

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Farmacia robot: cómo la automatización cambiará
radicalmente la distribución de medicamentos**
Trabajo Experimental

Diego Ricardo Larrea Cueva
Carlos Daniel Navarrete Gando

Ingeniería Electrónica

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

Quito, 11 de diciembre de 2015

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

**Farmacia robot: cómo la automatización cambiará radicalmente la
distribución de medicamentos**

Diego Ricardo Larrea Cueva

Carlos Daniel Navarrete Gando

Calificación:

Nombre del profesor, Título académico

Diego Benítez, Ph.D.

Firma del profesor

Quito, 11 de diciembre de 2015

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.

Firma del estudiante: _____

Nombres y apellidos: Diego Ricardo Larrea Cueva

Código: 00103810

Cédula de Identidad: 1717260234

Firma del estudiante: _____

Nombres y apellidos: Carlos Daniel Navarrete Gando

Código: 00106283

Cédula de Identidad: 1713937421

Lugar y fecha: Quito, diciembre de 2015

RESUMEN

La distribución es un punto crítico en la cadena de suministro de fármacos dado que los mismos requieren: un manejo preciso de inventario, poca manipulación y un ambiente controlado. Por otra parte los usuarios requieren de un sistema que sea confiable, con atención indefinida (24/7) y que disponga de varios puntos de venta. En el presente trabajo se ha desarrollado un prototipo de maquina dispensadora de medicamentos tomando en cuenta las necesidades expuestas y un presupuesto limitado. Se ha logrado crear un sistema que utiliza componentes de bajo costo que ofrece confiabilidad, estabilidad y precisión en su operación. Además el diseño asegura un control de inventario certero que reduce al mínimo la necesidad de que varios operadores tengan acceso a los medicamentos en bodega. Finalmente se ha garantizado las condiciones ambientales adecuadas para la conservación de los productos al interior del equipo. Se ha concluido que es posible desarrollar localmente una maquina dispensadora que satisfaga las necesidades actuales del sector farmacéutico a un costo apegado a la economía ecuatoriana. El prototipo puede ser rediseñado en un futuro para su fabricación en masa, lo cual implicaría la generación de un nueva forma de distribución de medicamentos en el mercado tanto para venta directa como indirecta al consumidor.

Palabras clave: Dispensador, automatización, medicinas, farmacia, inventario, Arduino MEGA, microcontrolador.

ABSTRACT

Distribution is a key point in medicine's supply chain. Drug's distribution requires: precise inventory control, limited human manipulation and specific environmental conditions. On the other hand, customers need a distribution system which is: reliable, 24/7 and with variety of selling points. Considering these needs and a limited budget, a medicine vending machine prototype has been developed with a low cost design that offers precision, stability, reliability and inventory control with less handling required. Finally, environmental product conditions have been satisfied with the integration of a temperature control system. In conclusion, it has been possible to develop a vending machine prototype that satisfies the Ecuadorian market necessities. For the future the prototype could be redesigned for mass production, which means that a new way of drugstore would be introduced to the existing market.

Key words: Vending Machine, automation, medicines, drugstore, inventory, Arduino MEGA, microcontroller.

TABLA DE CONTENIDO

Introducción.....	9
Desarrollo del Tema.....	12
Pruebas y Resultados.....	47
Conclusiones.....	53
Discusiones.....	55
Referencias bibliográficas.....	57
Anexo A: prototipo.....	62
Anexo B: Figuras Adicionales.....	71
Anexo C: Código Microcontrolador.....	75

ÍNDICE DE TABLAS

Tabla 1. Denominaciones y características de tiras LED SMD.....	26
Tabla 2. Lista y descripción de terminales del circuito diseñado.....	35
Tabla 3. Codificación de mensajes entre el microcontrolador y el microcomputador.....	38
Tabla 4. Consumo de corriente y potencia de motores Servo.....	49

ÍNDICE DE FIGURAS

Figura 1. Modelo General del Proyecto	10
Figura 2. Diagrama de bloques del sistema de control de la máquina.....	12
Figura 3. Diagrama esquemático de un motor de pasos de imán permanente bifásico...	17
Figura 4. Diagrama esquemático de motor de pasos híbrido.....	18
Figura 5. Diagrama de devanados con código por color de motor de paso híbrido.....	19
Figura 6. Diagrama de cableado mínimo para el modulo DRV8825 de motores a pasos.....	21
Figura 7. Estándar de simbología NEMA para interruptores de limite.....	22
Figura 8. Esquema de fotodiodo.....	24
Figura 9. Símbolo de fotodiodo.	24
Figura 10. Esquemático de fotodiodo conectado con polarización inversa a la fuente.....	25
Figura 11. Esquema de lazo con retroalimentación.....	27
Figura 12. Multiplexor 2:1 y su tabla de verdad.....	29
Figura 13. Esquemático de multiplexado para LEDs de 12V.....	30
Figura 14. Interruptor magnético con posiciones normalmente cerrado y normalmente abierto.....	31
Figura 15. Circuito para control de seguro eléctrico mediante un microcontrolador....	32
Figura 16. Diseño básico de fuente de alimentación DC con rectificador de onda completa y regulador de voltaje.....	33
Figura 17. Conexión de un regulador de voltaje.....	34
Figura 18. Vista frontal de la máquina en movimiento.....	62
Figura 19. Vista superior de la caja de entrega.....	63
Figura 20. Fin de carrera superior para el movimiento horizontal.....	64
Figura 21. Fin de carrera para el movimiento vertical.....	65
Figura 22. Luces de entrega exitosa.....	66
Figura 23. Luces de fallo de entrega.....	67
Figura 24. Luces indicadoras traseras.....	68
Figura 25. Ubicación del sensor de temperatura.....	69
Figura 26. Ubicación del ventilador en el prototipo.....	70
Figura 27. Ubicación de pines para shield RAMPS 1.4.....	71
Figura 28a. Esquema del circuito impreso diseñado (Parte 1).....	72
Figura 28b. Esquema del circuito impreso diseñado (Parte 2)	73
Figura 29. PCB diseñado.....	74

INTRODUCCIÓN

Justificación e Importancia del Proyecto

La falta de control en el manejo de almacenamiento y distribución de medicamentos en varios hospitales y centros médicos en el Ecuador ha llevado a que se den varios casos de robo de los mismos. Evidencia de ello es la sustracción de medicamentos del hospital del Instituto ecuatoriano de seguridad social (IESS) Teodoro Maldonado Carbo en Guayaquil en Febrero de 2015 (“Detienen sospechosos”, 2015). Además se encontraron medicamentos del IESS en una clínica clandestina en Ambato (“Medicamentos IESS”, 2015). De igual forma, en una entrevista con el Gerente General del Hospital de los Valles y profesor de la USFQ, Jaime Ocampo, se determinó que también existían problemas con el manejo de inventario en dicho hospital. Fue iniciativa de Jaime Ocampo la creación de un equipo de trabajo conformado por estudiantes para diseñar un dispensador automático de medicamentos que permita llevar de mejor manera el inventario de medicamentos y limitar el manejo de los mismos por parte de los funcionarios del hospital. En un principio el dispensador diseñado será utilizado como modelo en los sistemas médicos de la USFQ y dependiendo de su utilidad y funcionamiento se evaluará la posibilidad de producir dispensadores para ser distribuidos a varios hospitales del país.

Este es un proyecto multidisciplinario que combina el trabajo de estudiantes de ingeniería mecánica, electrónica y de sistemas de la universidad. En la Figura 1 se muestran las partes de las que está compuesto el proyecto.

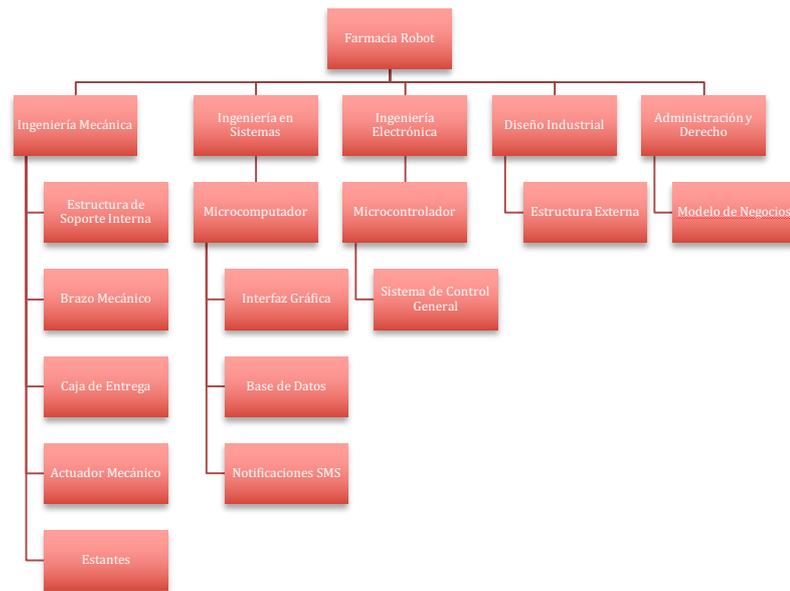


Figura 1. Modelo General del Proyecto

La parte mecánica del proyecto fue desarrollada como tesis por parte de Esteban Egüez, mientras que la parte de ingeniería en sistemas fue desarrollada por David Villacis. Además el proyecto cuenta con la asistencia de estudiantes de diseño gráfico, derecho y administración que colaboran con la parte estética de la máquina además de la parte legal y administrativa para la colocación de la máquina como producto en el mercado. Este trabajo se centra principalmente en la parte de diseño e implementación eléctrica y electrónica del dispensador.

Antecedentes

No se conoce que existan antecedentes de dispensadoras automáticas de medicamentos en los hospitales del Ecuador (Ocampo, comunicación personal, Diciembre 2014). A nivel internacional, existen varios mecanismos similares que han sido desarrollados por universidades o por empresas privadas. En el Centro Médico de la Universidad de California en San Francisco, UCSF, las medicinas almacenadas son manipuladas por un brazo robot que selecciona y empaqueta la medicinas de acuerdo a la prescripción médica de cada

paciente (Rush-Monroe, 2011). Un mecanismo similar de empaquetamiento fue diseñado por la empresa italiana Ingegneria Biomedica Santa Lucia para Leicester General Hospital, en el cual el código de barras de cada dosis está relacionado con el código de barras en el brazalete que tiene cada paciente (Buss, 2015). El dispensador del Wokingham Medical Centre tiene un funcionamiento similar, adicionalmente chequea la fecha de expiración de los medicamentos y de acuerdo a esto reordena el stock de medicinas (Nurse, 2014). Robot similares se han desarrollado en Penn State (Ingram, 2015), James Cook University Hospital (Hetherington, 2015) y Nash General Hospital (“Pharmacy robot”, 2011).

Objetivo del proyecto

Este trabajo tiene como finalidad el diseño e implementación de un sistema de control y alimentación para un dispensador automático de medicamentos mediante el uso de controladores y sensores.

Para lograr esta meta se establecieron los siguientes objetivos específicos:

1. Diseñar e implementar un sistema de control que permita el movimiento de un brazo mecánico para obtener y despachar los medicamentos almacenados.
2. Utilizar sensores para asegurar el correcto funcionamiento de la máquina.
3. Diseñar e implementar un sistema de control para mantener constante la temperatura de los medicamentos.
4. Diseñar e implementar luces indicativas para las distintas funciones de la máquina.
5. Utilizar comunicación serial para enviar y recibir instrucciones desde el computador de la máquina.
6. Diseñar un sistema de seguridad para la máquina.

DESARROLLO DEL TEMA

Modelo de Funcionamiento

El sistema de control general se implementó utilizando un microcontrolador Arduino MEGA. El microcontrolador utiliza comunicación serial para comunicarse con el microcomputador (Raspberry Pi), encargado del manejo de base de datos, interfaz gráfica y notificaciones SMS. El microcontrolador recibe los comandos desde el microcomputador y envía de vuelta mensajes de estado o de error.

El sistema de control general se divide en tres sistemas principales: sistema de entrega de medicamentos, sistema de seguridad y abastecimiento y sistema de control ambiental. El diagrama de bloques del sistema de control se muestra en la Figura 2.

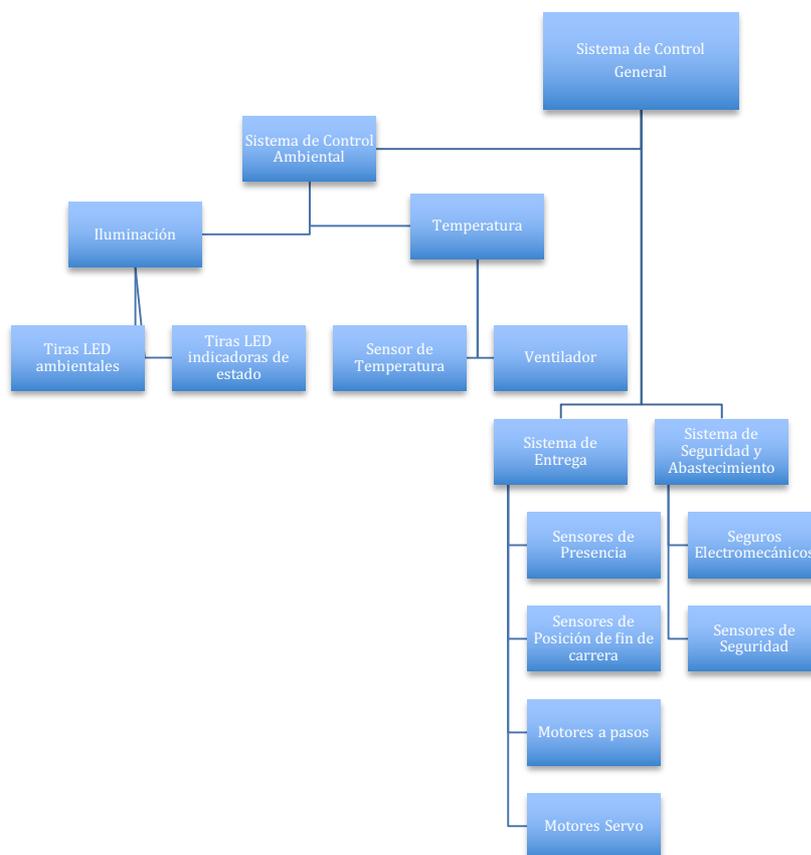


Figura 2. Diagrama de bloques del sistema de control de la máquina

El sistema de entrega de medicamentos esta compuesto por 2 motores de pasos de 12V en eje horizontal de coordenadas, 1 motores de pasos en eje vertical de coordenadas, 2 servo motores para actuador de brazo mecánico, sensores de presencia en actuador de brazo mecánico, sensores de fin de carrera horizontal y vertical. Este sistema toma los productos en las distintas posiciones según la petición que se ha ingresado por el usuario. De forma paralela a la operación del sistema de entrega de medicamento, se encuentran activos el sistema de control de temperatura y seguridad. El algoritmo del sistema de entrega de medicamentos es el siguiente:

Una vez que el suministro de energía se ha activado, el brazo mecánico procede a ubicarse en posición inicial ubicada en la esquina superior izquierda de la máquina. Durante este proceso el brazo se dirige hasta el tope mecánico horizontal y vertical, una vez que estos han sido accionados, se envía una señal de estado del microcontrolador al microcomputador que indica que la máquina se encuentra lista para operar.

Cuando el microcomputador envía la señal, el brazo mecánico procede a moverse hasta la posición indicada del medicamento. Una luz indicadora color blanco se enciende en el brazo mecánico para indicar que el mismo se encuentra en movimiento.

Cuando el brazo mecánico ha arribado a la coordenada solicitada, un actuador mecánico automatizado por un servo motor se acciona para que el medicamento caiga por acción de gravedad a la caja en el brazo mecánico. En el caso que el medicamento se encuentra dentro de la caja, una luz indicadora color verde se enciende dentro de la misma. Mientras que si existe algún problema con la posición del medicamento dentro de la caja o la misma no ha caído dentro de la caja, se enciende una luz indicadora color rojo.

Si no ha existido ningún error con el medicamento dentro de la caja, el brazo mecánico se dirige hasta la coordenada de posición de entrega de medicamento donde se

deposita el producto para que el usuario lo pueda recoger. Para aquello se acciona un compartimiento de la caja por medio de un servo motor de tal forma que el mismo se abra cuando el brazo mecánico ha llegado a la coordenada de entrega de medicamento. Si ha existido un problema con el medicamento dentro de la caja, el actuador mecánico hará un par de intentos para reacomodar el medicamento. Si no lo logra, se envía una señal de error de entrega desde el micro controlador y se reinicia el algoritmo de entrega. El microcomputador se encargará en este caso de alertar al usuario de que existe un problema con la entrega de ese medicamento, se procederá a continuar con el proceso de entrega del siguiente medicamento y se enviará una notificación vía SMS al supervisor de la máquina para la revisión del problema.

Una vez culminado el proceso de entrega, el algoritmo de entrega se reinicia para esperar una nueva petición de entrega de medicamento hasta el punto de selección de medicamento por parte del usuario.

El sistema de seguridad y abastecimiento está compuesto por un sensores magnéticos y un seguro electromecánico. El sistema de seguridad se encuentra en estado “stand by” mientras el sensor magnético ubicado en la puerta posterior de la maquina no se accione. Esto quiere decir que el sistema de entrega de medicamento puede operar de forma habitual. Si el sensor magnético se acciona debido a una apertura no autorizada de la puerta posterior, el sistema de entrega de medicamento se pausa y se envía una alerta inmediatamente al microcomputador. Este encenderá una alarma sonora y alertará visualmente o remotamente al usuario. El sistema de abastecimiento permite abrir el seguro eléctrico posterior de la máquina ante el pedido del microcomputador. Una vez abierto, el microcomputador enviará una señal al microcontrolador de la posición en la que se debe colocar el medicamento. El controlador encenderá una luz indicadora en la posición deseada. El operador deberá ingresar

al computador el número de medicamentos que va a colocar y deberá alertar al mismo cuando haya finalizado la colocación de un medicamento.

El sistema de control ambiental esta compuesto por: Tiras LED color azul, blanco, rojo y verde, sensor de temperatura LM35 y un ventilador de 12V. El sistema de temperatura verifica los valores de temperatura registrados por el sensor de temperatura y los compara con el valor de temperatura ideal. A partir de la diferencia de estos valores el sistema realiza un control PID y acciona un ventilador para el proceso de control de temperatura. La verificación de la temperatura se la hace continuamente si no existe otro requerimiento por parte del microcomputador, caso contrario el control de temperatura se lo realiza antes de procesar cada una de las acciones recibidas. El sistema de iluminación controla las luces indicativas para los procesos de entrega y abastecimiento. Así mismo permite el control por comando de las luces ambientales.

Marco Teórico

Motores a Pasos. En mecanismos donde se requiere una alta precisión de movimiento tales como máquinas de coordenadas o impresoras, los motores de paso son ideales dado que es posible moverlos un paso a la vez por cada pulso que se aplique. (Velasco, 2003)

Existen varios tipos de motores de pasos en el mercado, “Los motores a pasos vienen en una amplia variedad de diseños y configuraciones. Además de las configuraciones de reluctancia variable también existen configuraciones de imán permanente e híbridas.” (Fitzgerald, 2005). En términos generales este tipo de motor se elabora con el fin de producir un gran número de pasos por revolución con la mayor precisión posible, esto permite que sean empleados en sistemas de control digital de lazo abierto tales como impresoras y

graficadores. (Fitzgerald, 2005). Entre los más comunes se pueden encontrar de 50, 100 o 200 pasos por revolución, lo que equivale a 7.2, 3.6 o 1.8 grados por paso respectivamente. Las características esenciales del motor tales como el número de pasos o tamaño del mismo se encuentran descritos bajo la norma americana NEMA (National Electrical Manufacturers Association). (RepRap, 2014)

Como ya se dijo anteriormente, existen varias configuraciones de motores de pasos. Un motor de pasos de máquina de reluctancia variable, que se abrevia como VRM por sus siglas en inglés, está formado por un estator con bobinado de excitación con fases múltiples y rotor magnetizado. Dado que el torque es generado por la búsqueda de alineamiento del rotor con la onda de flujo magnético producida por parte del estator no es necesario que el rotor posea conductores. Los motores a pasos con esta configuración aprovechan el hecho que cuando una VRM es energizada en secuencia de una manera gradual apropiada, la máquina entonces gira un ángulo específico por paso. Existe también una variación de este tipo de configuración que se conoce motor de pasos de reluctancia variable de escalonamiento múltiple. En este caso el motor está compuesto por un conjunto de VRM monofásicas axialmente desplazadas y montadas en un eje único. (Fitzgerald, 2005)

Por otra parte también existe la configuración de motor de pasos bipolar con imán permanente y rotor bipolar. Este tipo de máquina es muy similar a una máquina de corriente alterna de imán permanente trifásica. No obstante a diferencia de la máquina de CA que impulsa una carga a una velocidad específica, el motor a pasos está diseñado para controlar la posición de una carga. La secuencia necesaria para que el rotor gire un ángulo, o se produzca un paso es la siguiente:

1. Corriente positiva en la fase 1
2. Corriente positiva de igual magnitud en ambas fases.

3. Corriente positiva en la fase 2
4. Corriente de igual magnitud en fase 1 y corriente positiva en fase 2.
5. Corriente negativa en fase 1
6. Repetir paso 1

Un motor de este tipo también puede operar de forma unipolar si las bobinas son excitadas alternadamente de manera independiente. (Fitzgerald, 2005). En la Figura 3 se muestra un diagrama de un motor a pasos con imán permanente de tipo bifásico.

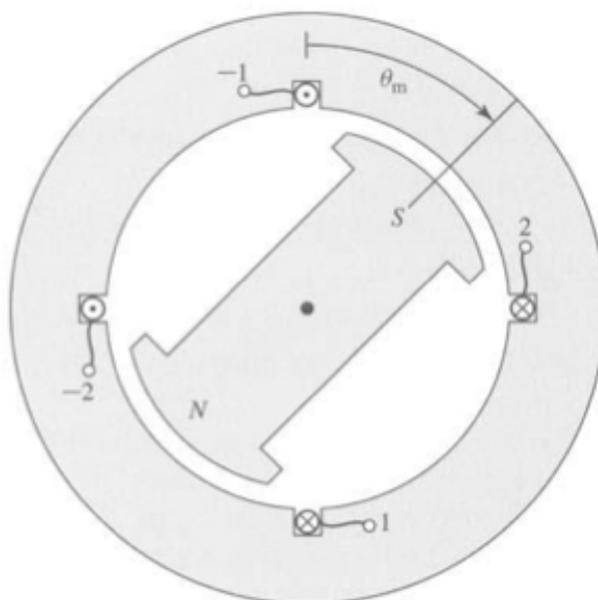


Figura 3. Diagrama esquemático de un motor de pasos de imán permanente bifásico. Tomado de Máquinas Eléctricas, Fitzgerald, p. 439

El torque para un motor de pasos con imán permanente como se muestra en la figura anterior esta dado por:

$$T_{mec} = T_0(i_1 \cos \theta_m + i_2 \sen \theta_m) \quad (1)$$

Donde T_0 es un valor constante que esta dado por las propiedades físicas del imán permanente y la geometría del motor. (Fitzgerald, 2005). El Torque producido por un motor de pasos es proporcional a la intensidad de campo magnético del devanado del estator, por

tanto para incrementar el torque del motor es necesario o añadir mas devanados o incrementar la corriente. (SGS-Thomson, 1995).

No obstante se debe tener mucho cuidado al momento de incrementar la corriente en los devanados del estator dado que se puede alcanzar una saturación del núcleo y también aumentar la temperatura del motor. Por tanto se debe aplicar un máximo de corriente según las especificaciones dadas por el fabricante del motor. Es claro que el circuito de control del motor requerirá un limitador de corriente. (SGS-Thomson, 1995).

Finalmente existe la configuración de motor de pasos híbrido. Esta última es una combinación de un motor de pasos con configuración de imán permanente y uno de reluctancia variable. En la Figura 4 se muestra un diagrama de un motor con esta configuración.

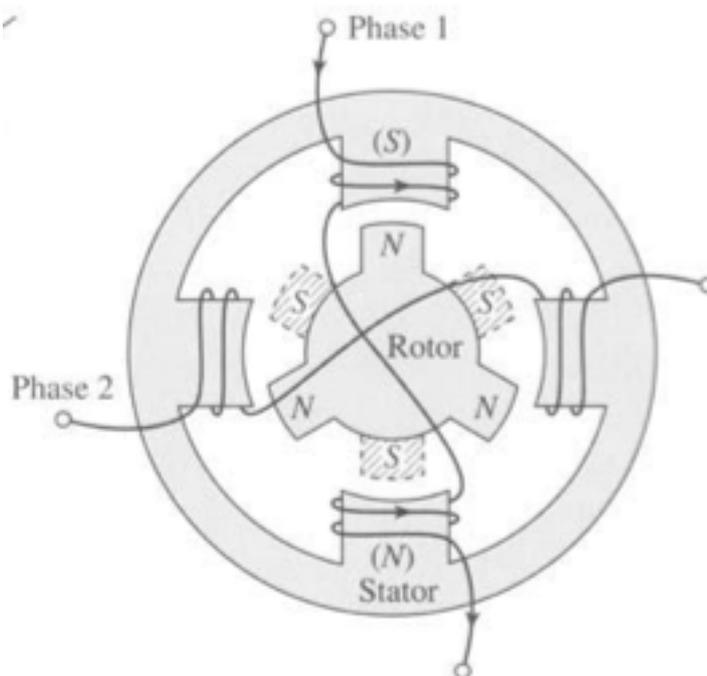


Figura 4. Diagrama esquemático de motor de pasos híbrido. Tomado de Máquinas Eléctricas, Fitzgerald, p. 444

En este tipo de configuración existen dos escalonamientos de rotor idénticos que están angularmente desplazados de forma axial una distancia de medio paso del polo del rotor.

Además los escalonamientos del rotor están separados por un imán permanente. Para el diagrama que ese muestra en la Figura 2 el estator posee un total de cuatro polos, dos por cada fase y ubicados de forma frontalmente opuesta. Como se puede observar cuando la fase 1 es excitada por una señal de entrada, de tal forma que el polo superior del estator es un polo sur al mismo tiempo que el inferior es norte. “La excitación del estator interactúa con el flujo del imán permanente del rotor para alinearlo con un polo en su extremo de polo norte verticalmente hacia arriba y un polo en el entre de su polo sur verticalmente hacia abajo.” (Fitzgerald, 2005).

En la Figura 5 se muestra un diagrama del bobinado de un motor de pasos bipolar:

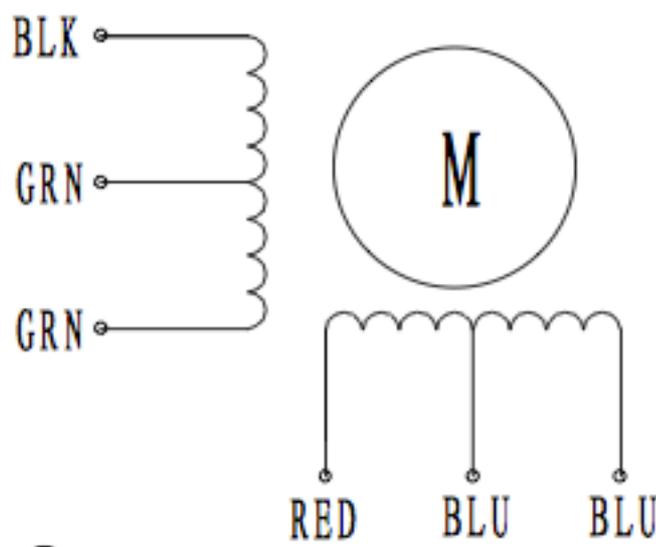


Figura 5. Diagrama de devanados con código por color de motor de paso híbrido. Tomado de Pololu Resources.

Como se puede observar de la figura anterior, el motor híbrido puede conectarse como bipolar o unipolar dependiendo de si los cables centrales de las bobinas se toman como punto común en la conexión (bipolar) o sencillamente no se los toma en cuenta (unipolar).

Las principales ventajas que ofrece un motor con este tipo de configuración son que se requiere menor excitación en el estator para la producción de torque y que el motor tiende a mantener su posición cuando se elimina dicha excitación. (Fitzgerald, 2005). Sin embargo

para una correcta operación del motor y evitar que se produzca un salto de paso se debe considerar que los motores deben vencer ciertas fuerzas inerciales además que el tiempo y frecuencia de los pulsos permitan al motor alcanzar el paso antes que se envíe un nuevo pulso. (Velasco, 2003) “Si la frecuencia de pulsos es muy elevada, el motor puede reacción de la siguientes formas: Puede que no realice ningún movimiento en absoluto, puede comenzar a vibrar pero sin llegar a girar, puede girar erráticamente o puede llegar a girar en sentido opuesto” (Velasco, 2003)

Controlador de motor de pasos. Como se ha podido ver los motores de pasos requieren una señal específica para su correcta operación. Por tanto el uso de un circuito que maneje esta señal, conocido como “driver” es necesario. “Los circuitos integrados han simplificado dramáticamente el manejo de motores de pasos” (SGS-Thomson, 1995). Un controlador típico para un motor de pasos esta compuesto por uno o varios puente H, un circuito limitador de corriente, transistores de tipo MOSFET y opcionalmente un indexador para micro-pasos (Texas Instruments, 2014).

Uno de los controladores con mayor uso en el mercado para motores a pasos es el DRV8825 de Texas Instruments. El controlador permite controlar un motor a pasos bipolar de hasta una corriente de salida de 2.2A por bobina. Además el controlador permite realizar 6 distintos tipos de resolución de pasos: paso completo, medio paso, $\frac{1}{4}$ de paso, $\frac{1}{8}$ de paso , $\frac{1}{16}$ de paso y $\frac{1}{32}$ de paso (Pololu Resources, 2015). El esquema de conexión del módulo para el driver DRV8825 se muestra en la Figura 6.

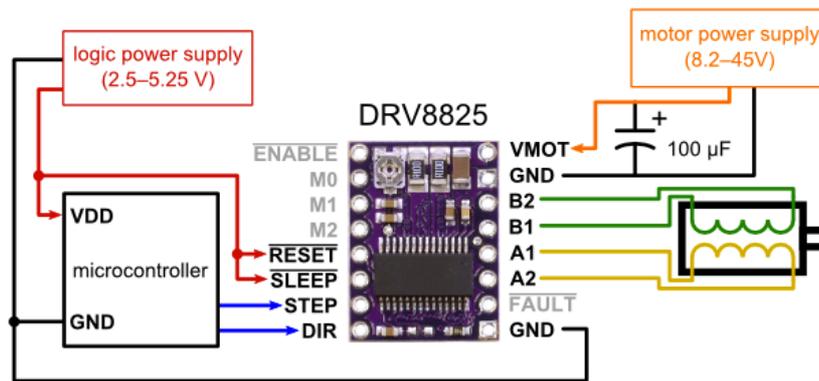


Figura 6. Diagrama de cableado mínimo para el módulo DRV8825 de motores a pasos. Tomado de:

Pololu Resources

Servomotores. Un motor servo es un dispositivo utilizado para posicionar superficies de control de forma angular. Por esta razón un servo está compuesto por un eje de rendimiento controlado, lo que quiere decir que su posición angular puede ser controlada por una señal codificada. Mientras esta señal se mantenga el motor servo mantiene la posición indicada. (El servomotor, 2015). “El ángulo está determinado por la duración de un pulso, a esto se le llama modulación codificada de pulsos” (El servomotor, 2015).

También es posible controlar la velocidad a la que gira un motor servo: “La cantidad de voltaje aplicado al motor es proporcional a la distancia que éste necesita viajar. Así, si el eje necesita regresar una distancia grande, el motor regresará a toda velocidad. Si este necesita regresar sólo una pequeña cantidad, el motor correrá a una velocidad más lenta.” (El servomotor, 2015)

Sensores mecánicos. Los sensores mecánicos o interruptores de límite se utilizan para detectar la presencia de un objeto en un punto en particular. Pueden ser activados por el movimiento de una maquinaria o por la presencia o ausencia de un objeto en específico. Los sensores mecánicos poseen un brazo de parachoques que varía según el tipo de aplicación.

Cuando el parachoques es accionado por la presencia de un objeto los contactos al interno del sensor cambian de posición, lo cual permite enviar una señal de presencia o no presencia de un objeto en determinado punto. (Herman, 2009)

El estándar NEMA de simbología para sensores mecánicos se muestra a en la Figura 7.

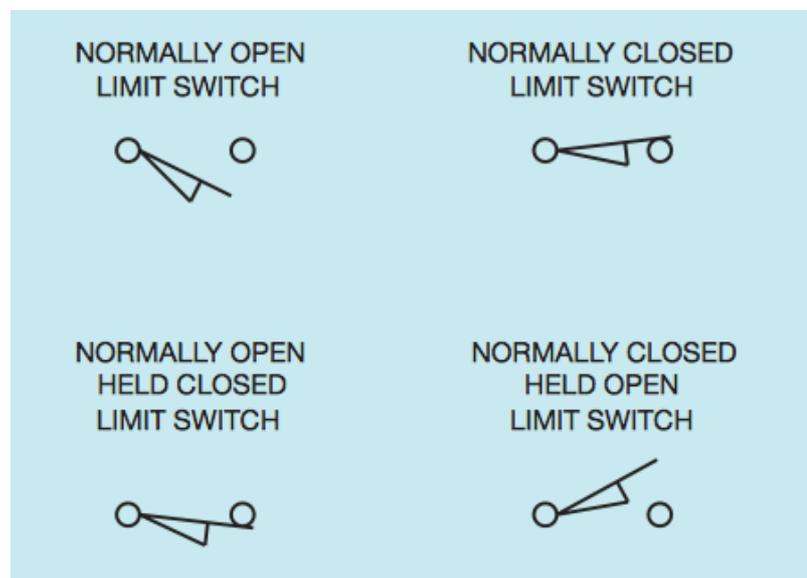


Figura 7. Estándar de simbología NEMA para interruptores de límite. Tomado de Herman.

Para aplicaciones donde se disponga de espacio limitado para los sensores se utiliza micro interruptores de límite. Además de su diseño compacto, este tipo de interruptor requiere una menor distancia de accionamiento en el embolo de activación, equivalente a el parachoques en los interruptores normales. Para ser accionado el embolo necesita ser desplazado únicamente 0.015 pulgadas aproximadamente. Además existen micro interruptores que disponen de un resorte para el retorno a su posición inicial después de ser accionados. (Herman, 2009)

Microcontrolador. Un microcontrolador es un chip que contiene en su interior un procesador (CPU), memoria no volátil para el programa (ROM o flash), memoria volátil para entradas y salidas (RAM), un reloj y una unidad de control de entradas y salidas (“Definition of: microcontroller”, s.f.).

Para este proyecto se utilizó una tarjeta Arduino Mega 2560, que contiene un microcontrolador ATmega2560 (“Arduino MEGA 2560”, s.f.). Cuenta con 54 entradas/salidas digitales, 16 entradas análogas, un cristal oscilador de 16MHz, memoria flash de 256 KB, SRAM de 8 KB y memoria EEPROM de 4KB (“Arduino MEGA 2560”, s.f.). Tiene un voltaje de operación de 5V y se alimenta con un voltaje de 6-20V (“Arduino MEGA 2560”, s.f.). Incluye un *bootloader*, un pedazo de *firmware* del microcontrolador que permite cargar nuevo código sin la necesidad de un programador de hardware externo (“Arduino MEGA 2560”, s.f.). El microcontrolador ATmega2569 incluye también cuatro UARTs para la comunicación serial TTL (*transistor-transistor logic*) de 5V. Un UART es un dispositivo periférico que permite la comunicación serial entre dos sistemas con una velocidad de transferencia de datos preestablecida (Harris & Harris, 2013).

Las tarjetas de marca Arduino se caracterizan por la posibilidad de utilizarlos junto con un “shield”. Un “shield” es una tarjeta que se conecta encima de la tarjeta Arduino para extender las capacidades de la misma (“Shields”, s.f.). Para este proyecto se utilizó un RepRap Arduino Mega Pololu Shield (abreviado RAMPS), el cual está diseñado para controlar impresoras 3D RepRap mediante el uso de controladores Pololu para motores de pasos (“Arduino Mega Pololu Shield”, s.f.). La facilidad que otorga este “shield” para trabajar con motores de pasos es la razón por la que se decidió utilizarlo.

Sensores de Presencia. Un fotodiodo esta conformado por un bolsillo de material dopado tipo P, el cual es fusionado a un substrato de tipo N de tal forma que se forma una unión tipo PN. El material tipo P tiene exposición directa a la luz por medio de una ventana. (Cox, 2001). En la Figura 8 se muestra la unión tipo PN de un fotodiodo y su ventana.

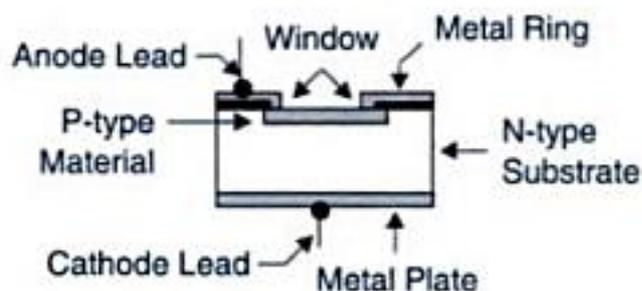


Figura 8. Esquema de fotodiodo. Tomado de Cox. J.

En la Figura 9 se muestra el símbolo utilizado para un fotodiodo en circuitos:



Figura 9. Símbolo de fotodiodo. Tomado de Cox. J

Se debe considerar que un fotodiodo esta diseñado para operar en polarización inversa. Esto significa que la parte dopada P debe conectarse al polo negativo de la fuente, mientras que la zona dopada N debe conectarse con el polo positivo. (Cox, 2001) Este tipo de polarización y el fotodiodo se muestra en el esquemático de la Figura 10.

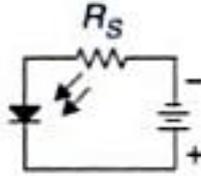


Figura 10. Esquemático de fotodiodo conectado con polarización inversa a la fuente. Tomado de Cox, J.

Cuando se realiza polarización inversa en la unión tipo PN, la misma se encuentra con polarización directa para los portadores minoritarios. Si se considera que la cantidad de corriente que fluye por la unión es una función del número portadores minoritarios presentes, y que en un fotodiodo la cantidad de portadores minoritarios depende directamente de la cantidad de luz que atraviesa la ventana, entonces se puede concluir que la corriente es una función directa de la luz que alcanza al fotodiodo. En otras palabras, a mayor intensidad de luz existe una menor resistencia de la polarización inversa. (Cox, 2001).

Luces LED. Existen varios tipos de LED en el mercado que pueden ser utilizados para la iluminación que varían según las características ambientales y otros factores. En el caso de las tiras de LED la sigla “SMD significa Surface Mounted Device o Dispositivo de Montaje Superficial” (Ledbox, 2012). Además existen varias denominaciones de SMD que especifican las propiedades técnicas del dispositivo, o LED para el caso. Para el presente análisis se realiza un enfoque especial a las tiras LED de tipo SMD3528.

Una tira LED SMD3528 se caracteriza por un consumo bajo y por ser de media potencia. Son las tiras de menor consumo que se pueden encontrar en el mercado, dado que su consumo promedio (W/metros) es muy bajo. Son de formato monocolor y además son la solución más económica para decoración de ambientes. (Ledbox, 2012).

A continuación se muestra una tabla donde se resume las principales características de las tiras LED, incluyendo las tipo SMD3528.

Tabla 1

Denominaciones y características de tiras LED SMD. Tomado de: Ledbox, 2012

	Led/metro	W/metro	Lm/metro	Tamaño
SMD3528	60 Led/m	4,8W/m	4Lm/m	9x6mm
SMD3014	120 Led/m	12W/m	8Lm/m	10x6,5mm
SMD5050	60 Led/m	14,4W/m	12Lm/m	10x6,5mm
SMD5050 High Power	60 Led/m	14,4W/m	20Lm/m	14x7mm

Como se puede observar en la Tabla 2, las tiras de LED SMD3528 consumen a penas 4,8W/m, es decir menos del 50% de consumo que las SMD3014. Por otra parte, como era de esperarse, su luminosidad es de 4Lm/m. Es decir a penas la mitad de luminosidad que las SMD3014. (Ledbox, 2012)

Sensores de Temperatura. En varias aplicaciones industriales es de vital importancia conocer el valor de temperatura. Un sensor de temperatura es un elemento diseñado de tal forma que una variable electrónicamente medible varíe según los cambios de temperatura. El método utilizado para medir temperatura es seleccionado según la precisión requerida y rangos de operación. Entre los métodos más conocidos están: expansión de metal, variación de resistencia, expansión por presión y otros sensores de estado solido. (Herman, 2009)

El LM35 es un circuito integrado diseñado para la medición precisa de temperatura. El voltaje de salida es linealmente proporcional con la temperatura en grados centígrados. La

ventaja con otros sensores con calibración en grados Kelvin es que no es necesario restar una constante al voltaje de salida para obtener la temperatura en escala de centígrados. El LM35 no requiere ningún tipo de calibración externa y posee una precisión de $\frac{1}{4}$ C a temperatura ambiente y $\frac{3}{4}$ C en valores cercanos a los -55 C y 150 C. Por otra parte el circuito es de bajo consumo en potencia y ocupa solamente $60\mu\text{A}$ de la fuente. (Texas Instruments LM35 Technical Description, 1999)

Control PID. El control PID es ampliamente utilizado para procesos industriales donde se requiere retroalimentación. Para aplicar un control PID adecuado se debe tener conocimiento sobre el sistema y decidir cual es la acción de control apropiada. Luego se proceda a ajustar los parámetros de constantes: proporcional, derivativa e integral, para estabilizar el sistema. (Araki, 2000)

El esquema de un lazo de control con retroalimentación se muestra en la Figura 11.

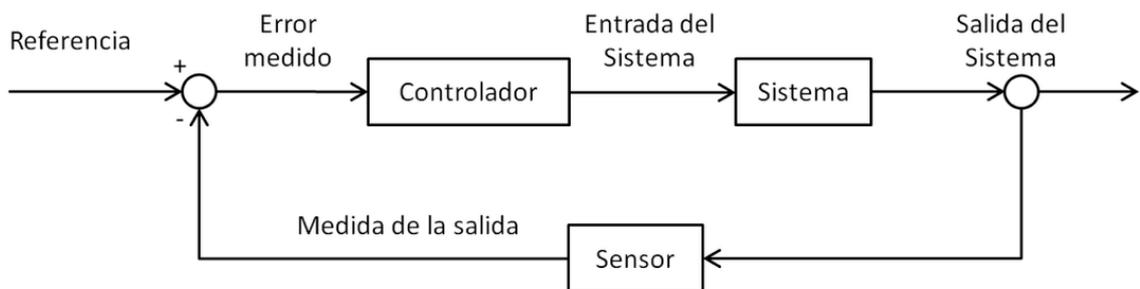


Figura 11. Esquema de lazo con retroalimentación. Licencia GNU de distribución libre. 2015.

Como se puede ver en la figura anterior, el error es igual al valor de referencia restado el valor medido por el sensor en la planta. La entrada del sistema pasa por una regla computacional en el controlador la cual se ajusta según los parámetros de diseño (Akai, 2000). Para el caso de un controlador PID los tres parámetros son los siguientes:

Parámetro P: Proporcional al error medido en el instante t , el cual es el error actual.

Parámetro I: Proporcional al integral del error medido hasta el instante t , es decir la acumulación de error en un periodo determinado de tiempo.

Parámetro D: Proporcional a la derivada del error en el instante t , es decir es la razón de cambio del error. (Akai, 2000)

Uso de Transistores como Interruptores. Un multiplexor, también conocido como “mux”, es un componente electrónico que selecciona como salida a una y solamente una de las múltiples señales de entrada según una señal de control. (Dean, 2010). El tipo de multiplexor se lo define según el número de entradas. Por ejemplo un multiplexor 2^n , posee 2^n entradas y requiere de n señales de control. (Debashis, 2010). Multiplexores de mayor tamaño pueden ser construidos ya sea utilizando técnicas lógica de suma de productos o una combinación de varios multiplexores más sencillos. (Harris, 2013). En la Figura 12 se muestra el esquemático para un multiplexor 2:1 y de forma adjunta su tabla de verdad respectiva.

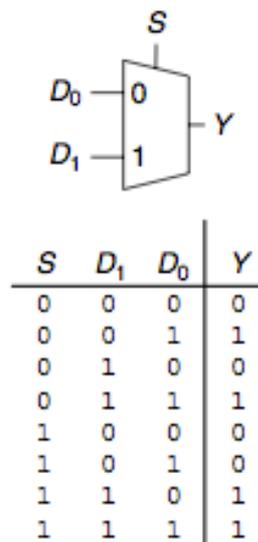


Figura 12. Multiplexor 2:1 y su tabla de verdad. Tomado de Harris & Harris. (2013).

En la Figura 12, S corresponde a la señal de control. D1 y D0 son las entradas y la salida del multiplexor.

La técnica de multiplexado puede ser utilizada para manejar grandes cantidades de LEDs. Para el caso en el cual se requiera controlar LEDs que operen con alimentación 12 voltios se puede hacer uso de transistores. Suponiendo el caso de que se van a controlar 25 LEDs es necesario utilizar 10 transistores, 5 del tipo NPN y 5 PNP. (García, 2015) La diagramación y circuito se muestra en la Figura 13.

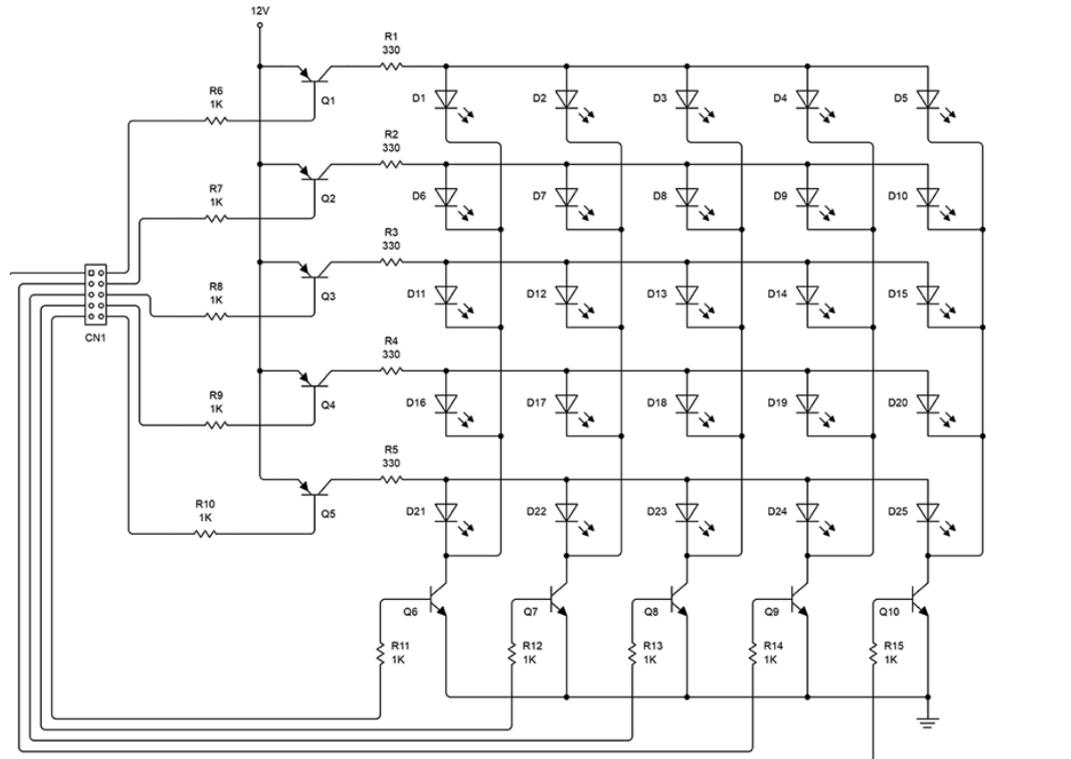


Figura 13. Esquemático de multiplexado para LEDs de 12V. Tomado de García, A.

Un MOSFET es un tipo de transistor compuesto por una unión Metal-Oxido. Son conocidos como Metal Oxide Field Effect Transistor. La principal ventaja de utilizar MOSFET es que requieren una corriente muy pequeña en su compuerta para activarlo. Y una vez activo, este puede entregar hasta 50 amperios para la carga. (The MOSFET, 2010)

Sensor magnético. Sensores de ventana pueden ser utilizados para un sistema de seguridad y alarma. Estos sensores consisten en un interruptor con dos contactos. Cuando un imán permanente se encuentra suficientemente cercano el interruptor cambia su posición al contacto normalmente cerrado, mientras que cuando el imán se aleja el interruptor regresa al contacto normalmente abierto. (LaLena, 2015) El sistema mencionado se muestra en la Figura 14.

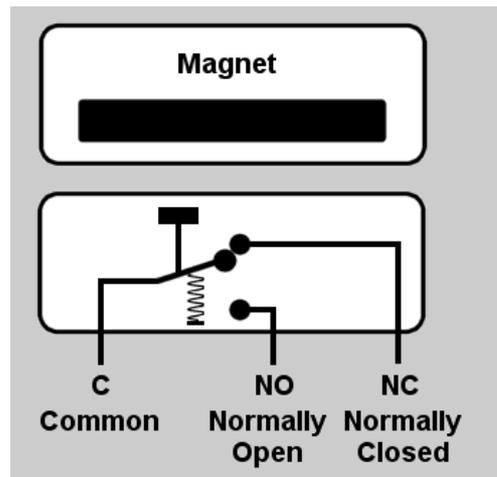


Figura 14. Interruptor magnético con posiciones normalmente cerrado y normalmente abierto. Tomado de LaLena.

Seguro Eléctrico. Un tipo de seguro eléctrico que puede ser activado con voltaje DC es aquel que es basado en solenoides. Un solenoide es una bobina hecha de hilo conductor que es capaz de producir un campo magnético (“Solenoid”, s.f.). En un seguro de este tipo el solenoide actúa como un electroimán que, cuando es energizado, atrae al seguro de metal hacia su centro permitiendo que se abra la puerta (“Lock-style Solenoid - 12VDC”, s.f.). Si el solenoide no está energizado, la puerta permanece cerrada y no se consume energía (“Lock-style Solenoid - 12VDC”, s.f.). Un seguro de este tipo utiliza alrededor de 650 mA a 12 V y tiene un tiempo de activación de 1 a 10 segundos (“Lock-style Solenoid - 12VDC”, s.f.). Para su implementación, se necesita utilizar un transistor de potencia y un diodo de protección (“Lock-style Solenoid - 12VDC”, s.f.), como se muestra en la Figura 15.

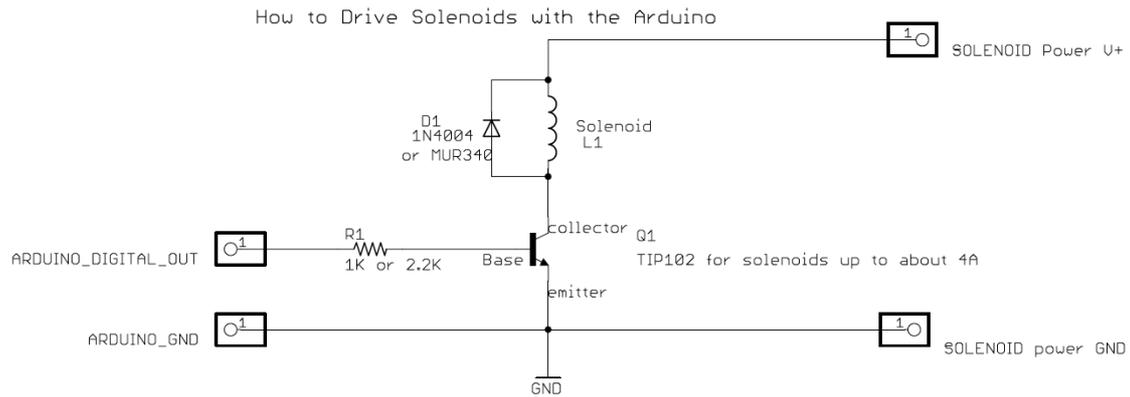


Figura 15. Circuito para control de seguro eléctrico mediante un microcontrolador. Tomado de “How to Drive Solenoids with the Arduino”. (s.f.).

Fuente de Alimentación. “Virtualmente todo equipo electrónico como computadoras, calculadores, TV e instrumentos son alimentados por una fuente de poder DC, ya sea una batería o una fuente de alimentación DC. La mayoría de estos equipos no solamente requieren un voltaje directo si no también un voltaje adecuadamente filtrado y regulado” (Texas Instruments, 2004).

Los equipos de conversión de energía eléctrica son clasificados en tres categorías: convertidor DC/DC, fuente de alimentación AC/DC y inversor DC/AC. Una fuente de alimentación AC/DC debe realizar las siguientes funciones: Rectificación, transformación de voltaje, filtrado, regulación, aislamiento y protección. (Texas Instruments, 2004). Sin embargo es posible realizar diseños más sencillos.

El diseño básico de una fuente de alimentación DC consiste en un rectificador de onda completa, compuesto por un arreglo de diodos, y un regulador de voltaje, para el cual se puede utilizar un circuito integrado como por ejemplo el LM317. (DuBois, 2009). El diseño se muestra en la Figura 16.

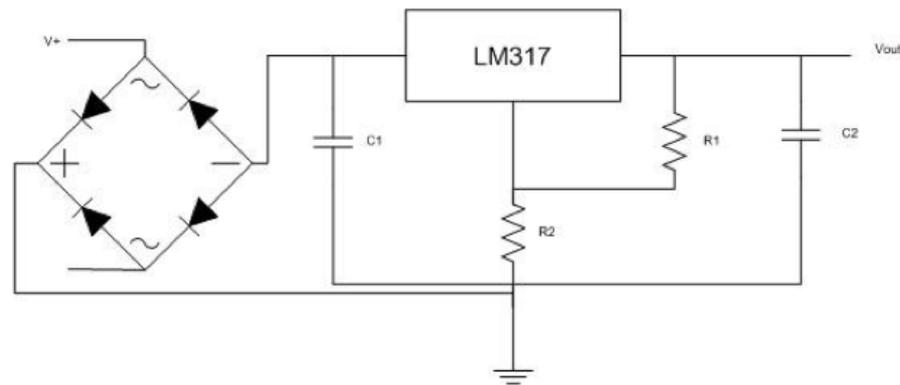


Figura 16. Diseño básico de fuente de alimentación DC con rectificador de onda completa y regulador de voltaje. Tomado de DuBois.

Reguladores de Voltaje. En términos generales, un regulador de voltaje es un circuito que provee una salida de voltaje constante independiente de las fluctuaciones de voltaje en la entrada (Poornachandra, Sasikala, y Khan, 2005). Un regulador de voltaje de circuito integrado tiene circuitos de fuente de referencia, amplificador comparador, dispositivo de control y de protección contra sobrecarga dentro del mismo integrado (Boylestad & Nashelsky, 2009).

Los reguladores de la serie 78XX proporcionan voltajes constantes de 5V a 24V (Boylestad & Nashelsky, 2009). El circuito integrado se conecta como se muestra en la Figura 17. Se utilizan capacitores para filtrar la señal de entrada y la de salida (Boylestad & Nashelsky, 2009).

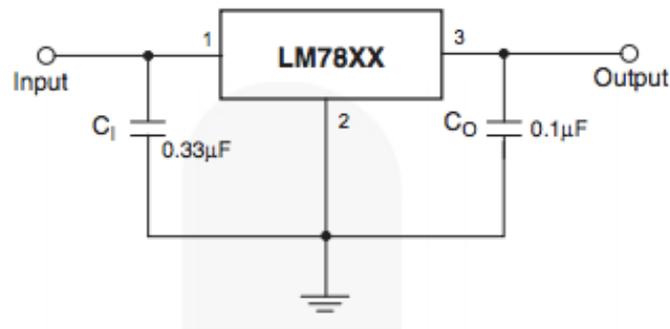


Figura 17. Conexión de un regulador de voltaje. Tomado de “3-Terminal 1 A Positive Voltage Regulator”, 2014

Descripción del diseño

Descripción General. El sistema electrónico encargado del movimiento y control general de la máquina está compuesto por tres partes principales:

1. Un microcontrolador Arduino Mega con un “shield” RAMPS 1.4.
2. Una circuito impreso que conecta las entradas/salidas del Arduino con los actuadores y sensores de la máquina.
3. Actuadores y sensores que permiten el funcionamiento de la máquina.

La ubicación de los pines de entrada y salida del Arduino con respecto al “shield” RAMPS 1.4 se muestran en la Figura 27 del Anexo B.

El esquema del circuito impreso se muestra en la Figura 28 del Anexo B. El circuito está alimentado por una fuente de poder de 12VDC. Se utilizan además reguladores 7805 y 7806 para alimentar con 5VDC y 6VDC, respectivamente, a los sensores y actuadores que

necesitan de alimentación con estos voltajes. En la Tabla 2 se detalla la función de cada una de las terminales del circuito, junto a donde van conectadas.

Tabla 2

Lista y descripción de terminales del circuito diseñado.

Terminal	Conectada a	Descripción
MOTXOUT	RAMPS driver eje Z pines 2B,2A,1A,1B	Señal de salida para el movimiento horizontal de motores.
MOTYOUT	RAMPS driver eje X pines 2B,2A,1A,1B	Señal de salida para el movimiento vertical de motores.
MOTUP	Motor mov. eje X superior	Señal para el movimiento de bobinas del motor horizontal superior.
MOTDOWN	Motor mov. eje X inferior	Señal para el movimiento de bobinas del motor horizontal inferior.
MOTY	Motor mov. Eje Y	Señal para el movimiento de bobinas del motor vertical.
TEMPIN	Microcontrolador pin A13	Señal de entrada de temperatura del sistema.
TEMP	Sensor LM35	Alimentación y medición del sensor de temperatura.
MGSECIN	Microcontrolador pin 19	Señal de entrada del estado de sensor magnético.
MGSEC	Seguro magnético	Conexión del sensor magnético de la puerta trasera al circuito lógico.
ENDSTOPINX	Microcontrolador pin 18	Señal de entrada de fines de carrera horizontales.
ENDSTOPINY	Microcontrolador pin 3	Señal de entrada de fin de carrera vertical.
ENDSTOPX	Fines de carrera eje X (en paralelo)	Conexión de los fines de carrera del eje X al circuito lógico.
ENDSTOPY	Fines de carrera eje Y	Conexión del fin de carrera del eje Y.
SERVOSOUT	Microcontrolador pin 11 y pin 6	Señales de salida para el movimiento de servomotores.
SERVOACT	Servomotor actuador	Alimentación y señal de control para servomotor de actuador.
SERVOCOM	Servomotor compuerta	Alimentación y señal de control para servomotor de compuerta.
IR LED	LED infrarrojo	Alimentación LED infrarrojo.
FOTOIN	Microcontrolador pin A15	Señal de entrada del sensor de presencia.
FOTOD	Fotodiodo infrarrojo	Conexión del fotodiodo (sensor de presencia) al circuito lógico.
REDLEDOUT	Microcontrolador pin 25	Señal de salida para control de luces indicativas de error de entrega.
REDLED	LEDs rojos (caja de entrega)	Conexión de luces de error al circuito de prendido y apagado.
GREENLEDOUT	Microcontrolador pin 23	Señal de salida para control de luces indicativas de entrega exitosa.
GREENLED	LEDs verdes (caja de entrega)	Conexión de luces de entrega exitosa al circuito de prendido y apagado.
BLUELEDOUT	Microcontrolador pin 27	Señal de salida para control de luces laterales.

BLUELED	LEDs azules (laterales)	Conexión de luces laterales al circuito de prendido y apagado.
FANOUT	Microcontrolador pin 29	Señal de salida para control de ventilador.
FAN	Ventilador	Conexión de ventilador al circuito de control.
LOCKOUT	Microcontrolador pin 31	Señal de salida para control de seguro eléctrico.
LOCK	Seguro eléctrico	Conexión de seguro eléctrico de la puerta trasera al circuito de control.
WHITELEDOUT	Microcontrolador pin 33	Señal de salida para control de luz indicativa de mov. del brazo
WHITELED	LEDs blancos (brazo)	Conexión de luces de mov. del brazo al circuito de prendido y apagado.
BLIGHTSOUT1	Microcontrolador pines 35,37,39,41,43,45,47,32,5,4	Señales de salida para control por filas de luces indicativas de abastecimiento
BLIGHTSOUT2	Microcontrolador pines 1,0,63,59,40,64,42,44,65,66	Señales de salida para control por columnas de luces indicativas de abastecimiento
BLIGHTSPLUS	Luces traseras (positivo)	Conexión del contacto positivo de las luces traseras al circuito de control.
BLIGHTSMINUS	Luces traseras (negativo)	Conexión del contacto negativo de las luces traseras al circuito de control.

El código utilizado para la programación del Arduino Mega se muestra en el Anexo C.

El programa se encuentra dividido en funciones de acuerdo a cada una de las acciones que realiza la máquina y que se detallan en las secciones siguientes.

Los sensores y actuadores de la máquina se encuentran ubicados de la siguiente manera:

- Dos motores de pasos ubicados en el borde derecho de la máquina encargados del movimientos horizontal del brazo mecánico.
- Un motor de pasos en la parte superior del brazo mecánico que permite el movimiento vertical de la caja de entrega.
- Un diodo infrarrojo y un fotodiodo en el interior de la caja actuando como sensor de presencia.

- Dos fines de carrera en el borde izquierdo de la máquina para limitar el movimiento horizontal del brazo y un fin de carrera en la parte superior del brazo para limitar el movimiento de la caja.
- Dos servomotores en la caja de entrega, uno para activar el actuador y otro para abrir la compuerta de entrega.
- Un sensor de temperatura y un ventilador en la parte superior de la máquina para el control de temperatura.
- Luces azules en los costados de la máquina para iluminación general, luces blancas en la parte trasera del brazo para indicación de movimiento, luces rojas y verdes dentro de la caja de entrega para alerta de error o entrega exitosa, luces verdes en la parte trasera para indicación de abastecimiento.
- Un sensor magnético y un seguro eléctrico en la puerta trasera de la máquina para control de seguridad.

Comunicación con el computador. La comunicación entre el microcontrolador utilizado y la computadora interna de la máquina se hace mediante comunicación serial por medio de un módulo TTL-USB PL2303HX. Para una comunicación efectiva entre las partes, se desarrolló la codificación serial que se muestra en la Tabla 3. Una explicación más detallada de la función que cumplen cada una de estas instrucciones se encuentra en las secciones siguientes.

Tabla 3

Codificación de mensajes entre el microcontrolador y el microcomputador.

Código HEX	Instrucción	Receptor	Descripción	Parámetro	Ejemplo Mensaje Final
x1	Encendido del sistema	Computador	Mensaje de inicio cuando se prende el microcontrolador.		x1x0
x2	Sistema listo	Computador	El microcontrolador está listo para recibir instrucciones.		x2x0
x3	OK (Orden Recibida/Ejecutada)	Computador	Instrucción ejecutada (o recibida en el caso de movimiento de motores).		x3x0
x4	Notificación de Temperatura	Computador	Notificación de la temperatura actual.	Temperatura	x4x18
x5	Advertencia de Temperatura	Computador	Advertencia de valor crítico de temperatura.	Temperatura	x5x1B
x6	Pedido de Temperatura	Arduino	Pedido de la temperatura actual.		x6x0
x7	Pedido de Reseteo	Arduino	Pedido para reiniciar el microcontrolador.		x7x0
x8	Desbloqueo de Puerta	Arduino	Pedido de desbloqueo de puerta trasera para el modo de abastecimiento.		x8x0
x9	Encendido de Luces	Arduino	Encendido de luces laterales.		x9x0
xA	Apagado de Luces	Arduino	Apagado de luces laterales.		xAx0
xC	Entrega exitosa	Computador	Se detectó el medicamento y se lo entregó exitosamente.		xCx0
xD	Entrega fallida	Computador	El medicamento no fue detectado.		xDx0
xE	Pedido primer medicamento	Arduino	Pedido para entregar medicamento desde posición inicial.	Posición de entrega (valor en X, valor en Y)	xEx42
xF	Pedido medicamentos subsiguientes	Arduino	Pedido para entregar medicamento desde posición de entrega.	Posición de entrega (valor en X, valor en Y)	xFx34
x10	Finalizar entrega	Arduino	Pedido para retornar desde posición de entrega a posición inicial.		x10x0
x11	Encendido luz de abastecimiento	Arduino	Encendido de luz en la posición a abastecer	Posición luz de abastecimiento (valor en X, valor en Y)	x11x42
x12	Apagado de luz de abastecimiento	Arduino	Apagado de luz en la posición a abastecer	Posición luz de abastecimiento	x12x34

				(valor en X, valor en Y)	
x13	Estado puerta	Arduino	Notificación de cambio de estado de la puerta trasera	Estado puerta (0=cerrado, 1=abierto)	x13x1

Control del Movimiento de Motores. La máquina cuenta con dos motores a pasos encargados del movimiento horizontal del brazo mecánico y un motor adicional para el movimiento vertical. Los motores de pasos están conectados directamente al circuito impreso, el cual a su vez los conecta con las salidas correspondientes del RAMPS. La conexión está hecha de tal manera que los dos motores para el movimiento horizontal reciban la misma señal de control desde uno de los controladores DRV8825. El otro controlador maneja el movimiento del motor vertical.

El control de la distancia recorrida tanto horizontal como verticalmente se lo realiza con la función *movMot()*. Esta función recibe como parámetros la dirección y el número de pasos en los que se moverán cada uno de los motores, en dirección horizontal y vertical. Basándose en evidencia experimental realizada, se determinó que la velocidad a la cual el sistema operaba sin problema era de 37.5 r/min en el arranque y 44.6 r/min en movimiento para el eje horizontal. Para el eje vertical, estos valores son de 18.75 r/min en arranque y 22.3 r/min en movimiento. Esta función está diseñada para emitir pulsos de forma casi simultánea a ambos controladores a la velocidad establecida, esto permite un movimiento simultáneo en dirección horizontal y vertical.

La máquina utiliza dos servomotores para recoger una medicina. Uno de los servos acciona un actuador en forma de varilla que empuja la caja del medicamento desde abajo de la bandeja haciendo que caiga por gravedad dentro de la caja de metal ubicada en el brazo mecánico. El otro servomotor permite abrir una compuerta en el otro extremo de la caja para que el medicamento caiga dentro del ducto de entrega. Ambos servomotores están

conectados de igual forma al circuito impreso. Están alimentados a 6 voltios que provee el regulador LM7806 y están conectados a la tierra común del circuito. La señal de control de ambos servos proviene directamente del Arduino a través del terminal *SERVOSOUT* en el circuito.

Para el control de los servomotores, se utilizó la librería *Servo* de dominio público. Esta librería facilita el control de los servomotores ya que se necesita pasar como parámetro el ángulo al que se desea mover a los servomotores. Los ángulos para abrir y cerrar la compuerta y para accionar y regresar al actuador mecánico se determinaron de forma experimental.

El primer uso del movimiento de motores en el funcionamiento de la máquina es cuando esta está recién encendida y se ejecuta la función *setup()* de Arduino. Además de definir variables y objetos de distintos procesos, en esta función se coloca al brazo en la posición inicial. Después de enviar la señal x1x0 al computador para indicar que ha comenzado la inicialización, se inicializa el servo del actuador hasta su posición inicial. De igual forma, se realiza abre y cierra la compuerta de la caja para que en caso de que algún medicamento haya quedado dentro de la misma al realizar el reset de la máquina, este medicamento caiga en la parte inferior de la máquina y no ocasione problemas con las siguientes entregas. Después de esto, el microcontrolador utiliza la función *movMot()* para mover el brazo hacia la izquierda y hacia arriba simultáneamente. El movimiento horizontal se da hasta que detectan los fines de carrera ubicados en el borde izquierdo de la máquina. Lo mismo sucede con el movimiento vertical cuando se activa el fin de carrera ubicado en la parte superior del mismo brazo mecánico. Esto garantiza que siempre el brazo se encuentre en la posición inicial después de este proceso. Para mejorar la eficiencia del proceso general, la detención del movimiento de los motores solo se hace cuando se produce un cambio en el estado de los fines de carrera. Esto ocasiona que, durante el desarrollo de la máquina, los

motores no se detuvieran si se hacía un *reset* a la máquina cuando alguno de los fines de carrera se encontraba activado. Por esta razón, antes de realizar el movimiento de inicialización, se realizan pequeños movimientos hacia abajo y hacia la derecha. Así se asegura que ninguno de los fines de carrera estén activados.

Cuando el microcontrolador recibe la señal serial del tipo *xExXX*, se ejecuta la función *stepper_ini()*. Para esta función, se asume que el actuador se encuentra en la posición inicial en la esquina superior izquierda de la máquina. El brazo se mueve desde la posición inicial hasta la posición deseada. Se activa el servo del actuador para recoger la medicina. Si los sensores de presencia detectaron la medicina, el brazo se mueve hasta la posición predefinida de entrega, donde se activa el otro servo para abrir la compuerta y entregar el medicamento. Se enviará la señal *xCx0* si se cumplió una entrega exitosa. Durante la construcción de la máquina se pudo apreciar que en ocasiones imperfecciones mecánicas o físicas de la máquina o de los medicamentos hacen que este último se que atascado y no caiga dentro de la caja metálica. Para evitar esto, se programó la función de tal manera que, si no se detecta el medicamento dentro de la caja, se realicen pequeños movimientos verticales (primero hacia arriba y luego hacia abajo) para mejorar la posición del brazo y volver a accionar el actuador. Si a pesar de esto no se detecta ningún medicamento, se envía la señal *xDx0* de error y, por convención, se mueve el brazo hasta la posición de entrega.

Cuando el microcontrolador recibe la señal serial del tipo *xFxXX*, se ejecuta la función *stepper()*. Esta función difiere de la función *stepper_ini()* en que se asume que antes de ejecutarse el brazo robótico se encuentra en la posición de entrega del medicamento. Esta distinción permite reducir el tiempo de entrega total de medicamentos ya que el brazo mecánico no tienen que volver a la posición inicial antes de buscar otro medicamento. Por lo demás, las funciones *stepper()* y *stepper_ini()* funcionan de forma idéntica.

Al recibir la señal $x10x0$, se ejecuta la función *regresar()*. Esta función regresa al brazo robótico a la posición inicial. El movimiento es similar al realizado al inicializar la máquina, de forma que el brazo se mueve hasta detectar los fines de carrera.

Sensores de Movimiento. Se cuenta con dos fines de carrera para el movimiento horizontal, los cuales están conectados en paralelo y en configuración normalmente cerrado, de forma que los dos deban estar activados para recibir una señal en el microcontrolador. Para el movimiento vertical se cuenta solo con un fin de carrera en configuración normalmente cerrado en la posición descrita en el apartado anterior. Los fines de carrera para limitar el movimiento horizontal y vertical se encuentran conectados en los terminales *ENDSTOPX* y *ENDSTOPY* de la placa, respectivamente. Para ambos terminales, se cuenta con un circuito similar como se aprecia en la Figura 28 del Anexo B. Un extremo está conectado directamente al terminal (*ENDSTOPX* o *ENDSTOPY* según sea el caso) conectado al microcontrolador. El mismo extremo del terminal está también conectado a 5 voltios a través de una resistencia de 10 k Ω . El otro extremo está conectado directamente a tierra. Este circuito permite que cuando se activen los fines de carrera, estos pasen a estado abierto y, por tanto, se tenga una señal de 5 voltios a la entrada del microcontrolador. El ruido en la señal a la entrada del microcontrolador se filtra mediante un capacitor de 1 μ F (valor determinado experimentalmente).

Los fines de carrera función como interrupciones en el código. Cuando se detecta un cambio lógico en el valor de los sensores, se interrumpe el código que se está ejecutando y se ejecutan las funciones *inicioX()* e *inicioY()*, respectivamente, que se encargan de detener el movimiento de los motores a pasos.

La máquina utiliza adicionalmente sensores de presencia para determinar si el actuador mecánico pudo obtener exitosamente el medicamento desde la posición de almacenamiento. El sistema de detección está conformado por un LED infrarrojo y un fotodiodo infrarrojo. El LED está conectado a la placa a través del terminal *IRLED*. El LED está alimentado por 5 voltios (obtenidos a través del regulador 7805), con la presencia de una resistencia limitadora de corriente de 150Ω . El circuito del fotodiodo es similar al de los fines de carrera, con una resistencia de $3.3k\Omega$ que conecta al terminal negativo del fotodiodo a 5V y el terminal positivo conectado a tierra. El fotodiodo se conecta a la placa a través del terminal *FOTOD*. El terminal negativo del fotodiodo está de igual forma conectado al terminal *FOTOIN*, el cual se conecta al microcontrolador.

La función *sensor()* del programa se encarga de discriminar la presencia o no del medicamento. Esta función toma como entrada el valor análogo de voltaje de la terminal *FOTOIN*. Si el valor medido es mayor o igual a un valor umbral (establecido experimentalmente), la función devuelve un valor booleano de *true*. Caso contrario, se devuelve el valor *false*. Esta función es utilizada durante la ejecución de las funciones *stepper_ini()* y *stepper()*, como se detalló en el apartado anterior.

Control de Temperatura. Este control se realiza utilizando un sensor de temperatura LM35. El sensor se conecta a la placa a través del terminal *TEMP* del circuito. El sensor está alimentado con 5V a través del pin 1 del terminal y está conectado a tierra a través del pin 3 del mismo. El pin 2 del terminal *TEMP* está conectado directamente al terminal *TEMPIN*, que se da la señal de temperatura para el microcontrolador.

Para disminuir la temperatura se utiliza como actuador un ventilador. Este se conecta al circuito a través del terminal *FAN* del circuito. El pin 1 de esta terminal está conectado a

12V, mientras que el pin 2 está conectado al drenaje de un MOSFET IRF530N. La fuente del MOSFET está conectada a tierra, mientras es la señal de control que proviene de la terminal *FANOUT* conectada al microcontrolador.

Para el control PID de la temperatura se utiliza como base la librería PID de Arduino (Beauregard, 2015). Adicionalmente se desarrollo la función *ventilador()* que se ejecuta cada vez que la función *loop()* empieza a correr. La función *ventilador()* lee el valor de entrada dado por el sensor de temperatura. Si este valor es superior al valor de advertencia preestablecido, se envía una señal al computador de tipo *x5xXX*, cuyo parámetro es la temperatura medida en hexadecimal. El valor medido de temperatura es utilizado como parámetro por objeto de la librería PID para compararlo con el valor deseado, que en este caso es 25°C, y obtener el error (Beauregard, 2015). Las constantes proporcional, diferencial e integral son valores constantes preestablecidos (Barnes, 2011). La salida que produce el control PID es utilizada para establecer el valor analógico de voltaje utilizado para mover el ventilador.

Adicionalmente a la función anterior, la función *medirTemperatura()* es ejecutada cuando se recibe un mensaje de tipo *x6x0*. Esta función lee la temperatura actual del sistema y la envía al computador con un mensaje de tipo *x4xXX*, cuyo parámetro es la temperatura medida en hexadecimal.

Control de Luces. El control de las diferentes luces con las que cuenta la máquina se realizan mediante el uso de transistores como interruptores. Las tiras de LED utilizadas para el proceso de entrega sirven para iluminación general (azules), movimiento del brazo (blancas), notificación de entrega exitosa (verdes) y notificación de error (rojas). Cada una de estas luces son controladas por un MOSFET IRF530N cuya señal de control está dada por el

microcontrolador, como se aprecia en el esquema de la Figura 28 del Anexo B. Como se aprecia en la Tabla 2, las terminales que se conectan al microcontrolador para el control de estas luces son *BLUELEDOUT*, *WHITELEDOUT*, *GREENLEDOUT* y *REDLEDOUT*, respectivamente.

Dentro del programa, el computador puede prender o apagar las luces laterales (azules) de la máquina utilizando los comandos *x9x0* y *xAx0*, respectivamente. Estas acciones son realizadas por la función *controlLuces()* y, en cualquier caso, se envía una señal de OK una vez realizada la acción. La luz blanca ubicada detrás del brazo mecánico se prende antes de realizar cualquier movimiento del brazo mecánico y se apaga cuando este finaliza. La luz verde indicativa ubicada dentro de la caja de entrega se prende cuando se ha detectado que el medicamento ha caído dentro de la caja durante el proceso de entrega. En cambio, la luz indicativa roja se prende cuando el medicamento no fue obtenido y la respectiva señal de error fue enviada.

El control de las luces indicativas traseras para el proceso de abastecimiento utiliza el esquema matricial de la Figura 11. Ya que se consideran 100 luces diferentes para cada posición de los medicamentos, el circuito cuenta con 10 transistores LM3904 y 10 transistores LM3906, como se muestra en el esquema de Figura 28 del Anexo B. Los terminales para la conexión tanto de las luces indicativas como de las señales de control para el modo abastecimiento se detallan en la Tabla 2.

La función *cargarMed()* utiliza los comandos del tipo *x11xXX* y *x12xXX* para prender y apagar, respectivamente, las luces de abastecimiento de una terminada posición. La función devuelve un mensaje de OK después de realizar esta operación.

Sensor magnético y seguro eléctrico. El circuito diseñado está programado también para controlar la seguridad de la puerta trasera de la máquina para su abastecimiento. Mediante el uso de un sensor magnético como el que se trató en la sección anterior permite saber el estado actual de la puerta. El sensor se conecta en la terminal *MGSEC* de la placa. El sensor cuenta con un circuito similar al de los fines de carrera, con un extremo conectado a la terminal de entrada al microcontrolador y a 5V (con una resistencia de $10k\Omega$) y el otro extremo conectado a tierra. Cuando la se cierra la puerta, la terminal *MGSECIN* que se conecta al Arduino se encuentra a 0V. Caso contrario, el sensor actúa como circuito abierto y la terminal *MGSECIN* se encuentra a 5V. Se colocó de igual forma un capacitor de $1\mu F$ para filtrar la señal de entrada al microcontrolador.

El sensor magnético función como una interrupción al código, como los fines de carrera. Inmediatamente después de producirse un cambio de estado en el sensor (de cerrado a abierto o viceversa) se ejecuta la función *magSec()*. Esta función envía al computador un mensaje de tipo *x13xX*, cuyo parámetro es 0 si la puerta está actualmente cerrada o 1 si está abierta.

La cerradura eléctrica se encuentra conectada al circuito por medio de la terminal *LOCK*. El voltaje entre las terminales se controla utilizando un MOSFET IRF530N. La señal de control para el MOSFET se obtiene desde el microcontrolador por medio de la terminal *LOCKIN*. Adicionalmente, se coloca un diodo 1N4004 por protección del solenoide (“How to Drive Solenoids with the Arduino”, s.f.).

Cuando el computador envía una señal *x8x0*, el microcontrolador ejecuta la función *unlock()*, que envía una señal lógica de 1 al terminal *LOCKIN* durante 1.2 segundos para que se pueda abrir el seguro eléctrico.

PRUEBAS Y RESULTADOS

Tal y como se ha visto en secciones anteriores, el equipo esta compuesto por un sistema de control general el cual se puede categorizar en: sistema de control ambiental, sistema de seguridad y sistema de entrega de medicamentos. Para el correcto funcionamiento del equipo en general se han llevado a cabo pruebas y experimentos para encontrar el punto de operación optimo de cada sistema. Una vez que se ha probado el funcionamiento de cada sistema se ha procedido a probarlos en conjunto y reevaluar dichos puntos de operación óptimos. Los procesos llevados a cabo se describen a continuación:

Sistema de control ambiental

Parámetros PID. Como se vio anteriormente, el controlador PID posee tres parámetros por defecto, estos son: Proporcional, Integral y Derivativo. La forma más eficiente de calibrar estos parámetros es experimental. La variación de dichas constantes puede hacer que el control sea más o menos agresivo. Si el control es más agresivo el mismo tendrá problemas para controlar variaciones pequeñas de temperatura. Del mismo modo si el control es muy dócil, este no podrá controlar cambios bruscos de temperatura. Por tanto es necesario encontrar un punto medio de funcionamiento donde el sistema de control pueda ser lo suficientemente sensible a cambios pequeños de temperatura pero al mismo tiempo sea capaz de controlar las variaciones considerables. Cabe mencionar que los parámetros deben tener una inclinación hacia los cambios sensibles, considerando que la variación de temperatura ocurre de forma lenta y los cambios bruscos de temperatura son menos probables. Para calibrar estos parámetros, se los ha ido variando uno por uno y se ha hecho variar la temperatura del sensor con una fuente de calor para que el control PID accione el ventilador. Por inspección visual del funcionamiento del ventilador se ha encontrado el punto optimo de

operación del control PID. Luego se ha probado el funcionamiento del sistema haciendo variar unos pocos grados el “set point” asignado y se ha comprobado con un termómetro digital que la acción del ventilador estabiliza la temperatura del sistema al valor de “set point”. Cabe destacar que para realizar estas pruebas se simuló las condiciones de la máquina utilizando un sistema cerrado con una bombilla como fuente de calor, el sensor LM35 y el ventilador como actuador. Los parámetros para el modelo real del prototipo podrían cambiar.

Punto óptimo de lectura del sensor de temperatura y ubicación de ventilador. Se ha realizado medición de voltaje del sensor de temperatura en distintos puntos del equipo para establecer donde se obtiene una mejor estimación de la temperatura promedio. Así mismo se ha probado en el diseño distintos puntos donde el ventilador pueda crear un mayor flujo de aire. Se ha probado que el punto óptimo de colocación del sensor de temperatura es en la parte interna de los racks donde se ubican los medicamentos, es decir en el centro del equipo. Sin embargo puesto que esto dificultaba la instalación del sensor y dado que los resultados obtenidos no variaban considerablemente, se ha optado por instalarlo en la parte superior del equipo. Por otra parte se ha comprobado que el ventilador puede distribuir una mayor cantidad de flujo de aire al colocarlo en la parte superior, es decir en el techo, del equipo. También se ha probado experimentalmente que es necesario incluir ventiladores con mayor capacidad y potencia. En ambientes con mayor concentración de humedad y calor también puede ser necesario que el ventilador sea remplazado por un aire acondicionado.

Consumo de potencia de luces LED ambientales. Dado que las luces LED ambientales, cuyo color es azul, deben estar encendidas en todo momento cuando la máquina

se encuentre energizada, es de suma importancia que el consumo de potencia de las mismas no afecte el rendimiento de elementos de otros sistemas como por ejemplo los motores de pasos, los cuales en reiteradas ocasiones deben mantener su posición y por ende tienen picos de consumo de corriente considerables. Ambos elementos, tanto las cintas LED como los motores, son los elementos de mayor consumo en el equipo. Por tanto, como una primera práctica experimental se ha procedido a alimentar los motores a pasos con una fuente de alimentación de laboratorio la cual permite dimensionar la cantidad de corriente consumida durante su funcionamiento. Se ha tomado nota del consumo de corriente de los motores y se ha llegado a los resultados de la Tabla 4.

Tabla 4
Consumo de corriente y potencia de motores Servo

Consumo por motor Servo	Standby	En movimiento
Corriente (A)	3 A	0.8 A
Consumo de potencia (W)	36 W	9.6W

Considerando que la fuente utilizada por el equipo puede otorgar hasta 360W en consumo de potencia y el consumo de los motores es el mostrado en la tabla anterior, se ha procedido a elegir el tipo de cinta de LED para la ambientación. Se ha probado la luminosidad ambiental con dos distintos tipos de tiras LED de bajo consumo: 4,8 W/m y de 12W/m. Se ha comprobado experimentalmente que aunque la cinta LED de 4.8W/m consume menor cantidad de potencia, es necesario colocar un mayor número de metros de cinta LED para alcanzar la misma luminosidad que con la cinta de 12W/m. Y como resultado al colocar la cinta de 12W/m se requiere menos consumo de potencia, por lo tanto se ha elegido la cinta LED de 12W/m.

Sistema de entrega de medicamentos

Frecuencia de motores. La frecuencia de la señal con la que operan los motores a pasos es de suma importancia para el correcto funcionamiento del equipo. Además el sistema requiere que los motores funcionen a la mayor rpm posible para poder entregar los medicamentos con la mayor rapidez. Para encontrar la frecuencia a la que los motores operan de forma correcta, se los ha conectado a un generador de señales en el laboratorio. Se ha ido variando la frecuencia de la señal hasta llegar al punto cuando los motores dejaban de funcionar por completo, esta frecuencia fue de 400Hz. Luego se ha comenzado a disminuir dicha frecuencia hasta llegar a un punto en que los motores funcionen de forma correcta nuevamente, de esta forma se ha encontrado la mayor frecuencia posible a la que pueden operar dichos motores. El valor elegido fue de 280Hz, a esta frecuencia los motores tienen una rpm de 84 rev/min. Luego se ha realizado esta misma prueba con los motores instalados y con la carga de los brazos mecánicos. Se ha encontrado que la frecuencia a la que operan los motores ha disminuido, sobre todo en el motor del eje vertical. Para solucionar este problema podría incluirse un motor adicional en el eje vertical para que asista al motor existente en el movimiento de la carga del brazo mecánico. Adicionalmente para dar una solución al problema rapidez de los motores, se los ha hecho trabajar con aceleración. En un inicio los motores operan a una frecuencia menor, una vez que la inercia del brazo mecánico ha sido superada se comienza a subir la frecuencia de la señal. Esto permitió mejorar la rpm a la que funcionan los motores en hasta un 25%.

Ángulo de movimiento de motores servo para actuador. Es de vital importancia para evitar que se estropeen las piezas mecánicas del actuador, donde se acciona una perilla que empuja los medicamentos a una caja con una compuerta que se abre cuando el actuador ha llegado a las coordenadas establecidas para entrega de medicamento, que los servos que

controlan tanto la perilla como la compuerta estén perfectamente calibrados para que se desplacen una cantidad exacta angular. Si el servo posee una mala calibración podría dañarse y en el peor de los casos echar a perder todo el mecanismo mecánico. Por esta razón se ha hecho variar los valores angulares de movimiento de ambos servo en pocos grados hasta establecer el punto óptimo. Después de algunas pruebas se ha concluido que el ángulo necesario para la posición inicial es: 145 grados para servo de la perilla y 160 grados para el servo de la compuerta. Para cuando se accionan ambos servos, es decir cuando se levanta la perilla o se abre la compuerta, los ángulos son: 25 grados para servo de la perilla y 66 grados para el servo de compuerta. Adicionalmente se ha realizado una prueba de movimiento continuo de los servo para comprobar que no exista calentamiento por fricción en las partes mecánicas ni motores.

Calibración de cantidad de pasos por posición en motores a pasos. Los motores a pasos deben moverse una cierta cantidad de pasos tanto en el eje vertical como en el horizontal para que el brazo mecánico llegue a una coordenada determinada. Para encontrar esta cantidad de pasos se ha definido la posición (1,0) (posición 1 horizontal, 0 vertical) y se ha ido variando los valores hasta el punto en el que la perilla del actuador del brazo mecánico se encuentre exactamente en el punto medio de los racks de dicha posición. Se ha concluido que los valores de pasos por posición son: 625 para X y 559 para Y. De tal forma que para llegar a la posición (5,6), los pasos requeridos en X son $625*5$ y $559*6$ en Y. Cabe mencionar que para mejorar la eficiencia de operación de los motores se los ha hecho funcionar con la opción de microstepping 1/32. En esta función, como se explico anteriormente, los motores se mueven 1/32 de paso en lugar de un paso completo. Esto permite que los motores operen de forma más silenciosa y precisa. Cuando los motores operan con microstepping para llegar a la posición (5,6), los pasos requeridos en X son $625*5*32$ y $559*6*32$ en Y.

Sistema de seguridad

Accionamiento de interruptores magnéticos. Con el fin de evitar que alarmas falsas se accionen, se ha realizado varias pruebas de desconexión del interruptor magnético ubicado en la puerta posterior. Adicionalmente se ha probado que no exista ningún efecto adverso en la operación de los mismos por campos magnéticos provenientes de los estatores de los motores de pasos. Como resultado se ha comprobado que los interruptores funcionan sin ningún inconveniente y la señal es recibida por el micro controlador sin ruido de consideración.

CONCLUSIONES

Con este proyecto se pudo lograr la creación de un sistema de control electrónico para la operación de un dispensador automático de medicamentos en base al presupuesto limitado con el que se contaba. El prototipo creado cumplió con las expectativas iniciales con las que fue creado. Su uso futuro dentro de la clínica de la Universidad San Francisco de Quito permitirá determinar las correcciones que pueda necesitar el sistema.

Con respecto a los sistemas específicos que conforman el sistema general de control, se obtuvieron las siguientes conclusiones:

- Se optimizó el movimiento de los motores de pasos de forma que estos actuarán a la mayor velocidad posible sin perder el torque necesario para mover el brazo mecánico. De igual forma se consiguió un movimiento horizontal y vertical simultáneo de los motores y se optimizó el código para eliminar movimientos innecesarios del brazo mecánico, lo cual redujo el tiempo total de entrega del medicamento.
- Con el uso de sensores de presencia y fines de carrera, se creó un control de lazo cerrado para el proceso de entrega de medicamentos que sea lo suficientemente confiable reduciendo la cantidad de errores y fallas al mínimo dentro de los parámetros de trabajo.
- Se implementó un sistema de control PID de temperatura en el microcontrolador con una revisión casi continua del nivel de temperatura de la máquina. La suspensión del control de temperatura se da únicamente para evitar el retardo de las otras operaciones de la máquina. Si no existe otra acción por realizarse, el control es continuo.
- Se diseñó un sistema de luces indicativas que mejoran la experiencia visual del usuario y le permiten saber de inmediato cuando existe un problema con la entrega de

medicamentos. De igual forma, se facilitó el trabajo del operador para el abastecimiento de la máquina por estos mismos medios.

- Con el uso de una comunicación serial codificada entre el microcontrolador y el computador encargado del control de inventario e interfaz, se pudo garantizar el correcto funcionamiento de la máquina en función de las instrucciones recibidas. De igual forma, el uso de mensajes de OK o de error permite alertar al usuario de errores de ejecución por medio de una interfaz gráfica.
- Se diseñó un sistema de seguridad capaz de controlar el acceso al interior de la máquina solo a personal autorizado mediante el uso de un seguro electromecánico y un sensor magnético.

DISCUSIONES

A pesar de que se cumplieron los objetivos establecidos para este proyecto, existen algunas mejoras que podrían optimizar el funcionamiento de un segundo prototipo:

- Se puede notar que la desventaja de un sistema que emplee motores a pasos es el bajo rendimiento en velocidad. Para aumentar la velocidad del mismo se podría utilizar una fuente de mayor voltaje o trabajar con motores DC con *encoders*.
- El uso de sensores de presencia en cada una de las posiciones de abastecimiento de la máquina permitiría hacer un conteo del número de medicamentos que son ingresados por el operador y compararlo con el número ingresado por el mismo operador mediante la interfaz gráfica. Esto aseguraría tener un control certero del inventario y evitaría posibles robos de medicamentos.
- Por restricciones de tiempo y presupuesto, no se pudo probar el sistema de control de temperatura en el prototipo real, en lugar de esto se simuló las condiciones en las que se encuentra la máquina. Una vez que se disponga del prototipo de la máquina totalmente armado, se deberá ajustar los parámetros del control PID para optimizar el control de temperatura. Además, se utilizó un solo sensor de temperatura y un solo ventilador para el control. Para un control más efectivo de temperatura se podrían colocar varios sensores de temperatura en diferentes ubicaciones dentro de la máquina. Al mismo tiempo se podría utilizar varios ventiladores distribuidos en la máquina o un sistema central de aire acondicionado.
- Por motivos de tiempo no ha sido posible la implementación del sistema de seguridad de la máquina, por lo que se desconoce si pueda haber posibles fallos en el sistema. Adicionalmente, para el prototipo, no se consideró la necesidad de una puerta

delantera para el mantenimiento de la máquina. Por esta razón, se debería también duplicar el sistema de seguridad para esta puerta.

- Además de los dispositivos utilizados para el sistema de seguridad, la adición de una sirena de alerta dentro del circuito podría mejorar la confiabilidad del mismo.
- Cabe resaltar que se utilizó una tarjeta Arduino y un shield RAMPS 1.4 para facilitar el desarrollo del sistema de control. Una vez que se haya optimizado el sistema se puede optar por fabricar una sola PCB que contenga el microcontrolador, los controladores de los motores y demás circuitos necesarios para el sistema. Esto eliminaría la existencia innecesaria de conductores entre el microcontrolador y el resto del circuito.

REFERENCIAS BIBLIOGRÁFICAS

- “3-Terminal 1 A Positive Voltage Regulator”. (2014). *Fairchild Semiconductor*. Recuperado el 24 de noviembre de 2015 de <https://www.fairchildsemi.com/datasheets/lm/LM7805.pdf>.
- “Arduino MEGA 2560”. (s.f.). *Arduino*. Recuperado el 24 de noviembre de 2015 de <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- “Arduino Mega Pololu Shield”. (s.f.). *RepRap*. Recuperado el 25 de noviembre de 2015 de http://reprap.org/wiki/Arduino_Mega_Pololu_Shield.
- “Definition of: microcontroller”. (s.f.). *PC Magazine*. Recuperado el 24 de noviembre de 2015 de <http://www.pcmag.com/encyclopedia/term/46924/microcontroller>.
- “Detienen a 7 sospechosos del robo de medicamentos en hospital del IESS.” (2015). *Ecuavisa*. Recuperado el 8 de abril de 2015 de <http://www.ecuavisa.com/articulo/noticias/actualidad/100636-detienen-7-sospechosos-del-robo-medicamentos-hospital-del-iess>.
- “El servomotor”. (Sin fecha). Recuperado el 24 de noviembre de 2015 de <http://www.info-ab.uclm.es/labelec/solar/electronica/elementos/servomotor.htm>
- “High Torque Hybrid Stepping Motor Specifications”. Recuperado el 7 de abril de 2015 de <https://www.pololu.com/file/0J672/SY57STH56-1006A.pdf>.
- “How to Drive Solenoids with Arduino”. (s.f.). *Arduino*. Recuperado el 24 de noviembre de 2015 de http://playground.arduino.cc/uploads/Learning/solenoid_driver.pdf
- “Lock-style Solenoid - 12VDC”. (s.f.). *Adafruit*. Recuperado el 24 de noviembre de 2015 de <https://www.adafruit.com/products/1512>.
- “Medicamentos del IESS fueron hallados en una clínica clandestina de Ambato”. (2015). *ElComercio.com*. Recuperado el 8 de abril de 2015 de

<http://www.elcomercio.com/tendencias/medicamentos-iess-clinicaclandestina-ambato-delito.html>.

“Nema Motor”. (sin fecha). *RepRap Org*. Recuperado el 7 de Abril de 2015 de

http://reprap.org/wiki/NEMA_Motor.

“Pharmacy robot streamlines medication dispensing process”. (2011). *Nash UNC Health Care*. Recuperado el 10 de abril de 2015 de

http://www.nhcs.org/press_releases/articletype/articleview/articleid/136.

“RAMPS 1.4”. (s.f.). *RepRap*. Recuperado el 25 de noviembre de 2015 de

http://reprap.org/wiki/RAMPS_1.4.

“Shields”. (s.f.). *Arduino*. Recuperado el 25 de noviembre de 2015 de

<https://www.arduino.cc/en/Main/ArduinoShields>.

“Solenoid”. (s.f.). *Real Academia Española*. Recuperado el 24 de noviembre de 2015 de

<http://buscon.rae.es/drae/srv/search?val=solenoid>

Akai, M. (2000). “Control Systems, Robotics, And Automation”. Kyoto University Japan.

Recuperado el 21 de noviembre de 2015 de

<http://www.eolss.net/ebooks/Sample%20Chapters/C18/E6-43-03-03.pdf>

Barnes, C. (2011). “Arduino Powered Smart Fan Controller”. *Barnesian*. Recuperado el 25 de noviembre de 2015 de <https://www.barnesian.com/arduino-powered-smart-fan-controller>.

controller.

Beauregard, B. (s.f.). “Arduino PID Library”. *Arduino*. Recuperado el 20 de noviembre de

<http://playground.arduino.cc/Code/PIDLibrary>.

Boylestad, R., & Nashelsky, L. (2009). *Electrónica: Teoría de circuitos y dispositivos electrónicos*. México: Pearson Education.

- Buss, C. (2015). "Robots to pack drugs for hospital patients". *Leicester Mercury*. Recuperado el 10 de abril de 2015 de <http://www.leicestermercury.co.uk/Robots-pack-drugs-hospital-patients/story-25943265-detail/story.html>.
- Cox, J. (2001). *Fundamentals of linear electronics: integrated and discrete*. Segunda Edición.
- Dean, T. (2010). *Network + Guide to Networks*.
- Debashis, D. (2010). *Basic Electronics*.
- DuBois, J. (2009). "Design a simple DC power supply". Recuperado el 21 de noviembre de 2015 de <http://www.egr.msu.edu/classes/ece480/capstone/spring09/group06/power%20supply%20app%20note.pdf>
- Fitzgerald, A. (2005). *Máquinas Eléctricas*. Sexta Edición. Mc Graw Hill.
- García, A. (2015). El Multiplexado de LEDs con Arduino. Recuperado el 20 de julio de 2015 de <http://panamahitek.com/el-multiplexado-de-leds-con-arduino>
- Harris, D., & Harris, S. (2013). *Digital design and computer architecture*. Amsterdam: Morgan Kaufmann.
- Herman, S. *Industrial Motor Control*. Sexta Edición. Delmar Cengage Learning.
- Hetherington, G. (2015). "Robotic dispenser is the modern arm of medicine". *The Northern Echo*. Recuperado el 10 de abril de 2015 de http://www.thenorthernecho.co.uk/news/11796726.Robotic_dispenser_is_the_modern_arm_of_medicine.
- Ingram, L. (2015). "Robot at University Health Services is a counting machine". *Penn State News*. Recuperado el 10 de abril de <http://news.psu.edu/story/342735/2015/02/03/impact/robot-university-health-services-counting-machine>.

LaLena, M. (2015) “Tamper Proof Security System Wiring”. Structured Home Wiring.

Recuperado el 24 de noviembre de 2015 de:

<http://www.structuredhomewiring.com/SecuritySystem/TamperProofWiring/>

Ledbox. (2012). “Tiras Led 220V. Diferencias entre SMD3528, SMD3014 y SMD 5050”.

Recuperado el 28 de julio de: <http://blog.ledbox.es/informacion-led/diferencias-entre-los-modelos-de-tiras-led-flexibles-con-smd-3528-y-smd-5050>.

Nurse, J. (2014). “Rose Street robot whirr-king hard for Wokingham patients”. *Getreading*.

Recuperado el 10 de abril de 2015 de <http://www.getreading.co.uk/news/local-news/rose-street-robot-whirr-king-hard-6804122>.

Poornachandra, S., Sasikala, B. y Khan, A. (2005). *Introduction to Electrical, Electronics and Communication Engineering*. Nueva Delhi: Lama Publications.

Rush-Monroe, K. (2011). “New UCSF Robotic Pharmacy Aims to Improve Patient Safety”.

UCSF. Recuperado el 10 de abril de 2015 de

<http://www.ucsf.edu/news/2011/03/9510/new-ucsf-robotic-pharmacy-aims-improve-patient-safety>.

SGS-Thomson Microelectronics. *Stepper Motor Driving*. Recuperado el 8 de abril de 2015 de

http://users.ece.utexas.edu/~valvano/Datasheets/Stepper_ST.pdf.

Stephen Herman. (2009) *Industrial Motor Control*. Sexta Edición, Delmar Cengage Learning.

Texas Instruments. *Stepper Motor Controller IC*. Recuperado el 8 de Abril de 2015 de

<https://www.pololu.com/file/0J590/drv8825.pdf>.

The MOSFET. (2010). How the MOSFET Works. Recuperado el 3 de agosto de 2015 de

<http://www.talkingelectronics.com/projects/MOSFET/MOSFET.html>.

Velasco, H. (Sin fecha). *Motores Paso a Paso*. Fecha de acceso: 7 de Abril de 2015.

Recuperado de

<http://www.docentes.unal.edu.co/hfvelascop/docs/CLASES/DIGITALES2/LABORATORIO/Motor%20Paso%20a%20Paso.pdf>

ANEXO A: PROTOTIPO



Figura 18. Vista frontal de la máquina en movimiento.

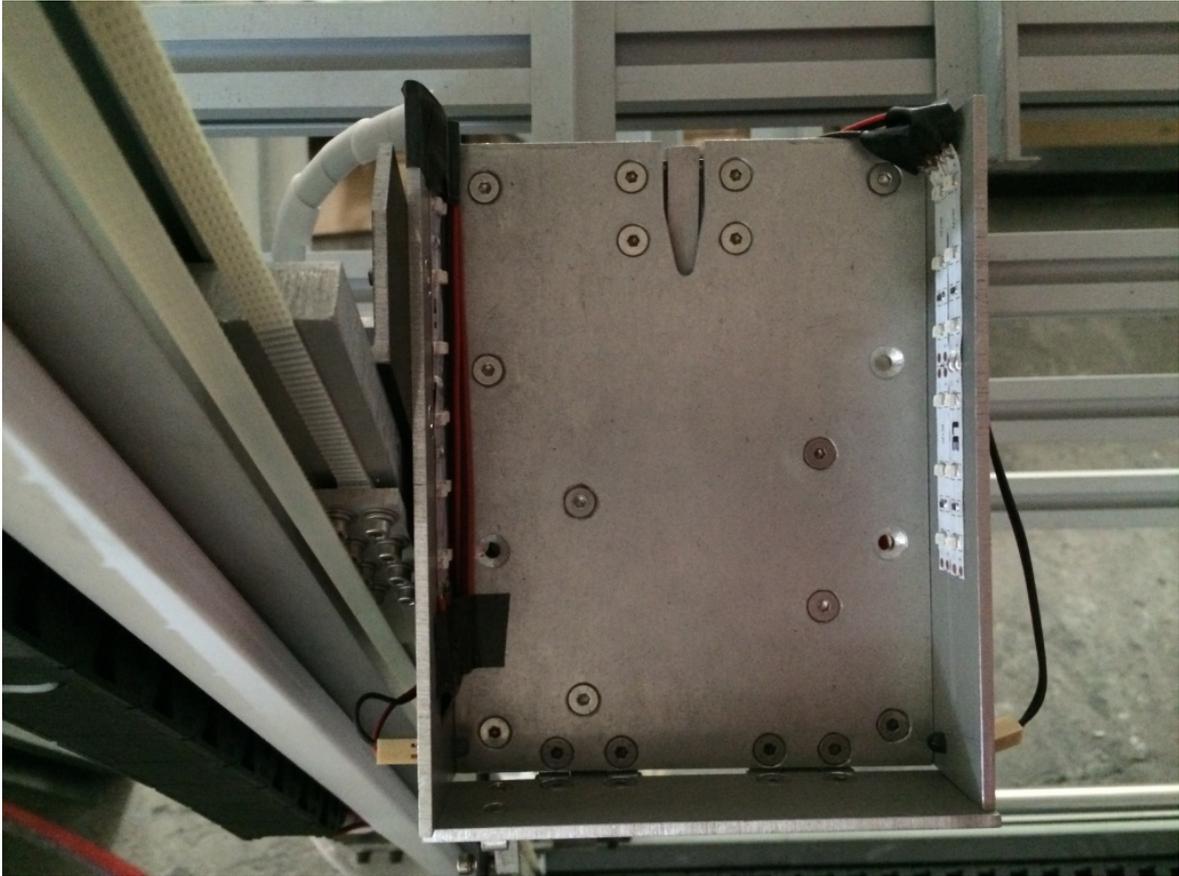


Figura 19. Vista superior de la caja de entrega. El fotodiodo y el LED IR se observan en los costados en la parte inferior de la caja. El actuador está en la parte superior y la compuerta en la parte inferior de la misma.

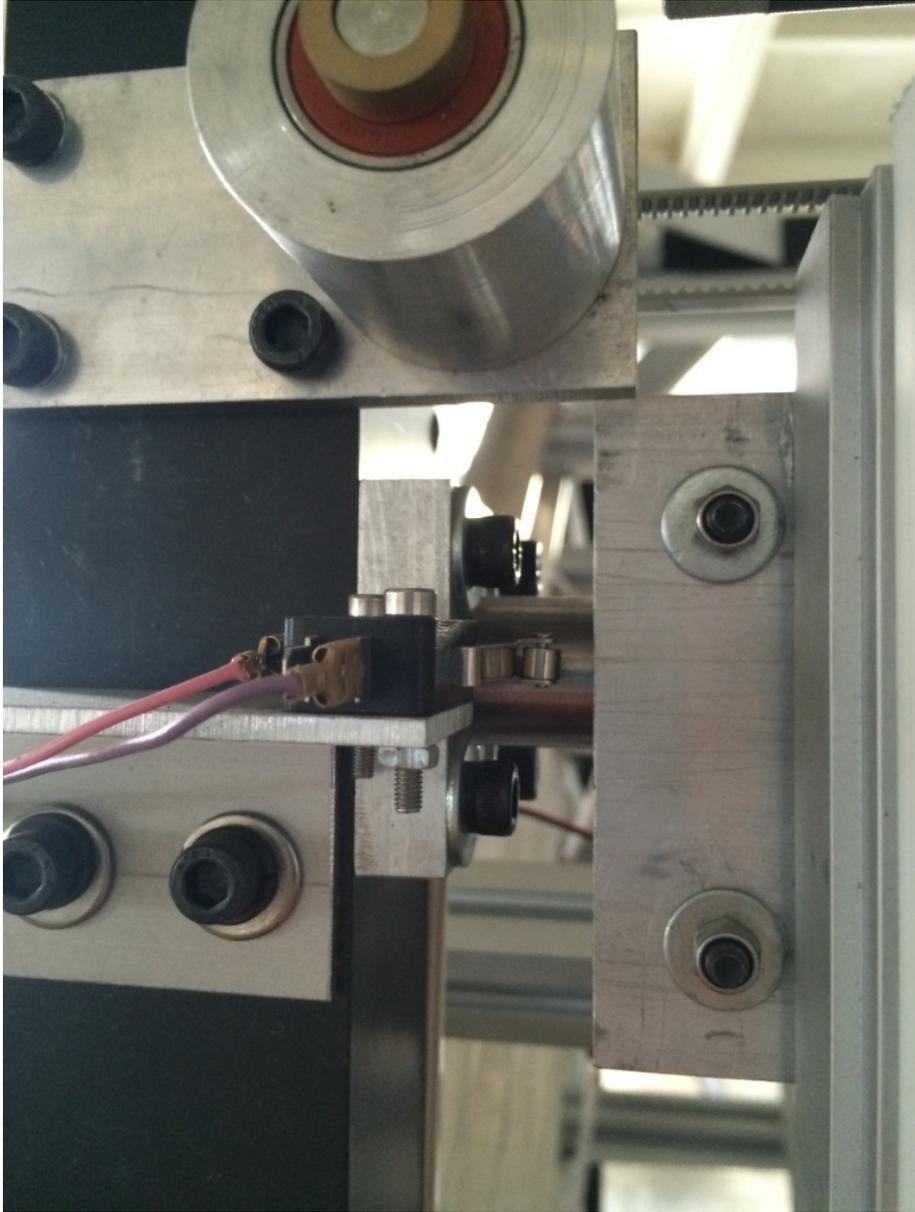


Figura 20. Fin de carrera superior para el movimiento horizontal.



Figura 21. Fin de carrera para el movimiento vertical. Este fin de carrera está ubicado en el brazo mecánico.



Figura 22. Luces de entrega exitosa. Las luces verdes se prenden ante la presencia del medicamento en la caja.

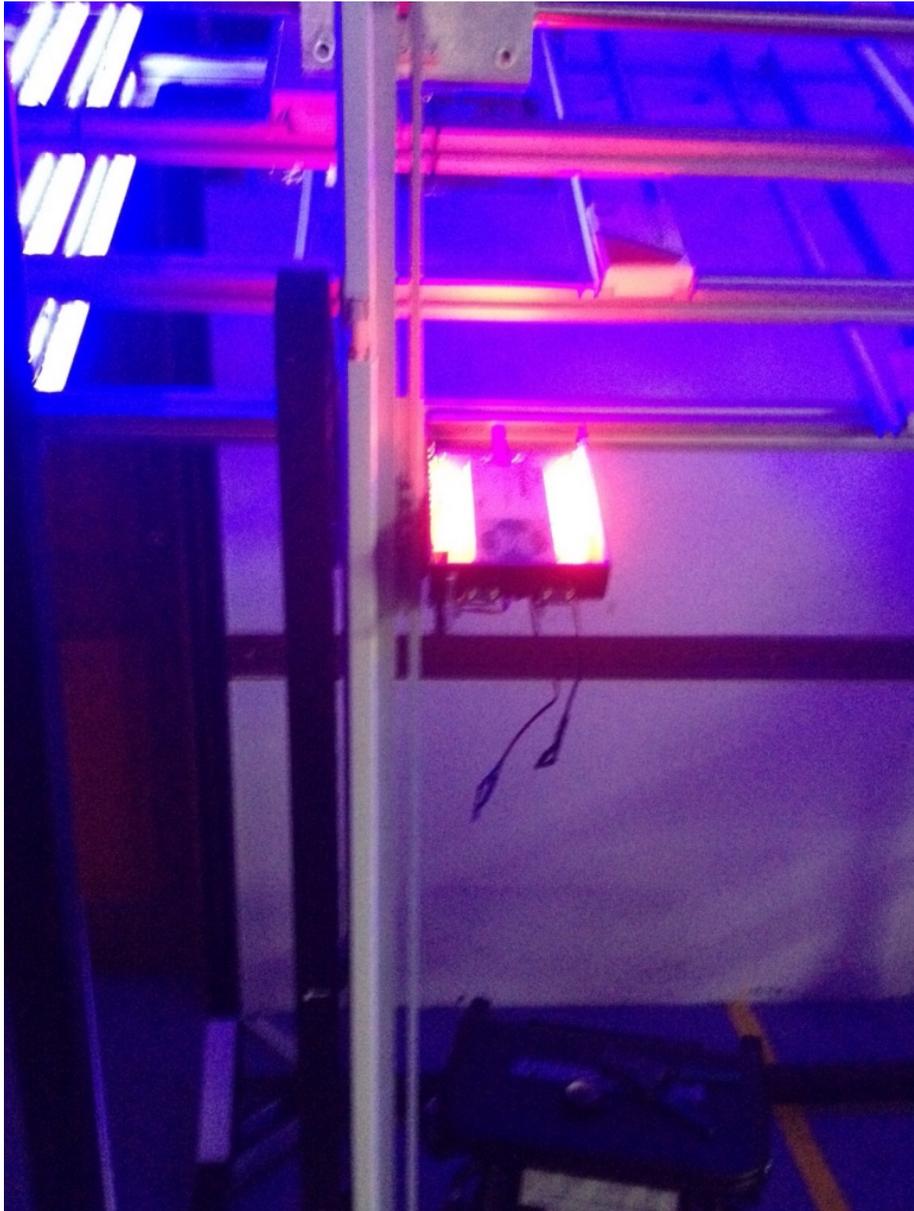


Figura 23. Luces de fallo de entrega. Las luces rojas se prenden cuando no se detecta el medicamento en el proceso de entrega.



Figura 24. Luces indicadoras traseras. Indican al operador donde colocar el medicamento seleccionado.



Figura 25. Ubicación del sensor de temperatura. Se encuentra en la parte superior trasera de la máquina. En esta imagen el sensor LM35 no está colocado.



Figura 26. Ubicación del ventilador en el prototipo. Se encuentra en la esquina superior derecha en la parte de atrás de la máquina. Para la versión final se planea colocarlo en la parte superior central.

ANEXO B: FIGURAS ADICIONALES

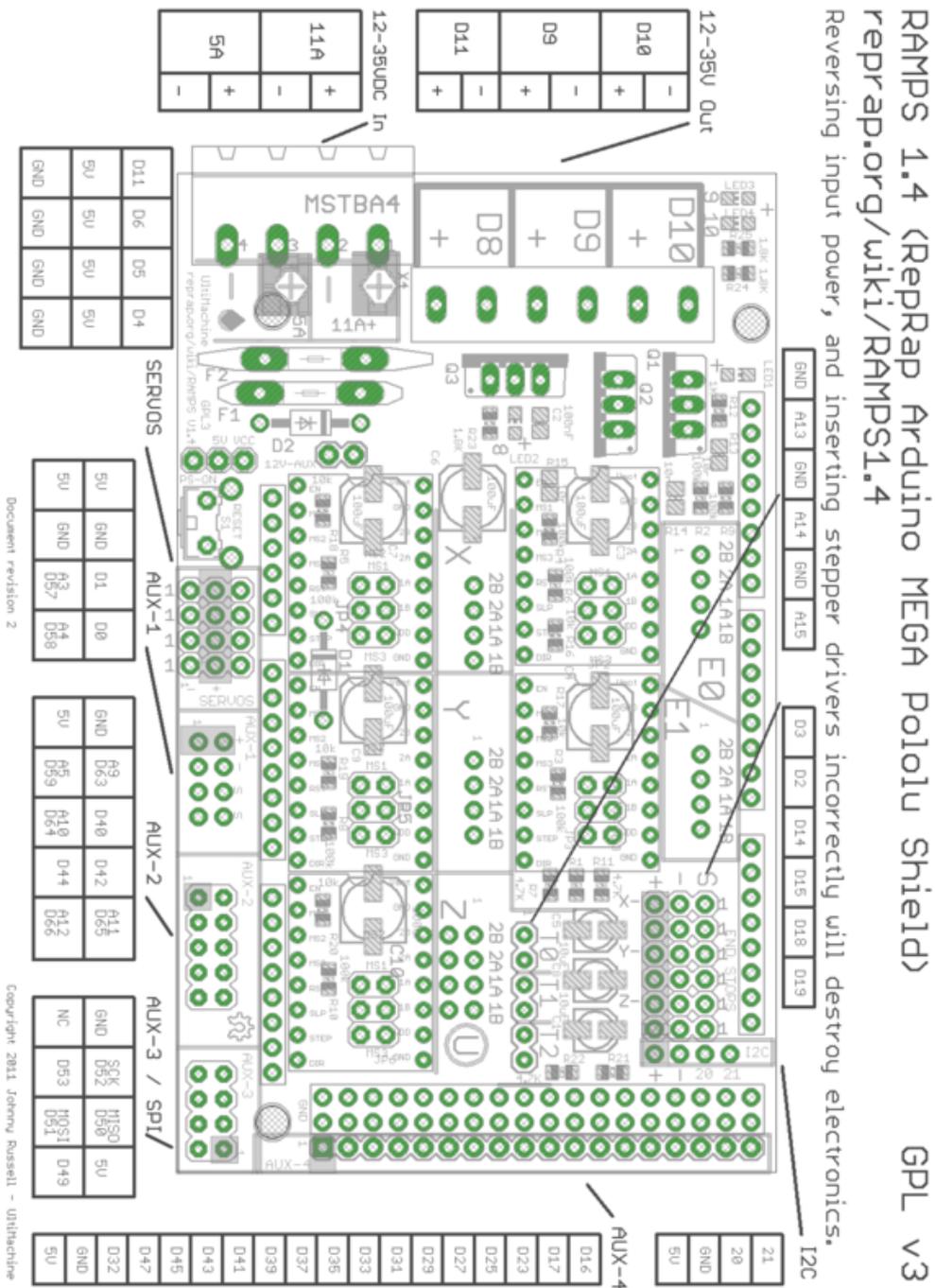


Figura 27. Ubicación de pines para shield RAMPS 1.4. Tomado de "RAMPS 1.4", s.f.

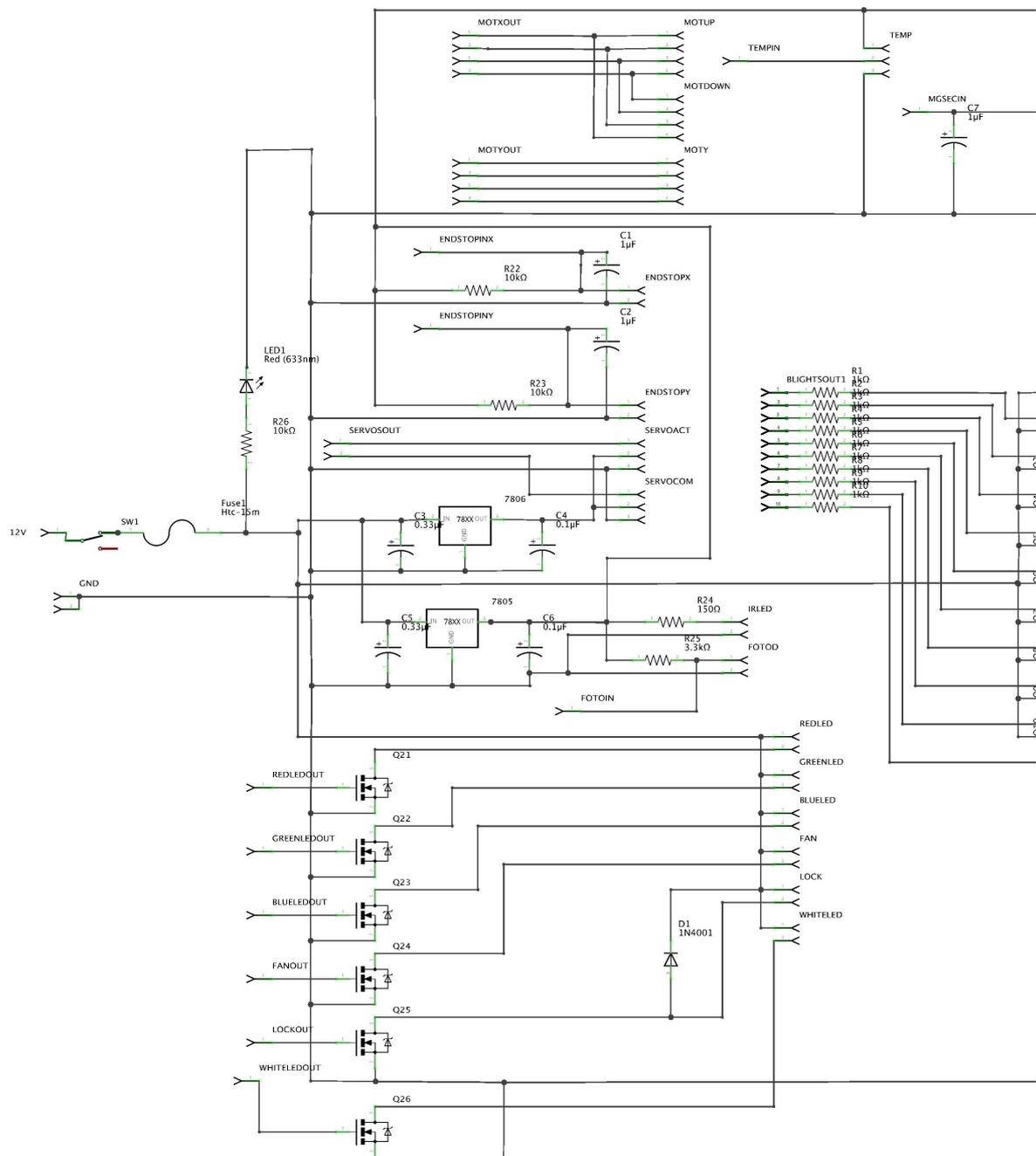


Figura 28a. Esquema del circuito impreso diseñado (Parte 1)

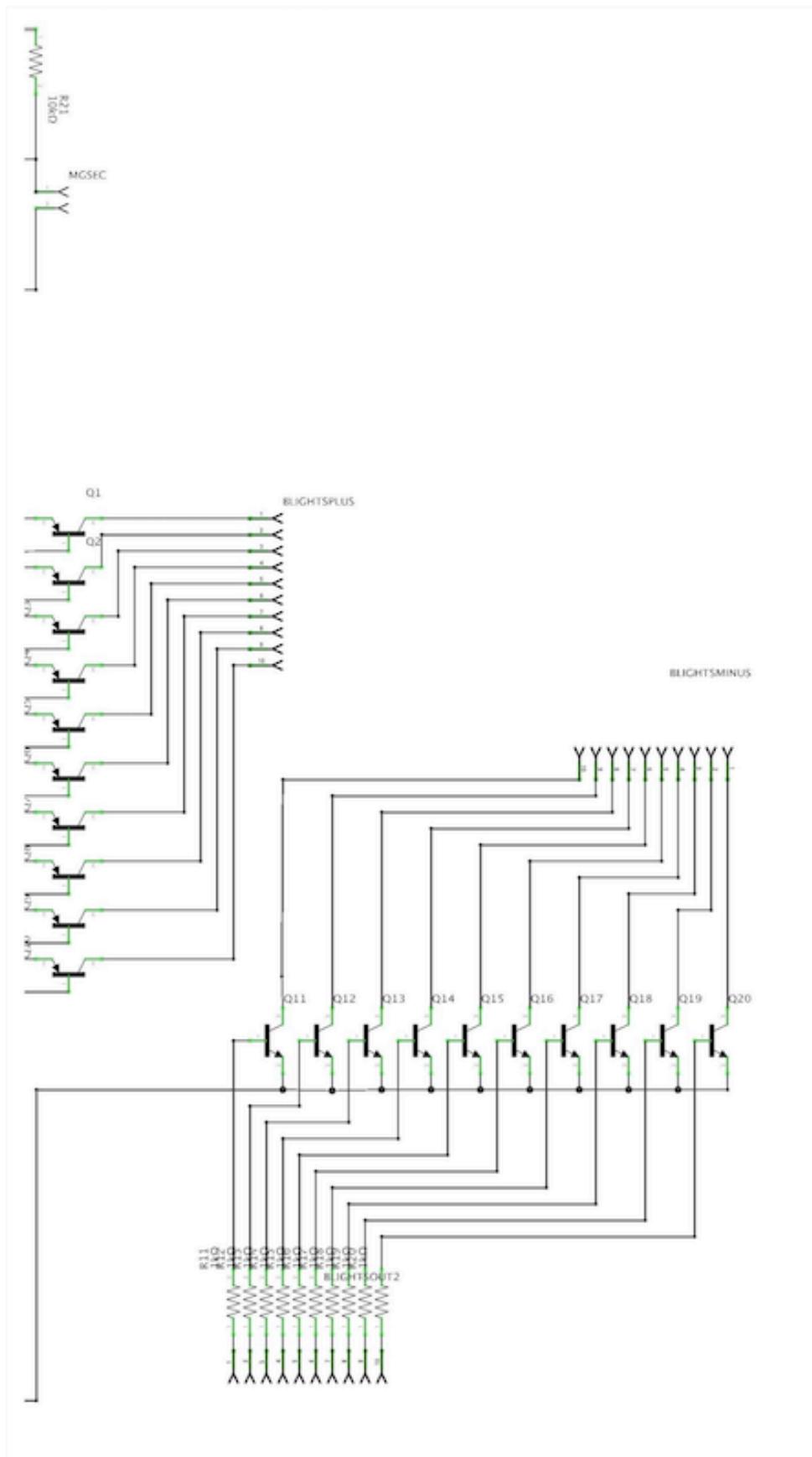


Figura 28b. Esquema del circuito impreso diseñado (Parte 2)

ANEXO C: CÓDIGO MICROCONTROLADOR

```

#include <Servo.h>
#include <PID_v1.h>
#include <stdlib.h>

//Motores Servo y Stepper*****
#define pinEnableY 38 // Enable Motor Y (X en Shield)
#define pinStepY 54 // Step Motor Y
#define pinDirY 55 // Direccion Motor Y
#define pinEnableX 62 // Enable Motor X (Z en shield)
#define pinStepX 46 // Step Motor X
#define pinDirX 48 // Direccion Motor X
#define pinServo1 11 //Control Servo
#define pinServo2 6 //Control Servo

#define pinVent 29
#define LM35 A13

#define pinFotoDiodo A15

#define pinLedVerde 23
#define pinLedRojo 25
#define pinLedAzul 27
#define pinLock 31
#define pinLedBlanco 33

boolean interrupcionX = false;
boolean interrupcionY = false;
const long numPasosX = 625 * 32;
const long numPasosY = 559 * 32; //570
const long entregaX = 7 * numPasosX;
const long entregaY = 4 * numPasosY;
const long x0 = 260 * 32;
const long y0 = 22 * 32; //22
Servo servo1;

```

```

int posServo1 = 0;

Servo servo2;

int posServo2 = 0;

const int limiteFotDio = 750;

int pinesLucesX[10] = {35, 37, 39, 41, 43, 45, 47, 32, 5, 4};
int pinesLucesY[10] = {1, 0, 63, 59, 40, 64, 42, 44, 65, 66};
//*****

//Declaracion Reset *****

void(* resetFunc) (void) = 0;

//Ventilador*****

double Setpoint, warning, Input, Output;

double Kp = 40, Ki = 1.5, Kd = 10;

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, REVERSE);

int temperatura;

void stepper_ini(int);

void ventilador();

//*****
//***** S E T U P *****
//*****

void setup() {

// Comunicacion Serial*****

//Serial.begin(9600);

Serial2.begin(9600); //9600 baudios

Serial2.print("x01x00\n"); //arduino encendido

```

```

// Interrupciones*****
attachInterrupt(5, inicioX, RISING); //Interrupcion externa pin 18
attachInterrupt(1, inicioY, RISING); //Interrupcion externa pin 3
attachInterrupt(4, magSec, CHANGE); //Interrupcion externa pin 19

// Luces y Seguros*****

pinMode(pinLedVerde, OUTPUT);
digitalWrite(pinLedVerde, LOW);
pinMode(pinLedRojo, OUTPUT);
digitalWrite(pinLedRojo, LOW);
pinMode(pinLedAzul, OUTPUT);
digitalWrite(pinLedAzul, LOW);
pinMode(pinLedBlanco, OUTPUT);
digitalWrite(pinLedBlanco, LOW);
pinMode(pinLock, OUTPUT);
digitalWrite(pinLock, LOW);

// Pin Motores *****
pinMode( pinEnableX, OUTPUT);
pinMode( pinDirX , OUTPUT);
pinMode( pinStepX , OUTPUT );
pinMode( pinEnableY, OUTPUT );
pinMode( pinDirY , OUTPUT);
pinMode( pinStepY , OUTPUT);

//*****
digitalWrite(pinLedBlanco, HIGH);

movMot(HIGH, HIGH, 1000, 1000);

digitalWrite(pinLedBlanco, LOW);

// Servo *****

```

```

servo1.attach(pinServo1); //ACTUADOR
servo2.attach(pinServo2); //COMPUERTA
servo1.write(145);
servo2.write(160);
delay(1000);
servo2.write(66);
delay(1000);

//*****Ventilador *****
temperatura = (5.0 * analogRead(LM35) * 100) / 1024;
Input = temperatura;
Setpoint = 25;
warning = Setpoint + 5;

myPID.SetMode(AUTOMATIC);

pinMode(pinVent, OUTPUT);
pinMode(LM35, INPUT);

// Encerado*****

digitalWrite( pinDirX , LOW); // Direccion Motor X para Retroceder
digitalWrite( pinDirY , LOW); // Direccion Motor Y para Retroceder

//Retroceder 10000 Steps

digitalWrite(pinLedBlanco, HIGH);

movMot(LOW, LOW, 10000000, 10000000);

digitalWrite(pinLedBlanco, LOW);

Serial2.print("\x02\x00\n"); //arduino listo
}

```

```

//*****
//***** L O O P *****
//*****

void loop() {

  int senalSerial = 0;

  long i = 0;

  String senal;

  byte senalHex[2];

  digitalWrite( pinStepX , LOW); // Inicializacion de Step X
  digitalWrite( pinStepY , LOW); // Inicializacion de Step Y

  //CONTROL TEMPERATURA
  ventilador();

  //LEER SERIAL
  if (Serial2.available()) {
    senal = Serial2.readString();
    codSenal(senal, &senalHex[0]);

    switch (senalHex[0]) {
      case 0x6:
        medirTemperatura();
        break;
      case 0x7:
        resetFunc();
        break;
      case 0x8:
        unlock();
        break;
      case 0x9:
        controlLuces(true);
        break;
      case 0xA:
        controlLuces(false);
        break;
    }
  }
}

```

```

case 0xE:
    stepper_ini(senalHex[1]);
    break;
case 0xF:
    stepper(senalHex[1]);
    break;
case 0x10:
    regresar();
    break;
case 0x11:
    cargarMed(true, senalHex[1]);
    break;
case 0x12:
    cargarMed(false, senalHex[1]);
    break;
}
}
}

// MOV INICIAL MOTORES*****
void stepper_ini(byte movimiento) {

    long stepsX = 0;          // Inicializacion de Pasos X
    long stepsY = 0;          // Inicializacion de Pasos Y

    stepsX = floor(movimiento / 16) * numPasosX;
    stepsY = (movimiento % 16) * numPasosY;

    digitalWrite(pinLedBlanco, HIGH);
    long i = 0;

    Serial2.print("\x03\x00\n"); //OK

    movMot(HIGH, HIGH, stepsX + x0, stepsY + y0);

    //Mover servol

```

```

servo1.write(25);

delay(1000);

servo1.write(145);

delay(1000);

boolean hayMed = sensor();
if (hayMed) //Comprobar si cayo medicina
  digitalWrite(pinLedVerde, HIGH);
else {
  movMot(HIGH, LOW, 0, numPasosY / 13);
  servo1.write(25);
  delay(1000);
  servo1.write(145);
  delay(1000);
  hayMed = sensor();
  if (hayMed) {
    digitalWrite(pinLedVerde, HIGH);
    movMot(HIGH, HIGH, 0, numPasosY / 13);
  }
  else {
    movMot(HIGH, HIGH, 0, 2 * numPasosY / 13);
    servo1.write(25);
    delay(1000);
    servo1.write(145);
    delay(1000);
    hayMed = sensor();
    if (hayMed) {
      digitalWrite(pinLedVerde, HIGH);
      movMot(HIGH, LOW, 0, numPasosY / 13);
    }
    else {
      digitalWrite(pinLedRojo, HIGH);
      movMot(HIGH, LOW, 0, numPasosY / 13);
    }
  }
}
}
}

```

```

//Definir pasos regreso
long pasosEntX = stepsX - entregaX;
long pasosEntY = stepsY - entregaY;

boolean dirX, dirY;
if (pasosEntY < 0) {
  dirY = HIGH;
  pasosEntY = -pasosEntY;
}
else
  dirY = LOW;

if (pasosEntX < 0) {
  dirX = HIGH;
  pasosEntX = -pasosEntX;
}
else
  dirX = LOW;

movMot(dirX, dirY, pasosEntX, pasosEntY);

digitalWrite(pinLedBlanco, LOW);
digitalWrite(pinLedVerde, LOW);
digitalWrite(pinLedRojo, LOW);

if (hayMed) {
  //Mover servo2
  servo2.write(160);
  delay(1000);
  servo2.write(66);
  delay(1000);
  Serial2.print("x0Cx00\n"); //Recepcion exitosa
}

```

```

else Serial2.print("x0Dx00\n"); //NO HAY MEDICAMENTO

}

// MOV NORMAL MOTORES*****
void stepper(byte movimiento) {

    long i = 0;

    long stepsX = 0;          // Inicializacion de Pasos X
    long stepsY = 0;          // Inicializacion de Pasos Y
    boolean dirX, dirY;

    stepsX = floor(movimiento / 16) * numPasosX;
    stepsY = (movimiento % 16) * numPasosY;

    //Definir pasos regreso
    long pasosPosX = stepsX - entregaX;
    long pasosPosY = stepsY - entregaY;
    if (pasosPosY < 0) {
        dirY = LOW;
        pasosPosY = -pasosPosY;
    }
    else
        dirY = HIGH;

    Serial2.print("x03x00\n"); //OK

    digitalWrite(pinLedBlanco, HIGH);

    if (pasosPosX < 0) {
        dirX = LOW;
        pasosPosX = -pasosPosX;
    }
    else
        dirX = HIGH;

```

```

movMot(dirX, dirY, pasosPosX, pasosPosY);

//Mover servo1
servo1.write(25);
delay(1000);
servo1.write(145);
delay(1000);

boolean hayMed = sensor();
if (hayMed) //Comprobar si cayo medicina
  digitalWrite(pinLedVerde, HIGH);
else {
  movMot(HIGH, LOW, 0, numPasosY / 13);
  servo1.write(25);
  delay(1000);
  servo1.write(145);
  delay(1000);
  hayMed = sensor();
  if (hayMed) {
    digitalWrite(pinLedVerde, HIGH);
    movMot(HIGH, HIGH, 0, numPasosY / 13);
  }
  else {
    movMot(HIGH, HIGH, 0, 2 * numPasosY / 13);
    servo1.write(25);
    delay(1000);
    servo1.write(145);
    delay(1000);
    hayMed = sensor();
    if (hayMed) {
      digitalWrite(pinLedVerde, HIGH);
      movMot(HIGH, LOW, 0, numPasosY / 13);
    }
    else {
      digitalWrite(pinLedRojo, HIGH);
      movMot(HIGH, LOW, 0, numPasosY / 13);
    }
  }
}

```

```

    }
  }
}

//Definir pasos regreso
long pasosEntX = stepsX - entregaX;
long pasosEntY = stepsY - entregaY;
if (pasosEntY < 0) {
  dirY = HIGH;
  pasosEntY = -pasosEntY;
}
else
  dirY = LOW;

if (pasosEntX < 0) {
  dirX = HIGH;
  pasosEntX = -pasosEntX;
}
else
  dirX = LOW;

movMot(dirX, dirY, pasosEntX, pasosEntY);

digitalWrite(pinLedVerde, LOW);
digitalWrite(pinLedRojo, LOW);
digitalWrite(pinLedBlanco, LOW);

if (hayMed) {
  //Mover servo2
  servo2.write(160);
  delay(1000);
  servo2.write(66);
  delay(1000);
  Serial2.print("x0Cx00\n"); //Recepcion exitosa
}
else Serial2.print("x0Dx00\n"); //NO HAY MEDICAMENTO

```

```

}

// REGRESAR*****
void regresar() {
    long i = 0;

    Serial2.print("x03x00\n"); //OK //OK

    digitalWrite(pinLedBlanco, HIGH);

    movMot(LOW, LOW, 10000000, 10000000);

    digitalWrite(pinLedBlanco, LOW);
}

// VENTILADOR*****
void ventilador()
{
    Input = temperatura = (5.0 * analogRead(LM35) * 100) / 1024;

    if (temperatura > warning) {
        Serial2.print("x05x" + String(byte(temperatura), HEX) + "\n"); //warning temperatura
    }

    myPID.Compute();
    analogWrite(pinVent, Output);
}

// SENSOR MEDICAMENTO*****
boolean sensor() {
    int fotodiodo = analogRead(pinFotoDiodo);
    if (fotodiodo >= limiteFotDio)
    {
        return true;
    }
    if (fotodiodo < limiteFotDio)
    {

```

```

    return false;
}
}

// ABRIR PUERTA TRASERA*****
void unlock() {
    digitalWrite(pinLock, HIGH);
    delay(1200);
    digitalWrite(pinLock, LOW);
    Serial2.print("x03x00\n"); //OK
}

// TEMPERATURA*****
void medirTemperatura() {
    temperatura = (5.0 * analogRead(LM35) * 100) / 1024;
    Serial2.print("x04x" + String(byte(temperatura), HEX) + "\n"); //enviar temperatura
}

//CONTROL LUCES LATERALES *****
void controlLuces(boolean estado) {
    if (estado)
        digitalWrite(pinLedAzul, HIGH);
    else
        digitalWrite(pinLedAzul, LOW);

    Serial2.print("x03x00\n"); //OK
}

//CONTROL LUCES POSTERIORES *****
void cargarMed(boolean estado, int posicion) {

    int posX = floor(posicion / 16);
    int posY = posicion % 16;

    digitalWrite(pinesLucesX[posX], estado);
    digitalWrite(pinesLucesY[posY], estado);
}

```

```

Serial2.print("\x03\x00\n"); //OK
}

//CONVERSION STRING A HEX *****
void codSenal(String senal, byte *senalHex) {
    String numero1 = ('0' + senal.substring(0, 3));
    String numero2 = ('0' + senal.substring(3, 6));
    const char* numero11 = numero1.c_str();
    const char* numero21 = numero2.c_str();
    senalHex[0] = strtol (numero11, NULL, 16);
    senalHex[1] = strtol (numero21, NULL, 16);
}

// INTERRUPCION X*****
void inicioX()
{
    interrupcionX = true;
}

// INTERRUPCION Y*****
void inicioY()
{
    interrupcionY = true;
}

// SEGURO MAGNETICO*****
void magSec()
{
    byte estadoSec = byte(digitalRead(19)); //pin seguro magnetico
    Serial2.print("\x13x" + String(estadoSec) + "\n"); //CAMBIO SEGURO MAGNETICO
}

//MOVIMIENTO MOTORES*****
void movMot(boolean dirX, boolean dirY, long stepsX, long stepsY) {
    int delayX = 0;
    int delayY = 0;
}

```

```

long i = 0;

interrupcionX = false;
interrupcionY = false;

digitalWrite( pinDirX , dirX);
digitalWrite( pinDirY , dirY);
digitalWrite( pinStepX, LOW );
digitalWrite( pinStepY, LOW );

long limMay, limMen;
boolean yMen = true, onX, onY;

if ((stepsX / 2) > stepsY) {
    limMay = stepsX / 2;
    limMen = stepsY;
    yMen = true;
}
else {
    limMay = stepsY;
    limMen = stepsX / 2;
    yMen = false;
}

onX = HIGH;
onY = HIGH;

for ( i = 0; i < limMay; i++) {

    if (i < 5000) {
        digitalWrite( pinStepX, onX );
        digitalWrite( pinStepY, onY);
        delayMicroseconds(50);
        digitalWrite( pinStepX, LOW );
        delayMicroseconds(50);
        digitalWrite( pinStepX, onX );
        digitalWrite( pinStepY, LOW );
        delayMicroseconds(50);
    }
}

```

```
digitalWrite( pinStepX, LOW );
delayMicroseconds(50);
}
else {
digitalWrite( pinStepX, onX );
digitalWrite( pinStepY, onY );
delayMicroseconds(42);
digitalWrite( pinStepX, LOW );
delayMicroseconds(42);
digitalWrite( pinStepX, onX );
digitalWrite( pinStepY, LOW );
delayMicroseconds(42);
digitalWrite( pinStepX, LOW );
delayMicroseconds(42);
}
if ((interrupcionX) || ((i >= limMen - 1) && !yMen) ) onX = LOW;
if ((interrupcionY) || ((i >= limMen - 1) && yMen) ) onY = LOW;
if (onX == LOW && onY == LOW) break;
}
}
```