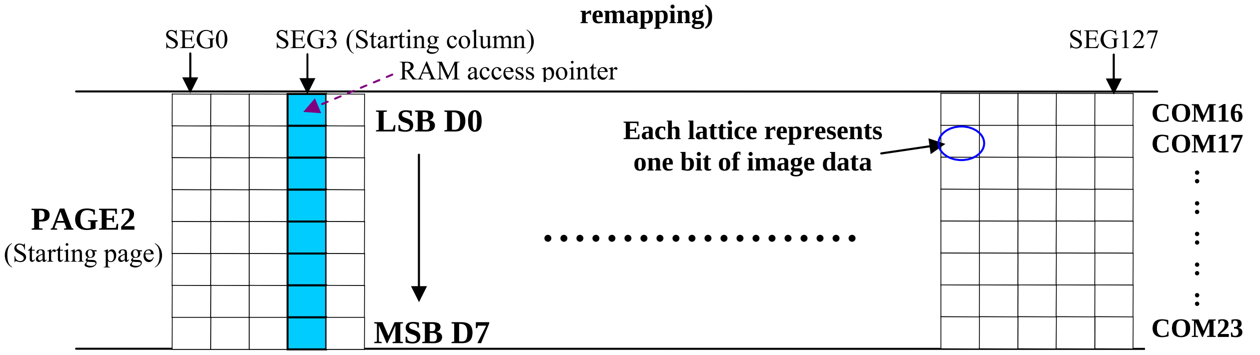
	COL0	COL 1	.....	COL 126	COL 127
PAGE0					
PAGE1					
:	:	:	:	:	:
PAGE6					
PAGE7					

In normal display data RAM read or write and page addressing mode, the following steps are required to define the starting RAM access pointer location:

- Set the page start address of the target display location by command B0h to B7h.
- Set the lower start column address of pointer by command 00h~0Fh.
- Set the upper start column address of pointer by command 10h~1Fh.

For example, if the page address is set to B2h, lower column address is 03h and upper column address is 10h, then that means the starting column is SEG3 of PAGE2. The RAM access pointer is located as shown in Figure 10-2. The input data byte will be written into RAM position of column 3.

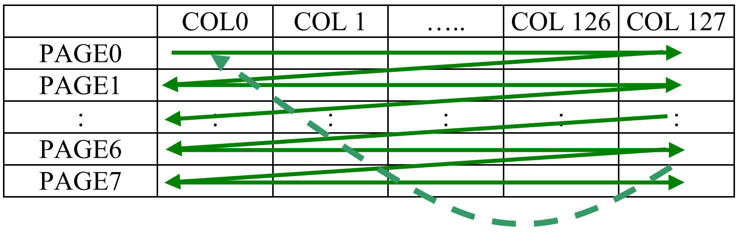
Figure 10-2 : Example of GDDRAM access pointer setting in Page Addressing Mode (No row and column-remapping)



Horizontal addressing mode (A[1:0]=00b)

In horizontal addressing mode, after the display RAM is read/written, the column address pointer is increased automatically by 1. If the column address pointer reaches column end address, the column address pointer is reset to column start address and page address pointer is increased by 1. The sequence of movement of the page and column address point for horizontal addressing mode is shown in Figure 10-3. When both column and page address pointers reach the end address, the pointers are reset to column start address and page start address (Dotted line in Figure 10-3.)

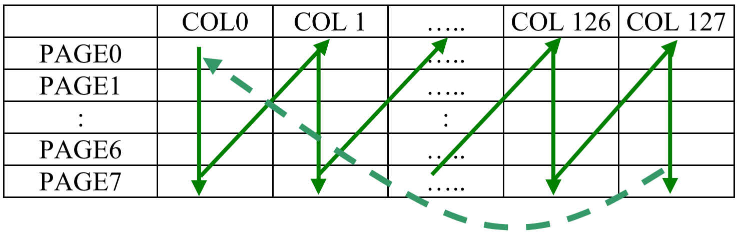
Figure 10-3 : Address Pointer Movement of Horizontal addressing mode



Vertical addressing mode: (A[1:0]=01b)

In vertical addressing mode, after the display RAM is read/written, the page address pointer is increased automatically by 1. If the page address pointer reaches the page end address, the page address pointer is reset to page start address and column address pointer is increased by 1. The sequence of movement of the page and column address point for vertical addressing mode is shown in Figure 10-4. When both column and page address pointers reach the end address, the pointers are reset to column start address and page start address (Dotted line in Figure 10-4.)

Figure 10-4 : Address Pointer Movement of Vertical addressing mode



In normal display data RAM read or write and horizontal / vertical addressing mode, the following steps are required to define the RAM access pointer location:

- Set the column start and end address of the target display location by command 21h.
- Set the page start and end address of the target display location by command 22h.

Example is shown in Figure 10-5.

10.1.4 Set Column Address (21h)

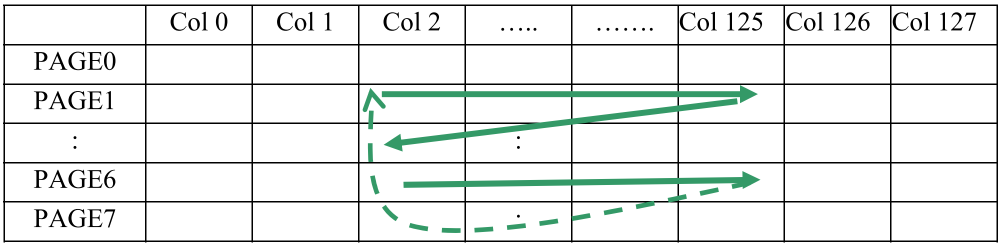
This triple byte command specifies column start address and end address of the display data RAM. This command also sets the column address pointer to column start address. This pointer is used to define the current read/write column address in graphic display data RAM. If horizontal address increment mode is enabled by command 20h, after finishing read/write one column data, it is incremented automatically to the next column address. Whenever the column address pointer finishes accessing the end column address, it is reset back to start column address and the row address is incremented to the next row.

10.1.5 Set Page Address (22h)

This triple byte command specifies page start address and end address of the display data RAM. This command also sets the page address pointer to page start address. This pointer is used to define the current read/write page address in graphic display data RAM. If vertical address increment mode is enabled by command 20h, after finishing read/write one page data, it is incremented automatically to the next page address. Whenever the page address pointer finishes accessing the end page address, it is reset back to start page address.

The figure below shows the way of column and page address pointer movement through the example: column start address is set to 2 and column end address is set to 125, page start address is set to 1 and page end address is set to 6; Horizontal address increment mode is enabled by command 20h. In this case, the graphic display data RAM column accessible range is from column 2 to column 125 and from page 1 to page 6 only. In addition, the column address pointer is set to 2 and page address pointer is set to 1. After finishing read/write one pixel of data, the column address is increased automatically by 1 to access the next RAM location for next read/write operation (*solid line in Figure 10-5*). Whenever the column address pointer finishes accessing the end column 125, it is reset back to column 2 and page address is automatically increased by 1 (*solid line in Figure 10-5*). While the end page 6 and end column 125 RAM location is accessed, the page address is reset back to 1 and the column address is reset back to 2 (*dotted line in Figure 10-5*). .

Figure 10-5 : Example of Column and Row Address Pointer Movement



10.1.6 Set Display Start Line (40h~7Fh)

This command sets the Display Start Line register to determine starting address of display RAM, by selecting a value from 0 to 63. With value equal to 0, RAM row 0 is mapped to COM0. With value equal to 1, RAM row 1 is mapped to COM0 and so on.

Refer to Table 10-1 for more illustrations.

10.1.7 Set Contrast Control for BANK0 (81h)

This command sets the Contrast Setting of the display. The chip has 256 contrast steps from 00h to FFh. The segment output current increases as the contrast step value increases.

10.1.8 Set Segment Re-map (A0h/A1h)

This command changes the mapping between the display data column address and the segment driver. It allows flexibility in OLED module design. Please refer to Table 9-1.

This command only affects subsequent data input. Data already stored in GDDRAM will have no changes.

10.1.9 Entire Display ON (A4h/A5h)

A4h command enable display outputs according to the GDDRAM contents.
If A5h command is issued, then by using A4h command, the display will resume to the GDDRAM contents.
In other words, A4h command resumes the display from entire display "ON" stage.
A5h command forces the entire display to be "ON", regardless of the contents of the display data RAM.

10.1.10 Set Normal/Inverse Display (A6h/A7h)

This command sets the display to be either normal or inverse. In normal display a RAM data of 1 indicates an "ON" pixel while in inverse display a RAM data of 0 indicates an "ON" pixel.

10.1.11 Set Multiplex Ratio (A8h)

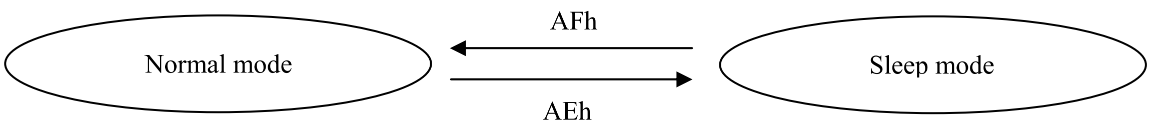
This command switches the default 63 multiplex mode to any multiplex ratio, ranging from 16 to 63. The output pads COM0~COM63 will be switched to the corresponding COM signal.

10.1.12 Set Display ON/OFF (AEh/AFh)

These single byte commands are used to turn the OLED panel display ON or OFF.
When the display is ON, the selected circuits by Set Master Configuration command will be turned ON.
When the display is OFF, those circuits will be turned OFF and the segment and common output are in V_{SS} state and high impedance state, respectively. These commands set the display to one of the two states:

- AEh : Display OFF
- AFh : Display ON

Figure 10-6 :Transition between different modes



10.1.13 Set Page Start Address for Page Addressing Mode (B0h~B7h)

This command positions the page start address from 0 to 7 in GDDRAM under Page Addressing Mode. Please refer to Table 9-1 and Section 10.1.3 for details.

10.1.14 Set COM Output Scan Direction (C0h/C8h)

This command sets the scan direction of the COM output, allowing layout flexibility in the OLED module design. Additionally, the display will show once this command is issued. For example, if this command is sent during normal display then the graphic display will be vertically flipped immediately. Please refer to Table 10-3 for details.

10.1.15 Set Display Offset (D3h)

This is a double byte command. The second command specifies the mapping of the display start line to one of COM0~COM63 (assuming that COM0 is the display start line then the display start line register is equal to 0).

For example, to move the COM16 towards the COM0 direction by 16 lines the 6-bit data in the second byte should be given as 010000b. To move in the opposite direction by 16 lines the 6-bit data should be given by 64 – 16, so the second byte would be 100000b. The following two tables (Table 10-1, Table 10-2) show the example of setting the command C0h/C8h and D3h.

Table 10-1 : Example of Set Display Offset and Display Start Line with no Remap

	Output												
	64		64		64		56		56		56		Set MUX ratio(A8h)
	Normal		Normal		Normal		Normal		Normal		Normal		COM Normal / Remapped (C0h / C8h)
Hardware	0		8		0		0		8		0		Display offset (D3h)
pin name	0		0		8		0		0		8		Display start line (40h - 7Fh)
COM0	Row0	RAM0	Row8	RAM8	Row0	RAM8	Row0	RAM0	Row8	RAM8	Row0	RAM8	
COM1	Row1	RAM1	Row9	RAM9	Row1	RAM9	Row1	RAM1	Row9	RAM9	Row1	RAM9	
COM2	Row2	RAM2	Row10	RAM10	Row2	RAM10	Row2	RAM2	Row10	RAM10	Row2	RAM10	
COM3	Row3	RAM3	Row11	RAM11	Row3	RAM11	Row3	RAM3	Row11	RAM11	Row3	RAM11	
COM4	Row4	RAM4	Row12	RAM12	Row4	RAM12	Row4	RAM4	Row12	RAM12	Row4	RAM12	
COM5	Row5	RAM5	Row13	RAM13	Row5	RAM13	Row5	RAM5	Row13	RAM13	Row5	RAM13	
COM6	Row6	RAM6	Row14	RAM14	Row6	RAM14	Row6	RAM6	Row14	RAM14	Row6	RAM14	
COM7	Row7	RAM7	Row15	RAM15	Row7	RAM15	Row7	RAM7	Row15	RAM15	Row7	RAM15	
COM8	Row8	RAM8	Row16	RAM16	Row8	RAM16	Row8	RAM8	Row16	RAM16	Row8	RAM16	
COM9	Row9	RAM9	Row17	RAM17	Row9	RAM17	Row9	RAM9	Row17	RAM17	Row9	RAM17	
COM10	Row10	RAM10	Row18	RAM18	Row10	RAM18	Row10	RAM10	Row18	RAM18	Row10	RAM18	
COM11	Row11	RAM11	Row19	RAM19	Row11	RAM19	Row11	RAM11	Row19	RAM19	Row11	RAM19	
COM12	Row12	RAM12	Row20	RAM20	Row12	RAM20	Row12	RAM12	Row20	RAM20	Row12	RAM20	
COM13	Row13	RAM13	Row21	RAM21	Row13	RAM21	Row13	RAM13	Row21	RAM21	Row13	RAM21	
COM14	Row14	RAM14	Row22	RAM22	Row14	RAM22	Row14	RAM14	Row22	RAM22	Row14	RAM22	
COM15	Row15	RAM15	Row23	RAM23	Row15	RAM23	Row15	RAM15	Row23	RAM23	Row15	RAM23	
COM16	Row16	RAM16	Row24	RAM24	Row16	RAM24	Row16	RAM16	Row24	RAM24	Row16	RAM24	
COM17	Row17	RAM17	Row25	RAM25	Row17	RAM25	Row17	RAM17	Row25	RAM25	Row17	RAM25	
COM18	Row18	RAM18	Row26	RAM26	Row18	RAM26	Row18	RAM18	Row26	RAM26	Row18	RAM26	
COM19	Row19	RAM19	Row27	RAM27	Row19	RAM27	Row19	RAM19	Row27	RAM27	Row19	RAM27	
COM20	Row20	RAM20	Row28	RAM28	Row20	RAM28	Row20	RAM20	Row28	RAM28	Row20	RAM28	
COM21	Row21	RAM21	Row29	RAM29	Row21	RAM29	Row21	RAM21	Row29	RAM29	Row21	RAM29	
COM22	Row22	RAM22	Row30	RAM30	Row22	RAM30	Row22	RAM22	Row30	RAM30	Row22	RAM30	
COM23	Row23	RAM23	Row31	RAM31	Row23	RAM31	Row23	RAM23	Row31	RAM31	Row23	RAM31	
COM24	Row24	RAM24	Row32	RAM32	Row24	RAM32	Row24	RAM24	Row32	RAM32	Row24	RAM32	
COM25	Row25	RAM25	Row33	RAM33	Row25	RAM33	Row25	RAM25	Row33	RAM33	Row25	RAM33	
COM26	Row26	RAM26	Row34	RAM34	Row26	RAM34	Row26	RAM26	Row34	RAM34	Row26	RAM34	
COM27	Row27	RAM27	Row35	RAM35	Row27	RAM35	Row27	RAM27	Row35	RAM35	Row27	RAM35	
COM28	Row28	RAM28	Row36	RAM36	Row28	RAM36	Row28	RAM28	Row36	RAM36	Row28	RAM36	
COM29	Row29	RAM29	Row37	RAM37	Row29	RAM37	Row29	RAM29	Row37	RAM37	Row29	RAM37	
COM30	Row30	RAM30	Row38	RAM38	Row30	RAM38	Row30	RAM30	Row38	RAM38	Row30	RAM38	
COM31	Row31	RAM31	Row39	RAM39	Row31	RAM39	Row31	RAM31	Row39	RAM39	Row31	RAM39	
COM32	Row32	RAM32	Row40	RAM40	Row32	RAM40	Row32	RAM32	Row40	RAM40	Row32	RAM40	
COM33	Row33	RAM33	Row41	RAM41	Row33	RAM41	Row33	RAM33	Row41	RAM41	Row33	RAM41	
COM34	Row34	RAM34	Row42	RAM42	Row34	RAM42	Row34	RAM34	Row42	RAM42	Row34	RAM42	
COM35	Row35	RAM35	Row43	RAM43	Row35	RAM43	Row35	RAM35	Row43	RAM43	Row35	RAM43	
COM36	Row36	RAM36	Row44	RAM44	Row36	RAM44	Row36	RAM36	Row44	RAM44	Row36	RAM44	
COM37	Row37	RAM37	Row45	RAM45	Row37	RAM45	Row37	RAM37	Row45	RAM45	Row37	RAM45	
COM38	Row38	RAM38	Row46	RAM46	Row38	RAM46	Row38	RAM38	Row46	RAM46	Row38	RAM46	
COM39	Row39	RAM39	Row47	RAM47	Row39	RAM47	Row39	RAM39	Row47	RAM47	Row39	RAM47	
COM40	Row40	RAM40	Row48	RAM48	Row40	RAM48	Row40	RAM40	Row48	RAM48	Row40	RAM48	
COM41	Row41	RAM41	Row49	RAM49	Row41	RAM49	Row41	RAM41	Row49	RAM49	Row41	RAM49	
COM42	Row42	RAM42	Row50	RAM50	Row42	RAM50	Row42	RAM42	Row50	RAM50	Row42	RAM50	
COM43	Row43	RAM43	Row51	RAM51	Row43	RAM51	Row43	RAM43	Row51	RAM51	Row43	RAM51	
COM44	Row44	RAM44	Row52	RAM52	Row44	RAM52	Row44	RAM44	Row52	RAM52	Row44	RAM52	
COM45	Row45	RAM45	Row53	RAM53	Row45	RAM53	Row45	RAM45	Row53	RAM53	Row45	RAM53	
COM46	Row46	RAM46	Row54	RAM54	Row46	RAM54	Row46	RAM46	Row54	RAM54	Row46	RAM54	
COM47	Row47	RAM47	Row55	RAM55	Row47	RAM55	Row47	RAM47	Row55	RAM55	Row47	RAM55	
COM48	Row48	RAM48	Row56	RAM56	Row48	RAM56	Row48	RAM48	-	-	Row48	RAM56	
COM49	Row49	RAM49	Row57	RAM57	Row49	RAM57	Row49	RAM49	-	-	Row49	RAM57	
COM50	Row50	RAM50	Row58	RAM58	Row50	RAM58	Row50	RAM50	-	-	Row50	RAM58	
COM51	Row51	RAM51	Row59	RAM59	Row51	RAM59	Row51	RAM51	-	-	Row51	RAM59	
COM52	Row52	RAM52	Row60	RAM60	Row52	RAM60	Row52	RAM52	-	-	Row52	RAM60	
COM53	Row53	RAM53	Row61	RAM61	Row53	RAM61	Row53	RAM53	-	-	Row53	RAM61	
COM54	Row54	RAM54	Row62	RAM62	Row54	RAM62	Row54	RAM54	-	-	Row54	RAM62	
COM55	Row55	RAM55	Row63	RAM63	Row55	RAM63	Row55	RAM55	-	-	Row55	RAM63	
COM56	Row56	RAM56	Row0	RAM0	Row56	RAM0	-	-	Row0	RAM0	-	-	
COM57	Row57	RAM57	Row1	RAM1	Row57	RAM1	-	-	Row1	RAM1	-	-	
COM58	Row58	RAM58	Row2	RAM2	Row58	RAM2	-	-	Row2	RAM2	-	-	
COM59	Row59	RAM59	Row3	RAM3	Row59	RAM3	-	-	Row3	RAM3	-	-	
COM60	Row60	RAM60	Row4	RAM4	Row60	RAM4	-	-	Row4	RAM4	-	-	
COM61	Row61	RAM61	Row5	RAM5	Row61	RAM5	-	-	Row5	RAM5	-	-	
COM62	Row62	RAM62	Row6	RAM6	Row62	RAM6	-	-	Row6	RAM6	-	-	
COM63	Row63	RAM63	Row7	RAM7	Row63	RAM7	-	-	Row7	RAM7	-	-	
Display examples	(a)		(b)		(c)		(d)		(e)		(f)		

(a)

(b)

(c)

(d)

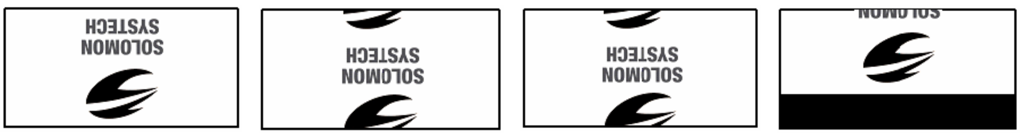
(e)

(f)

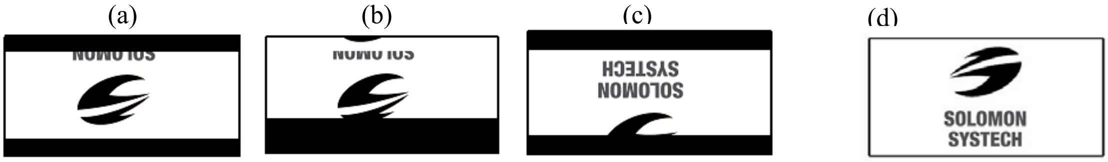
(RAM)

Table 10-2 :Example of Set Display Offset and Display Start Line with Remap

Hardware pin name	Output														
	64		64		64		48		48		48		48		Set MUX ratio(A8h)
	Remap		Remap		Remap		Remap		Remap		Remap		Remap		COM Normal / Remapped (C0h / C8h)
	0		8		0		0		8		0		8		Display offset (D3h)
	0		0		8		0		0		8		16		Display start line (40h - 7Fh)
COM0	Row63	RAM63	Row7	RAM7	Row63	RAM7	Row47	RAM47	-	-	Row47	RAM55	-	-	
COM1	Row62	RAM62	Row6	RAM6	Row62	RAM6	Row46	RAM46	-	-	Row46	RAM54	-	-	
COM2	Row61	RAM61	Row5	RAM5	Row61	RAM5	Row45	RAM45	-	-	Row45	RAM53	-	-	
COM3	Row60	RAM60	Row4	RAM4	Row60	RAM4	Row44	RAM44	-	-	Row44	RAM52	-	-	
COM4	Row59	RAM59	Row3	RAM3	Row59	RAM3	Row43	RAM43	-	-	Row43	RAM51	-	-	
COM5	Row58	RAM58	Row2	RAM2	Row58	RAM2	Row42	RAM42	-	-	Row42	RAM50	-	-	
COM6	Row57	RAM57	Row1	RAM1	Row57	RAM1	Row41	RAM41	-	-	Row41	RAM49	-	-	
COM7	Row56	RAM56	Row0	RAM0	Row56	RAM0	Row40	RAM40	-	-	Row40	RAM48	-	-	
COM8	Row55	RAM55	Row63	RAM63	Row55	RAM63	Row39	RAM39	Row47	RAM47	Row39	RAM47	Row47	RAM63	
COM9	Row54	RAM54	Row62	RAM62	Row54	RAM62	Row38	RAM38	Row46	RAM46	Row38	RAM46	Row46	RAM62	
COM10	Row53	RAM53	Row61	RAM61	Row53	RAM61	Row37	RAM37	Row45	RAM45	Row37	RAM45	Row45	RAM61	
COM11	Row52	RAM52	Row60	RAM60	Row52	RAM60	Row36	RAM36	Row44	RAM44	Row36	RAM44	Row44	RAM60	
COM12	Row51	RAM51	Row59	RAM59	Row51	RAM59	Row35	RAM35	Row43	RAM43	Row35	RAM43	Row43	RAM59	
COM13	Row50	RAM50	Row58	RAM58	Row50	RAM58	Row34	RAM34	Row42	RAM42	Row34	RAM42	Row42	RAM58	
COM14	Row49	RAM49	Row57	RAM57	Row49	RAM57	Row33	RAM33	Row41	RAM41	Row33	RAM41	Row41	RAM57	
COM15	Row48	RAM48	Row56	RAM56	Row48	RAM56	Row32	RAM32	Row40	RAM40	Row32	RAM40	Row40	RAM56	
COM16	Row47	RAM47	Row55	RAM55	Row47	RAM55	Row31	RAM31	Row39	RAM39	Row31	RAM39	Row39	RAM55	
COM17	Row46	RAM46	Row54	RAM54	Row46	RAM54	Row30	RAM30	Row38	RAM38	Row30	RAM38	Row38	RAM54	
COM18	Row45	RAM45	Row53	RAM53	Row45	RAM53	Row29	RAM29	Row37	RAM37	Row29	RAM37	Row37	RAM53	
COM19	Row44	RAM44	Row52	RAM52	Row44	RAM52	Row28	RAM28	Row36	RAM36	Row28	RAM36	Row36	RAM52	
COM20	Row43	RAM43	Row51	RAM51	Row43	RAM51	Row27	RAM27	Row35	RAM35	Row27	RAM35	Row35	RAM51	
COM21	Row42	RAM42	Row50	RAM50	Row42	RAM50	Row26	RAM26	Row34	RAM34	Row26	RAM34	Row34	RAM50	
COM22	Row41	RAM41	Row49	RAM49	Row41	RAM49	Row25	RAM25	Row33	RAM33	Row25	RAM33	Row33	RAM49	
COM23	Row40	RAM40	Row48	RAM48	Row40	RAM48	Row24	RAM24	Row32	RAM32	Row24	RAM32	Row32	RAM48	
COM24	Row39	RAM39	Row47	RAM47	Row39	RAM47	Row23	RAM23	Row31	RAM31	Row23	RAM31	Row31	RAM47	
COM25	Row38	RAM38	Row46	RAM46	Row38	RAM46	Row22	RAM22	Row30	RAM30	Row22	RAM30	Row30	RAM46	
COM26	Row37	RAM37	Row45	RAM45	Row37	RAM45	Row21	RAM21	Row29	RAM29	Row21	RAM29	Row29	RAM45	
COM27	Row36	RAM36	Row44	RAM44	Row36	RAM44	Row20	RAM20	Row28	RAM28	Row20	RAM28	Row28	RAM44	
COM28	Row35	RAM35	Row43	RAM43	Row35	RAM43	Row19	RAM19	Row27	RAM27	Row19	RAM27	Row27	RAM43	
COM29	Row34	RAM34	Row42	RAM42	Row34	RAM42	Row18	RAM18	Row26	RAM26	Row18	RAM26	Row26	RAM42	
COM30	Row33	RAM33	Row41	RAM41	Row33	RAM41	Row17	RAM17	Row25	RAM25	Row17	RAM25	Row25	RAM41	
COM31	Row32	RAM32	Row40	RAM40	Row32	RAM40	Row16	RAM16	Row24	RAM24	Row16	RAM24	Row24	RAM40	
COM32	Row31	RAM31	Row39	RAM39	Row31	RAM39	Row15	RAM15	Row23	RAM23	Row15	RAM23	Row23	RAM39	
COM33	Row30	RAM30	Row38	RAM38	Row30	RAM38	Row14	RAM14	Row22	RAM22	Row14	RAM22	Row22	RAM38	
COM34	Row29	RAM29	Row37	RAM37	Row29	RAM37	Row13	RAM13	Row21	RAM21	Row13	RAM21	Row21	RAM37	
COM35	Row28	RAM28	Row36	RAM36	Row28	RAM36	Row12	RAM12	Row20	RAM20	Row12	RAM20	Row20	RAM36	
COM36	Row27	RAM27	Row35	RAM35	Row27	RAM35	Row11	RAM11	Row19	RAM19	Row11	RAM19	Row19	RAM35	
COM37	Row26	RAM26	Row34	RAM34	Row26	RAM34	Row10	RAM10	Row18	RAM18	Row10	RAM18	Row18	RAM34	
COM38	Row25	RAM25	Row33	RAM33	Row25	RAM33	Row9	RAM9	Row17	RAM17	Row9	RAM17	Row17	RAM33	
COM39	Row24	RAM24	Row32	RAM32	Row24	RAM32	Row8	RAM8	Row16	RAM16	Row8	RAM16	Row16	RAM32	
COM40	Row23	RAM23	Row31	RAM31	Row23	RAM31	Row7	RAM7	Row15	RAM15	Row7	RAM15	Row15	RAM31	
COM41	Row22	RAM22	Row30	RAM30	Row22	RAM30	Row6	RAM6	Row14	RAM14	Row6	RAM14	Row14	RAM30	
COM42	Row21	RAM21	Row29	RAM29	Row21	RAM29	Row5	RAM5	Row13	RAM13	Row5	RAM13	Row13	RAM29	
COM43	Row20	RAM20	Row28	RAM28	Row20	RAM28	Row4	RAM4	Row12	RAM12	Row4	RAM12	Row12	RAM28	
COM44	Row19	RAM19	Row27	RAM27	Row19	RAM27	Row3	RAM3	Row11	RAM11	Row3	RAM11	Row11	RAM27	
COM45	Row18	RAM18	Row26	RAM26	Row18	RAM26	Row2	RAM2	Row10	RAM10	Row2	RAM10	Row10	RAM26	
COM46	Row17	RAM17	Row25	RAM25	Row17	RAM25	Row1	RAM1	Row9	RAM9	Row1	RAM9	Row9	RAM25	
COM47	Row16	RAM16	Row24	RAM24	Row16	RAM24	Row0	RAM0	Row8	RAM8	Row0	RAM8	Row8	RAM24	
COM48	Row15	RAM15	Row23	RAM23	Row15	RAM23	-	-	Row7	RAM7	-	-	Row7	RAM23	
COM49	Row14	RAM14	Row22	RAM22	Row14	RAM22	-	-	Row6	RAM6	-	-	Row6	RAM22	
COM50	Row13	RAM13	Row21	RAM21	Row13	RAM21	-	-	Row5	RAM5	-	-	Row5	RAM21	
COM51	Row12	RAM12	Row20	RAM20	Row12	RAM20	-	-	Row4	RAM4	-	-	Row4	RAM20	
COM52	Row11	RAM11	Row19	RAM19	Row11	RAM19	-	-	Row3	RAM3	-	-	Row3	RAM19	
COM53	Row10	RAM10	Row18	RAM18	Row10	RAM18	-	-	Row2	RAM2	-	-	Row2	RAM18	
COM54	Row9	RAM9	Row17	RAM17	Row9	RAM17	-	-	Row1	RAM1	-	-	Row1	RAM17	
COM55	Row8	RAM8	Row16	RAM16	Row8	RAM16	-	-	Row0	RAM0	-	-	Row0	RAM16	
COM56	Row7	RAM7	Row15	RAM15	Row7	RAM15	-	-	-	-	-	-	-	-	
COM57	Row6	RAM6	Row14	RAM14	Row6	RAM14	-	-	-	-	-	-	-	-	
COM58	Row5	RAM5	Row13	RAM13	Row5	RAM13	-	-	-	-	-	-	-	-	
COM59	Row4	RAM4	Row12	RAM12	Row4	RAM12	-	-	-	-	-	-	-	-	
COM60	Row3	RAM3	Row11	RAM11	Row3	RAM11	-	-	-	-	-	-	-	-	
COM61	Row2	RAM2	Row10	RAM10	Row2	RAM10	-	-	-	-	-	-	-	-	
COM62	Row1	RAM1	Row9	RAM9	Row1	RAM9	-	-	-	-	-	-	-	-	
COM63	Row0	RAM0	Row8	RAM8	Row0	RAM8	-	-	-	-	-	-	-	-	
Display examples	(a)		(b)		(c)		(d)		(e)		(f)		(g)		



(a) (b) (c) (d)



(e) (f) (g) (RAM)

10.1.16 Set Display Clock Divide Ratio/ Oscillator Frequency (D5h)

This command consists of two functions:

- Display Clock Divide Ratio (D)(A[3:0])
  Set the divide ratio to generate DCLK (Display Clock) from CLK. The divide ratio is from 1 to 16, with reset value = 1. Please refer to section 8.3 for the details relationship of DCLK and CLK.

- Oscillator Frequency (A[7:4])
  Program the oscillator frequency Fosc that is the source of CLK if CLS pin is pulled high. The 4-bit value results in 16 different frequency settings available as shown below. The default setting is 1000b.

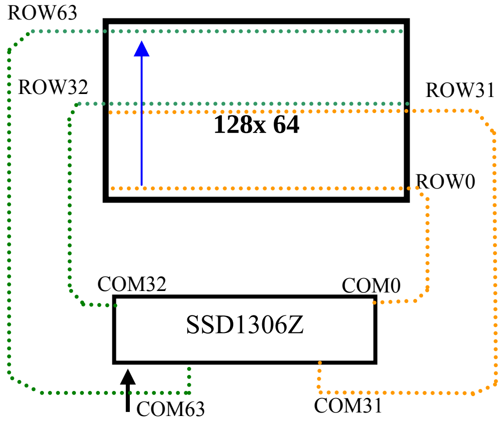
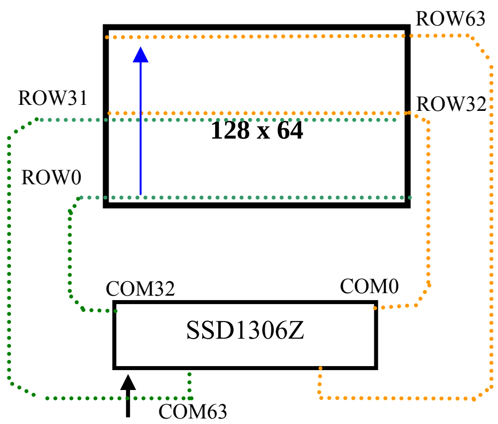
10.1.17 Set Pre-charge Period (D9h)

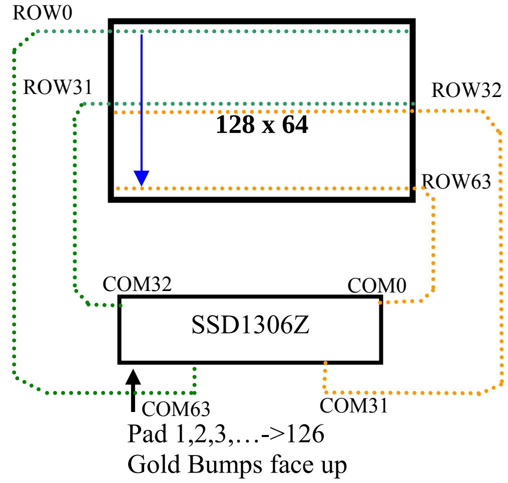
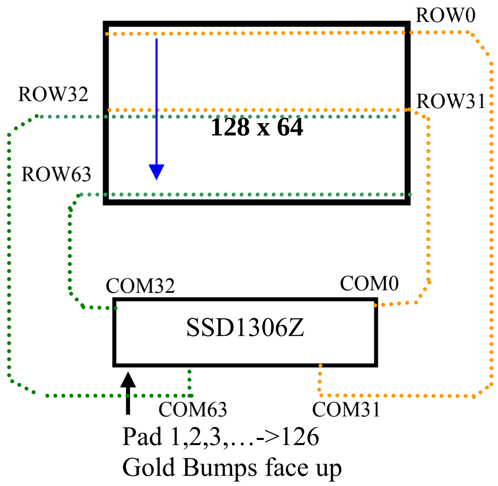
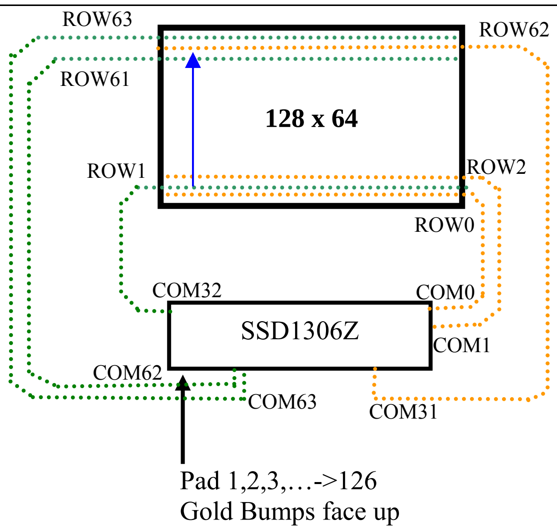
This command is used to set the duration of the pre-charge period. The interval is counted in number of DCLK, where RESET equals 2 DCLKs.

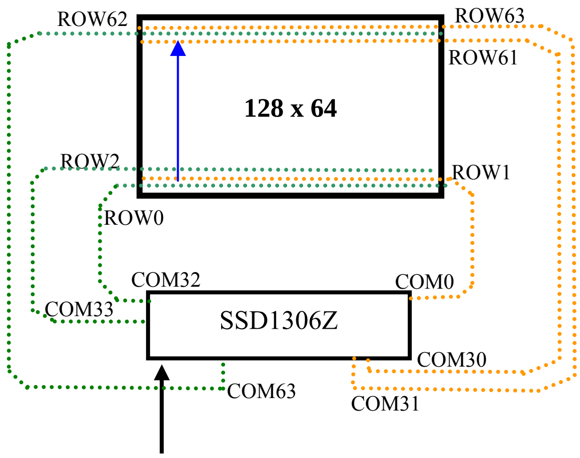
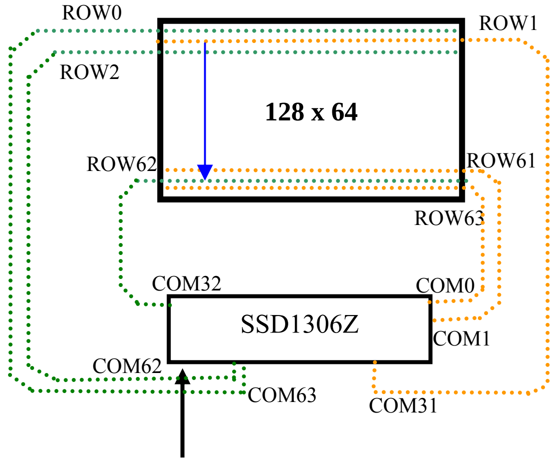
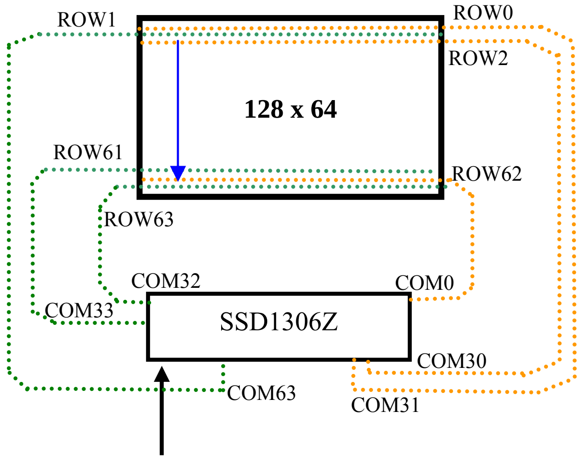
10.1.18 Set COM Pins Hardware Configuration (DAh)

This command sets the COM signals pin configuration to match the OLED panel hardware layout. The table below shows the COM pin configuration under different conditions (for MUX ratio =64):

Table 10-3 : COM Pins Hardware Configuration

Conditions	COM pins Configurations
1 Sequential COM pin configuration (DAh A[4] =0) COM output Scan direction: from COM0 to COM63 (C0h) Disable COM Left/Right remap (DAh A[5] =0)	 Pad 1,2,3,…->126 Gold Bumps face up
2 Sequential COM pin configuration (DAh A[4] =0) COM output Scan direction: from COM0 to COM63 (C0h) Enable COM Left/Right remap (DAh A[5] =1)	 Pad 1,2,3,…->126 Gold Bumps face up

Conditions	COM pins Configurations
3 Sequential COM pin configuration (DAh A[4] =0) COM output Scan direction: from COM63 to COM0 (C8h) Disable COM Left/Right remap (DAh A[5] =0)	
4 Sequential COM pin configuration (DAh A[4] =0) COM output Scan direction: from COM63 to COM0 (C8h) Enable COM Left/Right remap (DAh A[5] =1)	
5 Alternative COM pin configuration (DAh A[4] =1) COM output Scan direction: from COM0 to COM63 (C0h) Disable COM Left/Right remap (DAh A[5] =0)	

Conditions	COM pins Configurations
6 Alternative COM pin configuration (DAh A[4] =1) COM output Scan direction: from COM0 to COM63 (C0h) Enable COM Left/Right remap (DAh A[5] =1)	
7 Alternative COM pin configuration (DAh A[4] =1) COM output Scan direction: from COM63 to COM0(C8h) Disable COM Left/Right remap (DAh A[5] =0)	
8 Alternative COM pin configuration (DAh A[4] =1) COM output Scan direction: from COM63 to COM0(C8h) Enable COM Left/Right remap (DAh A[5] =1)	

10.1.19 Set V_{COMH} Deselect Level (DBh)

This command adjusts the V_{COMH} regulator output.

10.1.20 NOP (E3h)

No Operation Command

10.1.21 Status register Read

This command is issued by setting D/C# ON LOW during a data read (See Figure 13-1 to Figure 13-2 for parallel interface waveform). It allows the MCU to monitor the internal status of the chip. No status read is provided for serial mode.

10.2 Graphic Acceleration Command

10.2.1 Horizontal Scroll Setup (26h/27h)

This command consists of consecutive bytes to set up the horizontal scroll parameters and determines the scrolling start page, end page and scrolling speed.

Before issuing this command the horizontal scroll must be deactivated (2Eh). Otherwise, RAM content may be corrupted.

The SSD1306 horizontal scroll is designed for 128 columns scrolling. The following two figures (Figure 10-7, Figure 10-8, Figure 10-9) show the examples of using the horizontal scroll:

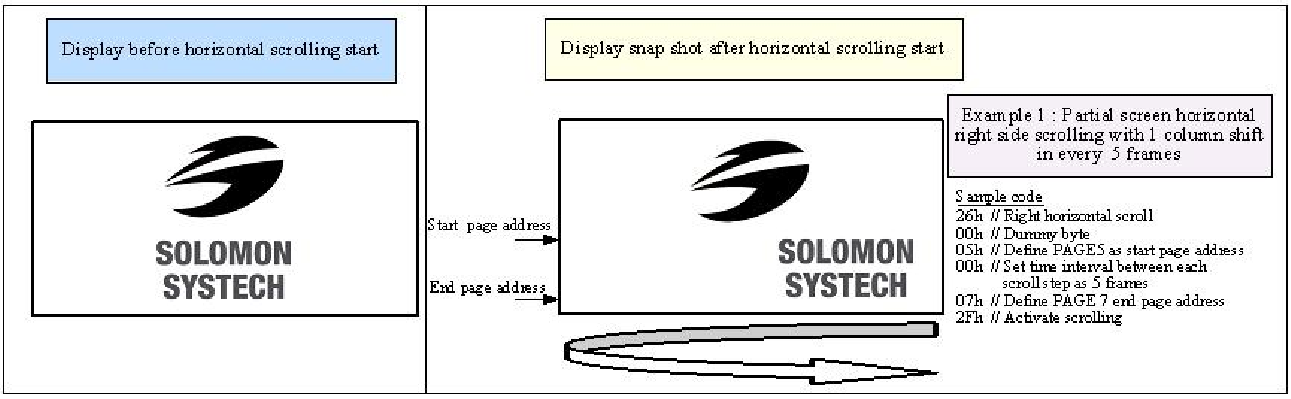
Figure 10-7 : Horizontal scroll example: Scroll RIGHT by 1 column

Original Setting	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	:	:	:	SEG122	SEG123	SEG124	SEG125	SEG126	SEG127
After one scroll step	SEG127	SEG0	SEG1	SEG2	SEG3	SEG4	:	:	:	SEG121	SEG122	SEG123	SEG124	SEG125	SEG126

Figure 10-8 : Horizontal scroll example: Scroll LEFT by 1 column

Original Setting	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	:	:	:	SEG122	SEG123	SEG124	SEG125	SEG126	SEG127
After one scroll step	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	:	:	:	SEG123	SEG124	SEG125	SEG126	SEG127	SEG0

Figure 10-9 : Horizontal scrolling setup example



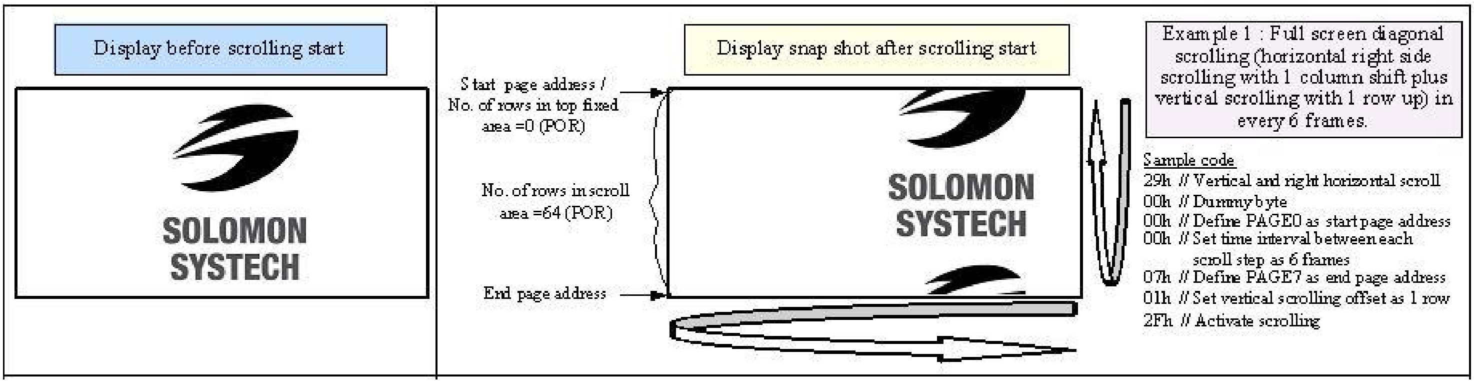
10.2.2 Continuous Vertical and Horizontal Scroll Setup (29h/2Ah)

This command consists of 6 consecutive bytes to set up the continuous vertical scroll parameters and determines the scrolling start page, end page, scrolling speed and vertical scrolling offset.

The bytes B[2:0], C[2:0] and D[2:0] of command 29h/2Ah are for the setting of the continuous horizontal scrolling. The byte E[5:0] is for the setting of the continuous vertical scrolling offset. All these bytes together are for the setting of continuous diagonal (horizontal + vertical) scrolling. If the vertical scrolling offset byte E[5:0] is set to zero, then only horizontal scrolling is performed (like command 26/27h).

Before issuing this command the scroll must be deactivated (2Eh). Otherwise, RAM content may be corrupted. The following figure (Figure 10-10) show the example of using the continuous vertical and horizontal scroll:

Figure 10-10 : Continuous Vertical and Horizontal scrolling setup example



10.2.3 Deactivate Scroll (2Eh)

This command stops the motion of scrolling. After sending 2Eh command to deactivate the scrolling action, the ram data needs to be rewritten.

10.2.4 Activate Scroll (2Fh)

This command starts the motion of scrolling and should only be issued after the scroll setup parameters have been defined by the scrolling setup commands :26h/27h/29h/2Ah . The setting in the last scrolling setup command overwrites the setting in the previous scrolling setup commands.

The following actions are prohibited after the scrolling is activated

1. RAM access (Data write or read)
2. Changing the horizontal scroll setup parameters

10.2.5 Set Vertical Scroll Area(A3h)

This command consists of 3 consecutive bytes to set up the vertical scroll area. For the continuous vertical scroll function (command 29/2Ah), the number of rows that in vertical scrolling can be set smaller or equal to the MUX ratio.

11 MAXIMUM RATINGS

Table 11-1 : Maximum Ratings (Voltage Referenced to VSS)

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to +4	V
V_{CC}		0 to 16	V
V_{SEG}	SEG output voltage	0 to V_{CC}	V
V_{COM}	COM output voltage	0 to 0.9*V_{CC}	V
V_{in}	Input voltage	V_{SS}-0.3 to V_{DD}+0.3	V
T_A	Operating Temperature	-40 to +85	ºC
T_{stg}	Storage Temperature Range	-65 to +150	ºC

Maximum ratings are those values beyond which damages to the device may occur. Functional operation should be restricted to the limits in the Electrical Characteristics tables or Pin Description section

This device may be light sensitive. Caution should be taken to avoid exposure of this device to any light source during normal operation. This device is not radiation protected.

12 DC CHARACTERISTICS

Condition (Unless otherwise specified):

Voltage referenced to V_{SS}
V_{DD} = 1.65 to 3.3V
T_A = 25°C

Table 12-1 : DC Characteristics

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
V_{CC}	Operating Voltage	-	7	-	15	V
V_{DD}	Logic Supply Voltage	-	1.65	-	3.3	V
V_{OH}	High Logic Output Level	I_{OUT} = 100uA, 3.3MHz	0.9 x V_{DD}	-	-	V
V_{OL}	Low Logic Output Level	I_{OUT} = 100uA, 3.3MHz	-	-	0.1 x V_{DD}	V
V_{IH}	High Logic Input Level	-	0.8 x V_{DD}	-	-	V
V_{IL}	Low Logic Input Level	-	-	-	0.2 x V_{DD}	V
$I_{CC, SLEEP}$	$I_{CC,}$ Sleep mode Current	V_{DD} = 1.65V~3.3V, V_{CC} = 7V~15V Display OFF, No panel attached	-	-	10	uA
$I_{DD, SLEEP}$	$I_{DD,}$ Sleep mode Current	V_{DD} = 1.65V~3.3V, V_{CC} = 7V~15V Display OFF, No panel attached	-	-	10	uA
I_{CC}	V_{CC} Supply Current V_{DD} = 2.8V, V_{CC} = 12V, I_{REF} = 12.5uA No loading, Display ON, All ON	Contrast = FFh	-	430	780	uA
I_{DD}	V_{DD} Supply Current V_{DD} = 2.8V, V_{CC} = 12V, I_{REF} = 12.5uA No loading, Display ON, All ON		-	50	150	uA
I_{SEG}	Segment Output Current V_{DD}=2.8V, V_{CC}=12V, I_{REF}=12.5uA, Display ON.	Contrast=FFh	-	100	-	uA
		Contrast=AFh	-	69	-	
		Contrast=3Fh	-	25	-	
Dev	Segment output current uniformity	Dev = $(I_{SEG} – I_{MID})/I_{MID}$ $I_{MID} = (I_{MAX} + I_{MIN})/2$ I_{SEG}[0:131] = Segment current at contrast = FFh	-3	-	+3	%
Adj. Dev	Adjacent pin output current uniformity (contrast = FF)	Adj Dev = (I[n]-I[n+1]) / (I[n]+I[n+1])	-2	-	+2	%

13 AC CHARACTERISTICS

Conditions:

Voltage referenced to V_{SS}
V_{DD}=1.65 to3.3V
T_A = 25°C

Table 13-1 : AC Characteristics

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
F_{OSC} (1)	Oscillation Frequency of Display Timing Generator	V_{DD} = 2.8V	333	370	407	kHz
F_{FRM}	Frame Frequency for 64 MUX Mode	128x64 Graphic Display Mode, Display ON, Internal Oscillator Enabled	-	F_{OSC} x 1/(DxKx64) (2)	-	Hz
RES#	Reset low pulse width		3	-	-	us

Note

(1) F_{OSC} stands for the frequency value of the internal oscillator and the value is measured when command D5h A[7:4] is in default value.

(2) D: divide ratio (default value = 1)
K: number of display clocks (default value = 54)
Please refer to Table 9-1 (Set Display Clock Divide Ratio/Oscillator Frequency, D5h) for detailed description

Table 13-2 : 6800-Series MCU Parallel Interface Timing Characteristics

(V_{DD} - V_{SS} = 1.65V to 3.3V, T_A = 25°C)

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	300	-	-	ns
t_{AS}	Address Setup Time	0	-	-	ns
t_{AH}	Address Hold Time	0	-	-	ns
t_{DSW}	Write Data Setup Time	40	-	-	ns
t_{DHW}	Write Data Hold Time	7	-	-	ns
t_{DHR}	Read Data Hold Time	20	-	-	ns
t_{OH}	Output Disable Time	-	-	70	ns
t_{ACC}	Access Time	-	-	140	ns
PW_{CSL}	Chip Select Low Pulse Width (read) Chip Select Low Pulse Width (write)	120 60	-	-	ns
PW_{CSH}	Chip Select High Pulse Width (read) Chip Select High Pulse Width (write)	60 60	-	-	ns
t_R	Rise Time	-	-	40	ns
t_F	Fall Time	-	-	40	ns

Figure 13-1 : 6800-series MCU parallel interface characteristics

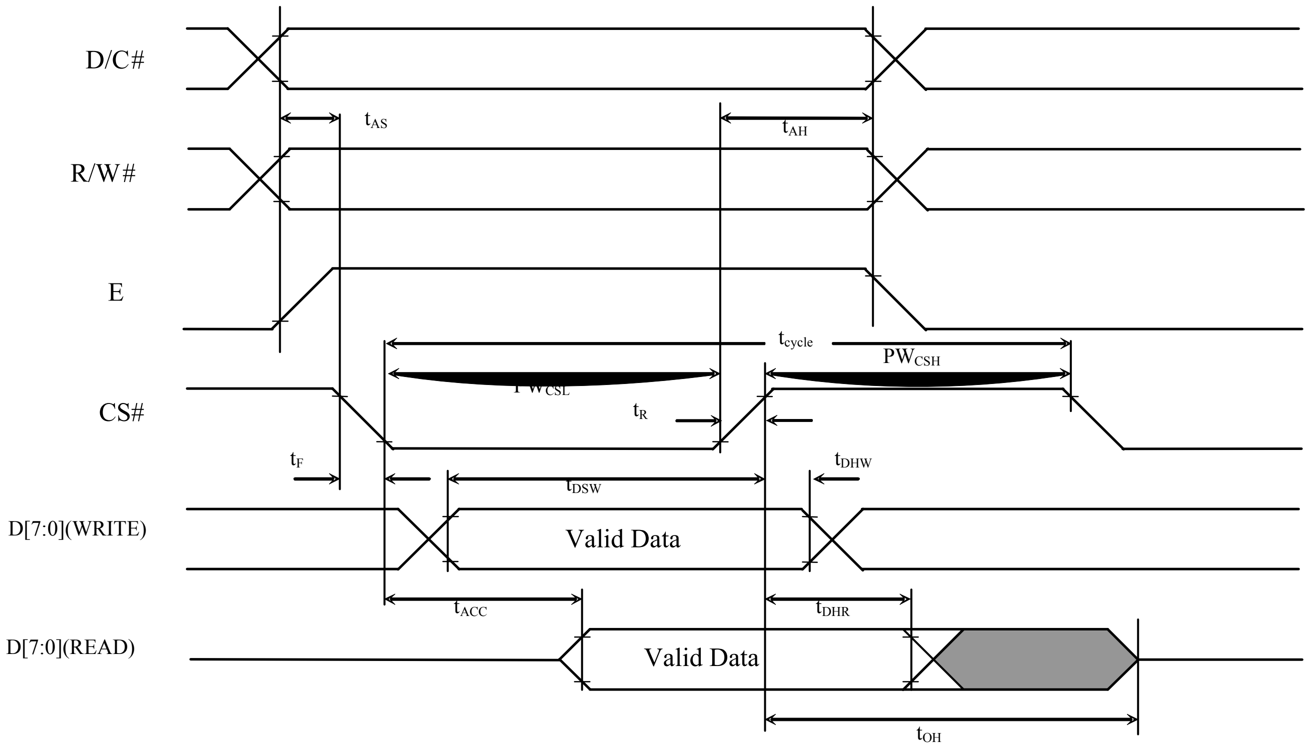


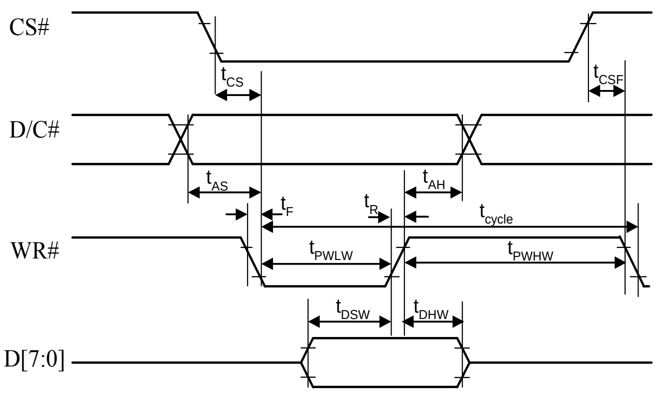
Table 13-3 : 8080-Series MCU Parallel Interface Timing Characteristics

(V_{DD} - V_{SS} = 1.65V to 3.3V, T_A = 25°C)

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	300	-	-	ns
t_{AS}	Address Setup Time	10	-	-	ns
t_{AH}	Address Hold Time	0	-	-	ns
t_{DSW}	Write Data Setup Time	40	-	-	ns
t_{DHW}	Write Data Hold Time	7	-	-	ns
t_{DHR}	Read Data Hold Time	20	-	-	ns
t_{OH}	Output Disable Time	-	-	70	ns
t_{ACC}	Access Time	-	-	140	ns
t_{PWLR}	Read Low Time	120	-	-	ns
t_{PWLW}	Write Low Time	60	-	-	ns
t_{PWHR}	Read High Time	60	-	-	ns
t_{PWHW}	Write High Time	60	-	-	ns
t_R	Rise Time	-	-	40	ns
t_F	Fall Time	-	-	40	ns
t_{CS}	Chip select setup time	0	-	-	ns
t_{CSH}	Chip select hold time to read signal	0	-	-	ns
t_{CSF}	Chip select hold time	20	-	-	ns

Figure 13-2 : 8080-series parallel interface characteristics

Write Cycle



Read cycle

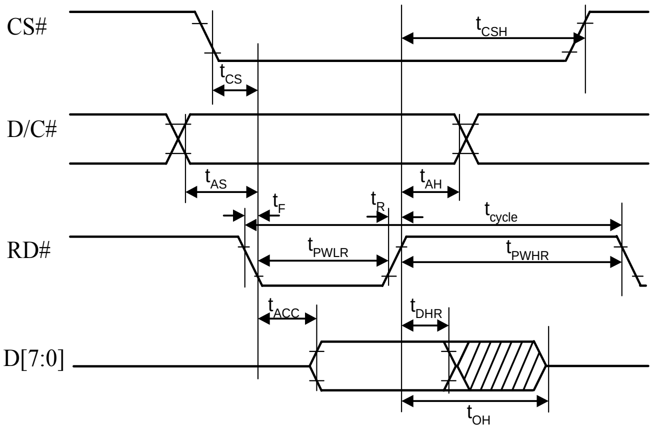
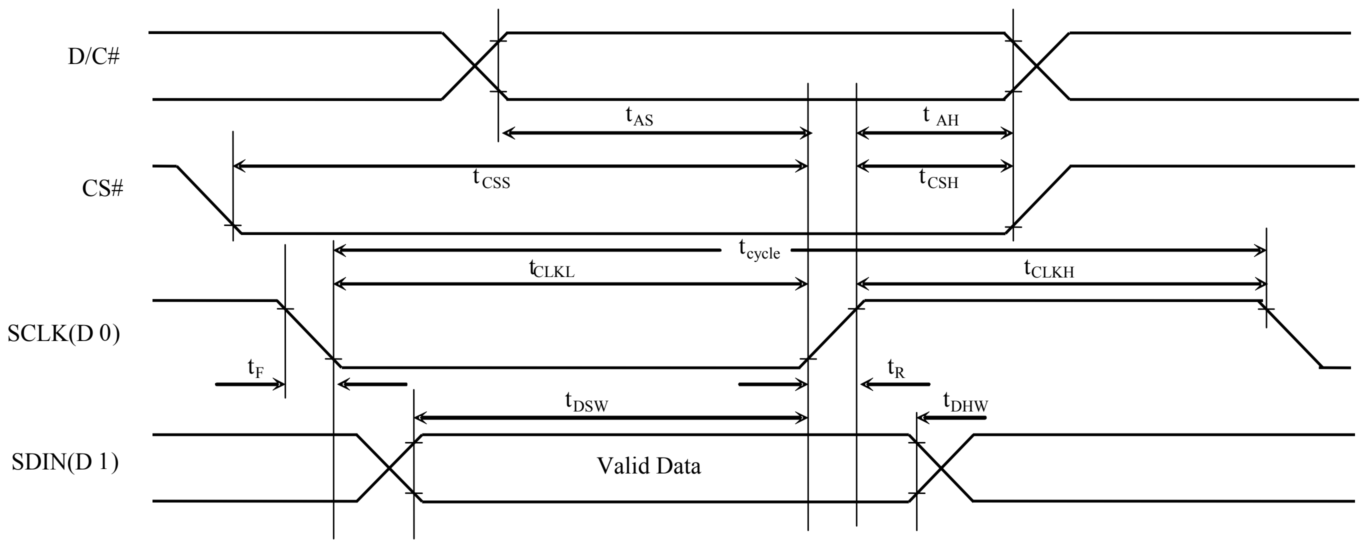


Table 13-4 : 4-wire Serial Interface Timing Characteristics

(V_{DD} - V_{SS} = 1.65V to 3.3V, T_A = 25°C)

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	100	-	-	ns
t_{AS}	Address Setup Time	15	-	-	ns
t_{AH}	Address Hold Time	15	-	-	ns
t_{CSS}	Chip Select Setup Time	20	-	-	ns
t_{CSH}	Chip Select Hold Time	10	-	-	ns
t_{DSW}	Write Data Setup Time	15	-	-	ns
t_{DHW}	Write Data Hold Time	15	-	-	ns
t_{CLKL}	Clock Low Time	20	-	-	ns
t_{CLKH}	Clock High Time	20	-	-	ns
t_R	Rise Time	-	-	40	ns
t_F	Fall Time	-	-	40	ns

Figure 13-3 : 4-wire Serial interface characteristics



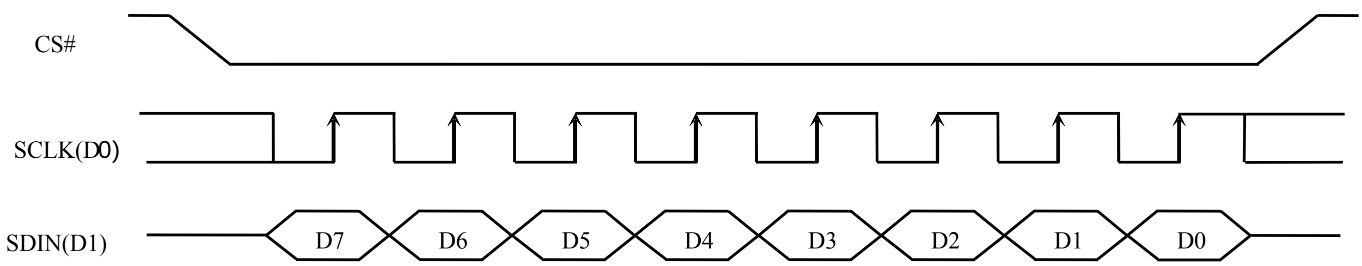
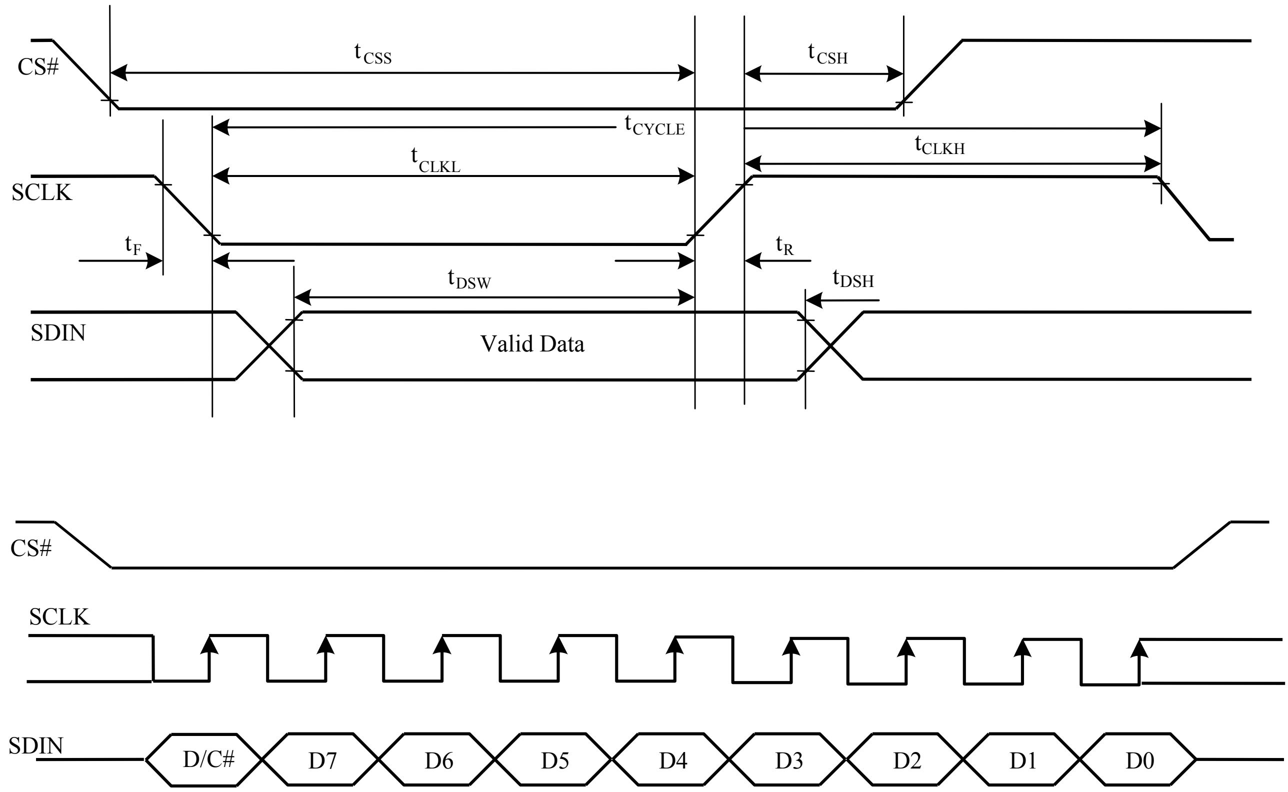


Table 13-5 : 3-wire Serial Interface Timing Characteristics

(V_{DD} - V_{SS} = 1.65V to 3.3V, T_A = 25°C)

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	100	-	-	ns
t_{CSS}	Chip Select Setup Time	20	-	-	ns
t_{CSH}	Chip Select Hold Time	10	-	-	ns
t_{DSW}	Write Data Setup Time	15	-	-	ns
t_{DHW}	Write Data Hold Time	15	-	-	ns
t_{CLKL}	Clock Low Time	20	-	-	ns
t_{CLKH}	Clock High Time	20	-	-	ns
t_R	Rise Time	-	-	40	ns
t_F	Fall Time	-	-	40	ns

Figure 13-4 : 3-wire Serial interface characteristics



Conditions:

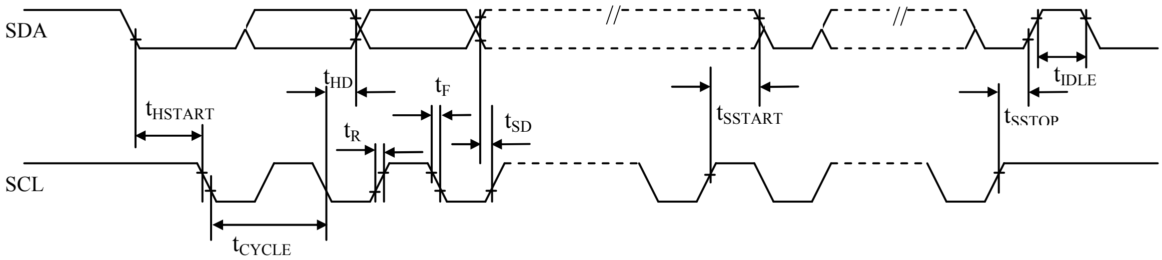
V_{DD} - V_{SS} = V_{DD} - V_{SS} = 1.65V to 3.3V

T_A = 25°C

Table 13-6 :I²C Interface Timing Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	2.5	-	-	us
t_{HSTART}	Start condition Hold Time	0.6	-	-	us
t_{HD}	Data Hold Time (for "SDA_{OUT}" pin)	0	-	-	ns
	Data Hold Time (for "SDA_{IN}" pin)	300	-	-	ns
t_{SD}	Data Setup Time	100	-	-	ns
t_{SSTART}	Start condition Setup Time (Only relevant for a repeated Start condition)	0.6	-	-	us
t_{SSTOP}	Stop condition Setup Time	0.6	-	-	us
t_R	Rise Time for data and clock pin	-	-	300	ns
t_F	Fall Time for data and clock pin	-	-	300	ns
t_{IDLE}	Idle Time before a new transmission can start	1.3	-	-	us

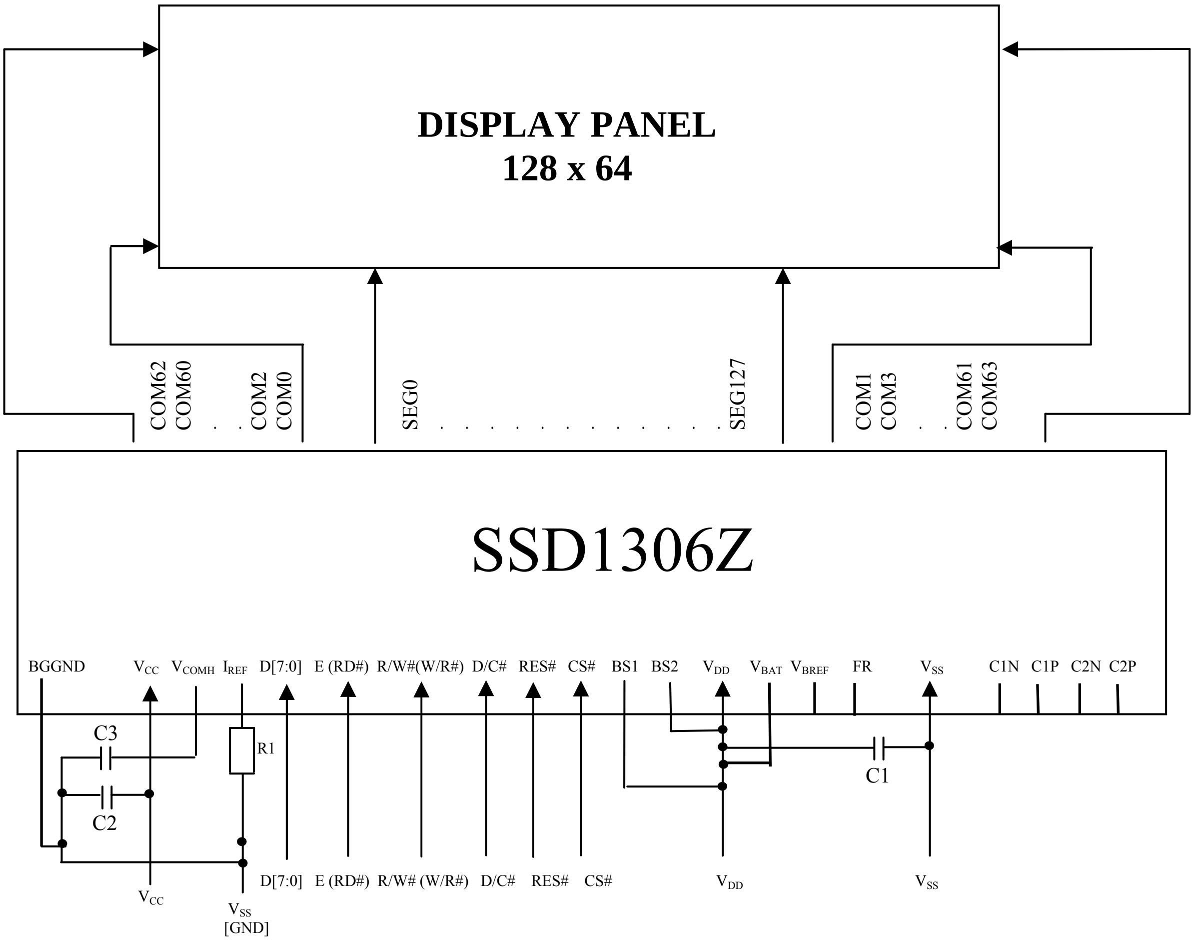
Figure 13-5 : I²C interface Timing characteristics



14 Application Example

Figure 14-1 : Application Example of SSD1306Z

The configuration for 8080-parallel interface mode is shown in the following diagram:
(V_{DD}=2.8V, V_{CC} =12V, I_{REF}=12.5uA)



Pin connected to MCU interface: D[7:0], E, R/W#, D/C#, CS#, RES#
Pin internally connected to V_{SS}: BS0, CL
Pin internally connected to V_{DD}: CLS
C2P, C2N, C1P, C1N, V_{BREF}, FB should be left open.

C1: 1.0uF [(1)]
C2: 2.2uF [(1)]
C3: 2.2uF [(1)]

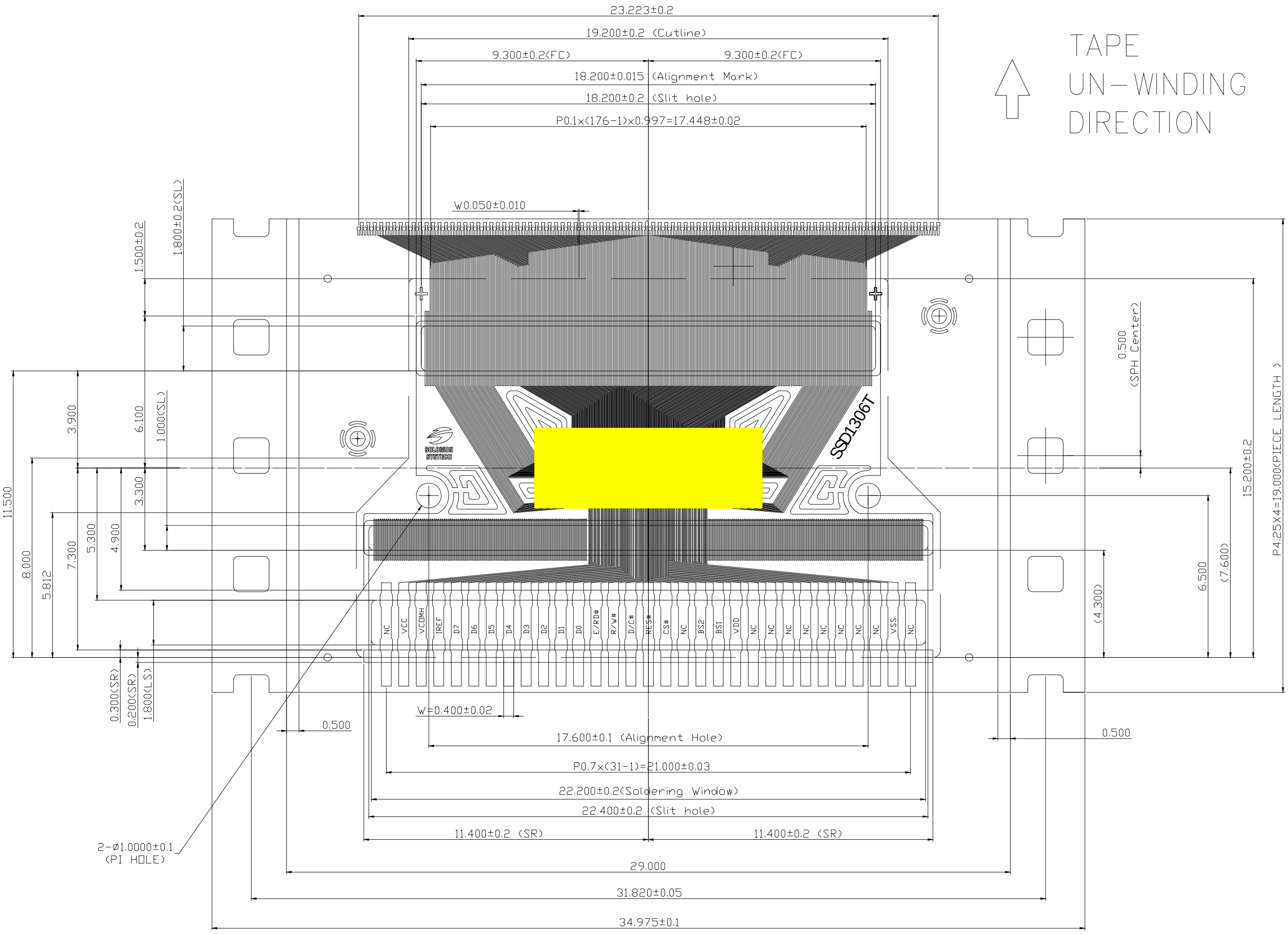
Voltage at I_{REF} = V_{CC} – 2.5V. For V_{CC} = 12V, I_{REF} = 12.5uA:
R1 = (Voltage at I_{REF} - V_{SS}) / I_{REF}
= (12-2.5) / 12.5u
=760KΩ

Note
[(1)] The capacitor value is recommended value. Select appropriate value against module application.

15 PACKAGE INFORMATION

15.1 SSD1306TR1 Detail Dimension

Figure 15-1 SSD1306TR1 Detail Dimension



Specification:

1. GENERAL TOLERANCE: ±0.05 mm

2. MATERIAL

   PI: 75 ± 8 um

   CU: 15 ± 3 um

   ADHESIVE: 12 ± 3um

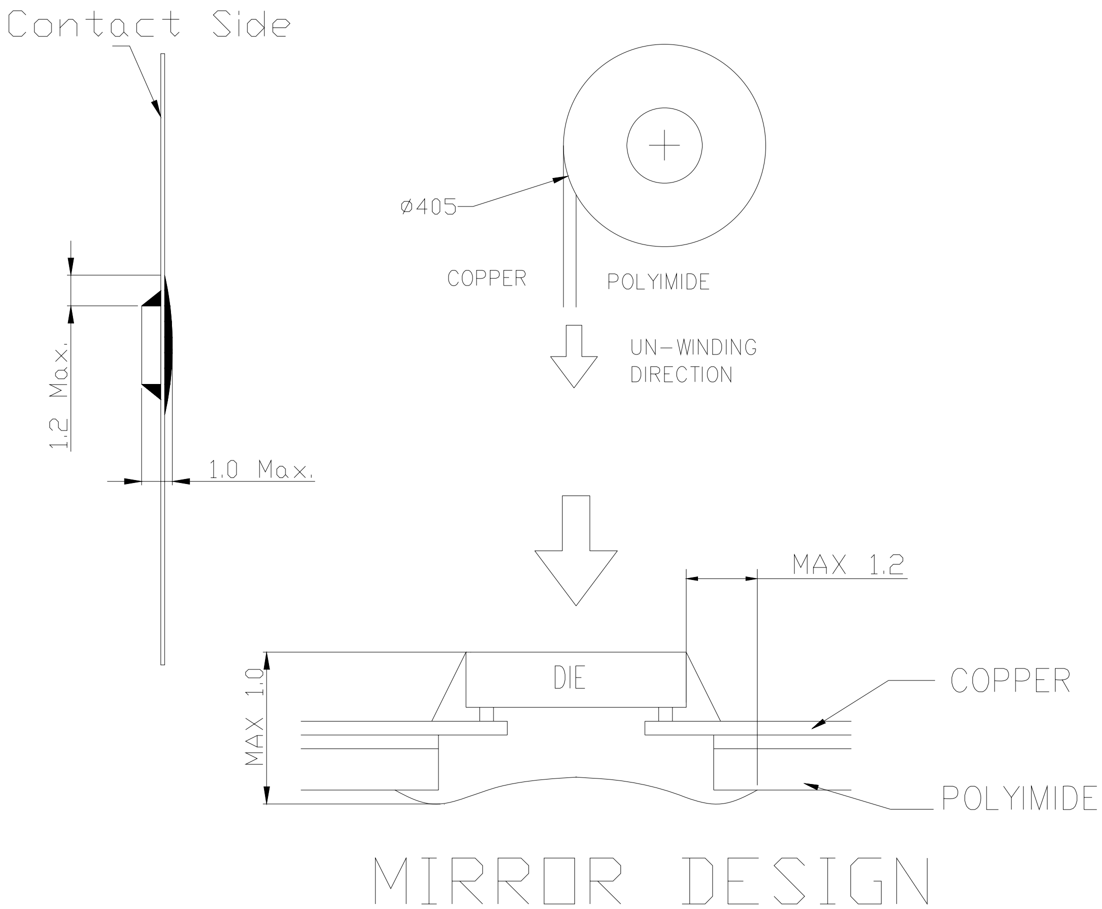
   SR: 26 ± 14 um

   TOLERANCE ± 0.200 mm

   FLEX COATING: Min10 um

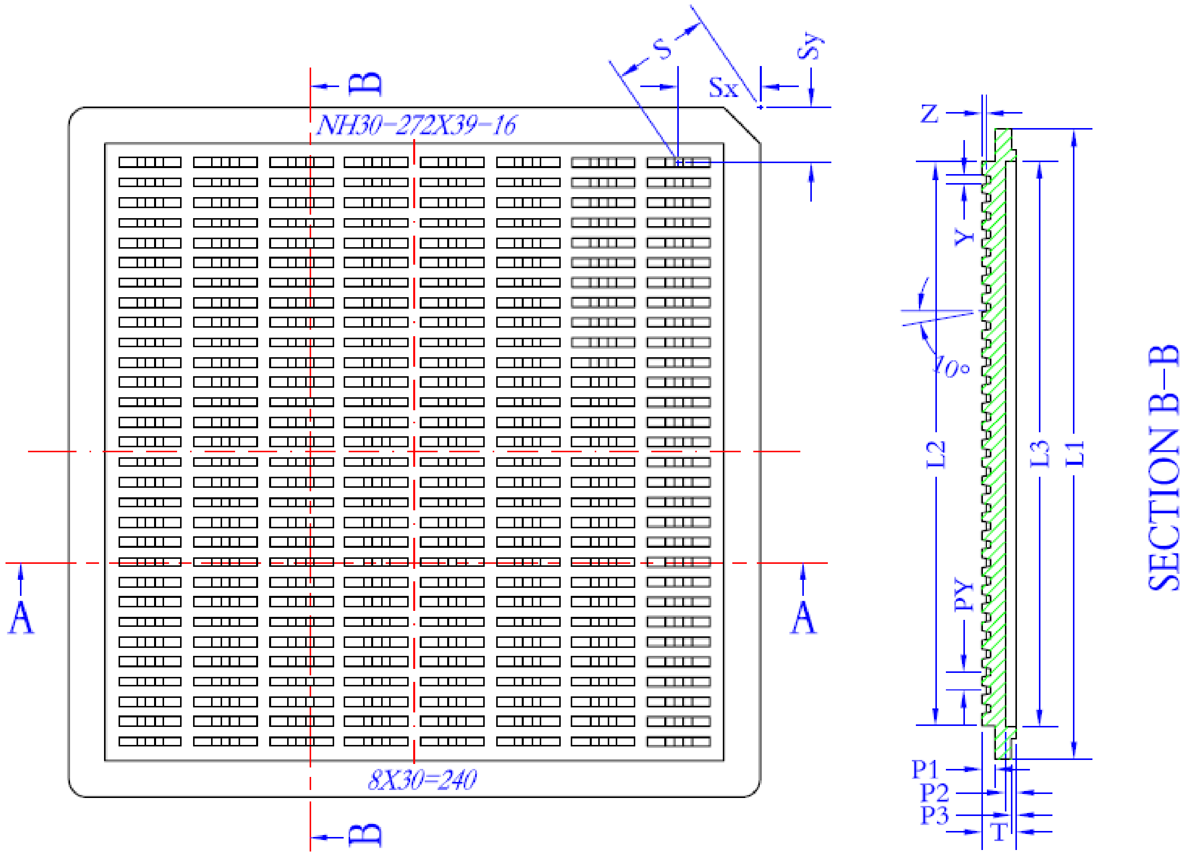
3. Plating : Sn 0.20 ±0.05 um

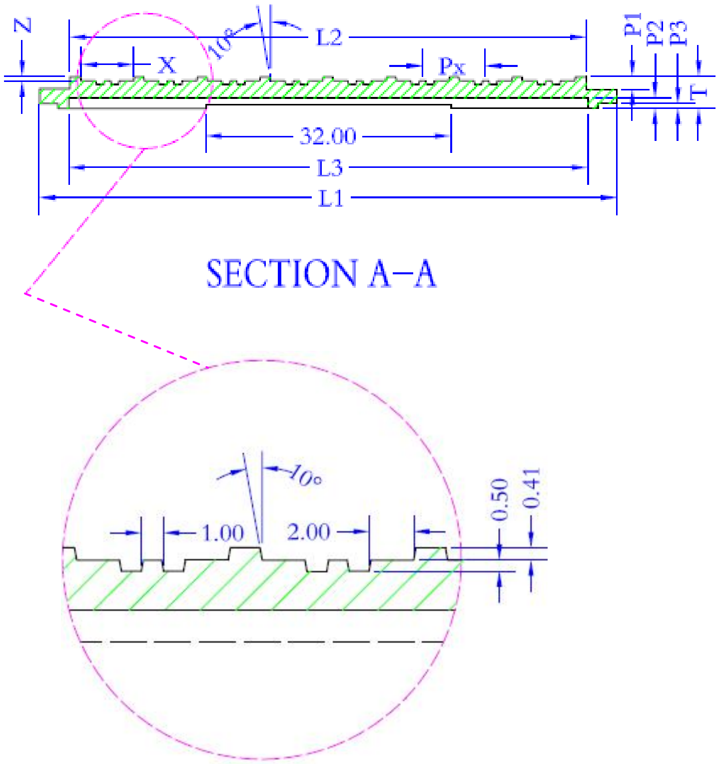
4. TAPESITE: 4 SPH,19 mm


Contact Side
Ø405
COPPER
POLYIMIDE
UN-WINDING
DIRECTION
1.2 Max.
1.0 Max.
MAX 1.2
DIE
COPPER
MAX 1.0
POLYIMIDE
MIRROR DESIGN

15.2 SSD1306Z Die Tray Information

Figure 15-2 : SSD1306Z die tray information





SPEC	
	MM
(L1)	76.00±0.10
(L2)	68.10±0.10
(L3)	68.20±0.10
(T)	4.20 ±0.05
(SX)	8.95
(SY)	6.10
(S)	10.83
(X)	6.91 ±0.05
(Y)	1.00 ±0.05
(Z)	0.41 ±0.05
(Px)	8.30±0.05
(Py)	2.20 ±0.05
(Nx)	8
(Ny)	30
(N)	240
(P1)	1.60
(P2)	1.40
(P3)	0.65

All Solomon Systech Products complied with six (6) hazardous substances limitation requirement per European Union (EU) "Restriction of Hazardous Substance (RoHS) Directive (2002/95/EC)" and China standard "电子信息产品污染控制标识要求 (SJ/T11364-2006)" with control Marking Symbol . Hazardous Substances test report is available upon requested.

http://www.solomon-systech.com

SSD1306

Application Note

128 x 64 Dot Matrix
OLED/PLED Segment/Common Driver with Controller

This document contains information on a new product. Specifications and information herein are subject to change without notice.

TABLE OF CONTENTS

TABLE OF FIGURES

1 Introduction

This application note of SSD1306 is written to explain the charge pump regulator function of SSD1306. SSD1306 is a single-chip CMOS OLED/PLED driver with controller for organic / polymer light emitting diode dot-matrix graphic display system. It consists of 128 segments and 64 commons. This IC is designed for Common Cathode type OLED panel.

For the detailed characteristics of the driver IC, please refer to SSD1306 datasheet.

2 Charge Pump Regulator

The internal regulator circuit in SSD1306 accompanying only 2 external capacitors can generate a 7.5V voltage supply, $V_{CC,}$ from a low voltage supply input, V_{BAT}. The V_{CC} is the voltage supply to the OLED driver block. This is a switching capacitor regulator circuit, designed for handheld applications. This regulator can be turned on/off by software command setting.

- Power supply
  - V_{DD} = 1.65V to 3.3V,< V_{BAT} for IC logic
  - V_{BAT} = 3.3V to 4.2V for charge pump regulator circuit

- Pins description for related pins of the charge pump regulator
  - V_{BAT} – Power supply for charge pump regulator circuit.

Status	V_{BAT}	V_{DD}	V_{CC}
Enable charge pump	Connect to external V_{BAT} source	Connect to external V_{DD} source	A capacitor should be connected between this pin and V_{SS}
Disable charge pump	Connect with V_{DD} pin	Connect to external V_{DD} source	Connect to external V_{CC} source

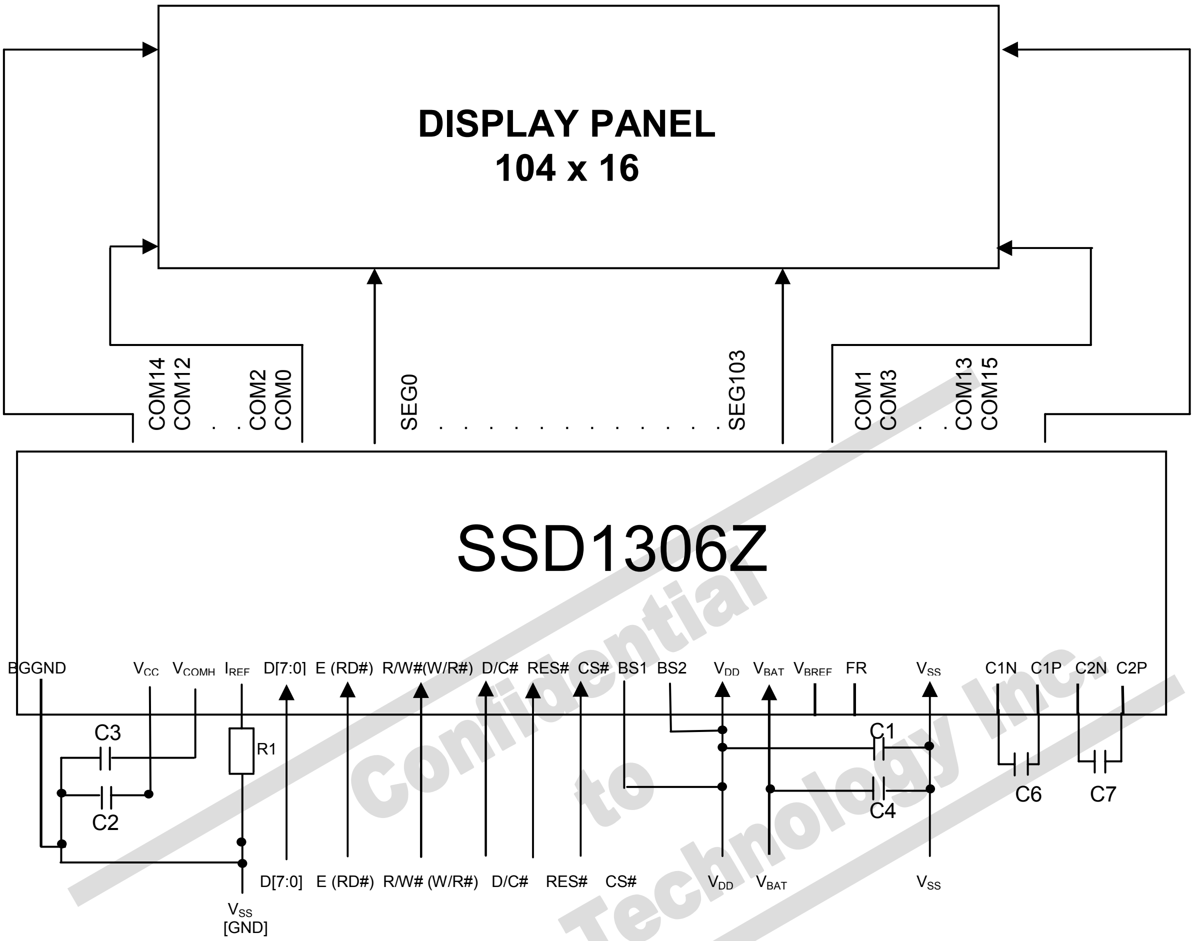
  - C1P/C1N – Pin for charge pump capacitor; Connect to each other with a capacitor
  - C2P/C2N – Pin for charge pump capacitor; Connect to each other with a capacitor

2.1 Command Table for Charge Bump Setting

1. Charge Pump Command Table											
D/C#	**Hex**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Command**	**Description**
0	8D	1	0	0	0	1	1	0	1	Charge Pump Setting	A[2] = 0b, Disable charge pump(RESET) A[2] = 1b, Enable charge pump during display on **Note** (1) The Charge Pump must be enabled by the following command: 8Dh ; Charge Pump Setting 14h ; Enable Charge Pump AFh; Display ON
0	A[7:0]	*	*	0	1	0	A_2	0	0		

Figure 1 : Application Example of SSD1306Z with charge bump

The configuration for 8080-parallel interface mode is shown in the following diagram:
(V_{DD}= 1.65V ~ 3.3V, < V_{BAT} , V_{BAT}=3.3V~4.2V, I_{REF}=12.5uA)



Pin connected to MCU interface: D[7:0], E, R/W#, D/C#, CS#, RES#
Pin internally connected to V_{SS}: BS0, CL
Pin internally connected to V_{DD}: CLS
VBREF, FR should be left open.

C1, C4, C6, C7: 1.0uF [(1)]
C2, C3: 2.2uF [(1)]

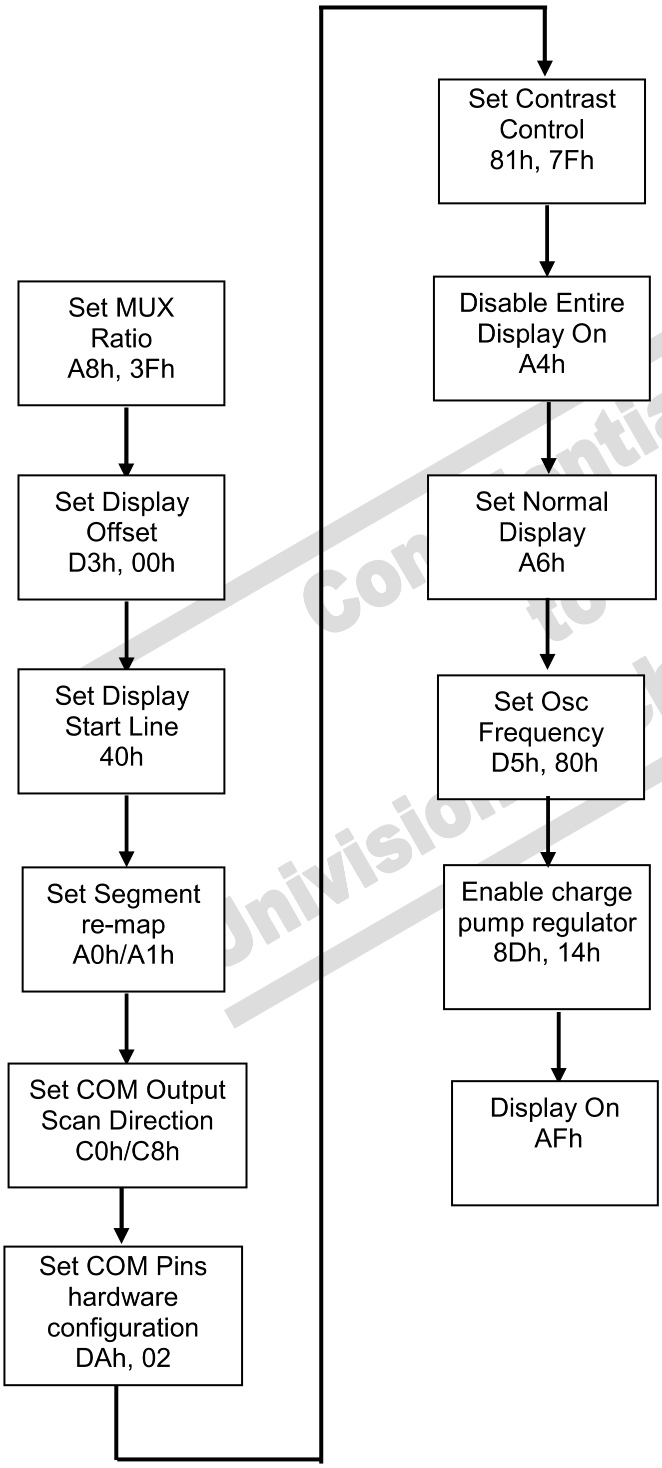
Voltage at I_{REF} = V_{CC} – 2.5V. For V_{CC} = 7.5V, I_{REF} = 12.5uA:
R1 = (Voltage at I_{REF} - V_{SS}) / I_{REF}
= (7.5-2.5) / 12.5u
=400KΩ

Note
[(1)] The capacitor value is recommended value. Select appropriate value against module application.

3 Software Configuration

SSD1306 has internal command registers that are used to configure the operations of the driver IC. After reset, the registers should be set with appropriate values in order to function well. The registers can be accessed by MPU interface in either 6800, 8080, SPI type with D/C# pin pull low or using I^2C interface. Below is an example of initialization flow of SSD1306. The values of registers depend on different condition and application.

Figure 2 : Software Initialization Flow Chart



All Solomon Systech Products complied with six (6) hazardous substances limitation requirement per European Union (EU) "Restriction of Hazardous Substance (RoHS) Directive (2002/95/EC)" and China standard "电子信息产品污染控制标识要求 (SJ/T11364-2006)" with control Marking Symbol . Hazardous Substances test report is available upon requested.

http://www.solomon-systech.com

DS3231

Extremely Accurate I2C-Integrated RTC/TCXO/Crystal

General Description

The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I2C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V_{CC} to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a µP reset.

Benefits and Features

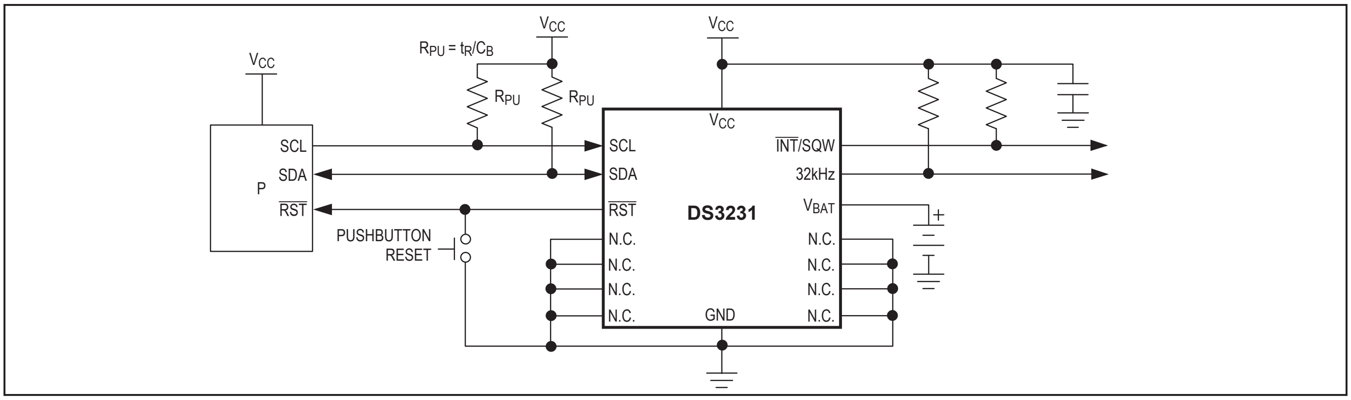
- Highly Accurate RTC Completely Manages All Timekeeping Functions
  - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
  - Accuracy ±2ppm from 0°C to +40°C
  - Accuracy ±3.5ppm from -40°C to +85°C
  - Digital Temp Sensor Output: ±3°C Accuracy
  - Register for Aging Trim
  - RST Output/Pushbutton Reset Debounce Input
  - Two Time-of-Day Alarms
  - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
  - Fast (400kHz) I2C Interface
- Battery-Backup Input for Continuous Timekeeping
  - Low Power Operation Extends Battery-Backup Run Time
  - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground ....-0.3V to +6.0V
Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1)73°C/W
Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1) ....23°C/W
Operating Temperature Range
DS3231S ....0°C to +70°C
DS3231SN... -40°C to +85°C

Junction Temperature...+125°C
Storage Temperature Range .... -40°C to +85°C
Lead Temperature (soldering, 10s) ....+260°C
Soldering Temperature (reflow, 2 times max) ....+260°C
(see the *Handling*, *PCB Layout*, *and Assembly* section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to **www.maximintegrated.com/thermal-tutorial**.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

($T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}		V_{CC} + 0.3	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		0.3 x V_{CC}	V

Electrical Characteristics

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	V_{CC} = 3.63V			200	µA
			V_{CC} = 5.5V			300	
Standby Supply Current	I_{CCS}	I²C bus inactive, 32kHz output on, SQW output off (Note 5)	V_{CC} = 3.63V			110	µA
			V_{CC} = 5.5V			170	
Temperature Conversion Current	$I_{CCSCONV}$	I²C bus inactive, 32kHz output on, SQW output off	V_{CC} = 3.63V			575	µA
			V_{CC} = 5.5V			650	
Power-Fail Voltage	V_{PF}			2.45	2.575	2.70	V
Logic 0 Output, 32kHz, $\overline{\text{INT}}$/SQW, SDA	V_{OL}	I_{OL} = 3mA				0.4	V
Logic 0 Output, $\overline{\text{RST}}$	V_{OL}	I_{OL} = 1mA				0.4	V
Output Leakage Current 32kHz, $\overline{\text{INT}}$/SQW, SDA	I_{LO}	Output high impedance		-1	0	+1	µA
Input Leakage SCL	I_{LI}			-1		+1	µA
$\overline{\text{RST}}$ Pin I/O Leakage	I_{OL}	$\overline{\text{RST}}$ high impedance (Note 6)		-200		+10	µA
V_{BAT} Leakage Current (V_{CC} Active)	I_{BATLKG}				25	100	nA

Electrical Characteristics (continued)

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Output Frequency	f_{OUT}	V_{CC} = 3.3V or V_{BAT} = 3.3V			32.768		kHz
Frequency Stability vs. Temperature (Commercial)	$\Delta f/f_{OUT}$	V_{CC} = 3.3V or V_{BAT} = 3.3V, aging offset = 00h	0°C to +40°C			±2	ppm
			>40°C to +70°C			±3.5	
Frequency Stability vs. Temperature (Industrial)	$\Delta f/f_{OUT}$	V_{CC} = 3.3V or V_{BAT} = 3.3V, aging offset = 00h	-40°C to <0°C			±3.5	ppm
			0°C to +40°C			±2	
			>40°C to +85°C			±3.5	
Frequency Stability vs. Voltage	Δf/V				1		ppm/V
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C		0.7		ppm
			+25°C		0.1		
			+70°C		0.4		
			+85°C		0.8		
Temperature Accuracy	Temp	V_{CC} = 3.3V or V_{BAT} = 3.3V		-3		+3	°C
Crystal Aging	$\Delta f/f_O$	After reflow, not production tested	First year		±1.0		ppm
			0–10 years		±5.0		

Electrical Characteristics

(**V_{CC} = 0V, V_{BAT} = 2.3V to 5.5V**, $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Battery Current	I_{BATA}	$\overline{EOSC}$ = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V_{BAT} = 3.63V			70	µA
			V_{BAT} = 5.5V			150	
Timekeeping Battery Current	I_{BATT}	$\overline{EOSC}$ = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V_{BAT} (Note 5)	V_{BAT} = 3.63V		0.84	3.0	µA
			V_{BAT} = 5.5V		1.0	3.5	
Temperature Conversion Current	I_{BATTC}	$\overline{EOSC}$ = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V_{BAT}	V_{BAT} = 3.63V			575	µA
			V_{BAT} = 5.5V			650	
Data-Retention Current	I_{BATTDR}	$\overline{EOSC}$ = 1, SCL = SDA = 0V, +25°C				100	nA

AC Electrical Characteristics

($V_{CC} = V_{CC(MIN)}$ to $V_{CC(MAX)}$ or $V_{BAT} = V_{BAT(MIN)}$ to $V_{BAT(MAX)}$, $V_{BAT} > V_{CC}$, $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Note 2)

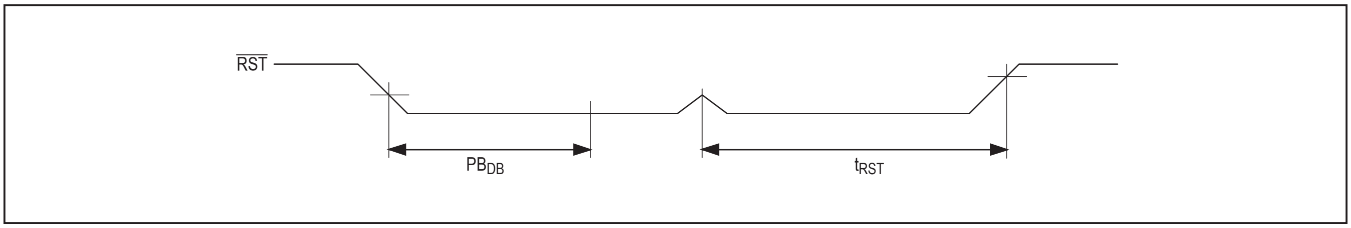
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}	Fast mode	100		400	kHz
		Standard mode	0		100	
Bus Free Time Between STOP and START Conditions	t_{BUF}	Fast mode	1.3			µs
		Standard mode	4.7			
Hold Time (Repeated) START Condition (Note 7)	$t_{HD:STA}$	Fast mode	0.6			µs
		Standard mode	4.0			
Low Period of SCL Clock	t_{LOW}	Fast mode	1.3			µs
		Standard mode	4.7			
High Period of SCL Clock	t_{HIGH}	Fast mode	0.6			µs
		Standard mode	4.0			
Data Hold Time (Notes 8, 9)	$t_{HD:DAT}$	Fast mode	0		0.9	µs
		Standard mode	0		0.9	
Data Setup Time (Note 10)	$t_{SU:DAT}$	Fast mode	100			ns
		Standard mode	250			
START Setup Time	$t_{SU:STA}$	Fast mode	0.6			µs
		Standard mode	4.7			
Rise Time of Both SDA and SCL Signals (Note 11)	t_R	Fast mode	$20 + 0.1C_B$		300	ns
		Standard mode			1000	
Fall Time of Both SDA and SCL Signals (Note 11)	t_F	Fast mode	$20 + 0.1C_B$		300	ns
		Standard mode			300	
Setup Time for STOP Condition	$t_{SU:STO}$	Fast mode	0.6			µs
		Standard mode	4.7			
Capacitive Load for Each Bus Line	C_B	(Note 11)			400	pF
Capacitance for SDA, SCL	$C_{I/O}$			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t_{SP}			30		ns
Pushbutton Debounce	PB_{DB}			250		ms
Reset Active Time	t_{RST}			250		ms
Oscillator Stop Flag (OSF) Delay	t_{OSF}	(Note 12)		100		ms
Temperature Conversion Time	t_{CONV}			125	200	ms

Power-Switch Characteristics

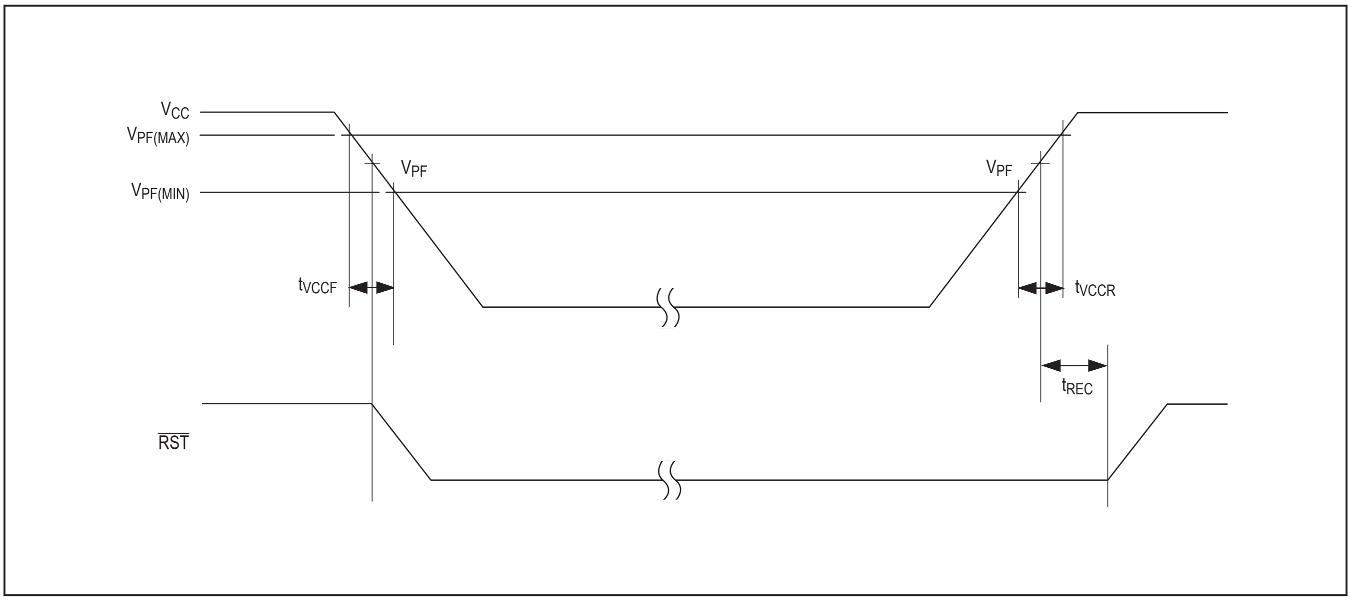
($T_A = T_{MIN}$ to T_{MAX})

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{CC} Fall Time; $V_{PF(MAX)}$ to $V_{PF(MIN)}$	t_{VCCF}		300			µs
V_{CC} Rise Time; $V_{PF(MIN)}$ to $V_{PF(MAX)}$	t_{VCCR}		0			µs
Recovery at Power-Up	t_{REC}	(Note 13)		250	300	ms

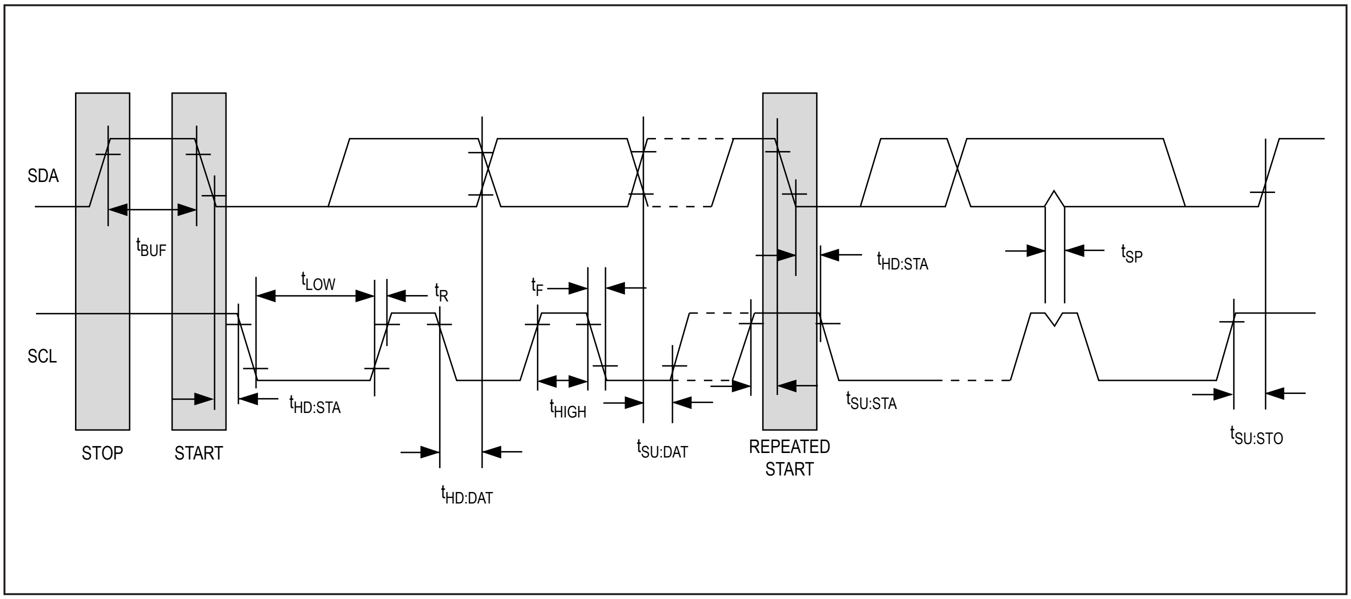
Pushbutton Reset Timing



Power-Switch Timing



Data Transfer on I2C Serial Bus



WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

Note 2: Limits at -40°C are guaranteed by design and not production tested.

Note 3: All voltages are referenced to ground.

Note 4: I_{CCA}—SCL clocking at max frequency = 400kHz.

Note 5: Current is the averaged input current, which includes the temperature conversion current.

Note 6: The $\overline{RST}$ pin has an internal 50kΩ (nominal) pullup resistor to V_{CC}.

Note 7: After this period, the first clock pulse is generated.

Note 8: A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the $V_{IH(MIN)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 9: The maximum $t_{HD:DAT}$ needs only to be met if the device does not stretch the low period (t_{LOW}) of the SCL signal.

Note 10: A fast-mode device can be used in a standard-mode system, but the requirement $t_{SU:DAT} \geq$ 250ns must then be met. This is automatically the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line $t_{R(MAX)} + t_{SU:DAT}$ = 1000 + 250 = 1250ns before the SCL line is released.

Note 11: C_B—total capacitance of one bus line in pF.

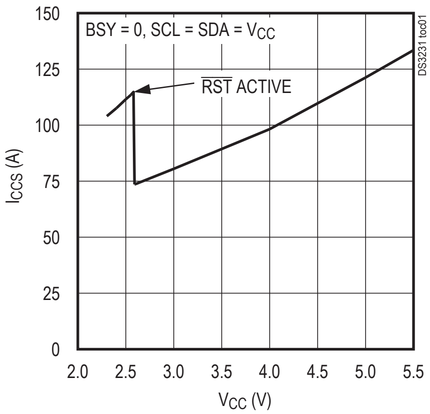
Note 12: The parameter t_{OSF} is the period of time the oscillator must be stopped for the OSF flag to be set over the voltage range of 0.0V ≤ V_{CC} ≤ $V_{CC(MAX)}$ and 2.3V ≤ V_{BAT} ≤ 3.4V.

Note 13: This delay applies only if the oscillator is enabled and running. If the $\overline{EOSC}$ bit is a 1, t_{REC} is bypassed and $\overline{RST}$ immediately goes high. The state of $\overline{RST}$ does not affect the I2C interface, RTC, or TCXO.

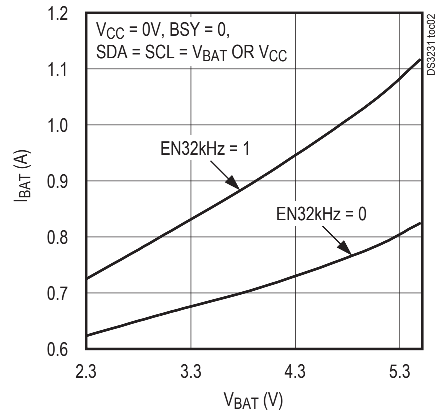
Typical Operating Characteristics

(V_{CC} = +3.3V, T_A = +25°C, unless otherwise noted.)

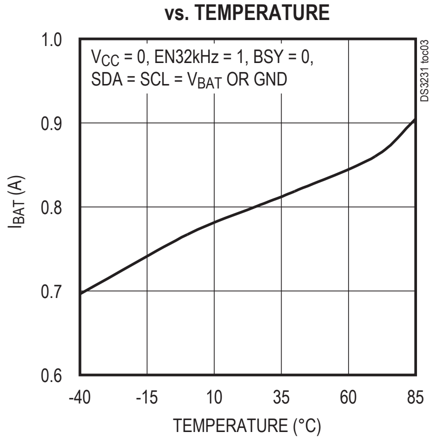
STANDBY SUPPLY CURRENT vs. SUPPLY VOLTAGE



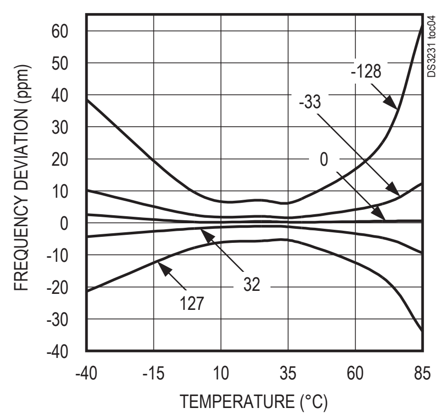
SUPPLY CURRENT vs. SUPPLY VOLTAGE



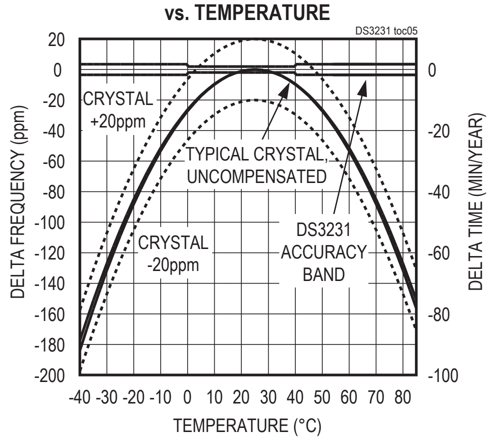
SUPPLY CURRENT vs. TEMPERATURE



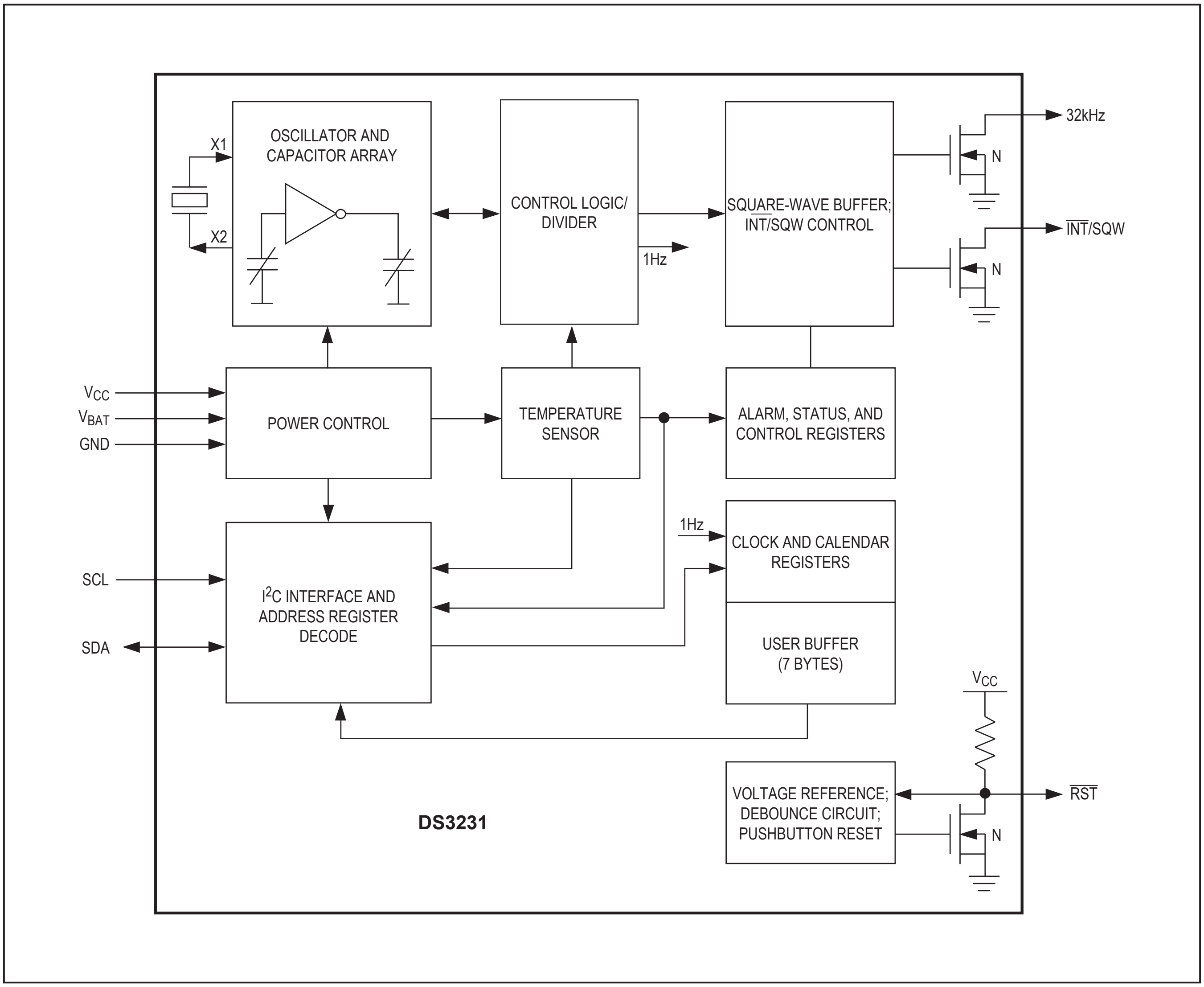
FREQUENCY DEVIATION vs. TEMPERATURE vs. AGING VALUE



DELTA TIME AND FREQUENCY vs. TEMPERATURE



Block Diagram



Pin Description

PIN	NAME	FUNCTION
1	32kHz	32kHz Output. This open-drain pin requires an external pullup resistor. When enabled, the output operates on either power supply. It may be left open if not used.
2	V_{CC}	DC Power Pin for Primary Power Supply. This pin should be decoupled using a 0.1µF to 1.0µF capacitor. If not used, connect to ground.
3	$\overline{INT}$/SQW	Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor connected to a supply at 5.5V or less. This multifunction pin is determined by the state of the INTCN bit in the Control Register (0Eh). When INTCN is set to logic 0, this pin outputs a square wave and its frequency is determined by RS2 and RS1 bits. When INTCN is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the $\overline{INT}$/SQW pin (if the alarm is enabled). Because the INTCN bit is set to logic 1 when power is first applied, the pin defaults to an interrupt output with alarms disabled. The pullup voltage can be up to 5.5V, regardless of the voltage on V_{CC}. If not used, this pin can be left unconnected.
4	$\overline{RST}$	Active-Low Reset. This pin is an open-drain input/output. It indicates the status of V_{CC} relative to the V_{PF} specification. As V_{CC} falls below V_{PF}, the $\overline{RST}$ pin is driven low. When V_{CC} exceeds V_{PF}, for t_{RST}, the $\overline{RST}$ pin is pulled high by the internal pullup resistor. The active-low, open-drain output is combined with a debounced pushbutton input function. This pin can be activated by a pushbutton reset request. It has an internal 50kΩ nominal value pullup resistor to V_{CC}. No external pullup resistors should be connected. If the oscillator is disabled, t_{REC} is bypassed and $\overline{RST}$ immediately goes high.
5–12	N.C.	No Connection. Must be connected to ground.
13	GND	Ground
14	V_{BAT}	Backup Power-Supply Input. When using the device with the V_{BAT} input as the primary power source, this pin should be decoupled using a 0.1µF to 1.0µF low-leakage capacitor. When using the device with the V_{BAT} input as the backup power source, the capacitor is not required. If V_{BAT} is not used, connect to ground. The device is UL recognized to ensure against reverse charging when used with a primary lithium battery. Go to **www.maximintegrated.com/qa/info/ul**.
15	SDA	Serial Data Input/Output. This pin is the data input/output for the I^2C serial interface. This open-drain pin requires an external pullup resistor. The pullup voltage can be up to 5.5V, regardless of the voltage on V_{CC}.
16	SCL	Serial Clock Input. This pin is the clock input for the I^2C serial interface and is used to synchronize data movement on the serial interface. Up to 5.5V can be used for this pin, regardless of the voltage on V_{CC}.

Detailed Description

The DS3231 is a serial RTC driven by a temperature-compensated 32kHz crystal oscillator. The TCXO provides a stable and accurate reference clock, and maintains the RTC to within ±2 minutes per year accuracy from -40°C to +85°C. The TCXO frequency output is available at the 32kHz pin. The RTC is a low-power clock/calendar with two programmable time-of-day alarms and a programmable square-wave output. The $\overline{INT}$/SQW provides either an interrupt signal due to alarm conditions or a square-wave output. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an $\overline{AM}$/PM indicator. The internal registers are accessible though an I^2C bus interface.

A temperature-compensated voltage reference and comparator circuit monitors the level of V_{CC} to detect power failures and to automatically switch to the backup supply when necessary. The $\overline{RST}$ pin provides an external pushbutton function and acts as an indicator of a power-fail event.

Operation

The block diagram shows the main elements of the DS3231. The eight blocks can be grouped into four functional groups: TCXO, power control, pushbutton function, and RTC. Their operations are described separately in the following sections.

32kHz TCXO

The temperature sensor, oscillator, and control logic form the TCXO. The controller reads the output of the on-chip temperature sensor and uses a lookup table to determine the capacitance required, adds the aging correction in AGE register, and then sets the capacitance selection registers. New values, including changes to the AGE register, are loaded only when a change in the temperature value occurs, or when a user-initiated temperature conversion is completed. Temperature conversion occurs on initial application of V_{CC} and once every 64 seconds afterwards.

Power Control

This function is provided by a temperature-compensated voltage reference and a comparator circuit that monitors the V_{CC} level. When V_{CC} is greater than V_{PF}, the part is powered by V_{CC}. When V_{CC} is less than V_{PF} but greater than V_{BAT}, the DS3231 is powered by V_{CC}. If V_{CC} is less than V_{PF} and is less than V_{BAT}, the device is powered by V_{BAT}. See Table 1.

Table 1. Power Control

SUPPLY CONDITION	ACTIVE SUPPLY
V_{CC} < VPF, V_{CC} < V_{BAT}	V_{BAT}
V_{CC} < VPF, V_{CC} < V_{BAT}	V_{CC}
V_{CC} < VPF, V_{CC} < V_{BAT}	V_{CC}
V_{CC} > V_{PF}, V_{CC} > V_{BAT}	V_{CC}

To preserve the battery, the first time V_{BAT} is applied to the device, the oscillator will not start up until V_{CC} exceeds V_{PF}, or until a valid I^2C address is written to the part. Typical oscillator startup time is less than one second. Approximately 2 seconds after V_{CC} is applied, or a valid I^2C address is written, the device makes a temperature measurement and applies the calculated correction to the oscillator. Once the oscillator is running, it continues to run as long as a valid power source is available (V_{CC} or V_{BAT}), and the device continues to measure the temperature and correct the oscillator frequency every 64 seconds.

On the first application of power (V_{CC}) or when a valid I^2C address is written to the part (V_{BAT}), the time and date registers are reset to 01/01/00 01 00:00:00 (DD/MM/YY DOW HH:MM:SS).

V_{BAT} Operation

There are several modes of operation that affect the amount of V_{BAT} current that is drawn. While the device is powered by V_{BAT} and the serial interface is active, active battery current, I_{BATA}, is drawn. When the serial interface is inactive, timekeeping current (I_{BATT}), which includes the averaged temperature conversion current, I_{BATTC}, is used (refer to Application Note 3644: *Power Considerations for Accurate Real-Time Clocks* for details). Temperature conversion current, I_{BATTC}, is specified since the system must be able to support the periodic higher current pulse and still maintain a valid voltage level. Data retention current, I_{BATTDR}, is the current drawn by the part when the oscillator is stopped ($\overline{EOSC}$ = 1). This mode can be used to minimize battery requirements for times when maintaining time and date information is not necessary, e.g., while the end system is waiting to be shipped to a customer.

Pushbutton Reset Function

The DS3231 provides for a pushbutton switch to be connected to the $\overline{RST}$ output pin. When the DS3231 is not in a reset cycle, it continuously monitors the $\overline{RST}$ signal for a low going edge. If an edge transition is detected, the DS3231 debounces the switch by pulling the $\overline{RST}$ low. After the internal timer has expired (PB_{DB}), the DS3231 continues to monitor the $\overline{RST}$ line. If the line is still low, the DS3231 continuously monitors the line looking for a rising edge. Upon detecting release, the DS3231 forces the $\overline{RST}$ pin low and holds it low for t_{RST}.

$\overline{RST}$ is also used to indicate a power-fail condition. When V_{CC} is lower than V_{PF}, an internal power-fail signal is generated, which forces the $\overline{RST}$ pin low. When V_{CC} returns to a level above V_{PF}, the $\overline{RST}$ pin is held low for approximately 250ms (t_{REC}) to allow the power supply to stabilize. If the oscillator is not running (see the *Power Control* section) when V_{CC} is applied, t_{REC} is bypassed and $\overline{RST}$ immediately goes high. Assertion of the $\overline{RST}$ output, whether by pushbutton or power-fail detection, does not affect the internal operation of the DS3231.

Real-Time Clock

With the clock source from the TCXO, the RTC provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an $\overline{AM}$/PM indicator.

The clock provides two programmable time-of-day alarms and a programmable square-wave output. The $\overline{INT}$/SQW pin either generates an interrupt due to alarm condition or outputs a square-wave signal and the selection is controlled by the bit INTCN.

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/$\overline{24}$	$\overline{AM}$/PM 20 Hour	10 Hour	Hour				Hours	1–12 + $\overline{AM}$/PM 00–23
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date		Date				Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00–59
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00–59
09h	A1M3	12/$\overline{24}$	$\overline{AM}$/PM 20 Hour	10 Hour	Hour				Alarm 1 Hours	1–12 + $\overline{AM}$/PM 00–23
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1–7
					Date				Alarm 1 Date	1–31
0Bh	A2M2	10 Minutes			Minutes				Alarm 2 Minutes	00–59
0Ch	A2M3	12/$\overline{24}$	$\overline{AM}$/PM 20 Hour	10 Hour	Hour				Alarm 2 Hours	1–12 + $\overline{AM}$/PM 00–23
0Dh	A2M4	DY/$\overline{DT}$	10 Date		Day				Alarm 2 Day	1–7
					Date				Alarm 2 Date	1–31
0Eh	$\overline{EOSC}$	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Figure 1. Timekeeping Registers

Note: *Unless otherwise specified, the registers' state is not defined when power is first applied.*

Address Map

Figure 1 shows the address map for the DS3231 timekeeping registers. During a multibyte access, when the address pointer reaches the end of the register space (12h), it wraps around to location 00h. On an I2C START or address pointer incrementing to location 00h, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to reread the registers in case the main registers update during a read.

I2C Interface

The I2C interface is accessible whenever either V_{CC} or V_{BAT} is at a valid level. If a microcontroller connected to the DS3231 resets because of a loss of V_{CC} or other event, it is possible that the microcontroller and DS3231 I2C communications could become unsynchronized, e.g., the microcontroller resets while reading data from the DS3231. When the microcontroller resets, the DS3231 I2C interface may be placed into a known state by toggling SCL until SDA is observed to be at a high level. At that point the microcontroller should pull SDA low while SCL is high, generating a START condition.

Clock and Calendar

The time and calendar information is obtained by reading the appropriate register bytes. Figure 1 illustrates the RTC registers. The time and calendar data are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the binary-coded decimal (BCD) format. The DS3231 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the $\overline{\text{AM}}$/PM bit with logic-high being PM. In the 24-hour mode, bit 5 is the 20-hour bit (20–23 hours). The century bit (bit 7 of the month register) is toggled when the years register overflows from 99 to 00.

The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on). Illogical time and date entries result in undefined operation.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any START and when the register pointer rolls over to zero. The time information is read from these secondary registers, while the clock continues to run. This eliminates the need to reread the registers in case the main registers update during a read.

The countdown chain is reset whenever the seconds register is written. Write transfers occur on the acknowledge from the DS3231. Once the countdown chain is reset, to avoid rollover issues the remaining time and date registers must be written within 1 second. The 1Hz square-wave output, if enabled, transitions high 500ms after the seconds data transfer, provided the oscillator is already running.

Alarms

The DS3231 contains two time-of-day/date alarms. Alarm 1 can be set by writing to registers 07h to 0Ah. Alarm 2 can be set by writing to registers 0Bh to 0Dh. The alarms can be programmed (by the alarm enable and INTCN bits of the control register) to activate the $\overline{\text{INT}}$/SQW output on an alarm match condition. Bit 7 of each of the time-of-day/date alarm registers are mask bits (Table 2). When all the mask bits for each alarm are logic 0, an alarm only occurs when the values in the timekeeping registers match the corresponding values stored in the time-of-day/date alarm registers. The alarms can also be programmed to repeat every second, minute, hour, day, or date. Table 2 shows the possible settings. Configurations not listed in the table will result in illogical operation.

The DY/$\overline{\text{DT}}$ bits (bit 6 of the alarm day/date registers) control whether the alarm value stored in bits 0 to 5 of that register reflects the day of the week or the date of the month. If DY/$\overline{\text{DT}}$ is written to logic 0, the alarm will be the result of a match with date of the month. If DY/$\overline{\text{DT}}$ is written to logic 1, the alarm will be the result of a match with day of the week.

When the RTC register values match alarm register settings, the corresponding Alarm Flag 'A1F' or 'A2F' bit is set to logic 1. If the corresponding Alarm Interrupt Enable 'A1IE' or 'A2IE' is also set to logic 1 and the INTCN bit is set to logic 1, the alarm condition will activate the $\overline{\text{INT}}$/SQW signal. The match is tested on the once-per-second update of the time and date registers.

Table 2. Alarm Mask Bits

DY/$\overline{\text{DT}}$	ALARM 1 REGISTER MASK BITS (BIT 7)				ALARM RATE
	A1M4	A1M3	A1M2	A1M1	
X	1	1	1	1	Alarm once per second
X	1	1	1	0	Alarm when seconds match
X	1	1	0	0	Alarm when minutes and seconds match
X	1	0	0	0	Alarm when hours, minutes, and seconds match
0	0	0	0	0	Alarm when date, hours, minutes, and seconds match
1	0	0	0	0	Alarm when day, hours, minutes, and seconds match

DY/$\overline{\text{DT}}$	ALARM 2 REGISTER MASK BITS (BIT 7)			ALARM RATE
	A2M4	A2M3	A2M2	
X	1	1	1	Alarm once per minute (00 seconds of every minute)
X	1	1	0	Alarm when minutes match
X	1	0	0	Alarm when hours and minutes match
0	0	0	0	Alarm when date, hours, and minutes match
1	0	0	0	Alarm when day, hours, and minutes match

Control Register (0Eh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	$\overline{\text{EOSC}}$	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE
POR:	0	0	0	1	1	1	0	0

Special-Purpose Registers

The DS3231 has two additional registers (control and status) that control the real-time clock, alarms, and square-wave output.

Control Register (0Eh)

Bit 7: Enable Oscillator ($\overline{\text{EOSC}}$). When set to logic 0, the oscillator is started. When set to logic 1, the oscillator is stopped when the DS3231 switches to V_{BAT}. This bit is clear (logic 0) when power is first applied. When the DS3231 is powered by V_{CC}, the oscillator is always on regardless of the status of the $\overline{\text{EOSC}}$ bit. When $\overline{\text{EOSC}}$ is disabled, all register data is static.

Bit 6: Battery-Backed Square-Wave Enable (BBSQW). When set to logic 1 with INTCN = 0 and $V_{CC} < V_{PF}$, this bit enables the square wave. When BBSQW is logic 0, the $\overline{\text{INT}}$/SQW pin goes high impedance when $V_{CC} < V_{PF}$. This bit is disabled (logic 0) when power is first applied.

Bit 5: Convert Temperature (CONV). Setting this bit to 1 forces the temperature sensor to convert the temperature into digital code and execute the TCXO algorithm to update the capacitance array to the oscillator. This can only happen when a conversion is not already in progress. The user should check the status bit BSY before forcing the controller to start a new TCXO execution. A user-initiated temperature conversion does not affect the internal 64-second update cycle.

A user-initiated temperature conversion does not affect the BSY bit for approximately 2ms. The CONV bit remains at a 1 from the time it is written until the conversion is finished, at which time both CONV and BSY go to 0. The CONV bit should be used when monitoring the status of a user-initiated conversion.

Bits 4 and 3: Rate Select (RS2 and RS1). These bits control the frequency of the square-wave output when the square wave has been enabled. The following table shows the square-wave frequencies that can be selected with the RS bits. These bits are both set to logic 1 (8.192kHz) when power is first applied.

SQUARE-WAVE OUTPUT FREQUENCY

RS2	RS1	SQUARE-WAVE OUTPUT FREQUENCY
0	0	1Hz
0	1	1.024kHz
1	0	4.096kHz
1	1	8.192kHz

Bit 2: Interrupt Control (INTCN). This bit controls the $\overline{\text{INT}}$/SQW signal. When the INTCN bit is set to logic 0, a square wave is output on the $\overline{\text{INT}}$/SQW pin. When the INTCN bit is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the $\overline{\text{INT}}$/SQW output (if the alarm is also enabled). The corresponding alarm flag is always set regardless of the state of the INTCN bit. The INTCN bit is set to logic 1 when power is first applied.

Bit 1: Alarm 2 Interrupt Enable (A2IE). When set to logic 1, this bit permits the alarm 2 flag (A2F) bit in the status register to assert $\overline{\text{INT}}$/SQW (when INTCN = 1). When the A2IE bit is set to logic 0 or INTCN is set to logic 0, the A2F bit does not initiate an interrupt signal. The A2IE bit is disabled (logic 0) when power is first applied.

Bit 0: Alarm 1 Interrupt Enable (A1IE). When set to logic 1, this bit permits the alarm 1 flag (A1F) bit in the status register to assert $\overline{\text{INT}}$/SQW (when INTCN = 1). When the A1IE bit is set to logic 0 or INTCN is set to logic 0, the A1F bit does not initiate the $\overline{\text{INT}}$/SQW signal. The A1IE bit is disabled (logic 0) when power is first applied.

Status Register (0Fh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	OSF	0	0	0	EN32kHz	BSY	A2F	A1F
POR:	1	0	0	0	1	X	X	X

Status Register (0Fh)

Bit 7: Oscillator Stop Flag (OSF). A logic 1 in this bit indicates that the oscillator either is stopped or was stopped for some period and may be used to judge the validity of the timekeeping data. This bit is set to logic 1 any time that the oscillator stops. The following are examples of conditions that can cause the OSF bit to be set:

1) The first time power is applied.
2) The voltages present on both V_{CC} and V_{BAT} are insufficient to support oscillation.
3) The $\overline{\text{EOSC}}$ bit is turned off in battery-backed mode.
4) External influences on the crystal (i.e., noise, leakage, etc.).

This bit remains at logic 1 until written to logic 0.

Bit 3: Enable 32kHz Output (EN32kHz). This bit controls the status of the 32kHz pin. When set to logic 1, the 32kHz pin is enabled and outputs a 32.768kHz square-wave signal. When set to logic 0, the 32kHz pin goes to a high-impedance state. The initial power-up state of this bit is logic 1, and a 32.768kHz square-wave signal appears at the 32kHz pin after a power source is applied to the DS3231 (if the oscillator is running).

Bit 2: Busy (BSY). This bit indicates the device is busy executing TCXO functions. It goes to logic 1 when the conversion signal to the temperature sensor is asserted and then is cleared when the device is in the 1-minute idle state.

Bit 1: Alarm 2 Flag (A2F). A logic 1 in the alarm 2 flag bit indicates that the time matched the alarm 2 registers. If the A2IE bit is logic 1 and the INTCN bit is set to logic 1, the $\overline{\text{INT}}$/SQW pin is also asserted. A2F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Bit 0: Alarm 1 Flag (A1F). A logic 1 in the alarm 1 flag bit indicates that the time matched the alarm 1 registers. If the A1IE bit is logic 1 and the INTCN bit is set to logic 1, the $\overline{\text{INT}}$/SQW pin is also asserted. A1F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Aging Offset

The aging offset register takes a user-provided value to add to or subtract from the codes in the capacitance array registers. The code is encoded in two's complement, with bit 7 representing the sign bit. One LSB represents one small capacitor to be switched in or out of the capacitance array at the crystal pins. The aging offset register capacitance value is added or subtracted from the capacitance value that the device calculates for each temperature compensation. The offset register is added to the capacitance array during a normal temperature conversion, if the temperature changes from the previous conversion, or during a manual user conversion (setting the CONV bit). To see the effects of the aging register on the 32kHz output frequency immediately, a manual conversion should be started after each aging register change.

Positive aging values add capacitance to the array, slowing the oscillator frequency. Negative values remove capacitance from the array, increasing the oscillator frequency.

The change in ppm per LSB is different at different temperatures. The frequency vs. temperature curve is shifted by the values used in this register. At +25°C, one LSB typically provides about 0.1ppm change in frequency.

Use of the aging register is not needed to achieve the accuracy as defined in the EC tables, but could be used to help compensate for aging at a given temperature. See the *Typical Operating Characteristics* section for a graph showing the effect of the register on accuracy over temperature.

Aging Offset (10h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data	Data	Data	Data	Data	Data	Data
POR:	0	0	0	0	0	0	0	0

Temperature Register (Upper Byte) (11h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data	Data	Data	Data	Data	Data	Data
POR:	0	0	0	0	0	0	0	0

Temperature Register (Lower Byte) (12h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Data	Data	0	0	0	0	0	0
POR:	0	0	0	0	0	0	0	0

Temperature Registers (11h–12h)

Temperature is represented as a 10-bit code with a resolution of 0.25°C and is accessible at location 11h and 12h. The temperature is encoded in two's complement format. The upper 8 bits, the integer portion, are at location 11h and the lower 2 bits, the fractional portion, are in the upper nibble at location 12h. For example, 00011001 01b = +25.25°C. Upon power reset, the registers are set to a default temperature of 0°C and the controller starts a temperature conversion. The temperature is read on initial application of V_{CC} or I^2C access on V_{BAT} and once every 64 seconds afterwards. The temperature registers are updated after each user-initiated conversion and on every 64-second conversion. The temperature registers are read-only.

I^2C Serial Data Bus

The DS3231 supports a bidirectional I^2C bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data is defined as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS3231 operates as a slave on the I^2C bus. Connections to the bus are made through the SCL input and open-drain SDA I/O lines. Within the bus specifications, a standard mode (100kHz maximum clock rate) and a fast mode (400kHz maximum clock rate) are defined. The DS3231 works in both modes.

The following bus protocol has been defined (Figure 2):

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data line while the clock line is high are interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain high.

START data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

STOP data transfer: A change in the state of the data line from low to high, while the clock line is high, defines a STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred bytewise and each receiver acknowledges with a ninth bit.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse, which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge-related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generat-

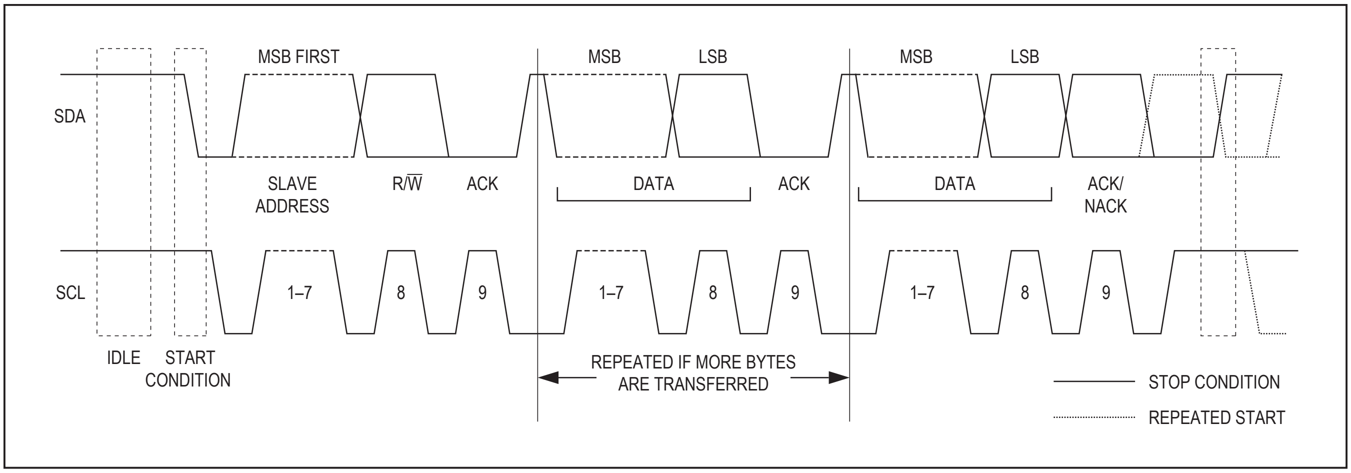


Figure 2. I²C Data Transfer Overview

ing an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line high to enable the master to generate the STOP condition.

Figures 3 and 4 detail how data transfer is accomplished on the I²C bus. Depending upon the state of the $R/\overline{W}$ bit, two types of data transfer are possible:

Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows a number of data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the

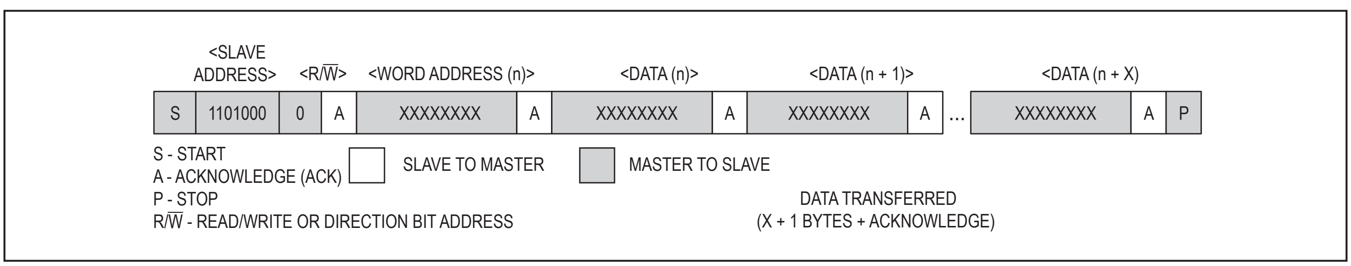


Figure 3. Data Write—Slave Receiver Mode

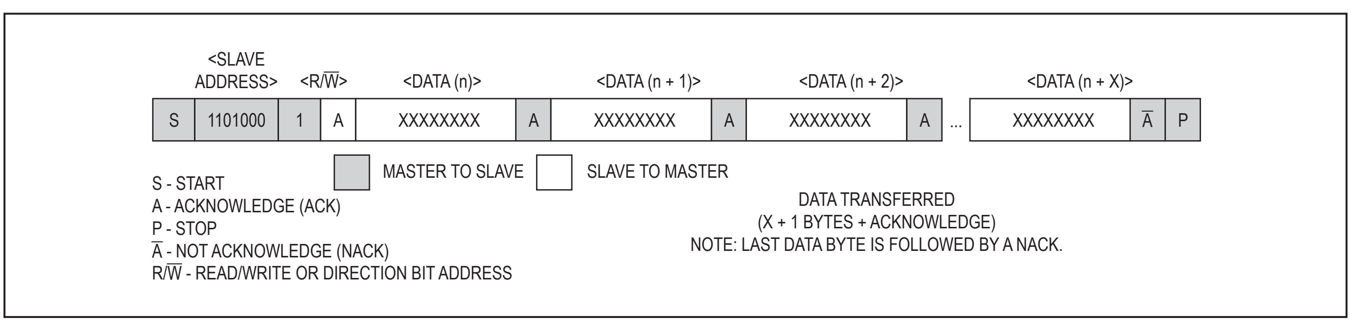


Figure 4. Data Read—Slave Transmitter Mode

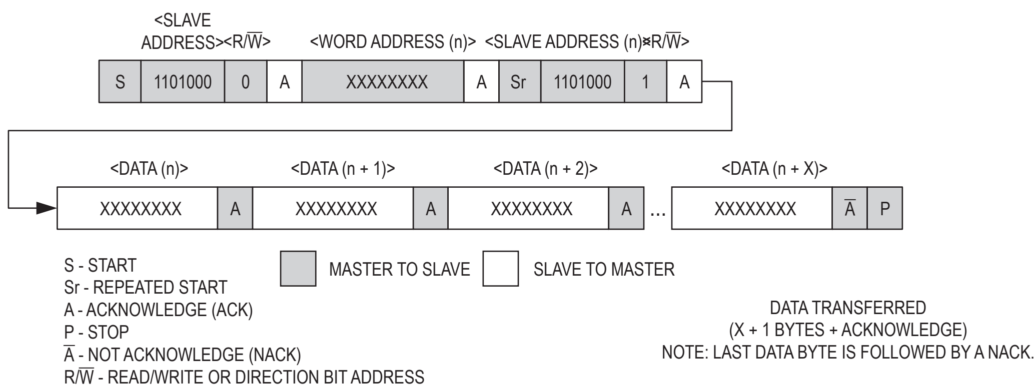


Figure 5. Data Write/Read (Write Pointer, Then Read)—Slave Receive and Transmit

last byte. At the end of the last received byte, a not acknowledge is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS3231 can operate in the following two modes:

Slave receiver mode (DS3231 write mode): Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/$\overline{W}$), which is 0 for a write. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. After the DS3231 acknowledges the slave address + write bit, the master transmits a word address to the DS3231. This sets the register pointer on the DS3231, with the DS3231 acknowledging the transfer. The master may then transmit zero or more bytes of data, with the DS3231 acknowledging each byte received. The register pointer increments after each data byte is transferred. The master generates a STOP condition to terminate the data write.

Slave transmitter mode (DS3231 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit indicates that the transfer direction is reversed. Serial data is transmitted on SDA by the DS3231 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates a START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/$\overline{W}$), which is 1 for a read. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. The DS3231 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode, the first address that is read is the last one stored in the register pointer. The DS3231 must receive a not acknowledge to end a read.

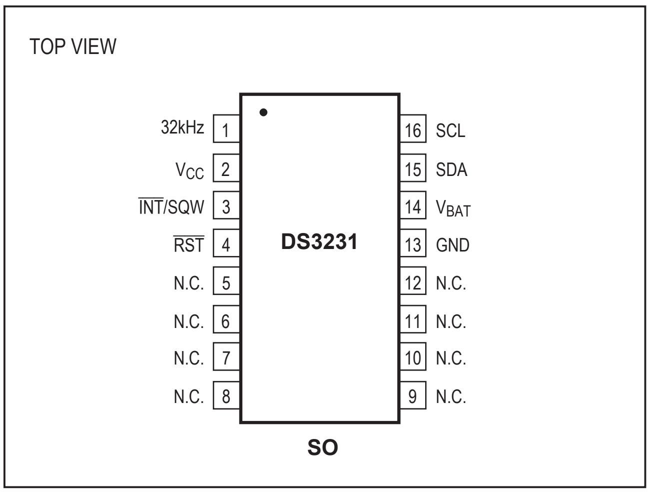
Handling, PCB Layout, and Assembly

The DS3231 package contains a quartz tuning-fork crystal. Pick-and-place equipment can be used, but precautions should be taken to ensure that excessive shocks are avoided. Ultrasonic cleaning should be avoided to prevent damage to the crystal.

Avoid running signal traces under the package, unless a ground plane is placed between the package and the signal line. All N.C. (no connect) pins must be connected to ground.

Moisture-sensitive packages are shipped from the factory dry packed. Handling instructions listed on the package label must be followed to prevent damage during reflow. Refer to the IPC/JEDEC J-STD-020 standard for moisture-sensitive device (MSD) classifications and reflow profiles. Exposure to reflow is limited to 2 times maximum.

Pin Configuration



Chip Information

SUBSTRATE CONNECTED TO GROUND

PROCESS: CMOS

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
DS3231S#	0°C to +70°C	16 SO
DS3231SN#	-40°C to +85°C	16 SO

#Denotes an RoHS-compliant device that may include lead (Pb) that is exempt under RoHS requirements. The lead finish is JESD97 category e3, and is compatible with both lead-based and lead-free soldering processes. A "#" anywhere on the top mark denotes an RoHS-compliant device.

Package Information

For the latest package outline information and land patterns (footprints), go to **www.maximintegrated.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
16 SO	W16#H2	21-0042	90-0107

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	1/05	Initial release.	—
1	2/05	Changed Digital Temp Sensor Output from ±2°C to ±3°C.	1, 3
		Updated *Typical Operating Circuit*.	1
		Changed T_A = -40°C to +85°C to T_A = T_{MIN} to T_{MAX}.	2, 3, 4
		Updated *Block Diagram*.	8
2	6/05	Added "UL Recognized" to Features; added lead-free packages and removed S from top mark info in *Ordering Information* table; added ground connections to the N.C. pin in the Typical Operating Circuit.	1
		Added "noncondensing" to operating temperature range; changed V_{PF} MIN from 2.35V to 2.45V.	2
		Added aging offset specification.	3
		Relabeled TOC4.	7
		Added arrow showing input on X1 in the *Block Diagram*.	8
		Updated pin descriptions for V_{CC} and V_{BAT}.	9
		Added the I2C Interface section.	10
		Figure 1: Added sign bit to aging and temperature registers; added MSB and LSB.	11
		Corrected title for rate select bits frequency table.	13
		Added note that frequency stability over temperature spec is with aging offset register = 00h; changed bit 7 from Data to Sign (Crystal Aging Offset Register).	14
		Changed bit 7 from Data to Sign (Temperature Register); correct pin definitions in *I2C Serial Data Bus* section.	15
		Modified the *Handing*, *PC Board Layout*, and *Assembly* section to refer to J-STD-020 for reflow profiles for lead-free and leaded packages.	17
3	11/05	Changed lead-free packages to RoHS-compliant packages.	1
4	10/06	Changed $\overline{RST}$ and UL bullets in *Features*.	1
		Changed EC condition "V_{CC} > V_{BAT}" to "V_{CC} = Active Supply (see Table 1)."	2, 3
		Modified Note 12 to correct t_{REC} operation.	6
		Added various conditions text to TOCs 1, 2, and 3.	7
		Added text to pin descriptions for 32kHz, V_{CC}, and $\overline{RST}$.	9
		Table 1: Changed column heading "Powered By" to "Active Supply"; changed "applied" to "exceeds V_{PF}" in the *Power Control* section.	10
		Indicated BBSQW applies to both SQW and interrupts; simplified temp convert description (bit 5); added "output" to $\overline{INT}$/SQW (bit 2).	13
		Changed the *Crystal Aging* section to the *Aging Offset* section; changed "this bit indicates" to "this bit controls" for the enable 32kHz output bit.	14
5	4/08	Added Warning note to EC table notes; updated Note 12.	6
		Updated the *Typical Operating Characteristics* graphs.	7
		In the *Power Control* section, added information about the POR state of the time and date registers; in the *Real-Time Clock* section, added to the description of the RST function.	10
		In Figure 1, corrected the months date range for 04h from 00–31 to 01–31.	11

Revision History (continued)

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
6	10/08	Updated the *Typical Operating Circuit*.	1
		Removed the V_{PU} parameter from the *Recommended DC Operating Conditions* table and added verbiage about the pullup to the *Pin Description* table for $\overline{INT}$/SQW, SDA, and SCL.	2, 9
		Added the Delta Time and Frequency vs. Temperature graph in the *Typical Operating Characteristics* section.	7
		Updated the *Block Diagram*.	8
		Added the *V_{BAT} Operation* section, improved some sections of text for the *32kHz TCXO* and *Pushbutton Reset Function* sections.	10
		Added the register bit POR values to the register tables.	13, 14, 15
		Updated the *Aging Offset* and *Temperature Registers (11h–12h)* sections.	14, 15
		Updated the I2C timing diagrams (Figures 3, 4, and 5).	16, 17
7	3/10	Removed the "S" from the top mark in the *Ordering Information* table and the *Pin Configuration* to match the packaging engineering marking specification.	1, 18
8	7/10	Updated the *Typical Operating Circuit*; removed the "Top Mark" column from the *Ordering Information*; in the *Absolute Maximum Ratings* section, added the theta-JA and theta-JC thermal resistances and Note 1, and changed the soldering temperature to +260°C (lead(Pb)-free) and +240°C (leaded); updated the functional description of the V_{BAT} pin in the *Pin Description*; changed the timekeeping registers 02h, 09h, and 0Ch to "20 Hour" in Bit 5 of Figure 1; updated the BBSQW bit description in the *Control Register (0Eh)* section; added the land pattern no. to the *Package Information* table.	1, 2, 3, 4, 6, 9, 11, 12, 13, 18
9	1/13	Updated *Absolute Maximum Ratings*, and last paragraph in *Power Control* section	2, 10
10	3/15	Revised *Benefits and Features* section.	1

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8

DESCRIPTION

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter.

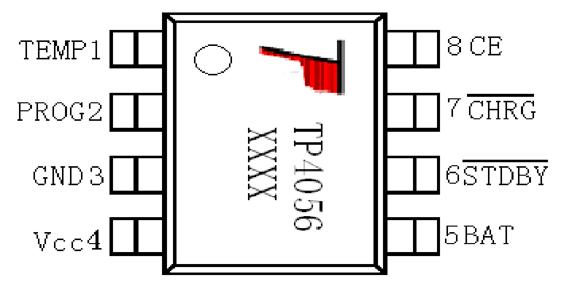
No blocking diode is required due to the internal PMOSFET architecture and have prevent to negative Charge Current Circuit. Thermal feedback regulates the charge current to limit the die temperature during high power operation or high ambient temperature. The charge voltage is fixed at 4.2V, and the charge current can be programmed externally with a single resistor. The TP4056 automatically terminates the charge cycle when the charge current drops to 1/10th the programmed value after the final float voltage is reached.

TP4056 Other features include current monitor, under voltage lockout, automatic recharge and two status pin to indicate charge termination and the presence of an input voltage.

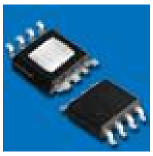
FEATURES

- Programmable Charge Current Up to 1000mA
- No MOSFET, Sense Resistor or Blocking Diode Required
- Complete Linear Charger in SOP-8 Package for Single Cell Lithium-Ion Batteries
- Constant-Current/Constant-Voltage
- Charges Single Cell Li-Ion Batteries Directly from USB Port
- Preset 4.2V Charge Voltage with 1.5% Accuracy
- Automatic Recharge
- two Charge Status Output Pins
- C/10 Charge Termination
- 2.9V Trickle Charge Threshold (TP4056)
- Soft-Start Limits Inrush Current
- Available Radiator in 8-Lead SOP Package, the Radiator need connect GND or impending

PACKAGE/ORDER INFORMATION



SOP－8

photo	
	**ORDER PART NUMBER** TP4056-42-SOP8-PP
	PART MARKING TP4056

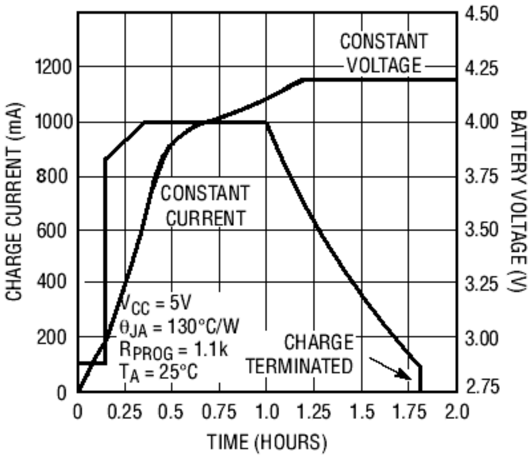
ABSOLUTE MAXIMUM RATINGS

- Input Supply Voltage(V_{CC})：-0.3V～8V
- TEMP：**-0.3V**～10V
- CE：**-0.3V**～10V
- BAT Short-Circuit Duration：Continuous
- BAT Pin Current：1200mA
- PROG Pin Current：1200uA
- Maximum Junction Temperature：145℃
- Operating Ambient Temperature Range：-40℃～85℃
- Lead Temp.(Soldering, 10sec)：260℃

APPLICATIONS

- Cellular Telephones, PDAs, GPS
- Charging Docks and Cradles
- Digital Still Cameras, Portable Devices
- USB Bus-Powered Chargers,Chargers

Complete Charge Cycle (1000mAh Battery)



TEMP(Pin 1) :Temperature Sense Input Connecting TEMP pin to NTC thermistor's output in Lithium ion battery pack. If TEMP pin's voltage is below 45% or above 80% of supply voltage V_{IN} for more than 0.15S, this means that battery's temperature is too high or too low, charging is suspended. The temperature sense function can be disabled by grounding the TEMP pin.

PROG(Pin 2): Constant Charge Current Setting and Charge Current Monitor Pin charge current is set by connecting a resistor R_{ISET} from this pin to GND. When in precharge mode, the ISET pin's voltage is regulated to 0.2V. When in constant charge current mode, the ISET pin's voltage is regulated to 2V.In all modes during charging, the voltage on ISET pin can be used to measure the charge current as follows:

$$I_{BAT} = \frac{V_{PROG}}{R_{PROG}} \times 1200 \quad (V_{PROG}=1V)$$

GND(Pin3): Ground Terminal

Vcc(Pin 4): Positive Input Supply Voltage V_{IN} is the power supply to the internal circuit. When V_{IN} drops to within 30mv of the BAT pin voltage, TP4056 enters low power sleep mode, dropping BAT pin's current to less than 2uA.

BAT(Pin5): Battery Connection Pin. Connect the positive terminal of the battery to BAT pin. BAT pin draws less than 2uA current in chip disable mode or in sleep mode. BAT pin provides charge current to the battery and provides regulation voltage of 4.2V.

$\overline{\text{STDBY}}$(Pin6): Open Drain Charge Status Output When the battery Charge Termination, the $\overline{\text{STDBY}}$ pin is pulled low by an internal switch, otherwise $\overline{\text{STDBY}}$ pin is in high impedance state.

$\overline{\text{CHRG}}$ (Pin7): Open Drain Charge Status Output When the battery is being charged, the $\overline{\text{CHRG}}$ pin is pulled low by an internal switch, otherwise $\overline{\text{CHRG}}$ pin is in high impedance state.

CE(Pin8): Chip Enable Input. A high input will put the device in the normal operating mode. Pulling the CE pin to low level will put the YP4056 into disable mode. The CE pin can be driven by TTL or CMOS logic level.

ELECTRICAL CHARACTERISTICS

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are at **T_A=25℃，V_{CC} =5V**，unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
V_{CC}	Input Supply Voltage		●	4.0	5	8.0	V
I_{CC}	Input Supply Current	Charge Mode, R_{PROG} = 1.2k StandbyMode(Charge Terminated) Shutdown Mode (R_{PROG} Not Connected,V_{CC} < V_{BAT}, or V_{CC} < V_{UV})	● ● ●		150 55 55	500 100 100	µA µA µA
V_{FLOAL}	Regulated Output (Float) Voltage	0℃≤T_A≤85℃，I_{BAT}=40mA		4.137	4.2	4.263	V
I_{BAT}	BAT Pin Current Text condition:VBAT=4.0V	R_{PROG} = 2.4k, Current Mode R_{PROG} = 1.2k, Current Mode Standby Mode, V_{BAT} = 4.2V	● ● ●	450 950 0	500 1000 −2.5	550 1050 −6	mA mA µA
I_{TRIKL}	Trickle Charge Current	V_{BAT}<V_{TRIKL}, R_{PROG}=1.2K	●	120	130	140	mA
V_{TRIKL}	Trickle Charge Threshold Voltage	R_{PROG}=1.2K, V_{BAT} Rising		2.8	2.9	3.0	V
V_{TRHYS}	Trickle Charge Hysteresis Voltage	R_{PROG}=1.2K		60	80	100	mV
T_{LIM}	Junction Temperature in Constant Temperature Mode				145		℃

indicator light state

Charge state	Red LED $\overline{\text{CHRG}}$	Greed LED $\overline{\text{STDBY}}$
charging	bright	extinguish
Charge Termination	extinguish	bright
Vin too low; Temperature of battery too low or too high; no battery	extinguish	extinguish
BAT PIN Connect 10u Capacitance; No battery	Greed LED bright，Red LED Coruscate T=1-4 S	

Rprog Current Setting

RPROG (k)	IBAT (mA)
10	130
5	250
4	300
3	400
2	580
1.66	690
1.5	780
1.33	900
1.2	1000

TYPICAL APPLICATIONS