

**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**Control y monitoreo de cargo buses: Investigación e implementación de un sistema de comunicación y mando**

**Andrés Benjamín Paladines Andrade**

Tesis de grado presentada como requisito para la obtención del título de  
Ingeniero en Electrónica de Control.

Quito, agosto de 2008

**Universidad San Francisco de Quito  
Colegio Politécnico**

**HOJA DE APROBACIÓN DE TESIS**

**Control y monitoreo de cargo buses: Investigación e  
implementación de un sistema de comunicación y mando**

**Andrés Benjamín Paladines Andrade**

Laurent Sass. Ph.D.  
Director de la Tesis y  
Miembro del Comité de Tesis

.....

Santiago Navarro. Ph.D.  
Miembro del Comité de Tesis

.....

René Játiva. Ph.D.  
Miembro del Comité de Tesis

.....

Fernando Romo. M.S.  
Decano del Colegio Politécnico

.....

Quito, agosto de 2008

© Derechos de autor

Andrés Benjamín Paladines Andrade

2008

## **Dedicatoria**

El presente trabajo está dedicado a toda la gente que me supo apoyar incondicionalmente y que me sigue apoyando durante el transcurso de mi vida. Sin su respaldo mis logros no tendrían sentido.

## **Agradecimientos**

Está de más agradecer a todas y cada una de las personas que hicieron posible la realización de esta obra, sin su apoyo y entrega culminar con esta tarea hubiera sido algo casi imposible de lograr. Entre estos personajes tan importantes vale resaltar a mis queridos padres que con su confianza, paciencia y esmero supieron alentarme para alcanzar esta meta tan anhelada, agradezco su preocupación día a día, su interés por mi avance personal y profesional y por el sin número de actos que me permitieron llegar a ser lo que soy. Agradezco de igual forma a mi guía profesional y director de tesis Dr. Ing. Laurent Sass por su respaldo y supervisión incondicional, el fue parte del desarrollo de este gran logro y definitivamente un gran ejemplo a seguir. Con igual importancia agradezco al Ing. Rodrigo Díaz que me brindó esa mano desinteresada durante todos y cada uno de los momentos que tomó la realización de esta tesis, contar con ese tipo de apoyo es algo incalculable que todo estudiante realmente necesitaría tener. Finalmente agradezco a todos mis amigos, familiares y demás personas que aportaron emocionalmente para que pueda culminar el proyecto.

## Resumen

EL presente trabajo está enfocado en crear un sistema práctico y efectivo para comunicar, controlar y monitorear una red de transporte overhead o grúas suspendidas automatizadas.

Un sistema de grúas suspendidas permite trasladar carga de un lugar a otro eficientemente. Al elevar la carga y trasladarla, los cargo buses permiten optimizar el espacio y evitar la manipulación innecesaria de los objetos. Una aplicación que se da a este tipo de grúas es en el ensamblaje de productos de gran tamaño y peso considerable.

Dentro de una línea de ensamblaje encontramos estaciones de trabajo en las cuales uno o varios operadores ensamblan ciertas piezas o llevan a cabo ciertos procesos. El movimiento del producto a través de la línea de ensamblaje es decir de estación a estación de ensamblado debe ser automatizado para mejorar la producción, y esto se logra con una intercomunicación de las estaciones y un control central. Además, para una empresa es necesario monitorear el conjunto de estaciones y el movimiento de los cargo buses a través del circuito para posteriormente llevar a cabo cálculos de productividad y controlar la producción.

En resumen, este proyecto logra intercomunicar y monitorear un definido número de estaciones además de automatizar y monitorear el movimiento de los cargo buses a través del circuito.

## **Abstract**

This work focuses on creating a practical and effective system to communicate, control and monitor an overhead suspended transportation network also known as suspended, automated cranes.

A system of suspended cranes allows for the efficient transportation of a load from one place to another. When the load is first elevated and then transferred, it allows for optimization of space and avoids unnecessary manipulation of objects. An application of this type of cranes appears in the assembly of products of important size and considerable weight.

Within an assembly line, one finds workstations in which one or several operators assemble certain pieces or carry out processes. The movement of the product through the line of assembly, or from one station to another, should be automated to improve the production, and this is obtained through communication between the stations and a control center.

In addition, for a company, it is necessary to monitor the set of stations and the movement of the suspended cranes through the circuit to then carry out calculations of productivity and control the production.

In summary, this project achieves the intercommunicating and monitoring of a defined number of stations while automating and monitoring the movement of the suspended cranes through the circuit.

# Tabla de Contenido

<b>Tabla de contenido</b> .....	viii
<b>Lista de figuras</b> .....	xii
<b>Introducción</b> .....	1
1.1 Sistemas de transporte overhead.....	2
1.2 Ciclo de movimiento de los cargobuses.....	3
1.3 Módulos de un sistema de cargobuses.....	5
1.4 Objetivo Principal del Proyecto.....	6
1.5 Objetivos Específicos.....	7
<b>Capítulo 2</b>	
<b>2. Base teórica</b> .....	8
2.1 Descripción del sistema de comunicación implementado.....	8
2.2 Comando de los módulos del sistema .....	14
2.2.1 ¿Que es un microcontrolador?.....	15
2.2.2 Consideraciones eléctricas para utilizar los microcontroladores PIC. ....	16
2.3 Medios de transmisión de datos y estándares de comunicación.....	18
2.3.1 Transmisión de datos por cable.....	18
2.3.1.1 Cable coaxial.....	19
2.3.1.2 Par trenzado.....	21
2.3.1.3 Fibra óptica.....	27
2.3.2 Transmisión de datos por infrarrojo.....	31
2.3.2.1 Tipos de transmisión infrarroja.....	33
2.3.3 Transmisión de datos por radio.....	34
2.3.3.1 ¿Cuándo utilizar radiofrecuencia?.....	35
2.3.3.2 Requisitos que debe cumplir un sistema de comunicación RF para ser fiable. ....	36
2.3.3.3 Factores a considerar en una transmisión de radio.....	37
2.3.3.4 Componentes de un link de radio.....	38



2.3.4 Comunicación Serial.....	41
2.3.4.1 ¿Qué es la comunicación serial?.....	41
2.3.4.2 Características principales de la comunicación.....	41
Serial.	
2.3.4.3 Lógica de transmisión para una comunicación .....	43
serial, norma RS232.	
2.3.4.4 Tipos de comunicación serial.....	45
2.3.5 Estándar EIA-485.....	47
2.3.5.1 Medio de comunicación del estandar EIA-485.....	49

### Capítulo 3

<b>3. Comunicación: módulo central – módulos de estaciones y paradas.</b> .....	54
3.1 El módulo de comunicación central, descripción física y .....	57
funcionamiento	
3.1.1 Descripción física del módulo central.....	60
3.1.2 Detalles de construcción y funcionamiento del módulo central....	61
3.2 Módulos de comunicación de estaciones y paradas .....	65
descripción física y funcionamiento.	
3.2.1 Descripción física de los módulos.....	66
3.2.2 Detalles de construcción y funcionamiento.....	68

### Capítulo 4

<b>4. Comunicación: módulo central – módulo de cargobuses.</b> .....	74
4.1 Módulo de comunicación de cargobuses, descripción .....	76
física y funcionamiento.	
4.1.1 Descripción física del módulo.....	79
4.1.2 Detalles de construcción y funcionamiento.....	81

### Capítulo 5

<b>5. Monitoreo del sistema con LabVIEW</b> .....	86
5.1 Conexión Serial entre el módulo central y la computadora.....	86
5.2 El Diagrama de Bloque del programa.....	88
5.2.1 Descripción del diagrama de bloque.....	91

5.3 El Panel de control.....	95
------------------------------	----

## Capítulo 6

6. Conclusiones.....	97
----------------------	----

Anexo A Programas .....	101
-------------------------	-----

A.1 Central de Control.....	102
A.1.1 Programa del PIC16F877A, encargado de controlar el movimiento de los cargobuses y transmitir la información a LabVIEW.	102
A.1.1.1 Algoritmo global del programa.....	111
A.1.1.2 Diagrama de flujo.....	115
A.1.2 Programa del PIC16F877A, encargado de presentar información del sistema en la pantalla LCD.	117
A.1.2.1 Algoritmo global del programa.....	124
A.1.2.2 Diagrama de flujo.....	126
A.1.3 Programa del PIC12F675, encargado de recibir información desde las estaciones y paradas.	127
A.1.3.1 Algoritmo global del programa.....	128
A.1.3.2 Diagrama de flujo.....	129
A.1.4 Programa del PIC12F675, encargado de enviar información hacia las estaciones y paradas.	130
A.1.4.1 Algoritmo global del programa.....	131
A.1.4.2 Diagrama de flujo.....	132
A.2 Estaciones.....	133
A.2.1 Programa del PIC16F628A, encargado de procesar toda las entradas y salidas del módulo.	133
A.2.1.1 Algoritmo global del programa.....	135
A.2.1.2 Diagrama de flujo.....	136
A.3 Paradas.....	138
A.3.1 Programa del PIC16F628A, encargado de procesar toda las entradas y salidas del módulo.	138
A.3.1.1 Algoritmo global del programa.....	139
A.3.1.2 Diagrama de flujo.....	141

A.4 Paradas y Estaciones.....	143
A.4.1 Programa del PIC12F675, encargado de recibir información ....	143
desde la central.	
A.4.1.1 Algoritmo global del programa.....	144
A.4.1.2 Diagrama de flujo.....	145
A.4.2 Programa del PIC12F675, encargado de enviar información ...	146
hacia la central.	
A.4.2.1 Algoritmo global del programa.....	147
A.4.2.2 Diagrama de flujo.....	148
A.5 Cargobus	
A.5.1 Programa del microcontrolador principal PIC16F877A.....	149
A.5.1.1 Algoritmo global del programa.....	160
A.5.1.2 Diagrama de flujo.....	163
A.5.2 Programa del PIC12F675 del primer cargobus encargado.....	169
de intercalar el uso de la frecuencia de transmisión RF.	
A.5.2.1 Algoritmo global del programa.....	170
A.5.2.2 Diagrama de flujo.....	171
A.5.3 Programa del PIC12F675 del segundo cargobus encargado....	172
de intercalar el uso de la frecuencia de transmisión RF.	
A.5.3.1 Algoritmo global del programa.....	173
A.5.3.2 Diagrama de flujo.....	174
<b>Anexo B Hojas de datos.....</b>	<b>175</b>
Microcontrolador PIC16F877A.....	176
Microcontrolador PIC16F628A.....	177
Microcontrolador PIC12F675.....	178
Transceiver SN75176B.....	179
Encoder HT12E.....	181
Decoder HT12D.....	182
<b>Bibliografía.....</b>	<b>183</b>

## Lista de Figuras

Figura 1.1	Línea de ensamblaje automatizada.....	1
Figura 1.2	Sistema de transporte por rieles para ensamblaje.....	2
Figura 1.3	Sistema de grúas suspendidas para ensamblaje.....	3
Figura 1.4	Secuencia de movimiento de los cargobuses.....	4
Figura 2.1	Boceto de la estructura del sistema de comunicación a..... implementar 1	10
Figura 2.2	Boceto de la estructura del sistema de comunicación a..... implementar 2	11
Figura 2.3	Boceto de la estructura del sistema de comunicación a..... implementar 3	11
Figura 2.4	Arquitectura implementada para comunicar los..... módulos de comunicación	13
Figura 2.5	Imagen de un microcontrolador PIC16F877A.....	15
Figura 2.6	Partes de un cable coaxial.....	19
Figura 2.7	Cable coaxial RG-58A/U.....	20
Figura 2.8	Cable coaxial grueso.....	21
Figura 2.9	Cable STP.....	22
Figura 2.10	Conector RJ49 para cable STP.....	23
Figura 2.11	Cable UTP.....	23
Figura 2.12	Conector RJ45.....	24
Figura 2.13	Cable FTP Figura 1.15.....	26
Figura 2.14	Fibras ópticas.....	27
Figura 2.15	Cable de fibra óptica con conector SC-DUPLEX.....	28
Figura 2.16	Fibra óptica tipo multimodo.....	30
Figura 2.17	Fibra óptica tipo monomodo.....	31
Figura 2.18	Comunicación infrarroja entre una estación fija y un AGVS.....	32
Figura 2.19	Elementos fundamentales de un link de radio.....	39
Figura 2.20	Transmisión serial asincrónica.....	43
Figura 2.21	Estructura de envío serial, correspondiente al carácter ..... ASCII "D" (%01000100) a 8N1 (8 bits de datos, 1 bit de	44

parada sin paridad), a 2400bits/seg.

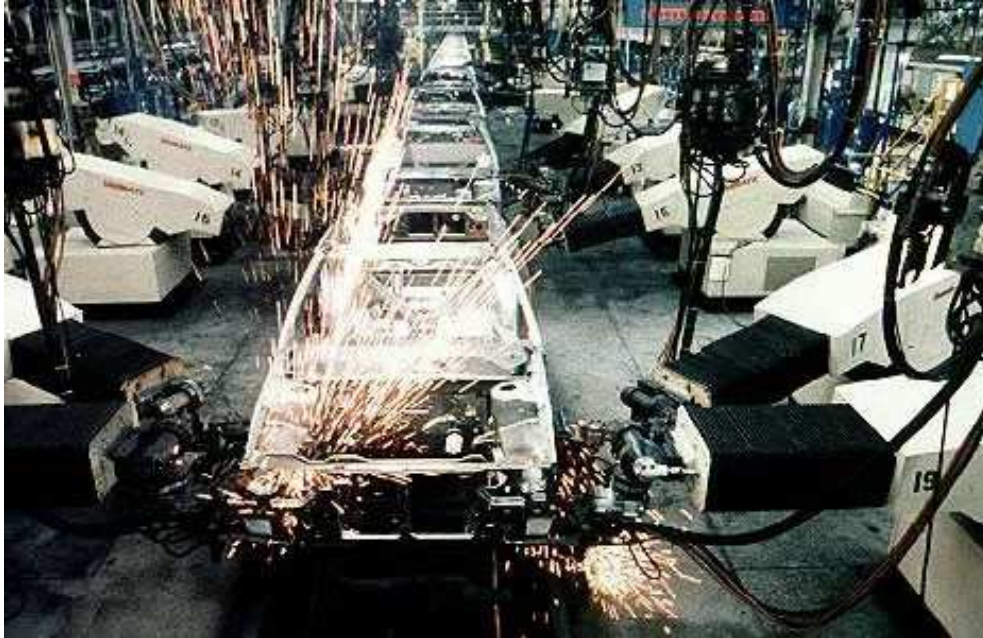
Figura 2.22 Red EIA-485 multidrop de dos conductores.....	50
Figura 2.23 Red EIA-485 multidrop de cuatro conductores.....	51
Figura 2.24 Posible instalación de resistores para minimizar el ruido.....	53
Figura 3.1 Arquitectura utilizada para comunicar el módulo de ..... comunicación central con los módulos de paradas y estaciones.	56
Figura 3.2 Lógica utilizada por la central de control para controlar el..... movimiento de los cargobuses.	59
Figura 3.3 Sistema de comunicación con el módulo central en ..... primer plano.	59
Figura 3.4 Módulo de comunicación central.....	61
Figura 3.5 Diagrama eléctrico del módulo de comunicación central.....	62
Figura 3.6 Modelo básico del sistema previamente armado en ..... Protoboard.	65
Figura 3.7 Módulo de comunicación de estaciones.....	67
Figura 3.8 Módulo de comunicación de paradas.....	68
Figura 3.9 Diagrama eléctrico del módulo de comunicación de estaciones... 69	
Figura 3.10 Diagrama eléctrico del módulo de comunicación de paradas..... 70	
Figura 3.11 Módulos de comunicación de estaciones y paradas ..... conectadas al sistema.	72
Figura 4.1 Representación del cargobus utilizado como esquema para ..... diseñar el módulo de comunicación.	78
Figura 4.2 Encapsulado y diagrama esquemático del IRM860S.....	80
Figura 4.3 Módulo de comunicación de cargobuses.....	81
Figura 4.4 Diagrama eléctrico del módulo de comunicación de ..... Cargobuses.	82
Figura 4.5 Módulo transmisor de RF, FST-3.....	84
Figura 4.6 Módulo receptor de RF, CZS-3.....	84
Figura 4.7 Transmisor de radiofrecuencia FST-3 y receptores ..... de infrarrojo IRM8601S, instalados en la tarjeta.	85

Figura 5.1 Computador personal conectado serialmente al módulo.....	87
de comunicación central a través de un cable UTP 5e.	
Figura 5.2 VISA Configure Serial Port (National Instruments LabVIEW).....	89
Figura 5.3 VISA Read (National Instruments LabVIEW).....	90
Figura 5.4 VISA Write (National Instruments LabVIEW).....	90
Figura 5.5 VISA Close (National Instruments LabVIEW).....	91
Figura 5.6 Diagrama de Bloque .....	94
Figura 5.7 Panel de control programado para el arranque y monitoreo.....	96
del sistema.	

## Introducción

Con el avance de la tecnología, en la actualidad se han automatizado un sinnúmero de procesos industriales. Procesos como: empaclar fundas de alimentos, ensamblar refrigeradoras, construir tarjetas electrónicas de computadoras, etc.

El principal objetivo de explotar la automatización es obtener mayor eficacia en un menor tiempo de producción y, en muchos casos controlar la calidad del producto. En la **figura 1.1** se puede apreciar el uso de la automatización en la industria automotriz; la imagen representa robots soldando piezas en la línea de producción de una fábrica estadounidense.



**Figura 1.1** Línea de ensamblaje automatizada (Biblioteca de Consulta Microsoft ®

Encarta ® 2005. © 1993-2004 Microsoft Corporation)

Uno de los procesos que se automatizan actualmente dentro de las fábricas es el transporte del producto. En muchos lugares se utilizan sistemas overhead para cumplir con este propósito.

## 1.1 Sistemas de transporte overhead

Un sistema overhead (sistema suspendido), en primer lugar eleva la carga y luego la transporta. Con ello se maximiza el espacio de trabajo utilizado y en muchos casos se mejora los tiempos de traslado del producto de etapa en etapa de producción.

En la industria automotriz, por ejemplo para ensamblar vehículos, se utilizan dos métodos de transporte dentro de la línea de ensamblaje: el primero es un gran riel que lleva los vehículos en construcción a nivel del piso, y el segundo es un grupo de grúas suspendidas (sistema overhead) que elevan el vehículo para luego transportarlo. En las **figuras 1.2 y 1.3** se puede observar claramente los dos métodos de transportación.



Figura 1.2 Sistema de transporte por rieles para ensamblaje (**BMW Group México**)



Con base en la información que precede se decidió enfocar la presente tesis en desarrollar un sistema de comunicación para grúas suspendidas, un sistema que permita controlar y monitorear el movimiento de los cargobuses a través de una línea de ensamblaje.

Es así que a continuación se presenta una breve explicación de éste método de transporte para tener una idea clara de las expectativas de la tesis.



**Figura 1.3 Sistema de grúas suspendidas para ensamblaje (motorpasion.com)**

## **1.2 Ciclo de movimiento de los cargobuses**

Los cargobuses o grúas suspendidas siguen un circuito predefinido. Muchas líneas de producción están conformadas por paradas y estaciones de ensamblaje ordenadas consecutivamente.

Se considera una estación como el lugar donde se lleva a cabo un determinado proceso de la cadena de producción, es decir donde se realizan tareas de ensamblaje, montaje de accesorios, pintura, etc. Una estación es un alto obligatorio para un cargobus.

A diferencia de una estación, en una parada no se realiza ningún proceso; una parada no es más que un determinado punto anterior a una estación donde se

puede detener un cargobus. El objetivo de una parada es agilizar el proceso de producción. Si un cargobus está listo para partir de una estación 1, y la siguiente estación 2 está ocupada por otro cargobus, entonces tiene que esperar hasta que ese cargobus la desocupe para poder avanzar. Por otro lado, si se coloca una parada entre las dos estaciones, el cargobus puede avanzar hasta la parada y desocupar la estación 1 para que otro cargobus que se encuentre en fila la utilice.

La lógica de movimiento de los cargobuses es la siguiente:

Todo cargobus parte de una primera parada. No puede haber dos cargobuses en una misma parada o en una misma estación.

Un cargobus puede partir de su posición original siempre y cuando no se encuentre otro en la siguiente posición. Al culminar el circuito, todo cargobus regresa a la parada de inicio. La **figura 1.4** representa el orden del sistema y el sentido de movimiento de los cargobuses.

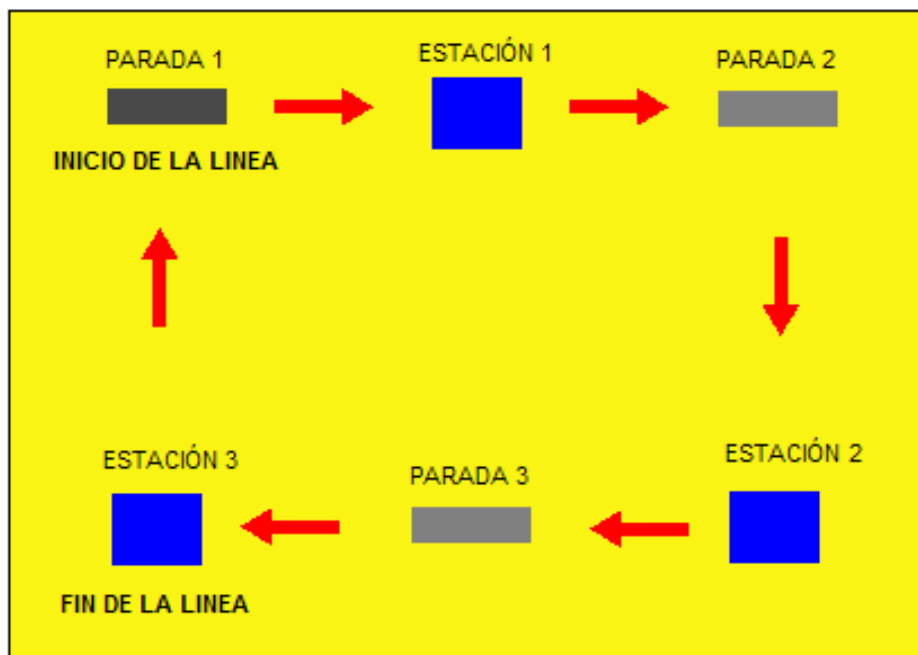


Figura 1.4 Secuencia de movimiento de los cargobuses

Cuando un cargobus llega a una estación, espera que el operario valide el descenso de la carga y luego comienza a bajar el vehículo. Cuando el operario termina de ensamblar las partes que le corresponden, valida el ascenso de la carga y el cargobus eleva la carga nuevamente. Si el cargobus está habilitado para continuar hacia la siguiente parada, parte automáticamente si no está habilitado, espera hasta que la central lo habilite y continúa el ciclo.

El cargobus puede tomar más de una velocidad a través de todo el circuito. Puede escoger entre una velocidad lenta al acercarse a una parada o estación o una velocidad rápida en los tramos más largos del circuito donde no necesita detenerse.

Quien controla el movimiento de los cargobuses es un módulo central que recibe toda la información del sistema, la procesa y ordena el desplazamiento de todos los cargobuses.

Generalmente todo sistema cuenta con una o varias computadoras donde se puede visualizar el estado de los cargobuses a través de la línea.

### **1.3 Módulos de un sistema de cargobuses**

De lo que antecede se puede definir los principales módulos de un sistema de cargobuses o grúas suspendidas:

- Las estaciones de ensamblaje, donde se arman las diferentes piezas del producto y se llevan a cabo diferentes procesos como pintura, suelda y control de fallas.
- Las paradas, donde los cargobuses esperan hasta que las estaciones siguientes se desocupen.

- Un módulo central que controla el movimiento de los cargobuses y de donde se puede monitorear el sistema.
- Los cargobuses que transportan los productos a través de todo el circuito.

La maqueta que forma parte de la tesis tiene nueve módulos de comunicación para simular un sistema real. Los módulos se encuentran distribuidos de la siguiente manera:

- Tres módulos para simular tres estaciones.
- Tres módulos para simular tres paradas.
- Un módulo central de comunicación.
- Dos módulos para simular dos cargobuses.

Adicionalmente se puede monitorear el sistema completo desde un computador personal.

Como parte fundamental de la tesis se implementaron tarjetas electrónicas individuales para la comunicación entre cada uno de los módulos y el control de diferentes procesos como la elevación de la carga en un cargobus.

Para tener en claro los alcances del proyecto, a continuación se presenta el objetivo principal y los objetivos específicos que rigen la tesis.

#### **1.4 Objetivo Principal del proyecto:**

En definitiva la presente tesis tiene como objetivo principal implementar un sistema de comunicación, control y monitoreo de cargobuses o grúas suspendidas automatizadas con el uso de la electrónica de control. En si, la

propuesta no se enfoca en crear cargobuses, si no en comunicarlos y controlarlos para que realicen un trabajo óptimo.

## **1.5 Objetivos específicos:**

El proyecto a desarrollarse permite inferir los siguientes objetivos específicos:

- Determinar que información necesita el cargobus para trasladarse entre paradas y estaciones.
- Encontrar la manera de transmitir dicha información a través de todo el sistema.
- Encontrar una forma eficiente para monitorear el sistema.
- Determinar qué estándar de comunicación industrial es el indicado para interconectar las estaciones y paradas con el módulo central, e identificar un protocolo óptimo para la transmisión de datos a una determinada distancia.
- Registrar la velocidad de cada cargobus a través de toda la línea de ensamblaje.
- Colocar indicadores que representen el ascenso y descenso de la carga en los cargobuses.
- Construir una pequeña maqueta donde se pueda visualizar el funcionamiento del sistema.
- Fabricar todos y cada uno de los módulos de la manera más profesional posible.

## **Capítulo 2**

### **Base teórica**

En este primer capítulo se presenta un resumen de los principales temas relacionados a la tesis, tópicos vinculados al control de módulos y a su comunicación.

Para tener una idea de por qué se seleccionó estos temas como base teórica del proyecto, a continuación se presenta, como primer tema del capítulo, una explicación del funcionamiento del sistema implementado.

Los temas tratados en este capítulo fueron tomados en cuenta tanto en la construcción de cada uno de los módulos como para la comunicación del sistema.

#### **2.1 Descripción del sistema de comunicación implementado**

El objetivo principal de comunicar la estación central con todo un circuito de ensamblaje formado por cargobuses, paradas y estaciones, es que cada cargobus se mueva de forma automática e independiente a través de la línea de ensamblaje. Para ello, la información transmitida desde la central hacia los cargobuses y viceversa debe seguir un camino apto y confiable. A continuación se explica la manera como se implementó la comunicación entre cada módulo del sistema.

Cuando un cargobus llega a una estación, la central de comunicación central debe informarse inmediatamente de que el cargobus ha llegado para

deshabilitar el avance de cualquier otro cargobus que llegue o se encuentre en la parada anterior.

Para ello, la estación detecta la presencia del cargobus y envía esta información hacia la central por cable trenzado UTP usando el estándar de comunicación industrial EIA-485.

De igual forma, cuando un cargobus parte de una estación, ésta informa inmediatamente a la central que se encuentra libre para recibir otro cargobus. Este mismo proceso se da cuando un cargobus llega a una parada.

Cuando un cargobus llega específicamente a una estación, espera a que el operario valide el descenso de la carga. Para que el cargobus reciba esta información desde la estación, debe existir algún tipo de transmisión de datos entre el cargobus y la estación. Es factible utilizar comunicación alámbrica para transmitir la información, si embargo debido a que el cargobus se traslada continuamente a través del circuito y sería imposible mantenerlo conectado todo el tiempo, sería necesario algún dispositivo mecánico que conecte la parada o estación con el cargobus cada vez que éste se estacione, dicho procedimiento incluiría ruido al sistema de comunicación y provocaría el deterioro frecuente de las piezas del dispositivo mecánico. Por lo tanto la forma más efectiva para realizar este proceso es comunicar ambos extremos vía infrarroja.

El cargobus tiene un receptor IR que toma el dato de habilitación del operario y lo envía a un microcontrolador para que luego éste de las ordenes para bajar la carga. De igual forma para subir la carga se lleva a cabo el mismo proceso.

También se utiliza la comunicación infrarroja para ordenar el arranque de un cargobus. Cuando los módulos de parada reciben la habilitación de partida de un cargobus desde el módulo central a través del cable UTP, ellos inmediatamente transmiten esa información vía infrarrojos hacia los cargobuses para que puedan arrancar hacia la siguiente estación.

En las siguientes **figuras** se puede apreciar un boceto de la estructura del sistema que se espera implementar.



Figura 2.1 Boceto de la estructura del sistema de comunicación a implementar 1



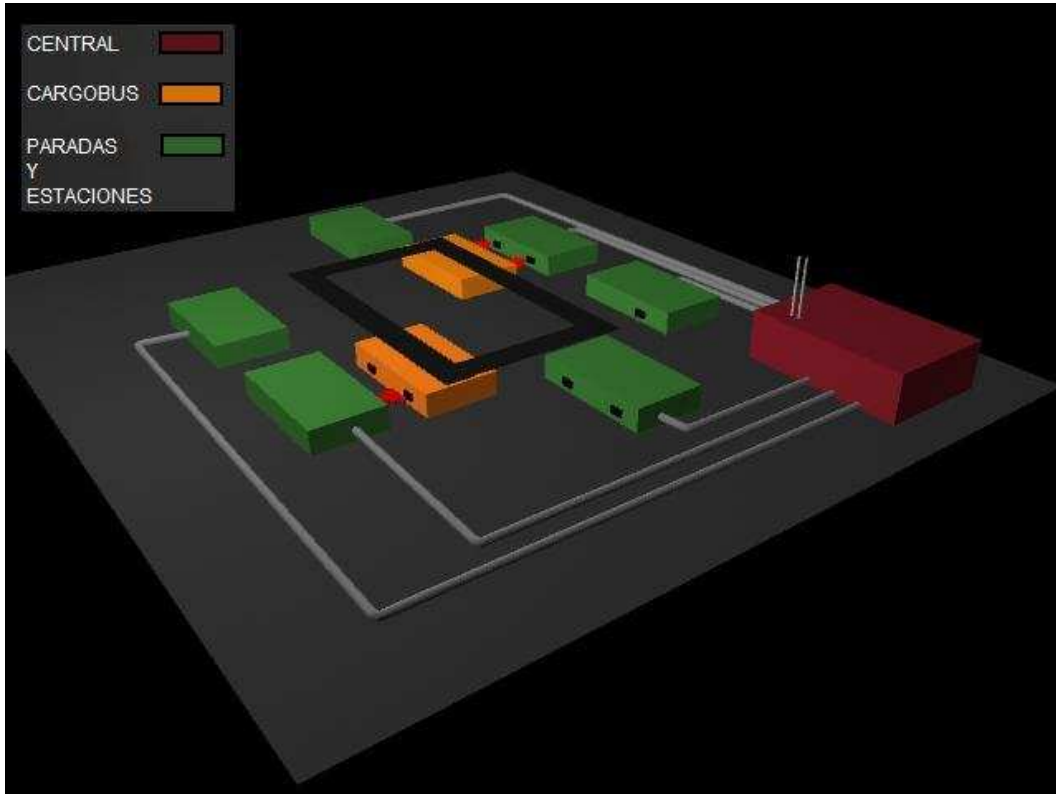


Figura 2.2 Boceto de la estructura del sistema de comunicación a implementar 2



Figura 2.3 Boceto de la estructura del sistema de comunicación a implementar 3

Para poder monitorear todo el sistema, es necesario contar además con la información de estado de los cargobuses en todo momento. Para ello se pudo haber utilizado una camino de transmisión similar al que fue empleado para habilitar la partida de los cargobuses, es decir, enviar la información desde el cargobus a las paradas y estaciones vía infrarrojos y posteriormente a la central por cable, sin embargo esta opción hubiese funcionado únicamente cuando los cargobuses se hubieran encontrado estacionados en frente de estas paradas, pero cuando se hubiesen encontrado trasladándose entre ellas, la comunicación se hubiese perdido y el sistema no hubiese funcionado.

Por esta razón y porque las grúas suspendidas están en continuo movimiento, se optó por utilizar radiofrecuencia como medio de transmisión de información no crítica y así contar con datos del estado de cada cargobus continuamente en la central.

Al momento de monitorear el sistema, se pensó primeramente en utilizar una tarjeta de adquisición de datos para transmitir la información hacia la computadora personal, sin embargo se optó por utilizar una transmisión serial entre el módulo central y el computador ya que cumplía con los objetivos de confiabilidad y velocidad de transmisión necesarios para el proyecto. Con la implementación de esta opción se pudo además economizar grandemente en la construcción de la maqueta.

La comunicación entre el sistema y el computador personal es bidireccional. Además de poder recibir datos en la computadora, se puede también arrancar de forma remota el sistema desde el computador personal.

En el siguiente esquema se puede apreciar la arquitectura implementada para comunicar los módulos de comunicación.

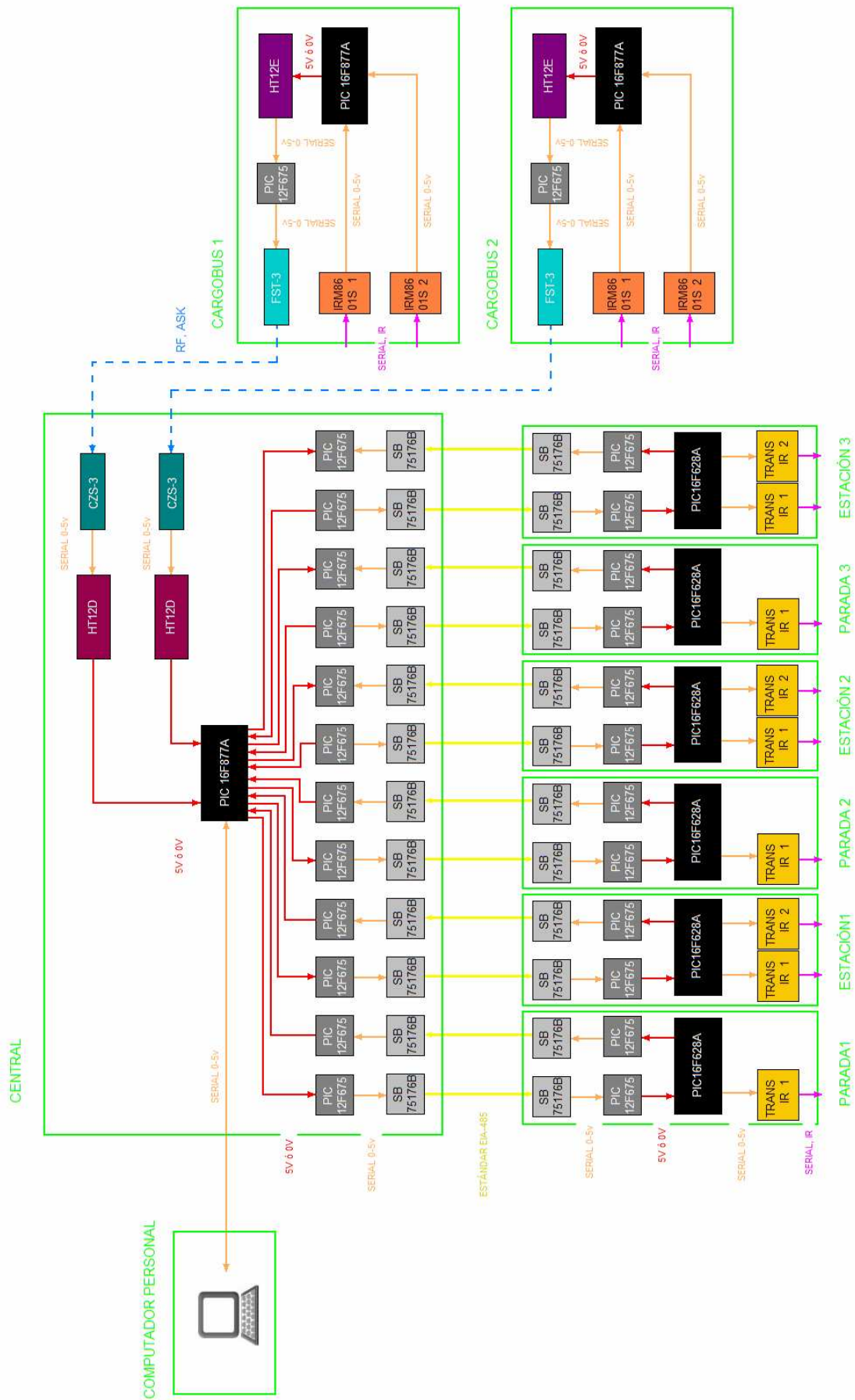


Figura 2.4 Arquitectura implementada para comunicar los módulos de comunicación

Más adelante en el transcurso del texto se explicará a detalle por que se escogió ésta arquitectura y que función cumple cada uno de los elementos del esquema.

## **2.2 Comando de los módulos del sistema**

Antes de comenzar a diseñar cada uno de los módulos, se pensó detenidamente en como controlar cada uno de ellos de una manera eficiente.

Una de las opciones fue utilizar PLC's (Programmable Logic Controllers) para comandar todos los procesos del sistema, un PLC en cada módulo. Sin embargo, luego de constatar que la inversión era relativamente alta y que las funciones en ciertos módulos no justificaban el uso de un PLC, se rechazó esta alternativa.

Para que el diseño sea productivo además de eficiente se decidió construir tarjetas electrónicas separadas para cada módulo, controladas por microcontroladores. Dichas tarjetas son las encargadas de comunicarse entre los módulos del sistema y realizar tareas de control.

Al utilizar microcontroladores se facilitó la construcción de cada módulo. Se evitó utilizar una gran cantidad de elementos discretos y circuitos integrados. Dentro de cada diseño se utilizó por lo menos un microcontrolador. Se los usó principalmente para procesar información de interruptores, recibir información de drivers de comunicación, procesar información para activar o desactivar relés y para crear una interfase entre el computador personal y el resto del sistema.

Para tener una idea clara de por que se utilizaron micros, a continuación se presenta un breve resumen de lo que es un microcontrolador.

### 2.2.1 ¿Que es un microcontrolador?

En términos simples un microcontrolador es un dispositivo que permite diseñar y construir circuitos electrónicos digitales de una manera sencilla sin la necesidad de utilizar gran cantidad de componentes discretos y circuitos integrados. La apariencia de un microcontrolador es la de un circuito integrado o chip. La **figura 2.5** presenta un microcontrolador con encapsulado PDIP (Plastic Dual In-Line Package) de la familia PIC de Microchip.



**Figura.2.5 Imagen de un microcontrolador PIC16F877A (solarbotics.com)**

Se han encontrado una infinidad de usos para los microcontroladores, por ejemplo: en el procesamiento de imagen y sonido, en la automatización y regulación de procesos industriales, en instrumentación, en equipos informáticos como periféricos y controladores, en la fabricación de aparatos para la medicina, etc.

Un microcontrolador posee dentro de su capsula básicamente un microprocesador, una memoria RAM, una memoria ROM que puede ser del tipo FLASH, así como también puertos de entrada y salida, decodificadores, convertidores análogo/digitales, buses de interfaz serie especializados como I<sup>2</sup>C y CAN, temporizadores, UART's, etc. [1].

En el mercado existe una gran variedad de marcas que fabrican diferentes clases de microcontroladores, sin embargo en este proyecto se optó por utilizar PIC's por su facilidad de programación y disponibilidad en el mercado ecuatoriano.

Los microcontroladores PIC (Peripheral Interface Controller) son producidos por Microchip Technology Inc, una de las empresas con mayor éxito en ventas de microcontroladores a nivel mundial [2].

Lo que caracteriza a los PIC's del resto de microcontroladores es la gran versatilidad, variedad, y velocidad de sus modelos, además del bajo consumo de potencia que tienen, su bajo precio y la facilidad de encontrar software para su programación.

Dentro de toda la implementación del proyecto se utilizaron básicamente tres modelos de esta firma, los PIC's 12F675, 16F628A y 16F877A. **En la sección anexos se puede observar las hojas de datos correspondientes a cada PIC.**

## **2.2.2 Consideraciones eléctricas para utilizar los microcontroladores PIC**

Para mantener los microcontroladores funcionando correctamente y para evitar que se dañen, a continuación se presentan una serie de consideraciones eléctricas que se tomaron muy en cuenta para elaborar las tarjetas electrónicas:

- Una de las razones para que un microcontrolador deje de funcionar es la electricidad estática que nuestro cuerpo produce. Para evitar posibles daños por electricidad estática, se utilizó una pinza y una manilla para transportar y manipular cada uno de los microcontroladores.
- Los PIC's utilizados en el proyecto no funcionan correctamente si el voltaje de polarización no es realmente estable. Por ello se utilizaron fuentes de alimentación de computadora para alimentar el sistema y así garantizar las mejores condiciones eléctricas. Además se instalaron capacitores de 1uF en paralelo con la alimentación de cada microcontrolador para evitar cualquier malfuncionamiento.
- La máxima corriente que soportan los microcontroladores PIC12F675, PIC16F628A y PIC16F877A, en cualquier pin de entrada/salida, es de 25mA y pueden entregar una corriente máxima de 25 mA [3].

Con un voltaje de polarización en cada módulo de 5V se utilizaron resistencias de 4.7K $\Omega$  en serie con cada pin para mantener la corriente de entrada dentro de los límites.

De esta forma la corriente que ingresa por cada pin I/O es de 1.06mA, muy por debajo del límite.

- Debido a que los osciladores internos de los microcontroladores no tienen una muy buena precisión, algo necesario en la transmisión de

datos seriales, se utilizaron cristales osciladores externos de 4MHz en todos los microcontroladores.

## **2.3 Medios de transmisión de datos y estándares de comunicación.**

En una planta de ensamblaje las líneas de producción pueden extenderse por largas distancias. Esto conlleva a que, por un lado el alejamiento entre estaciones y paradas sea considerable y por otro lado, que la distancia entre la central de control y todo el circuito de ensamblaje llegue a alcanzar algunos cientos de metros.

Como solución a esta inquietud se pensó en utilizar medios de transmisión muy utilizados en la industria: cable, luz infrarroja y radiofrecuencia, cada uno con sus ventajas y desventajas.

A continuación se presenta un corto resumen de las características más importantes de cada medio de transmisión junto con los estándares de comunicación utilizados.

### **2.3.1 Transmisión de datos por cable**

Existen varios tipos de cables que se utilizan para conectar dos o más dispositivos, entre los más conocidos están:

- Cable coaxial
- Doble par trenzado
- Fibra óptica

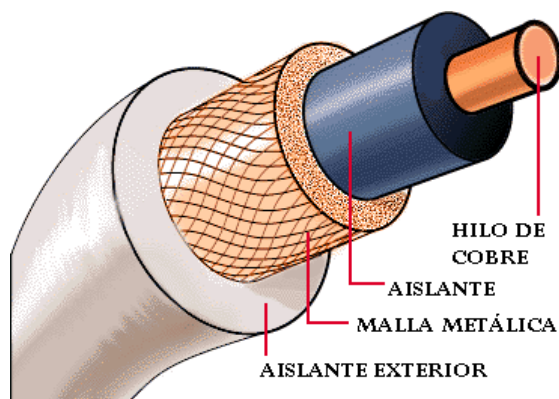


### 2.3.1.1 Cable coaxial

En sus inicios el cable coaxial tenía gran utilidad por su propiedad idónea de transmisión de voz, audio y video. Actualmente se lo utiliza mayoritariamente en redes de datos con topología de bus, como Ethernet y en transmisión de televisión por cable.

El cable coaxial está constituido por dos conductores concéntricos uno central formado por un hilo sólido o trenzado de cobre llamado núcleo, positivo o vivo; y uno exterior con forma de malla tubular de cobre trenzado que sirve como blindaje de ruido a demás de conducir las corrientes de retorno. Los dos conductores están separados por una capa aislante llamada dieléctrico [4].

En la **figura 2.6** se puede apreciar cada una de las partes de un cable coaxial común.



**Figura 2.6 Partes de un cable coaxial (<http://wiki.esliceu.com>)**

Gracias a su blindaje el cable coaxial permite cubrir grandes distancias a altas velocidades de transmisión, velocidades de hasta 10 Mbps.

La mayoría de los cables coaxiales tienen una impedancia característica de 50, 52, 75, o 93  $\Omega$ .

En el mercado comúnmente se utilizan dos tipos de cables coaxiales: el 10Base2 o cable coaxial delgado y el 10Base5 o cable coaxial grueso.

### **Cable coaxial delgado - 10Base2**

Se lo llama también Thinnet o CheaperNet. Como su nombre lo indica es un cable delgado de 6mm de diámetro de color blanco o gris. Se lo puede utilizar en la mayoría de redes por su flexibilidad. Puede transportar una señal por aproximadamente 185 metros con un mínimo de atenuación.

El cable delgado forma parte de la familia RG-58 cuya impedancia es de 50 ohms. Existen diferentes tipos de cables delgados y se diferencian principalmente por el tipo de núcleo que poseen. Por ejemplo el RG-58U contiene un solo hilo de cobre como núcleo mientras que el RG-58A/U tiene alambre trenzado (**Figura 2.7**) [34].



**Figura 2.7 Cable coaxial RG-58A/U (ecvv.com)**

### **Cable coaxial grueso - 10Base5**

Se lo conoce también como Thicknet o Thick Ethernet. Tiene un diámetro de 12 mm y 50 ohms de impedancia. Posee un núcleo más grueso que le permite transportar señales por largas distancias (**figura 2.8**), hasta 500 metros sin repetidores. Se lo utiliza generalmente para crear grandes troncales

de comunicación. Una troncal es la unión de varios segmentos de red en un gran LAN.

Aunque el cable coaxial grueso es caro, pesado, y algo difícil de instalar, es inmune a niveles corrientes de ruido eléctrico, lo que permite conservar la integridad de las señales [5].



**Figura 2.8 Cable coaxial grueso (wikilearning.com)**

### **2.3.1.2 Par trenzado**

El cable de par trenzado, en su forma más simple consiste en dos hilos de cobre o aluminio trenzados dentro de un cordón y cubiertos por un aislante. Generalmente, está compuesto por varios pares trenzados agrupados dentro de un mismo aislante [6].

El objetivo de trenzar el cable es reducir la interferencia eléctrica debido a pares adyacentes u otras fuentes de interferencia como motores, relés, etc. A pesar de ello el par trenzado es más sensible a perturbaciones externas que el cable coaxial, sin embargo es más usado debido a su menor costo, mayor flexibilidad y facilidad de instalación.

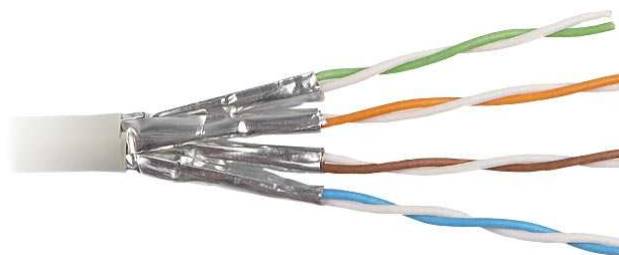
### **Tipos de cable par trenzado**

#### **Cable de par trenzado blindado, STP (Shielded Twisted Pair )**

En el cable STP, cada par trenzado posee una malla o pantalla metálica conductora que lo protege de interferencia electromagnética, ruido eléctrico e

interferencia de radiofrecuencia. En la **figura 2.9** se puede visualizar un cable STP.

Para que la malla actúe eficazmente debe estar interconectada con tierra en ambos extremos. Si la conexión de la malla metálica a tierra no se realiza adecuadamente o si existen discontinuidades en el material a través del cable, el blindaje puede servir como antena para recibir señales no deseadas.



**Figura 2.9 Cable STP (hyperlinesystems.com)**

A demás de tener protección de señales electromagnéticas que intenten ingresar a cada par trenzado, el blindaje minimiza la irradiación de ondas electromagnéticas emitidas desde dentro.

Desafortunadamente el cable STP es un cable grueso de difícil manipulación, lo que dificulta su instalación. Además su precio es algo elevado en comparación a un cable UTP.

El cable STP posee una impedancia de 150 ohms. Se lo suele utilizar en instalación de redes Token Ring porque reduce ampliamente la diafonía. La diafonía, en el caso de cables de pares trenzados se presenta generalmente debido a acoplamientos magnéticos entre los elementos que componen los circuitos perturbador y perturbado o como consecuencia de desequilibrios de admitancia entre los hilos de ambos circuitos [7].

Junto con el cable STP se suele utilizar conectores RJ49 (**Figura 2.10**).



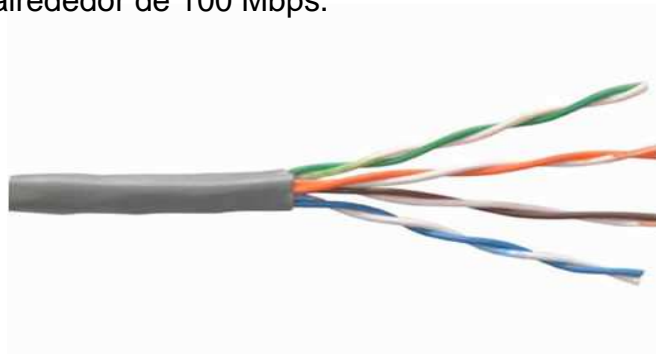
**Figura 2.10 Conector RJ49 para cable STP 34t.com)**

### **Cable de par trenzado sin blindaje, UTP (Unshielded Twisted Pair)**

El cable UTP es el tipo de cable de par trenzado más utilizado en redes locales. Es similar al cable STP pero con la diferencia de que no posee ningún tipo de pantalla metálica que recubra los pares. En la **figura 2.11** se puede visualizar un cable UTP.

Posee una impedancia característica de 100 ohms. El conector más frecuente que se utiliza con cable UTP es el RJ45 (**figura 2.12**), sin embargo también se pueden utilizar conectores RJ11, DB25, DB9, etc., dependiendo del adaptador de red.

La principal debilidad del cable UTP es su susceptibilidad a interferencias, principalmente cuando se transmite información a altas velocidades, velocidades de alrededor de 100 Mbps.



**Figura 2.11 Cable UTP (compustore.cl)**



**Figura2.12 Conector RJ45 (avancetelecom.com)**

Dependiendo de su atenuación, impedancia y capacidad de línea existen diferentes categorías de cable UTP.

### **Categorías del cable UTP:**

#### **Categoría 1:**

Alcanza una velocidad máxima de transmisión de 4 Mbps. Se la utiliza en telefonía tradicional para transmitir voz.

#### **Categoría 2:**

Permite transmitir datos con una velocidad máxima de 4Mbps. Se la emplea generalmente en redes Token Ring. Tiene 4 pares de hilos trenzados.

#### **Categoría 3:**

Tiene un máximo de transmisión de hasta 16Mbps. Se la utiliza para interconectar ordenadores y en redes Token Ring y 10BaseT. Contiene 4 pares trenzados.

#### **Categoría 4:**

Se la utiliza para redes de ordenadores tipo Token Ring con una velocidad de transmisión de 20Mbps. Tiene 4 pares trenzados en su estructura.

#### **Categoría 5:**

Es capaz de soportar comunicaciones de hasta 100Mbps. Se la utiliza en Ethernet (10Mbps), Fast Ethernet (100Mbps) y Token Ring (16Mbps).

Dependiendo de su velocidad de transmisión su atenuación viene dada por la **tabla 2.1** con una distancia referencial de 100 metros. Al igual que las otras categorías tiene 4 pares trenzados.

Velocidad de transmisión de datos	Nivel de atenuación
4 Mbps	13 dB
10 Mbps	20 dB
16 Mbps	25 dB
100 Mbps	67 dB

**Tabla 2.1 Atenuación del cable UTP categoría 5 (hispace.com)**

### **Categoría 5e:**

Es una mejora de la categoría 5 ya que minimiza las características de atenuación e interferencia de su antecesora. Permite una velocidad máxima de 1000 Mbps. Se la aplica en Gigabit Ethernet. Contiene 4 pares de hilos trenzados [8].

En la **tabla 2.2** se puede apreciar la distancia de utilización máxima recomendada para las categorías 3,4 y 5 con respecto al ancho de banda.

Ancho de banda	100 KHz	1MHz	20MHz	100MHz
<b>En categoría 3</b>	2km	500m	100m	no existe
<b>En categoría 4</b>	3km	600m	150m	no existe
<b>En categoría 5</b>	3km	700m	160m	100m

**Tabla 2.2 Distancia máxima del cable UTP categoría 5e con respecto al ancho de banda (hispace.com)**

### **Cable de par trenzado apantallado, FTP (Foil Screened Twisted Pair)**

Al igual que el cable UTP el FTP no posee láminas metálicas que recubren cada uno de los pares, sin embargo tiene una lámina metálica de aluminio que cubre todos los pares a la vez. Con esta pantalla se logra mejorar el nivel de protección ante interferencias externas. La **figura 2.13** muestra un cable FTP con su respectiva pantalla metálica.

Generalmente se utiliza cable FTP para transmitir datos, sin embargo también se lo puede utilizar para transmitir voz. Se suele instalar cable FTP en sistemas de cableado de edificios o en otros ambientes donde el ruido adyacente a los cables puede causar interferencia. El precio de un cable FTP oscila entre el precio de un cable UTP y el precio de un STP [9].

Tiene una impedancia característica de 120 ohms. Utiliza conectores RJ45.

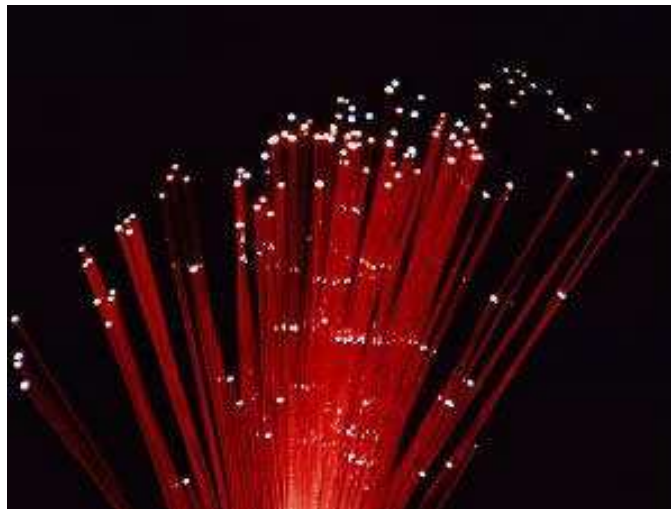


**Figura 2.13 Cable FTP (cablecom.es)**



### 2.3.1.3 Fibra óptica

Un cable de fibra óptica está compuesto por filamentos o fibras ópticas por donde se transmiten señales luminosas. En la **figura 2.14** se puede ver un ejemplo de fibra óptica.



**Figura 2.14** Fibras ópticas (<http://wiki.esliceu.com>)

En la industria de las comunicaciones, el cable de fibra óptica representa una alternativa frente al cable coaxial y el cable de par trenzado. Es así que, un cable con 8 fibras ópticas puede soportar la misma cantidad de datos que 60 cables de 1623 pares de cobre o 4 cables coaxiales; además es mucho más delgado y liviano que sus competidores y puede soportar una mayor distancia entre repetidores [10]. Existen diferentes tipos de conectores específicos para fibra óptica, en la **figura 2.15** se puede observar un cable con conectores de tipo SC-DUPLEX a cada extremo.



**Figura 2.15 Cable de fibra óptica con conector SC-DUPLEX (electronica-basica.com)**

Aunque el uso de la fibra óptica se ha dirigido más hacia comunicaciones de larga distancia, actualmente se hace más común su uso en redes de área local.

#### **Ventajas de la fibra óptica:**

- Es un cable liviano.
- Tiene inmunidad al ruido.
- Presenta baja atenuación.
- Es inmune a interferencias electromagnéticas.
- Conduce rayos luminosos y no señales eléctricas lo que permite su utilización en lugares peligrosos con riesgo de cortocircuitos.
- El ancho de banda es bastante amplio, en teoría de hasta 1THz ( $10^{12}$  Hz).
- Se pueden alcanzar velocidades de transmisión de hasta 10Tbps utilizando técnicas de multiplexación por división de frecuencias, donde cada haz de luz tiene una longitud de onda diferente que transmite a una velocidad de 10Gbps.

### **Desventajas de la fibra óptica:**

- Las fibras que componen el cable son frágiles. Se debe manipular el cable con cuidado.
- El precio de un cable de fibra óptica es relativamente superior al precio de otros cables de comunicación que conducen electricidad.
- En caso de rotura del cable, realizar un empalme es una tarea complicada.
- Al no conducir electricidad no es posible alimentar los repetidores que se encuentran en el camino.
- Los módulos transmisores y receptores son costosos.
- Es necesario efectuar procesos de conversión eléctrica-óptica.

### **Tipos:**

#### **Fibra multimodo**

Una fibra multimodo es aquella donde los haces de luz pueden circular por diferentes caminos y llegar al otro extremo.

El lugar por donde viajan los rayos de luz dentro del cable de fibra óptica se llama núcleo. En una fibra multimodo el diámetro del núcleo es lo suficientemente grande como para permitir que la luz tome varios trayectos a lo largo de la fibra.

La **figura 2.16** representa un ejemplo de los caminos que puede tomar la luz en una fibra multimodo. La luz puede tomar un sinnúmero de rutas para llegar a su destino.

Para emitir las señales se utilizan diodos láser de baja intensidad. Este tipo de fibra se utiliza comúnmente en transmisiones de corta distancia, menores a 1 kilómetro, aunque se pueden utilizar en longitudes de hasta 2 kilómetros.

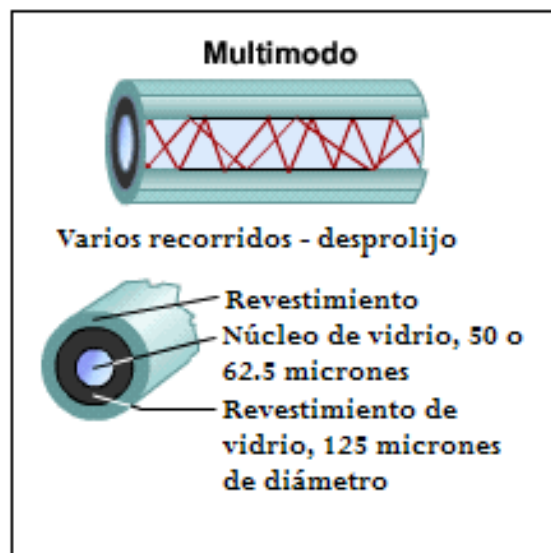


Figura 2.16 Fibra óptica tipo multimodo (fibraoptica hoy.com)

### Fibra monomodo

En una fibra monomodo la luz se propaga en un solo camino. Al reducir el diámetro del núcleo de la fibra a valores de hasta valores de 8.3 micrones, la luz toma un solo camino para su propagación que es en dirección paralela al eje de la fibra. La **figura 2.17** representa una fibra monomodo y el sentido de la luz dentro del núcleo.

Se diferencian de las fibras multimodo en que pueden alcanzar distancias de transmisión de hasta 100 kilómetros. Se utiliza un láser de alta intensidad para lograr la transmisión de datos [11].

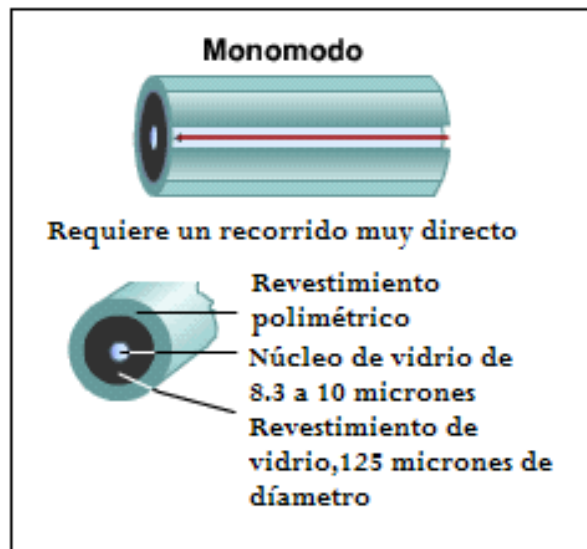
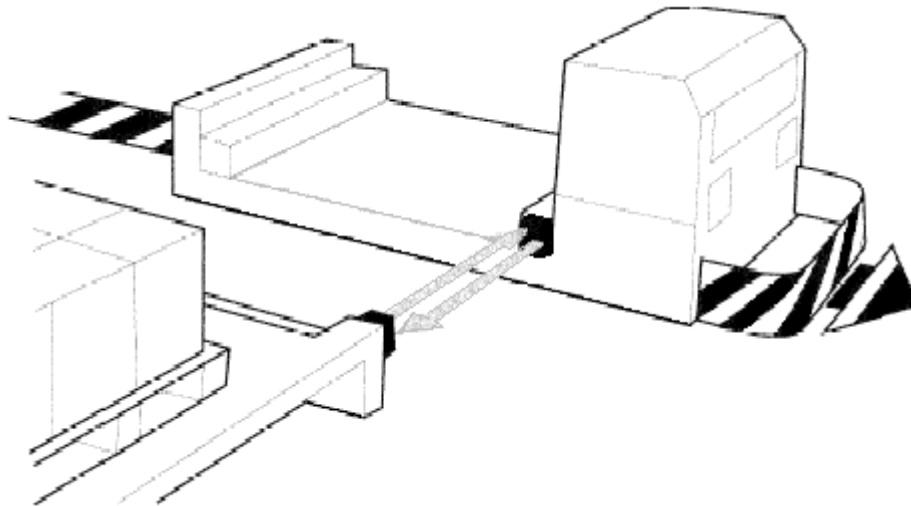


Figura 2.17 Fibra óptica tipo monomodo (fibropticaahoy.com)

### 2.3.2 Transmisión de datos por infrarrojo

En la actualidad son muchos los equipos inalámbricos que incorporan módulos de transmisión y recepción infrarroja. Entre los dispositivos más conocidos se puede encontrar celulares, impresoras, PDA's, módems, cámaras digitales, computadoras portátiles etc.

Debido a que el método de transmisión no envuelve ningún contacto y es insensible a interferencias, la transmisión infrarroja puede ser utilizada en ambientes industriales, en particular donde el movimiento es inevitable. Se puede utilizar éste tipo de transmisión en la comunicación de AGVS (Automatic Guided Vehicle Systems), Sistemas de Vehículos Automáticamente Guiados y evitar molestos cables sostenidos y desgaste de contactos al momento de intercambiar datos entre el AGVS y la estación fija (**figura 2.18**) [22].



**Figura 2.18 Comunicación infrarroja entre una estación fija y un AGVS (leuze.de).**

Se pueden utilizar infrarrojos para transmitir información de un punto a otro y para detectar obstáculos o presencia. Una línea de vista entre el transmisor y receptor es fundamental si se desea tener una transmisión de datos confiable; la distancia entre los módulos y la ubicación de los mismos dependerá de la naturaleza de la transmisión y del tipo de dispositivo a utilizar. Para transmitir la información se puede utilizar LED's de luz infrarroja de diferente potencia de emisión, tipo de lentilla, consumo de energía y frecuencia de operación.

Para recibir los datos es factible utilizar dispositivos ya equipados con elementos sensibles al infrarrojo, lentes, filtros de espectro y toda la lógica necesaria para distinguir señales moduladas de otras radiaciones de infrarrojo; además es más efectivo utilizar receptores que incluyan en su encapsulado las etapas de filtrado, amplificación y detección de errores.

### 2.3.2.1 Tipos de transmisión infrarroja

Existen tres formas de transmitir el rayo de luz para comunicar dos módulos. Los tipos de transmisión son: punto a punto, cuasi difuso y difuso.

En el tipo de **transmisión punto a punto** el emisor envía una señal totalmente direccional hacia el receptor, es decir, las estaciones deben estar ubicadas una frente a la otra para poder realizar la comunicación. Forzadamente debe existir línea de vista entre los módulos.

En el **modo cuasi difuso** la emisión ya no es direccionada sino más bien radial. El transmisor emite la señal en todas direcciones y ésta es reflejada hasta llegar al receptor. Las superficies donde el haz de luz se refleja se clasifican en pasivas y activas. Una superficie pasiva refleja la señal sin modificar su potencia es decir, no modifica ni amplifica la señal; por otro lado una superficie activa recibe la señal y devuelve amplificada. Se puede entender entonces que, para efectuar una transmisión infrarroja cuasi difusa pasiva, la potencia de transmisión del emisor debe ser mayor que la potencia necesaria en una comunicación infrarroja cuasi difusa activa donde la señal se amplifica en las superficies reflectantes.

Finalmente para el tipo de **transmisión difusa**, el haz de luz debe cubrir, en base a múltiples reflexiones, toda la zona donde el receptor se pueda encontrar, es decir, no es necesario tener línea de vista ni tampoco una sola superficie reflectante para entablar la comunicación. Sin embargo este modo de transmisión necesita mayor potencia para lograr mantener una conexión, puesto que un mayor número de rebotes sin llegar al receptor generan pérdidas.

En esta tesis, específicamente al momento de transmitir la información desde los módulos de paradas y estaciones hacia los módulos de cargobuses, se utiliza una transmisión infrarroja de tipo punto a punto. Debe existir línea de vista entre los dos extremos.

### **2.3.3 Transmisión de datos por radio**

Debido a que los cargobuses se encuentran la mayor parte del tiempo en movimiento, se hace realmente complicado utilizar cable para transferir los datos hacia la central. Como solución para ello se decidió usar comunicación por radiofrecuencia para transmitir la información menos crítica.

Dentro de la industria encontramos una infinidad de equipos que nos permiten monitorear y controlar procesos con precisión, facilidad y velocidad. Desde simples interruptores que manejan una señal todo o nada, hasta sofisticados dispositivos que utilizan protocolos de comunicación predefinidos para transmitir la información. Sin embargo todos estos equipos necesitan un medio a través del cual poder comunicarse.

Dentro de una planta la información que se maneja comúnmente corresponde a: señales digitales que indican interruptores activados, motores encendidos, habilitaciones; señales analógicas que proporcionan medidas continuas de presión, temperatura o velocidad; señales de pulsos que sirven para totalizar cantidades y, por último los protocolos de comunicación vía serie como MODBUS, que son empleados por PLC's y otros dispositivos. Toda esta información necesita ser trasladada desde los sensores, interruptores y PLC's hacia centrales de control donde la información será procesada y utilizada



consecutivamente. La radiofrecuencia permite trasladar toda esta información sin la necesidad de ningún tipo de cable de una manera rápida y confiable.

Aunque el medio más común para transmitir información en la industria sea el cable, la radiofrecuencia es una opción utilizada cada vez más en este campo.

### **2.3.3.1 ¿Cuándo utilizar radiofrecuencia?**

Existen ocasiones cuando es factible escoger entre utilizar cable o radiofrecuencia para transmitir la información, sin embargo en ciertos casos la transmisión de datos vía radio es la única manera de comunicar dos o más puntos separados, por ejemplo:

- Cuando la distancia entre equipos es realmente grande, puede ser de varios kilómetros, y se vuelve demasiado costoso utilizar cable para comunicarlos. La radiofrecuencia permite minimizar costos y evita extender grandes longitudes de cable.
- Cuando entre los equipos existe algún tipo de barrera física que sea imposible o muy difícil de atravesar. Por ejemplo: vías de tren, grandes corrientes de agua, espacios privados, etc. Si se emplea radiofrecuencia no es necesario invadir sitios con tendido de cables.
- Cuando uno o varios dispositivos están en movimiento. Si un equipo está en continuo movimiento es muy difícil conectarlo físicamente a un puerto estático. La radio soluciona este problema definitivamente.
- Cuando la instalación de cable representa el paro total o parcial de una maquinaria o dispositivo. Ciertas máquinas tienen un horario de trabajo casi ininterrumpido y realizar modificaciones en la instalación es algo que trae consigo el paro total de la máquina. Por ejemplo en una planta

hidroeléctrica, los sistemas que monitorean el nivel del agua trabajan continuamente para que las grandes turbinas generen energía , es así que realizar una nueva instalación del cableado conllevaría a un paro total del monitoreo. Con la radiofrecuencia no es necesario detener el sistema por completo y además se simplifica realizar modificaciones en los circuitos [23] pp.2

Es muy común dejar de lado la radiofrecuencia cuando se tienen aplicaciones críticas, esto debido a la exposición que tienen los canales de radio a cualquier tipo de interferencia, es por ello que a distancias cortas el cable es el medio más utilizado. Sin embargo cuando las aplicaciones no son críticas un sistema de radio puede funcionar maravillosamente.

### **2.3.3.2 Requisitos que debe cumplir un sistema de comunicación RF para ser fiable.**

Es importante que el canal por el cual se van a transmitir los datos esté libre de interferencias tanto naturales como artificiales. Encontrar la frecuencia ideal es primordial para transmitir información a través de cualquier ambiente. Por un lado las frecuencias bajas son susceptibles a presentar ruido de origen natural e industrial y por el otro las frecuencias altas son mayoritariamente afectadas por transmisiones de televisión, telefonía móvil, microondas, etc. No obstante existe una banda de frecuencias con baja presencia de ruido y poca interferencia que se puede utilizar para transmitir los datos, la banda de frecuencias entre los 400 y 500 MHz es la ideal. Las únicas interferencias importantes producidas dentro de esta banda son las causadas por dispositivos cercanos que utilizan la misma banda de frecuencias. Sin embargo, ciertas

normativas de gobierno restringen la potencia de transmisión de los dispositivos para evitar este tipo de interferencias [23] pp.3

Los equipos utilizados deben ser capaces de evitar la influencia de las señales producidas por otros dispositivos.

Para ello es necesario que el equipo pueda identificar una transmisión correcta y descarte cualquier interferencia producto de la transmisión de equipos cercanos. Esto se logra codificando la información en tramas y transmitiéndola digitalmente e incluyendo códigos de detección de errores. Como ejemplo, para crear una sofisticada red de comunicación inalámbrica se requiere de un sistema de software para control de redes, un protocolo que dirija el acceso a la red y detecte posibles errores [23] pp. 3-4

### **2.3.3.3 Factores a considerar en una transmisión de radio**

Los dispositivos a utilizar deben ser capaces de evitar la influencia del ruido, para ello se debe tener en cuenta los siguientes factores:

#### ***Duración de cada mensaje***

Si la transmisión toma menos tiempo en transmitirse, la posibilidad de que ésta sea afectada por alguna interferencia disminuye. Es por eso que se debe tomar en cuenta la longitud y la velocidad del mensaje para definir el tiempo que durará la transmisión.

#### ***Número de mensajes transmitidos***

Mientras más mensajes se transmitan, más ocupado estará el canal de radio, y por ende la interferencia entre sistemas próximos será mayor. Se

puede utilizar notificación de eventos para disminuir el número de mensajes transmitidos. Con notificación de eventos los paquetes de datos son transmitidos únicamente cuando una de las entradas del dispositivo transmisor detecta un cambio. Sin embargo puede darse el caso de que exista más de un dispositivo transmisor y que uno de ellos quiera transmitir cuando el otro ya lo está haciendo, en dicho caso es necesario que todos los dispositivos tengan la facilidad de detectar cuando otro dispositivo esta transmitiendo y esperar hasta que lo deje de hacer para proceder con la transmisión.

### ***Una detección de errores eficaz***

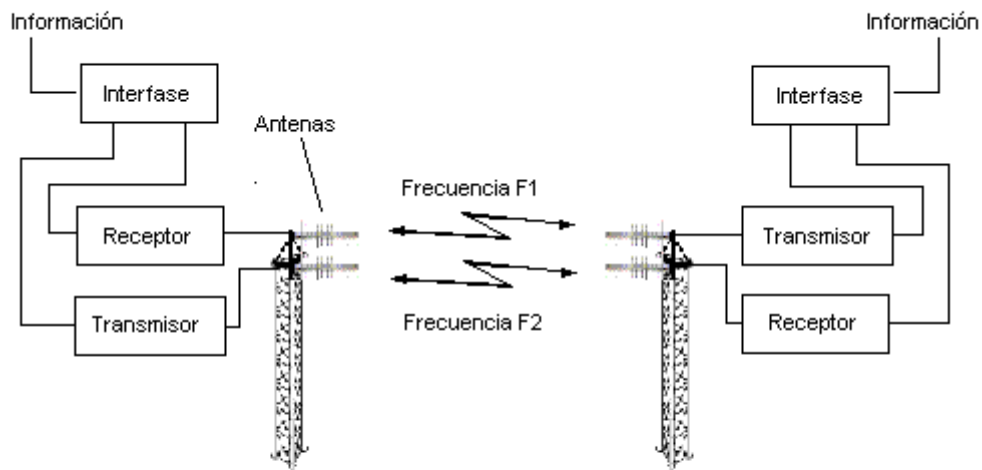
Con una rápida detección de errores se puede corregir fallas en la transmisión o en último caso detectarlas y notificar su presencia para una posterior reparación. Se puede utilizar algún tipo de handshaking para comprobar que los paquetes lleguen a su destino sin ningún tipo de error.

### **2.3.3.4 Componentes de un link de radio**

Un link de radio está constituido por los siguientes elementos:

- Antenas
- Transmisores
- Receptores
- Cableado
- Equipo de interfase

La **figura 2.19** representa claramente cada uno de los componentes nombrados.



**Figura 2.19 Elementos fundamentales de un link de radio full duplex**

A continuación se explica cada una de los elementos que constituyen el link.

### **Antena**

Es el dispositivo utilizado para irradiar o detectar las ondas electromagnéticas. Existen varios diseños de antenas disponibles, cada una de ellas irradia las señales a través de ondas electromagnéticas de diferentes maneras. El tipo de antena a utilizar en cualquier proyecto dependerá de la aplicación y del área de cobertura requerida [19] pp.278

### **Transmisor**

Es el elemento encargado de convertir todos los datos en una sola señal de alta frecuencia modulada o modificada. Luego de convertir la información la envía hacia la antena donde es irradiada al espacio libre como una onda electromagnética a una cierta frecuencia.

## **Receptor**

Es el dispositivo encargado de filtrar la señal recibida de la antena. El receptor luego de seleccionar la parte útil de la señal, la amplifica para posteriormente convertirla de vuelta a un grupo de datos casi idénticos a los enviados por el transmisor.

## **Cableado**

Los cables transportan la información entre los equipos de interfase, receptores, transmisores y antenas. Existen tres principales tipos de cables que se utilizan para conectar sistemas de radio:

- Cable coaxial para todas las conexiones de radio frecuencia.
- Cable par trenzado para transmitir datos.
- Cables de poder

## **Equipo de interfase**

El equipo de interfase permite transmitir los datos hacia los transmisores desde fuentes externas como teclados, interruptores, etc., y desde los receptores hacia fuentes externas como pantallas, leds, etc. Además controla el flujo de la información y permite monitorear los transmisores y receptores [19] pp.279

En ésta tesis se creó un link de radio de tipo simplex, es decir la comunicación entre los dos extremos fluye en un solo sentido. El módulo de cargobuses cumple el papel de transmisor y el módulo central de receptor. En el capítulo 4 se explica a detalle la comunicación RF entre éstos módulos.

## **2.3.4 Comunicación Serial**

### **2.3.4.1 ¿Qué es la comunicación serial?**

La comunicación Serial permite enviar y recibir bytes de información por un solo conductor, pero un solo bit a la vez. Este tipo de comunicación es más lenta que la comunicación paralela que permite transmitir bytes enteros de una sola vez a través de varios conductores, sin embargo el ahorro en el costo de cableado justifica ampliamente su utilización. El uso más común que se da a la transmisión serial es para el envío y recepción de datos ASCII. Existen dos tipos de comunicación serial; sincrónica y asincrónica. Para establecer una comunicación serial sincrónica es necesario enviar conjuntamente con la información una señal de reloj para mantener la transmisión. En la comunicación serial asincrónica la transmisión y recepción de datos se da sin la necesidad de utilizar una señal de reloj [12].

La comunicación serial asincrónica se completa con el uso de tres cables: la línea de transmisión (Tx), la de recepción (Rx) y la de referencia o tierra.

Para establecer una comunicación asincrónica el transmisor y el receptor deberán tener bases de tiempo iguales y niveles de tensión iguales, pudiendo ser compatibles con la lógica TTL/CMOS o voltajes comprendidos entre -25 y 25 voltios DC para el estándar RS-232C.

### **2.3.4.2 Características principales de la Comunicación Serial**

Entre las características principales de la transmisión serial están: la tasa de baudios, los bits de datos, los bits de parada o paro y los bits de paridad.

Para que dos puertos puedan comunicarse serialmente es de vital importancia que estos parámetros sean iguales.

El conjunto de bits de datos se establece generalmente en paquetes de 5, 7 u 8 bits.

La **tasa de baudios** es una unidad de medición definida para la comunicación que indica el número de bits transferidos por segundo. Es decir, 100 baudios representan una tasa de transmisión de 100 bits por segundo. En ocasiones cuando se refiere a ciclos de reloj, se está refiriendo a la tasa en baudios, es por eso que si un protocolo especifica una razón en baudios de 2000, el reloj se está ejecutándose a 2000 Hz. Esto quiere decir que el puerto serial realiza un muestreo de datos en la línea a 2000 Hz. La tasa de baudios es inversamente proporcional a la distancia entre dispositivos.

Los **bits de parada** tienen como finalidad señalar el término de la transmisión de un paquete. Es común utilizar de 1 a 2 bits de parada. Además de indicar el fin de la comunicación, los bits de parada dan un margen de error a las velocidades de reloj de los dispositivos. A medida que se utilizan más bits de parada la sincronización de relojes entre dispositivos debe ser mayor, sin embargo la razón de transmisión de datos es menor, es decir la comunicación se vuelve menos eficiente.

En la comunicación serial, los **bits de paridad** permiten realizar una detección de error simple. Existen cuatro tipos de paridad: pares, impares, marcados y espaciados.

Para paridad par el puerto serial fija en 1 el bit de paridad si el número de unos en el conjunto de bits de datos es impar, de esta forma, el número total de bits 1 (datos mas paridad) es par.



Para paridad impar el puerto serial fija en 1 el bit de paridad si el número de unos en el conjunto de bits de datos es par, de esta forma, el total de bits 1 es impar.

La paridad marcada y la paridad espaciada no revisan los bits de datos para determinar el nivel lógico del bit de paridad, simplemente fijan un nivel alto para la paridad marcada o baja para la paridad espaciada, de esta forma el receptor analiza el estado de un bit para determinar si el ruido ha corrompido la información o si los relojes del dispositivo transmisor y receptor están desincronizados [13].

En la **figura 2.20** se puede distinguir los componentes de una comunicación serial, el bit de inicio, los bits que componen la palabra o bits de datos, el bit de paridad y el bit de paro.

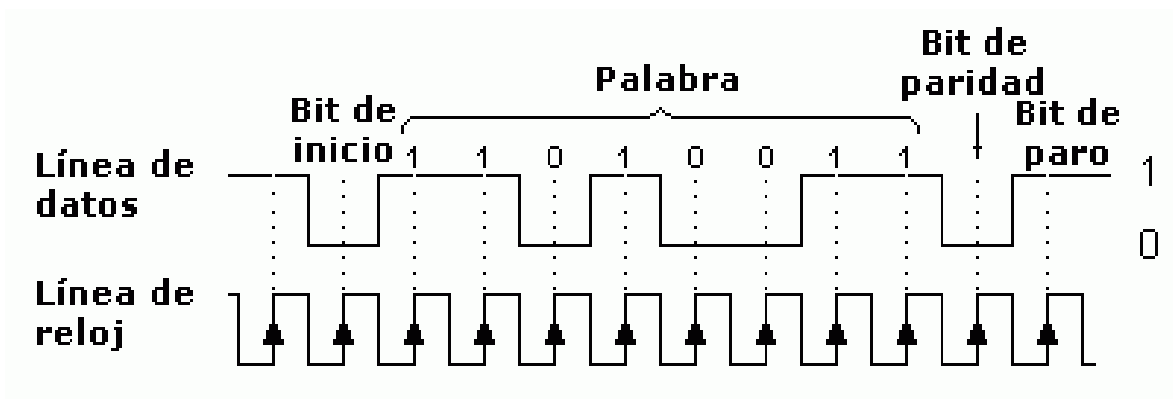
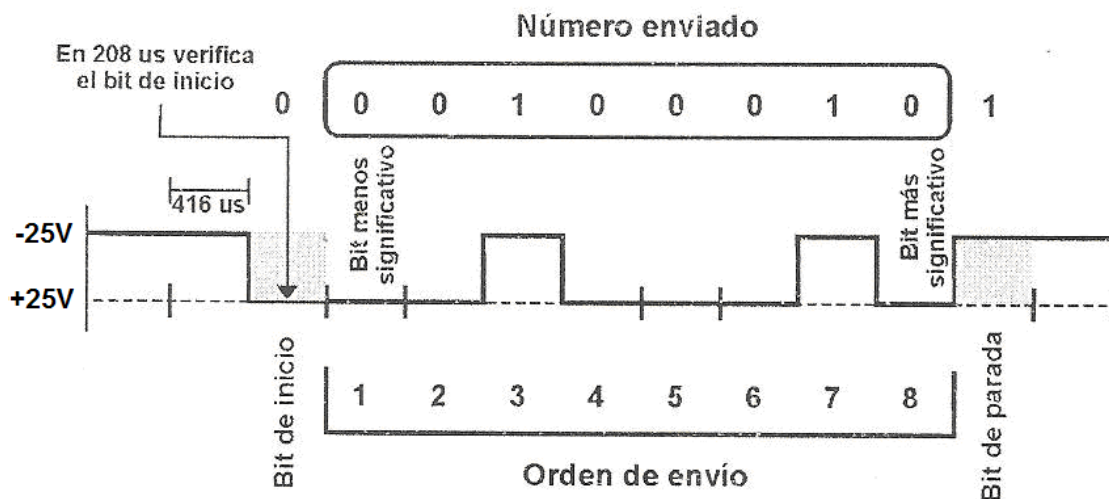


Figura 2.20 Transmisión serial asincrónica (ni.com)

### 2.3.4.3 Lógica de transmisión para una comunicación serial, norma RS232

A continuación se presenta un gráfico que muestra la forma de comunicación serial bajo la norma RS232.



**Figura 2.21 Estructura de envío serial, correspondiente al carácter ASCII "D" (%01000100) a 8N1 (8 bits de datos, 1 bit de parada sin paridad), a 2400bits/seg (Carlos A. Reyes, 128)**

Como se puede ver en la **figura 2.21**, la señal se mantiene en un nivel lógico alto mientras no se transmita ningún dato. Para comenzar a transmitir datos, el transmisor coloca el nivel de la señal en bajo durante el tiempo de envío de un bit (416 us para 2400bits/seg), este es el bit de arranque; a continuación, con el mismo intervalo de tiempo, comienza a transmitir los bits de datos desde los menos significativos hasta los más significativos. Al final de la transmisión de datos se envía el bit de paridad, si existiese, y por último el bit de parada. Una vez transmitida toda la secuencia, la línea vuelve a su estado alto inicial [14].

Debido a que el receptor no está sincronizado con el transmisor, para que una transmisión comience, éste debe estar siempre en espera de un cambio de estado, es decir el bit de arranque o inicio enviado por el transmisor. Una vez que el bit de inicio es recibido, medio bit después (208us), se vuelve a verificar

el nivel bajo de la señal para confirmar que se trata de una transmisión real y no un posible cambio de estado producto de ruido en la línea. Luego de confirmar el estado bajo por el tiempo de un bit, se comienzan a transmitir los datos hasta el bit de parada.

Para que la lectura de los datos sea correcta, ambos equipos, el transmisor y el receptor, deben tener la misma configuración de puerto, es decir la tasa de transmisión, el número de bits de datos, el bit de paridad y el número de bits de parada deben coincidir.

Para transmisiones a distancias superiores a los 2 metros, los datos transmitidos pueden no llegar correctos, en este caso se pueden utilizar estándares de transmisión como EIA-232 cuyos niveles de voltaje para la transmisión son: -5V a -15V en el transmisor y -3V a -25V en el receptor para representar una señal 1 lógica y +5V a +15V en el transmisor y +3V a +25V en el receptor para una señal 0 lógica. La diferencia en el voltaje entre transmisores y receptores se da con el objetivo de permitir un rango de error debido a caídas de voltaje en la línea [15].

Otro estándar de comunicación que se puede utilizar es el EIA-485, el cual será tratado detenidamente más adelante.

#### **2.3.4.4 Tipos de comunicación serial**

Los tipos utilizados de comunicación serial son: Simplex, Duplex, halfduplex o semi duplex y full duplex.

En la **comunicación serial simplex** la transmisión de datos es unidireccional. En este caso los extremos del sistema, transmisor y receptor

están plenamente definidos. El uso más frecuente que se da a este tipo de comunicación es en redes de radio difusión donde el transmisor es el único que envía información a los receptores y estos no necesitan enviar ningún tipo de información al transmisor para satisfacer los requerimientos de comunicación.

En el caso de la **comunicación duplex o half duplex** ambos extremos del sistema cumplen funciones de transmisor y receptor. Los datos se pueden desplazar en ambas direcciones pero no al mismo tiempo, es decir si un extremo cumple las veces de transmisor y el otro extremo las de receptor es necesario que primero se termine de transmitir los datos en ese sentido antes de intercambiar funciones entre extremos y ocupar el canal. La comunicación duplex o half duplex se utiliza en sistemas donde existen varios terminales y solo un controlador central.

La **comunicación full duplex** permite que los datos se desplacen en ambas direcciones al mismo tiempo. En este caso cada extremo cumple el papel de transmisor y receptor. Es indispensable que exista más de un canal para mantener una comunicación full duplex, en otras palabras, si la transmisión de datos es alámbrica se necesitaría de cuatro conductores para entablar la comunicación, o en el caso de una transmisión inalámbrica el uso de dos frecuencias diferentes [16].

### **2.3.5 Estándar EIA-485**

El estándar EIA-485 es uno de los más versátiles estándares EIA (Electronic Industries Alliance). Permite una comunicación serial por largas distancias y de forma segura entre dos o más dispositivos.

EIA-485 es básicamente un estándar de capa física, con especificaciones eléctricas para comunicaciones seriales half-duplex mono o multipunto sobre dos conductores de par trenzado.

Para transmitir la información a través de los conductores utiliza un sistema diferencial balanceado. Se llama balanceado porque la señal en el primer conductor es idealmente la opuesta exacta de la señal en el segundo conductor, es decir, si por un conductor se transmite un nivel alto, por el otro conductor se transmitirá un nivel bajo y viceversa [17].

El EIA-485 es una extensión del EIA-422. EIA-422 es un estándar de transmisión serial que utiliza señalización balanceada o diferenciada para transmitir los datos.

El estándar EIA-422 fue implementado con la finalidad de reducir el ruido en la línea. Al sustituir el cero común como punto de referencia para la transmisión, e implementar como factor determinante del valor de la señal a la diferencia de voltaje entre las líneas, el estándar EIA-422 permitió disminuir el ruido producto de corrientes inducidas y habilitó el uso de mayores tasas de transmisión.

Muchas de las características del EIA-422 se han mantenido y otras han sido modificadas para mejorar el estándar. Las características más importantes que guardan estos dos protocolos en común son la distancia y velocidad de transmisión. Por otro lado en el protocolo EIA-485 se incrementó el número de transmisores y receptores permitidos en la línea.

EIA-485 permite una conexión punto a punto entre dispositivos o una conexión de red de tipo multidrop o multipunto bajo dos conductores, esto quiere decir que, basta con conectar dispositivos en paralelo a los dos cables para que queden adheridos a la red.

Las características principales de transmisión del EIA-485 son:

- Tasa de transmisión de datos de sobre los 10 Mbps, la misma tasa que posee el protocolo EIA-422.
- Distancia de transmisión de sobre los 1200 m o 4000 pies, la misma distancia que abarca EIA-422.
- Conectividad para más de 32 posibles transmisores en la misma línea.
- Conectividad para más de 32 posibles receptores en la misma línea.

Sin embargo, la máxima tasa de transmisión y la máxima distancia no pueden ser alcanzadas al mismo tiempo. Por ejemplo, para un cable trenzado AWG número 24 la tasa máxima de transmisión a la distancia de 1200 m es de aproximadamente 90 kbps; y la máxima distancia que podría cubrir el cable a 10 Mbps es de menos de 6 metros. Es posible mejorar estas cifras si se utiliza cable de mejor calidad y terminales activos de estado sólido al final de las líneas, con el objeto de contrarrestar el ruido [19] pp.53

Como se vio, es factible incorporar a la línea hasta 32 transmisores, sin embargo, algunos fabricantes proveen de dispositivos que representan  $\frac{1}{2}$  o incluso  $\frac{1}{4}$  de dispositivo estándar, con ello se logra incrementar el número de dispositivos a 64 o 128 respectivamente. Si es necesario el uso de una mayor cantidad de transmisores, se pueden utilizar repetidores para ampliar la red.

### 2.3.5.1 Medio de comunicación del estándar EIA-485

A los dos conductores que componen la línea de transmisión se los conoce como A y B. Para cada señal a transferir el par de conductores transmiten, uno la señal original no invertida y el otro la señal invertida. A la línea invertida por regla general se la especifica con el índice A, mientras que a la no invertida con el índice B. Al conductor A también se lo conoce como A -, TxA y Tx+. De igual forma al conductor B también se lo llama B+, TxB y Tx-. El receptor evalúa solamente la diferencia de voltaje existente entre ambas líneas, de modo que perturbaciones en la línea de transmisión no falsifican la señal útil [18].

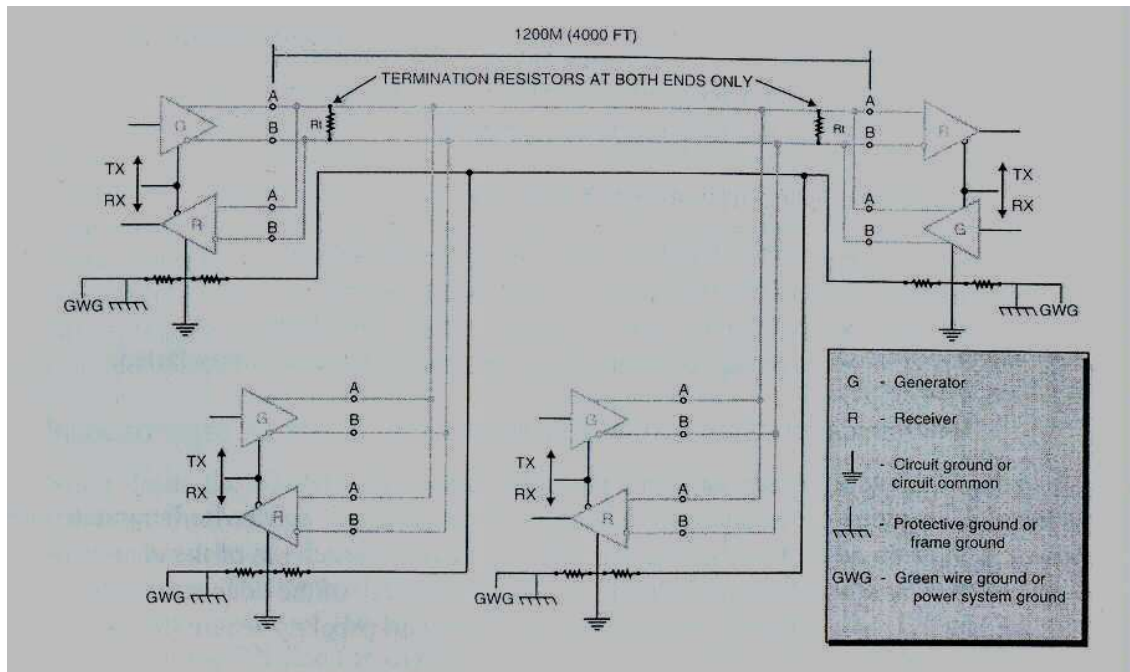
Para identificar cual conductor es el A y cual el B, en el estado apagado o MARK el voltaje en el conductor A es más negativo que el voltaje en el conductor B. En la **figura 2.22** se puede observar los dos conductores mencionados.

La diferencia de potencial entre los dos conductores debe ser de al menos 0.2 voltios positivos o negativos para una operación válida.

Cualquier voltaje comprendido entre +12 voltios y -7 voltios permitirá una correcta operación del receptor.

Sin embargo para mayor confiabilidad en la transmisión la asignación tensión de diferencia al estado lógico se define del modo siguiente:

- De -0.3V a -6V en el terminal A con respecto al B. Representa al 1 binario, estado apagado (OFF) o MARK.
- De +0.3V a +6V en el terminal A con respecto al B. Representa al 0 binario, estado encendido (ON) o SPACE.



**Figura 2.22 Red EIA-485 multidrop de dos conductores (Steve Mackay,55)**

Dentro de las características más importantes del estándar EIA-485 están sus tres estados de operación:

- 1 lógico
- 0 lógico
- Estado de alta impedancia

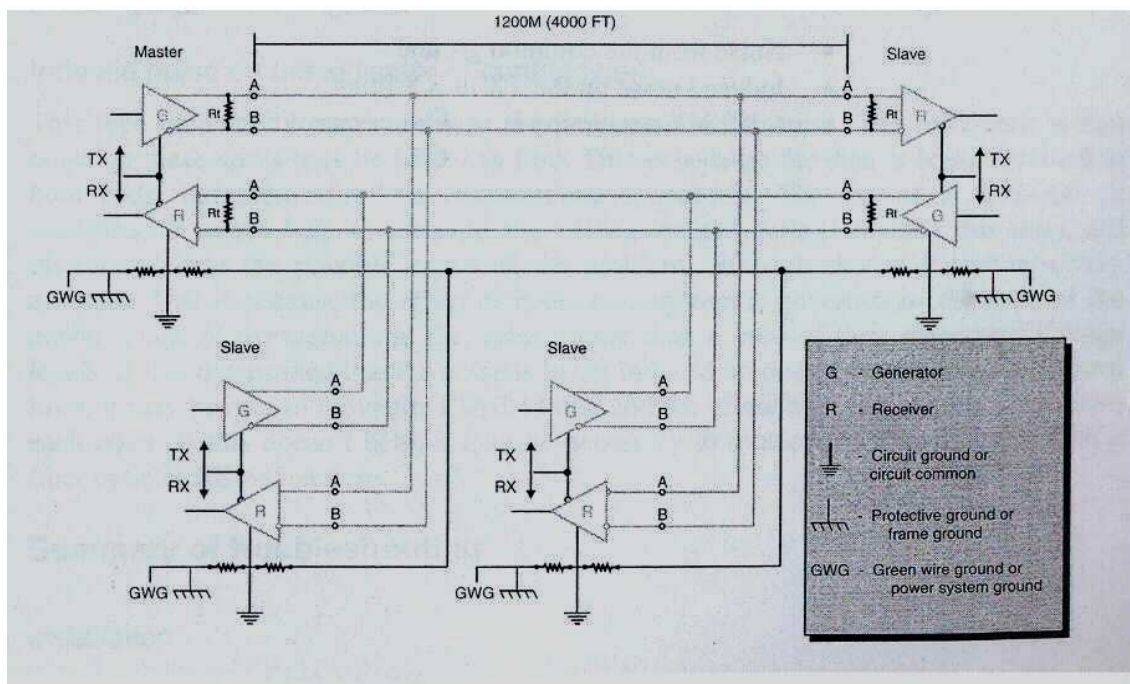
En el estado de alta impedancia la línea del transmisor o receptor, adherida al bus de datos, no presenta ninguna corriente, es decir aparece como si no existiera ningún driver de comunicación. Este estado puede ser inicializado con una señal sobre el pin de control del circuito integrado dentro del driver.

El objetivo de tener estos tres estados en la línea es darle la característica de multidrop a la red. Ya que solo un transmisor puede transmitir a la vez, es necesario que los módulos que no intervienen en la comunicación



permanezcan en un estado de alta impedancia mientras la transmisión toma lugar. Además es necesario que cada terminal en el sistema tenga su única dirección de localización, de esta manera se evitan conflictos con otros dispositivos [19] pp. 54

Además de crear una red con solo dos conductores, EIA-485 también puede ser implementado con cuatro hilos, la **figura 2.23** representa una red de cuatro conductores. Para este tipo de conexión es necesario que un nodo cumpla el papel de maestro y el resto de nodos el de esclavos. El nodo maestro puede comunicarse con todos los nodos esclavos, pero un nodo esclavo solo puede comunicarse con el nodo maestro. De igual forma un nodo esclavo no puede escuchar la respuesta de otro nodo al maestro.



**Figura 2.23** Red EIA-485 multidrop de cuatro conductores (Steve Mackay,55)

Para evitar algunos problemas como: ruido, voltajes de modo común y reflexiones, se suele conectar resistores a la línea.

Para reducir el ruido se utilizan resistores diagonales con valores entre 560 ohms y 4 k ohms. Estos resistores se conectan, uno entre la línea B+ y una fuente de +5V y otro entre A- y tierra (**figura 2.24**). No se debe utilizar voltajes superiores ya que cualquier voltaje a +12V podría provocar fallas en el sistema. Desafortunadamente en algunos casos estos resistores diagonales pueden incrementar ruido al sistema al mejorar su trayectoria desde tierra. Por esta razón es recomendable utilizar resistores diagonales solo si el fabricante lo indica [19] pp. 59

Los resistores que se utilizan para evitar voltajes de modo común tienen valores entre 100k y 200k ohms. Estos resistores van conectados entre cada línea y tierra. El voltaje que debe caer en cada resistor no debe superar el valor de 7 voltios medidos desde cada línea a tierra. En la **figura 2.24** se puede ver cada uno de los resistores mencionados. Como el caso de los resistores para reducir el ruido, las resistencias para voltajes de modo común pueden mejorar la trayectoria del ruido desde tierra, por ende, es recomendable utilizarlas bajo recomendación del fabricante [19] pp. 59-60

Para distancias cortas menores a 100 metros y para tasas bajas de transmisión menores a 9600 baudios, no es necesario ningún aditamento a las líneas de transmisión. Por otro lado si la distancia es considerable se suele utilizar un resistor en cada fin de línea como medida para reducir las reflexiones. Los resistores van conectados al final del bus entre las líneas. Ningún resistor va conectado a la conexión en paralelo de ningún dispositivo,

solo a los extremos del bus. Como en los casos anteriores es recomendable utilizar las resistencias de fin de línea bajo recomendación del fabricante [37].

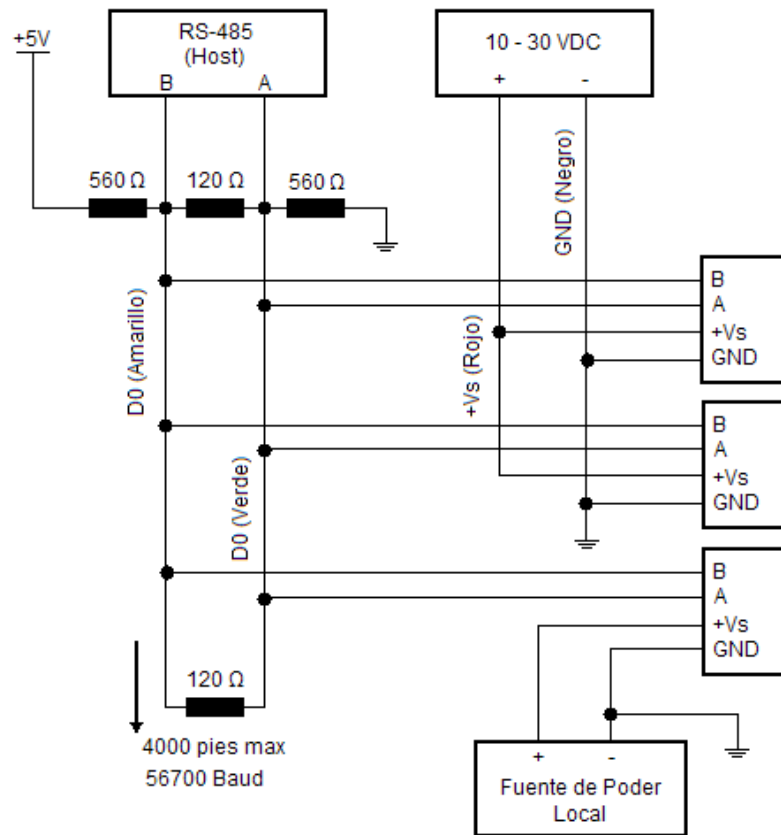


Figura 2.24 Posible instalación de resistores para minimizar el ruido (Steve Mackay,56)

## Capítulo 3

### **Comunicación: módulo central – módulos de estaciones y paradas**

En el presente capítulo se detalla la manera como se implementó la comunicación entre el módulo de comunicación central y los módulos de las estaciones y paradas; además se presenta detenidamente cada uno de ellos, su construcción y detalles de funcionamiento.

Debido a que la distancia entre el módulo de comunicación central y el circuito de ensamblaje puede llegar a ser considerable, se pensó en utilizar un estándar de comunicación industrial de capa física que permitiera extender cable por varios cientos de metros sin perder confiabilidad en la transmisión de datos con una razonable tasa de bits.

El estándar de comunicación industrial EIA-485 además de permitir una longitud de transmisión máxima aproximada de 1200 metros, longitud más que suficiente para cumplir con las necesidades del proyecto, permite enviar y recibir datos con un nivel de ruido mínimo debido al acoplamiento magnético en el alambre par trenzado y su balance diferencial de líneas . Fue así que se decidió utilizar este estándar para comunicar el módulo central del sistema con los módulos de paradas y estaciones.

Se utilizó cable UTP categoría 5e para interconectar las paradas y estaciones con el módulo central.

Para la comunicación módulo central - paradas y estaciones, primeramente se intentó transferir datos serialmente bajo el estándar EIA-485 directamente entre el microcontrolador principal del módulo de comunicación central y los microcontroladores principales de cada módulo de parada y estación utilizando transceivers para EIA-485 a cada extremo, sin embargo el tiempo de espera por parte de los microcontroladores para entablar la comunicación demoraba el resto de procesos que debían realizarse, procesos como encender relés o leds que necesitan un tiempo de respuesta rápido.

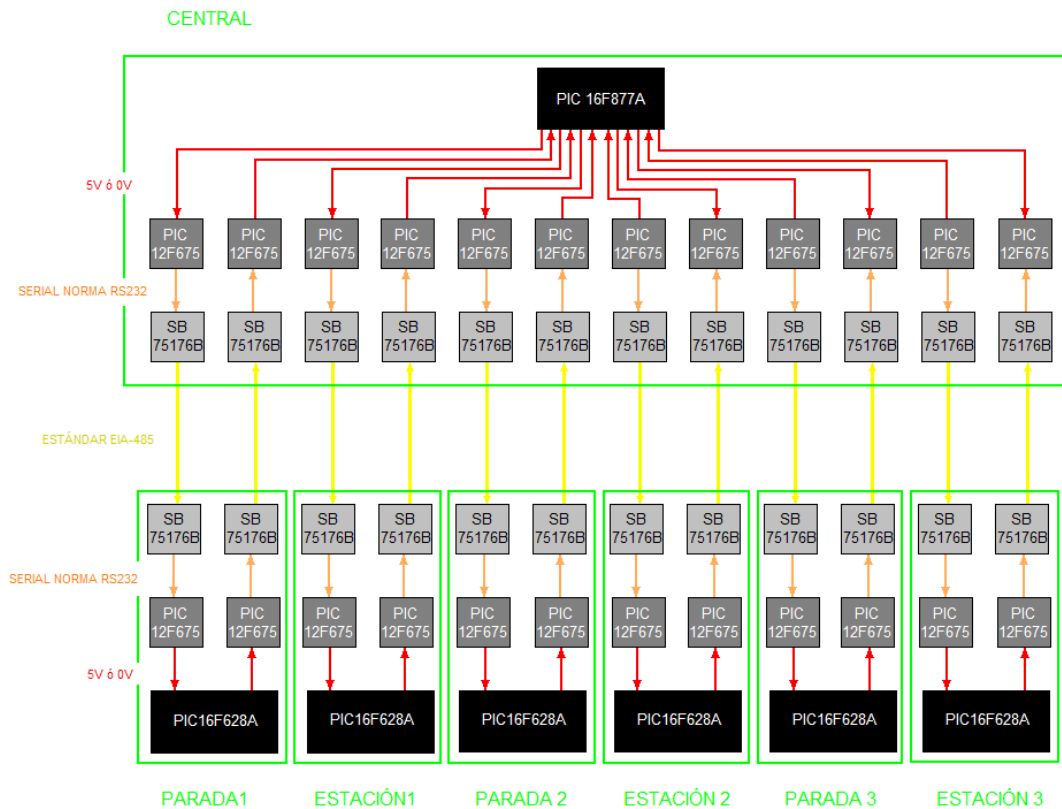
También se pretendió realizar una comunicación bidireccional EIA-485 a través de un solo par de conductores. Bajo programación se logró intercambiar la dirección de transmisión de datos automáticamente, sin embargo el resultado no cumplía con las expectativas de confiabilidad ni de velocidad de transmisión.

Finalmente se optó por realizar una comunicación punto a punto entre la central y cada una de las paradas y estaciones. Se decidió colocar pequeños microcontroladores a cada extremo para que se encarguen específicamente de enviar y recibir los datos. Es así que, en el módulo central un PIC16F877A como microcontrolador principal, está interconectado con 12 PIC's12F675 que son los encargados de entablar la comunicación, 6 transmiten la información y 6 la reciben.

Se utilizó el PIC16F877A como microcontrolador principal por sus cinco puertos disponibles para conectar una gran cantidad de entradas y salidas. El PIC12F675 fue usado por su bajo precio y pequeño tamaño. **Las hojas de**

datos de cada uno de ellos se encuentran disponibles en la sección anexos.

Seis PIC's 12F675 se encargan de enviar los datos. Cada uno recibe información del PIC central y la envía serialmente a un transceiver (dispositivo transmisor-receptor) SN75176B que modula la información para transmitirla consecutivamente a través de un par trenzado bajo el estándar EIA-485 hacia otro transceiver que decodificará la señal en la estación o parada respectiva. Para recibir la información se realiza el proceso inverso, seis transceiver decodifican las señales provenientes de las paradas y estaciones y las envían a los respectivos PIC's 12F675 para que luego éstos transmitan la información hacia el PIC central. En la **figura 3.1** se puede apreciar la arquitectura empleada para conectar el módulo central con los módulos de paradas y estaciones.



**Figura 3.1** Arquitectura utilizada para comunicar el módulo de comunicación central con los módulos de paradas y estaciones

Los pequeños micros, envían y reciben datos seriales en un formato estándar verdadero asincrónico usando 8 bits de datos, sin paridad y 1 stop bit a una velocidad de de 9600 bits/seg.

Como se verá más adelante en los programas, se diseño un pequeño protocolo de comunicación para procesar la información transmitida y recibida por los PIC12F675.

Básicamente para cada transmisión se envía un carácter ASCII con el formato especificado anteriormente. Si el carácter recibido es una "A" se considera una transmisión exitosa y se procede a realizar una secuencia específica de programa, si el carácter es una "E" se considera una transmisión exitosa y se procede con una secuencia diferente. Si el carácter recibido no es ni "A" ni "E" se toma como una comunicación fallida y se vuelve a solicitar un nuevo carácter.

### **3.1 El módulo de comunicación central, descripción física y funcionamiento**

El módulo central es el encargado de administrar toda la información del sistema. Recibe y envía datos desde y hacia las estaciones y paradas.

La central posee dos receptores de radiofrecuencia que se encargan de recibir las señales emitidas por dos transmisores colocados en los módulos de los cargobuses. Estos receptores envían la información al microcontrolador principal que procesa la información para presentarla en el computador personal y en una pantalla LCD.

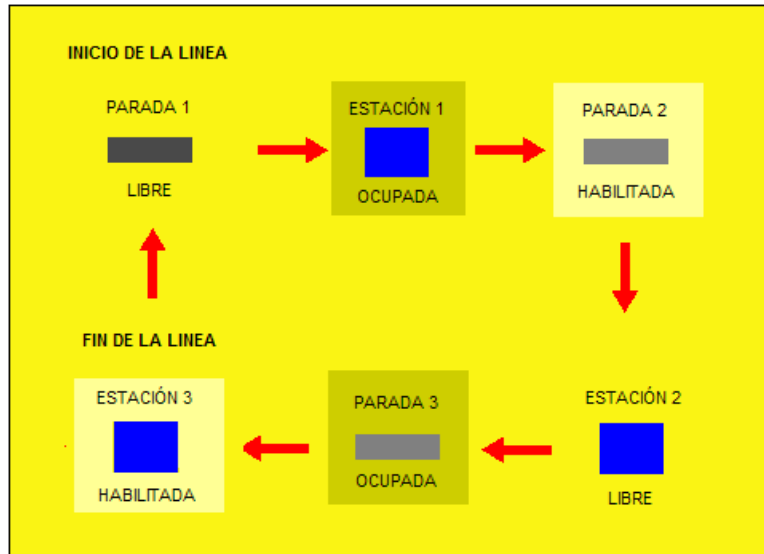
La información que recibe el módulo central, proveniente de la conexión alámbrica con los módulos de paradas y estaciones, se relaciona netamente con la disponibilidad que las paradas y estaciones tienen para recibir o no un cargobus. Es decir, la central se mantiene informada de que paradas o estaciones están ocupadas con cargobuses y que paradas o estaciones están disponibles. Con toda ésta información la central tiene la factibilidad de controlar el movimiento de cada cargobus.

La información que transmite la central hacia los módulos de paradas y estaciones está relacionada con el permiso de arranque que necesitan los cargobuses para partir hacia la siguiente parada o estación. En otras palabras, la central transmite indirectamente la autorización de partida desde las paradas y estaciones hacia los cargobuses para que puedan arrancar y continuar con el ciclo de ensamblaje.

Ahora, la lógica que utiliza la central para controlar el desplazamiento de las grúas suspendidas a través de la línea de ensamblaje es la siguiente:

En primer lugar el módulo central recibe la información de estado de cada parada y estación del circuito, es decir si están o no ocupadas con un cargobus, luego la procesa para poder decidir que paradas y estaciones recibirán la autorización de movimiento de cargobuses. Por ejemplo, si la parada 3 y la estación 1 están ocupadas, entonces la estación 2 y la parada 1 no reciben ninguna habilitación por parte de la central para que cualquier cargobus que estuviese detenido allí parta. Sin embargo si un cargobus estuviese estacionado en la parada 2 recibiría el permiso para continuar su camino hasta la estación 2 que si está desocupada. Para tener más claro la lógica descrita en el ejemplo, refiérase a la **figura 3.2**.

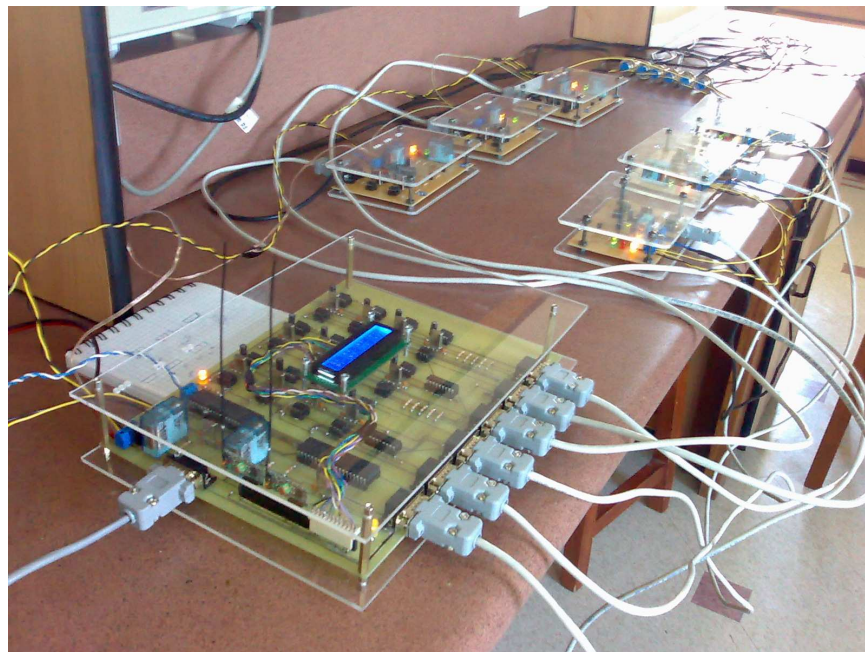




**Figura 3.2** Lógica utilizada por el módulo central para controlar el movimiento de los cargobuses.

A más de encargarse del control del sistema, el módulo central tiene la tarea de transferir la información del circuito hacia la computadora portátil para su monitoreo.

En la **figura 3.3** se puede observar en primer plano el módulo central conjuntamente con el resto de módulos implicados en la comunicación central – estaciones y paradas.



**Figura 3.3** Sistema de comunicación con el módulo central en primer plano.

### 3.1.1 Descripción física del módulo central

Cabe destacar que la tarjeta electrónica del módulo central, así como todas las demás tarjetas del resto de módulos del proyecto, fueron diseñadas, fabricadas y probadas en su totalidad por el autor de esta tesis.

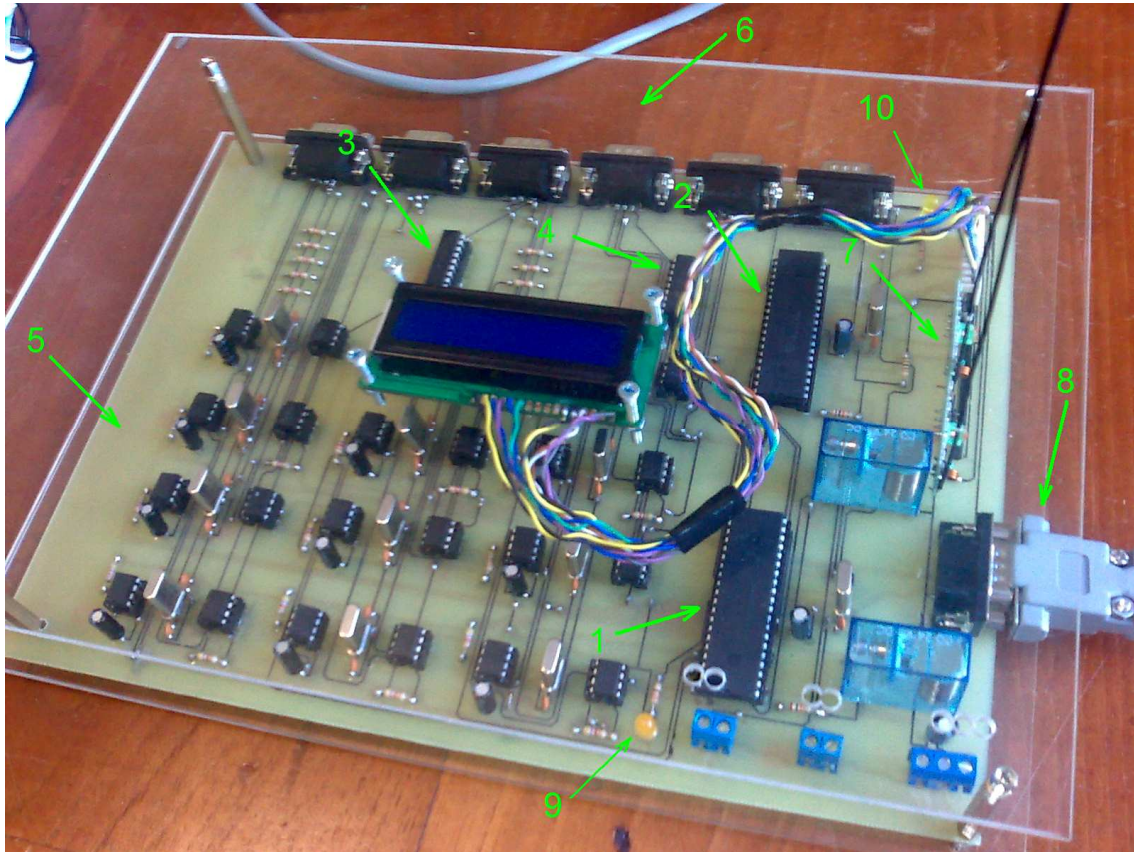
Las dimensiones del módulo central son: 21.4cm X 30.1cm X 5.2cm. La placa electrónica está ensamblada en una lámina de fibra de vidrio de doble lado y está protegida por dos planchas de acrílico transparente colocadas una a cada lado. El acrílico además de proteger la placa, le da un aspecto profesional al trabajo.

El módulo posee una bornera de tres entradas para conectar la alimentación de voltaje, 24VDC, 5VDC y tierra; tiene además dos borneras de dos entradas para conectar el selector de arranque (manual o automático) y el pulsador de arranque manual.

Posee seis conectores DB9 machos para comunicarse con las estaciones y paradas. Se utilizaron conectores DB9 por su bajo precio y por su facilidad de adquisición en el mercado. Además de los seis puertos, el módulo central tiene un séptimo puerto para conectar serialmente la computadora que monitorea y permite arrancar todo el sistema.

La central incluye además un módulo LCD donde se puede visualizar el estado de cada parada o estación y la velocidad que tiene cada cargobus.

El módulo posee dos pequeños alambres sólidos que reemplazan dos antenas tipo látigo de  $\frac{1}{4}$  de onda. Utilizar las antenas en la maqueta significaba una inversión no necesaria ya que éstas se usan para transmisiones superiores a 140 metros. La **figura 3.4** permite ver la descripción física descrita.



**Figura 3.4 Módulo de comunicación central.** 1. Microcontrolador principal PIC17F877A. 2. Microcontrolador PIC17F877A para control de pantalla LCD. 3 Y 4. Decodificadores de radiofrecuencia HT12D. 5. Microcontroladores PIC12F675 y transmisores-receptores SN75176B. 6. Puertos seriales para conexión de módulos de paradas y estaciones. 7. Receptores de RF CZS-3. 8. Puerto serial para comunicación con la PC. 9. Indicador de arranque del sistema. 10. Indicador de encendido del módulo de comunicación central.

### 3.1.2 Detalles de construcción y funcionamiento del módulo central

Antes de comenzar a construir el módulo se realizó un diseño previo para asegurar su correcto funcionamiento. La **figura 3.5** presenta el esquema eléctrico del módulo de comunicación central.

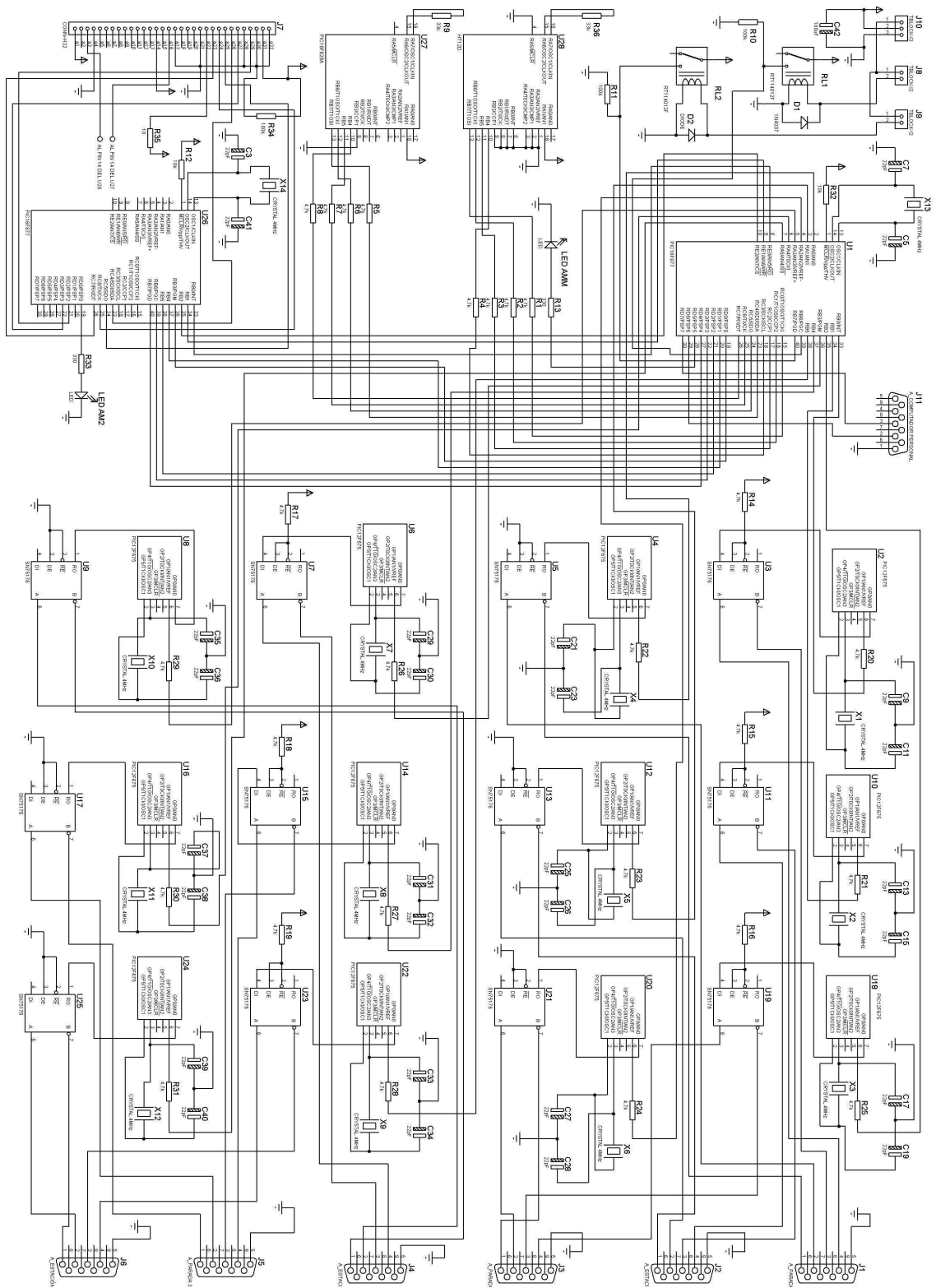


Figura 3.5 Diagrama eléctrico del módulo de comunicación central

El módulo está controlado básicamente por un microcontrolador principal, PIC16F877A, que se encarga de recibir la información de todo el sistema a través de las conexiones serial y RF establecidas.

El microcontrolador se encarga además de distribuir ordenes y datos hacia las estaciones, paradas y dispositivos de monitoreo usando el enlace de comunicación serial.

Como está mencionado anteriormente, doce microcontroladores PIC12F675 conectados al microcontrolador principal tienen la labor de recibir y transmitir datos hacia las estaciones y paradas. Estos microcontroladores están a su vez conectados con doce transmisores-receptores SN75176B para estándar EIA-485, que modulan los datos para poder transmitirlos bajo ese estándar.

La conexión entre el microcontrolador principal y los doce microcontroladores se realiza de pin a pin con una resistencia intermedia para limitar la corriente. No existe ningún tipo de protocolo en esta etapa de comunicación, únicamente se trata de niveles lógicos altos o bajos que son interpretados dependiendo del programa interno de cada microcontrolador.

Un PIC16F877A, diferente al microcontrolador principal, es el encargado de presentar en la pantalla LCD la velocidad de los cargobuses y el estado de las paradas y estaciones. Con el objetivo de utilizar una menor cantidad de pines el PIC central se comunica serialmente con éste microcontrolador.

Los programas escritos para cada microcontrolador se presentan en la sección anexos.

El módulo central tiene además dos decodificadores de radiofrecuencia HT12D que están conectados entre el microcontrolador principal y dos receptores de RF CZS-3.

El HT12D es un decodificador con aplicación a sistemas de control remoto. Recibe direcciones e información desde un codificador de su misma familia (en éste caso el HT12E) utilizando RF o IR como medios de transmisión.

Para recibir la transmisión, el decodificador en primer lugar compara continuamente tres veces la información serial entrante, con la dirección local, si no hay error o códigos que no coinciden, la información se decodifica y se transfiere a los pines de salida.

El HT12D decodifica palabras de 12 bits, compuestas por una dirección de 8 bits y una sección de datos de 4 bits. Es decir se pueden comandar hasta 256 dispositivos diferentes [28].

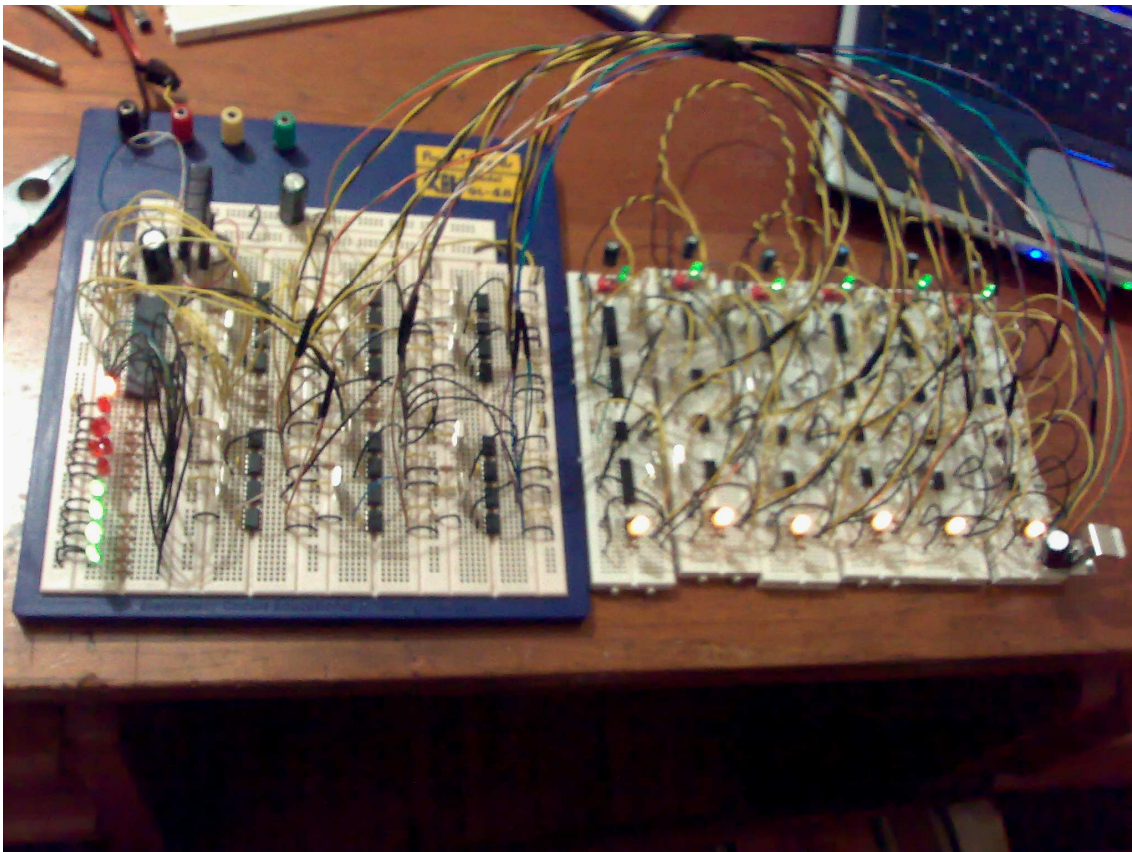
Los pines de salida de los dos HT12D están conectados directamente al microcontrolador principal utilizando resistencias de 4.7K para reducir la corriente de entrada.

Los receptores de RF se encargan de recibir la información proveniente de los dos módulos de comunicación de cargobuses. En el capítulo 4 se presenta un resumen de sus características.

En la **figura 3.4** se puede apreciar los elementos mencionados ensamblados en la tarjeta.

Para constatar que las características eléctricas de corriente y voltaje eran las correctas, anterior a la fabricación de la tarjeta electrónica se armó todo el módulo central en un protoboard. En definitiva todos los módulos que

componen el sistema de comunicación fueron armados en protoboards antes de construir sus respectivas tarjetas. En la **figura 3.6** se puede visualizar el modelo básico del módulo central armado en protoboard conjuntamente con los módulos de paradas y estaciones.



**Figura 3.6** Modelo básico del sistema previamente armado en protoboard

### **3.2 Módulos de comunicación de estaciones y paradas, descripción física y funcionamiento.**

Como se vio en la introducción, las estaciones y paradas de una línea de ensamble son los puntos específicos del circuito donde los cargobuses se pueden detener.

Los módulos de comunicación de las paradas y estaciones tienen la finalidad de informar a la central la posición de cada cargobus en la línea, es decir si un cargobus se encuentra o no estacionado en una determinada parada o estación, con esta información la central puede administrar el desplazamiento de los cargobuses.

Si un cargobus llega a una parada o estación es detectado inmediatamente por un sensor inductivo, esta información es enviada al microcontrolador principal del módulo de la estación o parada y transmitida luego a la central de control.

Para que un cargobus estacionado pueda partir, en primer lugar necesita ser habilitado por el módulo central. El módulo central primeramente envía el permiso de partida a las estaciones y paradas para que luego ellas transmitan de manera infrarroja el dato de habilitación a los cargobuses.

A diferencia de una parada, una estación además de transmitir el permiso de partida de manera infrarroja envía también el mando del operario por otro puerto infrarrojo. El mando del operario no es más que la orden de bajar y subir la carga.

### **3.2.1 Descripción física de los módulos**

Tanto los módulos de paradas como de estaciones tienen las mismas dimensiones: 12cm X 16cm X 3.7cm

Las placas electrónicas están fabricadas en baquelita de un solo lado y protegidas en ambos lados por dos planchas de acrílico transparente.

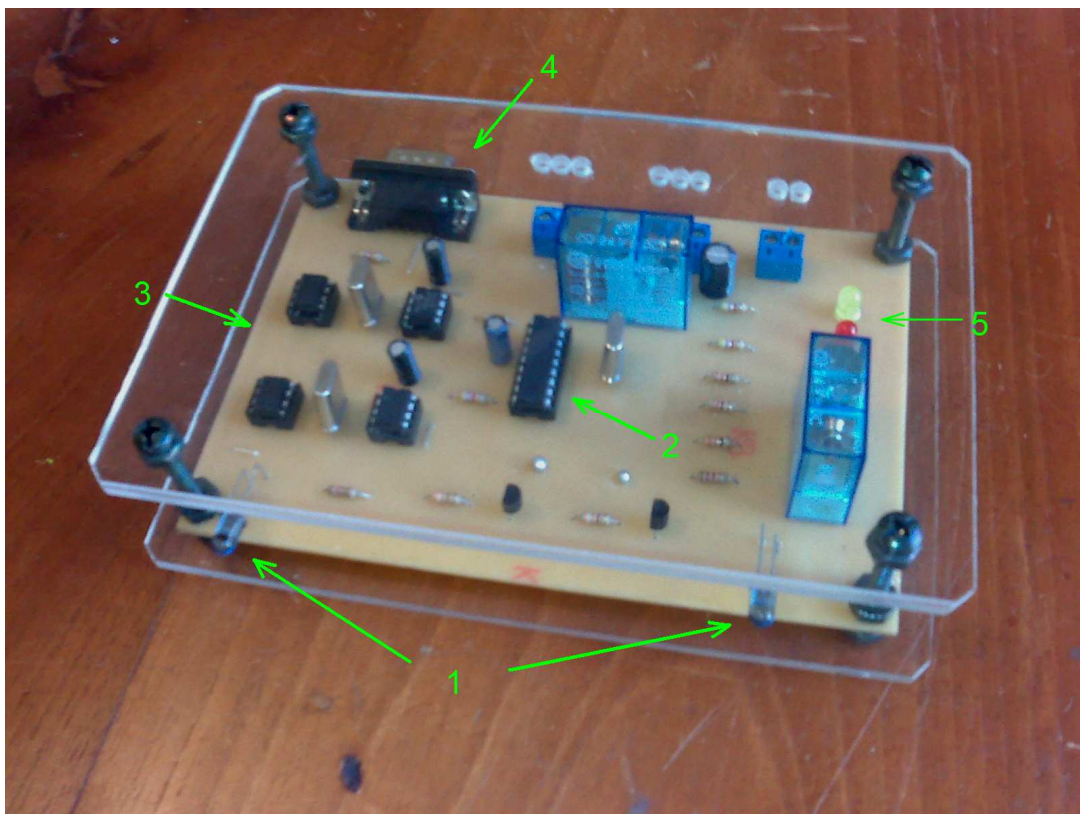
Cada módulo de parada y estación posee un puerto serial DB9 macho para conectarse con la central de control. Además tienen dos pequeñas



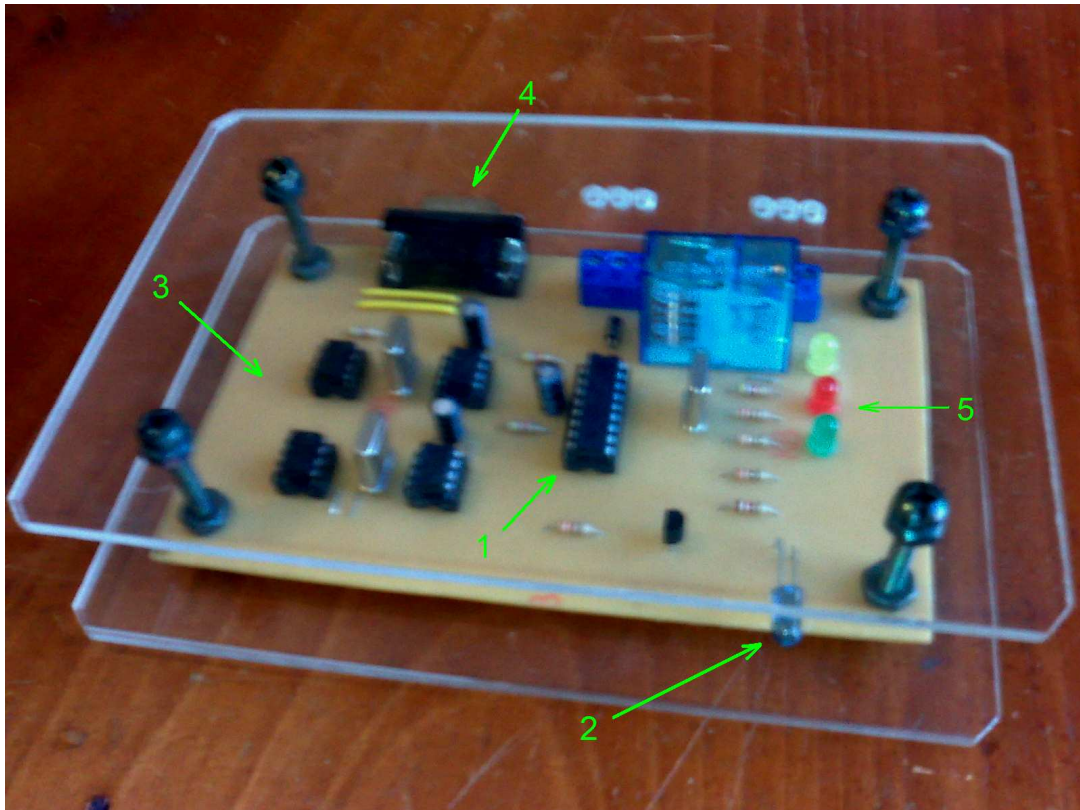
borneras, una para conectar la alimentación de voltaje y otra para conectar el sensor inductivo.

Para transmitir la información hacia el cargobus se instaló un diodo transmisor infrarrojo en cada módulo.

En el caso de los módulos de estación fue necesario colocar una bornera más para conectar el pulsador que enviará el mando de operario al microcontrolador principal. Además se incorporó otro diodo transmisor de infrarrojo para enviar la orden de operario desde la estación al cargobus. Las **figuras 3.7 y 3.8** permiten ver el aspecto físico de los módulos de comunicación de estación y parada respectivamente.



**Figura 3.7 Módulo de comunicación de estaciones.** 1. Diodos transmisores infrarrojos. 2. Microcontrolador principal PIC16F628A. 3. Microcontroladores PIC12F675 y transmisores-receptores SN75176B. 4. Puerto serial 5. Indicadores de estado de la estación,



**Figura 3.8 Módulo de comunicación de paradas.** 1. Microcontrolador principal PIC16F628A. 2. Diodo transmisor infrarrojo. 3. Microcontroladores PIC12F675 y transmisores-receptores SN75176B. 4. Puerto serial. 5. Indicadores de estado de la parada.

### 3.2.2 Detalles de construcción y funcionamiento

Al igual que con el módulo central, se realizaron diseños previos antes de armar la tarjetas electrónicas. Además se hicieron pruebas en protoboard para evitar posibles errores de diseño.

Las **figuras 3.9 y 3.10** presentan los esquemas eléctricos de los módulos de comunicación de estaciones y paradas respectivamente.

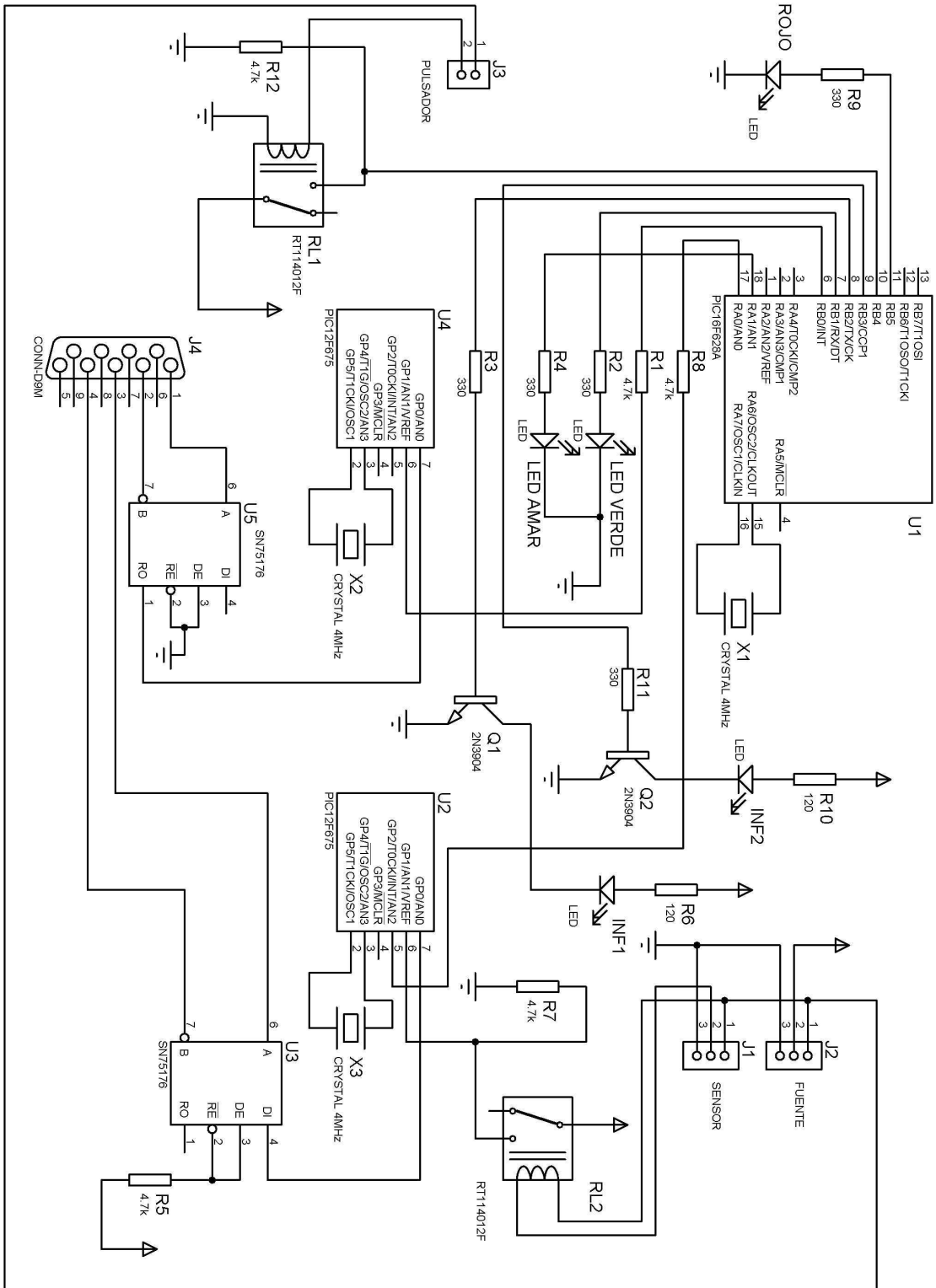
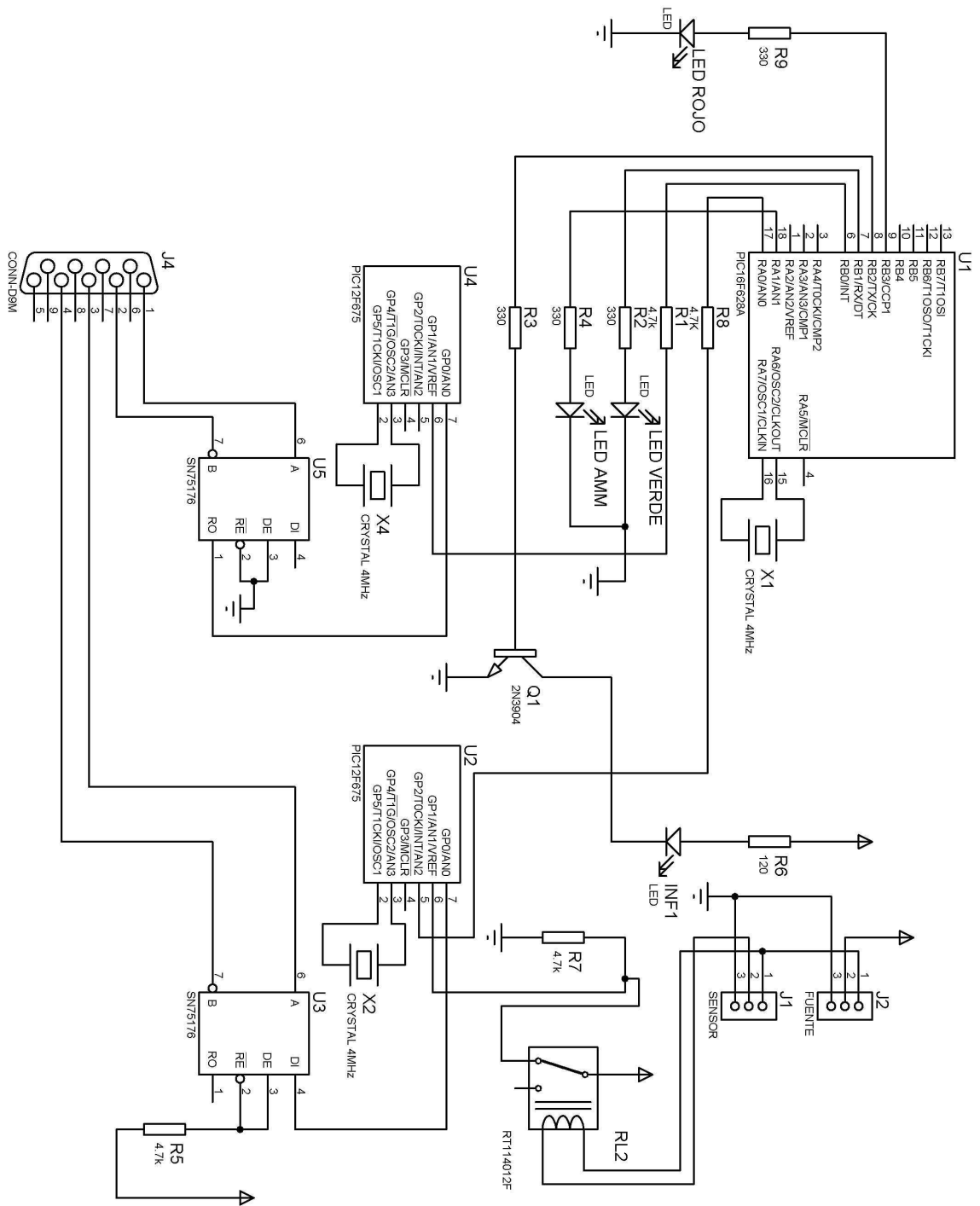


Figura 3.9 Diagrama eléctrico del módulo de comunicación de estaciones



Con relación a la comunicación EIA-485 con la central, un módulo de estación funciona de la misma forma que un módulo de parada. Reciben la habilitación de avance de cargobus desde el módulo central y envían su estado de uso; en otras palabras, reciben el permiso de disponibilidad para que un posible cargobus estacionado pueda partir, e informan si están o no ocupados por un cargobus.

Tanto en los módulos de las paradas como en los módulos de las estaciones, la señal que llega por el puerto DB9 desde el módulo de comunicación central es recibida en primer lugar por un transceiver SN75176B. Este elemento cambia la señal recibida con estándar EIA-485 de las dos líneas diferenciales balanceadas a una señal serial con referencia a tierra.

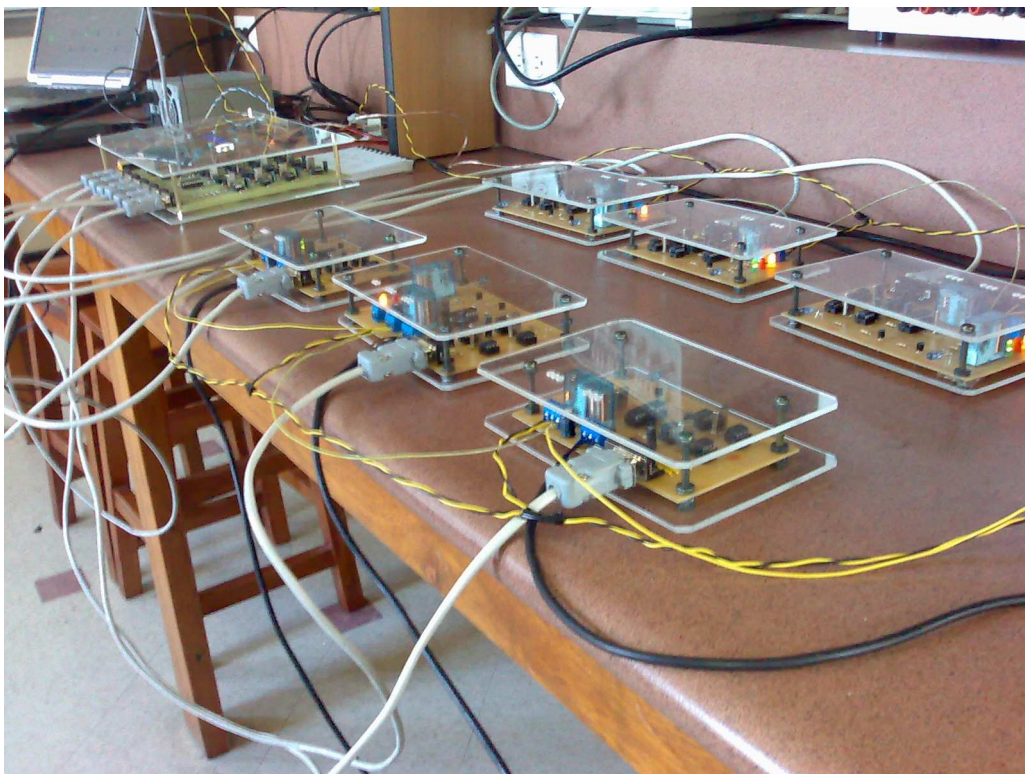
Bajo el nuevo estándar de comunicación, la información se transmite a un PIC18F675 que está continuamente esperando por datos. Cada PIC18F675 está conectado con el PIC central de cada módulo, un PIC16F628A. Es así como la información recibida desde la central llega al microcontrolador principal de cada módulo de paradas y estaciones.

Con la información de la central de control recibida y procesada, el microcontrolador principal de cada parada y estación ya puede enviar los datos hacia el cargobus vía infrarrojo, el PIC central utiliza el diodo emisor de infrarrojo y transmite serialmente los datos.

Para transmitir la información desde los módulos de paradas y estaciones hacia el módulo central de control, se utiliza un proceso similar al utilizado para recibir los datos. Dentro de cada módulo, el microcontrolador principal a través de un pin de salida transmite la información a un PIC18F675, éste PIC, dependiendo del nivel de voltaje recibido, envía serialmente la

información a un transceiver SN75176B que modifica el estándar de transmisión a EIA-485 para poder enviar la información hacia el módulo central a través de un puerto DB9 macho.

Para enviar la habilitación de avance desde las estaciones y paradas hacia el cargobus, ambos módulos utilizan transmisión vía infrarrojo. El dato recibido desde el módulo central por el puerto serial llega al microcontrolador principal de cada módulo, allí la información es procesada y enviada por infrarrojo al cargobus. En la **figura 3.11** se puede ver en primer plano los módulos de estaciones y paradas conectados al sistema.



**Figura 3.11 Módulos de comunicación de estaciones y paradas conectadas al sistema.**

En las estaciones de ensamblaje, además de transmitirse el permiso de avance por infrarrojo, también se transmite el mando del operario. Cuando un operario de una estación desea bajar la carga, aprieta un pulsador que se encuentra instalado muy cerca de él y la carga desciende. Si desea elevar la carga aprieta nuevamente el pulsador y la carga se eleva. Como el cargobus se encuentra separado de las estaciones y paradas, los datos se envían serialmente utilizando luz infrarroja.

El pulsador activa o no un determinado pin del microcontrolador principal a través de un relé. Se utilizó el relé con la finalidad de evitar cambios de estado indeseados en la entrada del microcontrolador producto de variaciones en la corriente al cerrarse el pulsador. Dependiendo del estado de la entrada, el PIC central habilita o no la transmisión serial hacia el cargobus a través del diodo emisor infrarrojo.

## Capítulo 4

### Comunicación: módulo central – módulo de cargobuses

En el presente capítulo se especifica la manera como se implementó la comunicación entre el módulo central de control y los módulos de cargobuses; conjuntamente se detalla las características del módulo de cargobus, su construcción y pormenores de su funcionamiento.

Para poder monitorear el estado de cada cargobus desde el módulo central es necesario que el módulo de comunicación de la grúa suspendida transmita continuamente datos a la central. Debido a que el cargobus pierde total contacto con el sistema de comunicación al momento en que parte de una parada o estación, es preciso transmitir la información de manera inalámbrica.

El ruido eléctrico que se produce en un ambiente industrial afecta notablemente la efectividad de una comunicación inalámbrica.

Los equipos eléctricos que producen cambios rápidos o picos en el voltaje son fuentes típicas de ruido. A continuación se presenta una lista de las principales causas de ruido eléctrico en un ambiente industrial:

- Grandes motores eléctricos encendiéndose
- Tubos de luz fluorescente
- Máquinas soldadoras
- Altos voltajes producto de fallas eléctricas [19] pp.20



Existen dispositivos de transmisión inalámbrica específicos para este tipo de ambiente, sin embargo sus precios son relativamente altos como para ser utilizados en este proyecto.

En su lugar, para implementar la comunicación, se utilizaron módulos de transmisión inalámbrica con aplicación a la robótica. Estos módulos están disponibles en el mercado nacional lo que facilita su adquisición y, además permiten una distancia de transmisión suficiente como para comprobar que el sistema funciona correctamente, aproximadamente 1000 metros.

La transmisión de datos inalámbrica es unidireccional, los datos fluyen desde el cargobus hacia la central de control. Es posible sin duda transmitir información desde la central al cargobus, es más, se puede realizar todo el control de movimiento del cargobus de manera inalámbrica, sin embargo esta opción no es muy recomendada ya que por un lado la información transmitida es realmente crítica, información relacionada a la habilitación de arranque de motores de elevación y traslación o activación de frenos; y por otro lado se necesitaría de dispositivos de transmisión dedicados a un ambiente industrial con la finalidad de asegurar una transmisión de datos cien por ciento segura.

La central de control debe recibir la información proveniente de dos cargobuses, para ello, dos receptores de RF instalados en la tarjeta electrónica se mantienen escuchando desde que el sistema arranca.

La información que transmite el cargobus, está relacionada con su velocidad de avance en la línea. Como se verá más adelante, un cargobus puede tomar dos velocidades a través del circuito dependiendo del tramo. Cada cargobus tiene instalado un transmisor de RF que envía continuamente información de su velocidad de avance.

Como ya sabemos el cargobus recibe a través de dos receptores de IR dos señales diferentes de las estaciones y paradas, más adelante se presentan sus detalles técnicos.

#### **4.1 Módulo de comunicación de cargobuses, descripción física y funcionamiento**

Como ya se sabe los cargobuses son grúas suspendidas que se trasladan a través de una línea de producción transportando productos en construcción. Estos productos son generalmente de gran tamaño y peso.

El cargobus puede tomar distintas velocidades dependiendo del tramo del circuito. Con el objetivo de ubicarse correctamente, a medida que se acerca a una parada o estación el cargobus disminuye su velocidad hasta que se detiene totalmente y por el otro lado a medida que se aleja aumenta su velocidad para acelerar el proceso.

Para tener un control efectivo de un cargobus es necesario en primer lugar mantener una comunicación segura con la central. Con este objetivo se diseñó un módulo individual de comunicación que recibe información desde las paradas y estaciones vía comunicación serial infrarroja y a la vez transmite datos a la central por radiofrecuencia.

El módulo de comunicación de cargobuses además de recibir y transmitir información también controla el descenso y ascenso de la carga. Cuando el cargobus se encuentra estacionado en una parada o estación, el módulo de comunicación tiene la facilidad de controlar los motores de elevación de un cargobus y así controlar el descenso o ascenso del producto.

Para diseñar el módulo, primeramente se analizó el funcionamiento de un cargobus real montado en una planta ensambladora local de vehículos. En base a la información recopilada, se decidió incorporar a la placa electrónica funciones para el control de los motores del cargobus.

El cargobus analizado posee un solo motor de traslación y dos motores de elevación. Para detectar que la carga ha subido, el cargobus incorpora dos encoders, uno en cada motor de elevación que dependiendo del número de vueltas que dan los motores determinan si la carga se encuentra arriba o abajo.

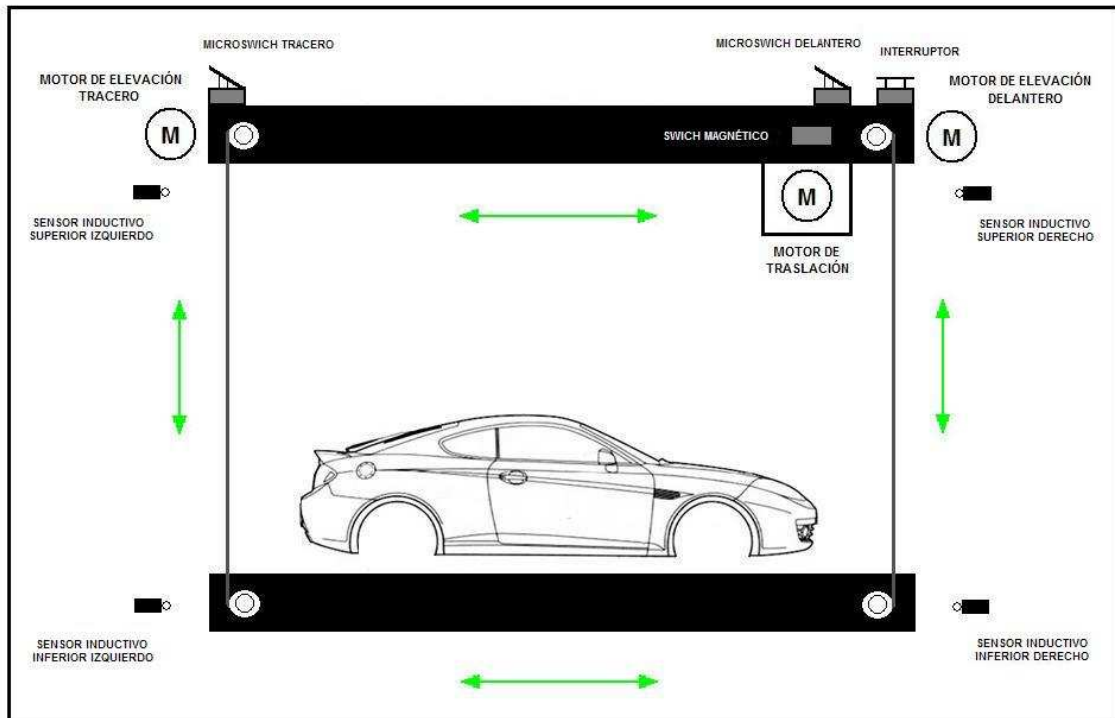
Para que el motor de traslación arranque conjuntamente con el cargobus, es necesario el permiso de la central de control, permiso que llega a través de las paradas o estaciones vía infrarrojo.

Cuando un cargobus está detenido específicamente en una estación, no puede avanzar hasta que la carga baje y suba. Incluso si la estación se encuentre transmitiendo el permiso de partida continuamente a través de los infrarrojos, el cargobus no parte hasta que el ciclo de descenso/ascenso de la carga termine.

Para aumentar la velocidad una vez que ha partido de una parada o estación, o disminuirla al momento de acercarse a una de ellas, un microswitch colocado estratégicamente en el cargobus, envía una señal al PIC central del módulo de cargobus para aumentar su velocidad.

En base a las características del cargobus descrito anteriormente se ideó el siguiente sistema para el cual se diseñó el módulo de comunicación.

El cargobus tiene cuatro sensores de presencia, que pueden ser reemplazados por microswitches, para detectar la posición de la carga. Dos sensores están colocados en la parte superior y dos en la parte inferior. En la **figura 4.1** se puede identificar los sensores descritos.



**Figura 4.1 Representación del cargobus utilizado como esquema para diseñar el módulo de comunicación.**

Para diferenciar si el cargobus llegó a una parada o una estación existe un switch magnético que se activa o no dependiendo el caso.

Para evitar fallas en la comunicación infrarroja, el cargobus abre el puerto de comunicación únicamente si el cargobus se aproxima a una parada o estación, si no lo hace, el puerto se mantiene cerrado. Un interruptor colocado en la parte superior es el encargado de decidir si el cargobus puede o no abrir el puerto.

Para variar su velocidad, el cargobus tiene dos microswitches, uno colocado en la parte superior delantera y otro en la parte superior trasera de la grúa. Al acercarse el cargobus a una parada o estación, un tope acciona el primer microswitch el cual envía inmediatamente una señal al microcontrolador para que disminuya la velocidad del motor de traslación. Al momento en que el

segundo microswitch se activa con el mismo tope, el cargobus se detiene por completo.

#### **4.1.1 Descripción física del módulo**

Las dimensiones de los módulos de cargobuses son: 20cm X 25cm X 3.7cm. La placa electrónica montada en cada módulo está ensamblada en una lámina de fibra de vidrio de un solo lado y al igual que con los otros módulos del sistema está protegida por dos planchas de acrílico transparente colocadas una a cada lado.

Para alimentar el módulo se instaló una bornera de dos entradas, 5VDC y tierra, que está colocada en la esquina inferior izquierda del módulo.

Para conectar el interruptor que abre el puerto de comunicación y el switch magnético que permite diferenciar el arribo a una estación del arribo a parada, se instalaron dos borneras colocadas a la derecha de la bornera de alimentación.

Dos borneras más permiten conectar los microswitches que varían la velocidad del cargobus.

Para acoplar los cuatro sensores de presencia que determinan la posición de la carga, se instalaron cuatro borneras que están colocadas en línea.

La placa posee siete salidas de relé para conectar el motor de traslación y los motores de elevación del cargobus. Dos para elevar la carga, dos para bajar la carga y tres para cada velocidad del motor de traslación. Las salidas de relé están conectadas a siete borneras distribuidas a lo largo de placa.

Tres leds instalados en la tarjeta de color rojo, amarillo y verde permiten visualizar la velocidad que tiene el cargobus. Cuatro leds más indican el sentido

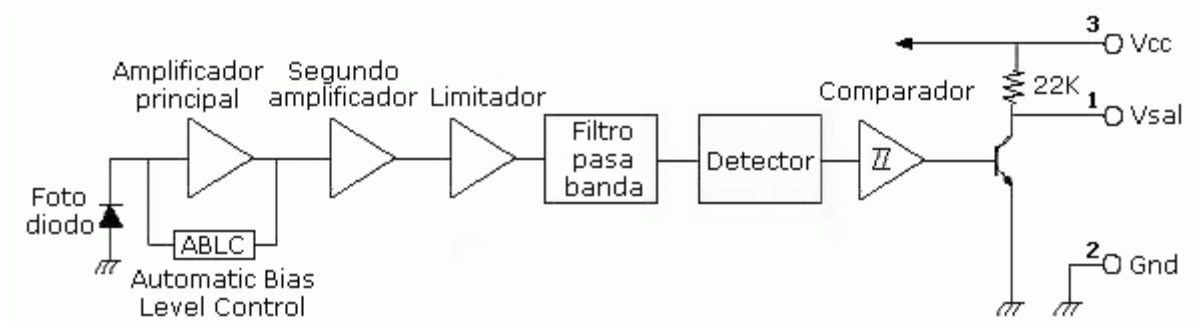
de giro de los motores de elevación, es decir si la carga desciende o se eleva; dos led verdes representan a los dos motores de elevación subiendo la carga y dos leds rojos representan a los dos motores bajando la carga.

El módulo de comunicación de cargobuses tiene instalado un transmisor de radiofrecuencia para emitir su velocidad.

El séptimo led de color rojo, led que se encuentra junto al transmisor, indica el momento durante el cual el módulo se encuentra irradiando su señal.

El último led, al igual que el resto de módulos, representa el arranque del microcontrolador. Este led es de color amarillo.

Para recibir la señal infrarroja proveniente de los dos transmisores instalados en las estaciones y paradas, los módulos de comunicación de cargobuses poseen dos receptores de infrarrojo. Los dispositivos utilizados son del tipo IRM8601S que vienen en un encapsulado similar al de un transistor TIP. En la **figura 4.2** se puede observar un receptor IRM8601S conjuntamente con su diagrama esquemático [35].



**Figura 4.2 Encapsulado y diagrama esquemático del IRM860S**

El receptor infrarrojo IRM8601S tiene las siguientes características:

Inmunidad a interferencias electromagnéticas, compatibilidad con tecnología TTL y CMOS, elevada sensibilidad, alta inmunidad a luz ambiente, bajo consumo de corriente.

En la **figura 4.3** se presenta el módulo de comunicación de cargobuses.



**Figura 4.3 Módulo de comunicación de cargobuses.** 1. Microcontrolador principal PIC16F877A. 2. Receptores infrarrojos IRM860S. 3. Transmisor de radiofrecuencia FST-3. 4. Encoder de RF HT12E. 5. Microcontrolador PIC12F675. 6. Indicador de encendido. 7. Indicadores de velocidad del cargobus. 8. Indicadores de asenso o descenso de carga. 9. Indicador de estado de la transmisión RF.

#### 4.1.2 Detalles de construcción y funcionamiento

Al igual que con el resto de módulos se realizaron diseños previos y pruebas en protoboard antes de armar la tarjetas electrónicas. La **figuras 4.4** presenta el diagrama eléctrico de las tarjetas.

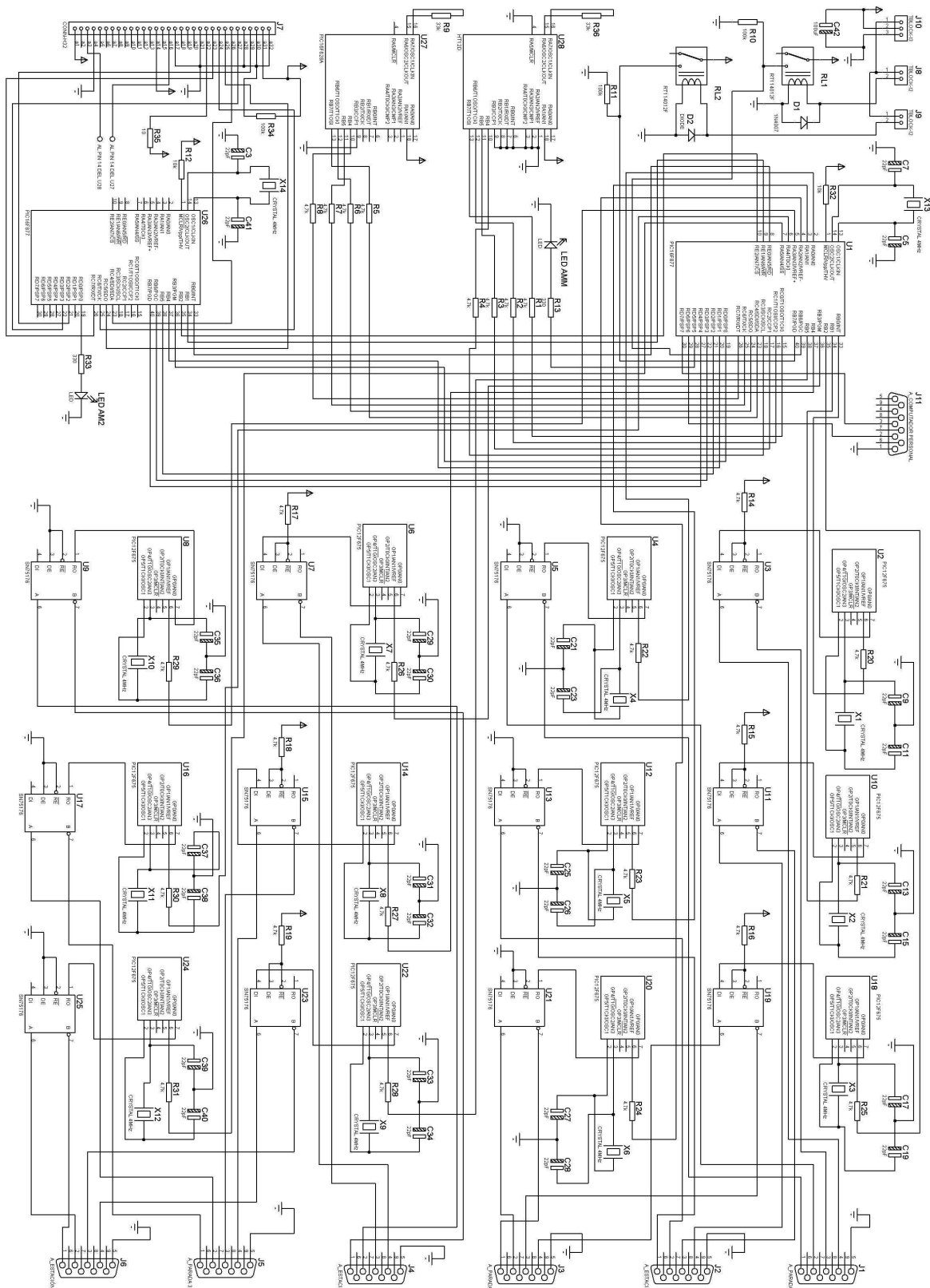


Figura 4.4 Diagrama eléctrico del módulo de comunicación de cargobuses



El módulo de cargobuses incorpora un microcontrolador PIC16F877A para comandar las acciones del cargobus. Este microcontrolador recibe las señales provenientes de los sensores y microswitches, y dependiendo del programa, activa o desactiva las salidas de relé.

Para activar los relés se utilizaron transistores NPN 2N3904 en corte y saturación, los transistores son excitados desde los pines de salida del microcontrolador.

Los dos receptores infrarrojos están conectados directamente al microcontrolador. Como resultado de la transmisión serial de datos, el PIC16F877A recoge de los receptores IR dos cadenas de bits que son leídas y procesadas inmediatamente.

Para transmitir la velocidad del cargobus, el microcontrolador está conectado a un encoder de radiofrecuencia HT12E que codifica la información para posteriormente enviarla al transmisor de radiofrecuencia FST-3.

El encoder HT12E es un codificador RF con aplicación a sistemas de control remoto. Envía direcciones e información hacia un decodificador de su misma familia (en éste caso el HT12D) utilizando RF o IR como medios de transmisión.

Las entradas de dirección y datos pueden ser colocadas en uno de los dos estados lógicos. El HT12E tiene ocho pines para introducir la dirección y cuatro para datos [30].

Los pines de datos están conectados directamente al microcontrolador principal.

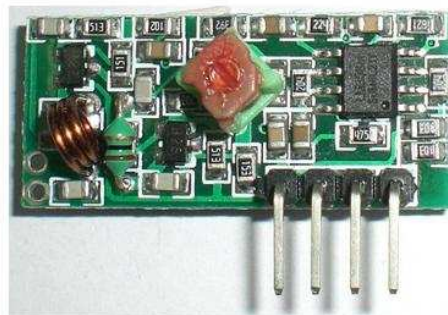
Los módulos de transmisión y recepción de RF utilizados en las tarjetas de comunicación de los cargobuses y de la central de control respectivamente, son de la marca SZSAW Electronic, el modelo del transmisor es el FST-3 y del receptor el CZS-3. La frecuencia en la cual operaran estos módulos es 433.92MHz, el tipo de modulación que utilizan es ASK (Amplitude – shift keying), modulación por desplazamiento de amplitud.

En la modulación ASK la amplitud de una señal portadora análoga representa la información a transmitir; la frecuencia y la fase de la señal portadora se mantienen constantes. En la señal modulada se puede representar por ejemplo el valor lógico 0 con la ausencia de una portadora, y el valor lógico 1 con una portadora de amplitud definida.

En las **figuras 4.5 y 4.6** se pueden apreciar los dos módulos



**Figura 4.5** Módulo transmisor de RF, FST-3

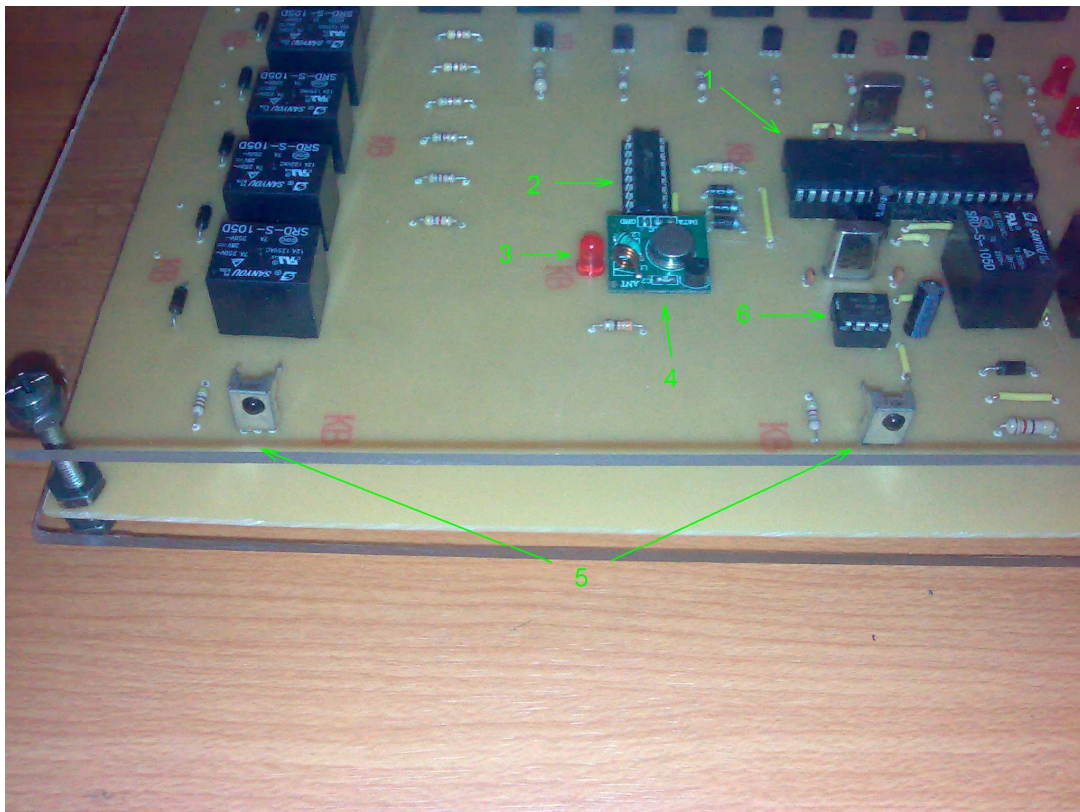


**Figura 4.6** Módulo receptor de RF, CZS-3

El módulo transmisor FST-3 funciona con un voltaje de alimentación de 3 a 12VDC, según las especificaciones, la distancia máxima de transmisión es de 1000m [31].

El módulo receptor CZS-3 funciona con un voltaje de 5VDC y tiene una sensibilidad de -103dBm [32].

En la **figura 4.7** se puede distinguir los dos receptores infrarrojos IRM8601S, y el trasmisor RF FST-3.



**Figura 4.7** Transmisor de radiofrecuencia FST-3 y receptores de infrarrojo IRM8601S, instalados en la tarjeta. 1. Microcontrolador principal PIC16F877A. 2. Encoder de RF HT12E. 3. Indicador de estado de la transmisión RF 4. Transmisor de radiofrecuencia FST-3. 5. Receptores infrarrojos IRM860S. 6. Microcontrolador PIC12F675.

## Capítulo 5

### Monitoreo del sistema con LabVIEW

En el presente capítulo se detalla como se implementó la comunicación entre el módulo central del proyecto y el computador personal que monitorea el estado del sistema. En definitiva se explica qué información se transmite al computador y como se logran intercomunicar ambas partes.

Se utilizo la herramienta de software orientada a objetos LabVIEW versión 8.2 para programar la interfaz de usuario. La licencia que se usó para correr el software fue la que se encuentra disponible exclusivamente para estudiantes de la Universidad San Francisco de Quito.

#### 5.1 Conexión Serial entre el módulo central y la computadora

Como ya se mencionó, el computador personal se comunica serialmente con el módulo central del sistema, no se utiliza ninguna tarjeta de adquisición de datos para transferir la información. En la **figura 5.1** se puede observar el computador en primer plano conectado al módulo de comunicación central.

El módulo central transmite una cadena de bits que contiene el estado de cada parada y estación conjuntamente con la velocidad de cada cargobus, LabVIEW a través del puerto serial del computador, recibe la cadena de bits y los separa para que dependiendo del programa se presente la información en el panel de control.



**Figura 5.1** Computador personal conectado serialmente al módulo de comunicación central a través de un cable UTP 5e.

Para que la comunicación sea efectiva es necesario que tanto el transmisor, en éste caso el microcontrolador del módulo central, como el receptor, el computador personal, se encuentren configurados con la misma velocidad, el mismo número de bits de datos, la misma cantidad de stop bits y si se incluye o no el bit de paridad.

Por otro lado, para el arranque remoto del sistema, el computador personal tiene la capacidad de transmitir serialmente hacia el microcontrolador principal una cadena de bits similar a la descrita anteriormente. El microcontrolador principal del módulo central de comunicación recibe los datos enviados desde el computador personal y dependiendo de ésta información, procede a arrancar

o no el sistema. Más adelante se especifica que contiene la cadena de bits citada.

La transmisión de datos entre ambos puntos es de tipo serial asincrónica con dato invertido a una velocidad de 9600 bit/seg usando 8 bits de datos, sin paridad y 1 stop bit.

En la sección anexos se puede observar el programa que corresponde al microcontrolador principal, allí se puede distinguir la configuración utilizada al momento de transmitir y recibir los datos.

A continuación se describe el diagrama de bloque programado en LabVIEW para recibir y transmitir la información desde el computador personal.

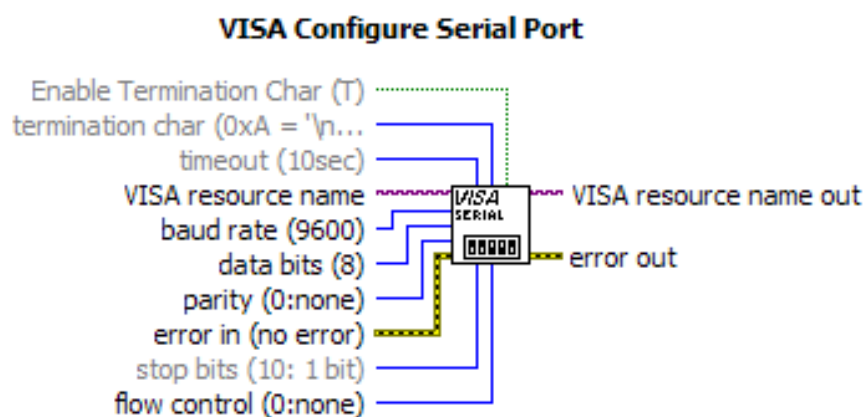
## **5.2 El diagrama de bloque del programa**

Para comenzar a recibir la cadena de bits desde el módulo central es necesario en primer lugar configurar el puerto de comunicación serial. Para configurar el puerto es necesario utilizar el Instrumento Virtual, VISA Configure Serial Port, que se encuentra en la paleta VISA del menú Instrument I/O.

VISA (Virtual Instrument Software Architecture), Arquitectura de Software para Instrumentos Virtuales, es un estándar para configurar, controlar, reparar y programar sistemas de instrumentación que abarcan dispositivos para GPIB, Bus de Interfase de Fines Generales, Comunicación Serial, Ethernet, RS232, interfases para USB, entre otros. VISA proporciona la interfase de programación entre el hardware y diferentes entornos de desarrollo como LabVIEW.

El estándar VISA unifica la industria de instrumentación para crear software que pueda ser interpretado y reusado por más tiempo, sin importar el tipo de operación del instrumento [33].

El VI, VISA Configure Serial Port permite especificar la tasa en baudios, los bits de datos, los bits de parada, los bits de paridad, entre otras características de la comunicación serial. Como parte de la programación, dentro del VISA Configure Serial Port se configura el puerto por el cual se comunica el computador, en este caso COM1 y además el tiempo de espera antes de cortar la comunicación luego de no recibir ninguna transmisión. En la **figura 5.2** se puede ver el VI VISA Configure Serial Port.

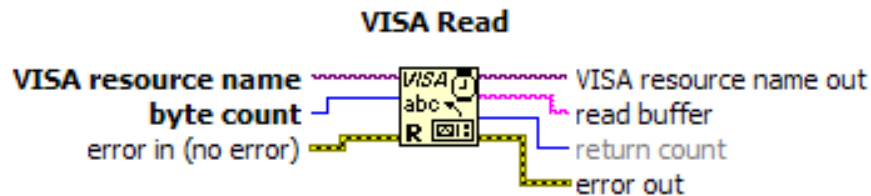


**Figura 5.2 VISA Configure Serial Port (National Instruments LabVIEW)**

Para recibir la cadena de bytes que es transmitida desde el módulo central se utilizó la función VISA Read que lee el número específico de bytes recibidos desde el dispositivo o interfase definida, en este caso el puerto serial COM1.

En la función VISA Read se especifica: el número de bytes a leer, el nombre de la fuente VISA, en este caso COM1, y la entrada de error. Con esta

información se obtiene una cadena de bytes de longitud igual a la especificada y una salida de error. En la **figura 5.3** se puede ver la función VISA Read.



**Figura 5.3 VISA Read (National Instruments LabVIEW)**

Para transmitir la información se utilizó la función VISA Write que escribe los datos del buffer de escritura en el dispositivo o interfase definida, en este caso el puerto serial COM1. Se debe especificar el nombre del receptor VISA en este caso COM1, el buffer con los datos respectivos y la entrada de error. En la **figura 5.4** se puede ver la función VISA Write.



**Figura 5.4 VISA Write (National Instruments LabVIEW)**

Es necesario cerrar cualquier sesión VISA abierta cuando se haya terminado la comunicación. Para finalizar con la sesión se utiliza la función VISA Close que termina cualquier evento especificado con el receptor VISA, es decir cierra la sesión con el dispositivo. En la **figura 5.5** se puede ver la función VISA Close.



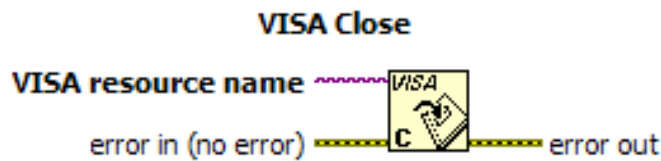


Figura 5.5 VISA Close (National Instruments LabVIEW)

### 5.2.1 Descripción del diagrama de bloque

Como se puede apreciar en la **figura 5.6** se utilizó el VI VISA Configure Serial Port y las funciones VISA Read, VISA Write y VISA Close para entablar la comunicación con el PIC central del módulo central de control del sistema.

Para mantener el programa dentro de un lazo repetitivo se utilizó un While Loop que contiene todas las funciones y VI's utilizadas. Se utilizó como Terminal Condicional el botón STOP del panel de control.

Para comenzar con la adquisición de datos y arrancar el programa se utilizó una estructura Case que, en estado verdadero abre el puerto de comunicación con VISA Configure Serial Port y en estado falso lo cierra con la función Visa Close.

Una vez que el programa arrancó es posible activar el sistema de comunicación de forma remota. La función VISA Write escribe en el puerto una cadena de datos que contiene dos posibles variables, el caracter ASCII A o el caracter ASCII B.

Dentro del programa los caracteres son escogidos a través de un Case Structure que tiene como selector de cambio de estado al pulsador de Arranque de Sistema. Se puede distinguir fácilmente el pulsador en **las figuras**

**5.6 y 5.7.** Al otro lado, para arrancar el sistema, el microcontrolador principal del módulo central recibe los caracteres y los interpreta según el programa.

Para recibir la cadena de bytes desde el módulo central, la función VISA Read se encarga de leer 15 bytes por vez y los retorna en el bufer de lectura.

Con los bytes en el buffer, el programa se encarga de separar en orden cada uno de ellos para compararlos con un caracter patrón. Se utiliza la función String Subset para retornar cada uno de los bytes especificados.

Como los caracteres vienen en orden desde el microcontrolador, el programa respeta ese mismo orden para su interpretación. El primer caracter es un carácter de aviso de recepción, se programó que fuese el carácter ASCII "B".

Los subsiguientes caracteres representan el estado de las paradas y estaciones conjuntamente con la velocidad de los cargobuses.

La asignación de los caracteres es la siguiente:

1. Segundo caracter representa la parada 0
2. Tercer caracter representa la estación 1
3. Cuarto caracter representa la parada 1
4. Quinto caracter representa la estación 2
5. Sexto caracter representa la parada 2
6. Séptimo caracter representa la estación 3
7. Octavo caracter representa al cargobus 1 parado
8. Noveno caracter representa al cargobus 1 con velocidad 1
9. Décimo caracter representa al cargobus 1 con velocidad 2
10. Décimo segundo caracter representa al cargobus 2 parado
11. Décimo tercer caracter representa al cargobus 2 con velocidad 1
12. Décimo cuarto caracter representa al cargobus 2 con velocidad 2

Los caracteres décimo primero y décimo quinto están reservados para un posible dato extra por cada cargobus.

Cada carácter ASCII que envía el microcontrolador principal puede ser "A" o "E". Si el carácter es una "A", un comparador lo registra y enciende un indicador, si el carácter es "E" el indicador se mantiene apagado.

Para contabilizar el número de carrocerías que parten de la parada 0, es decir el número de veces que se enciende el indicador de parada 0, se utilizaron los VI: Trigger and Gate y Statistics. Estos VI's extraen los pulsos positivos para posteriormente almacenarlos en un arreglo; para que éste proceso se de fue necesario convertir los datos de Array double de una sola fila provenientes del comparador a datos dinámicos, ya que los VI's necesitan ese formato para procesar la información. De igual forma para presentar los datos en el contador fue necesario convertirlos nuevamente a Array double de una fila.

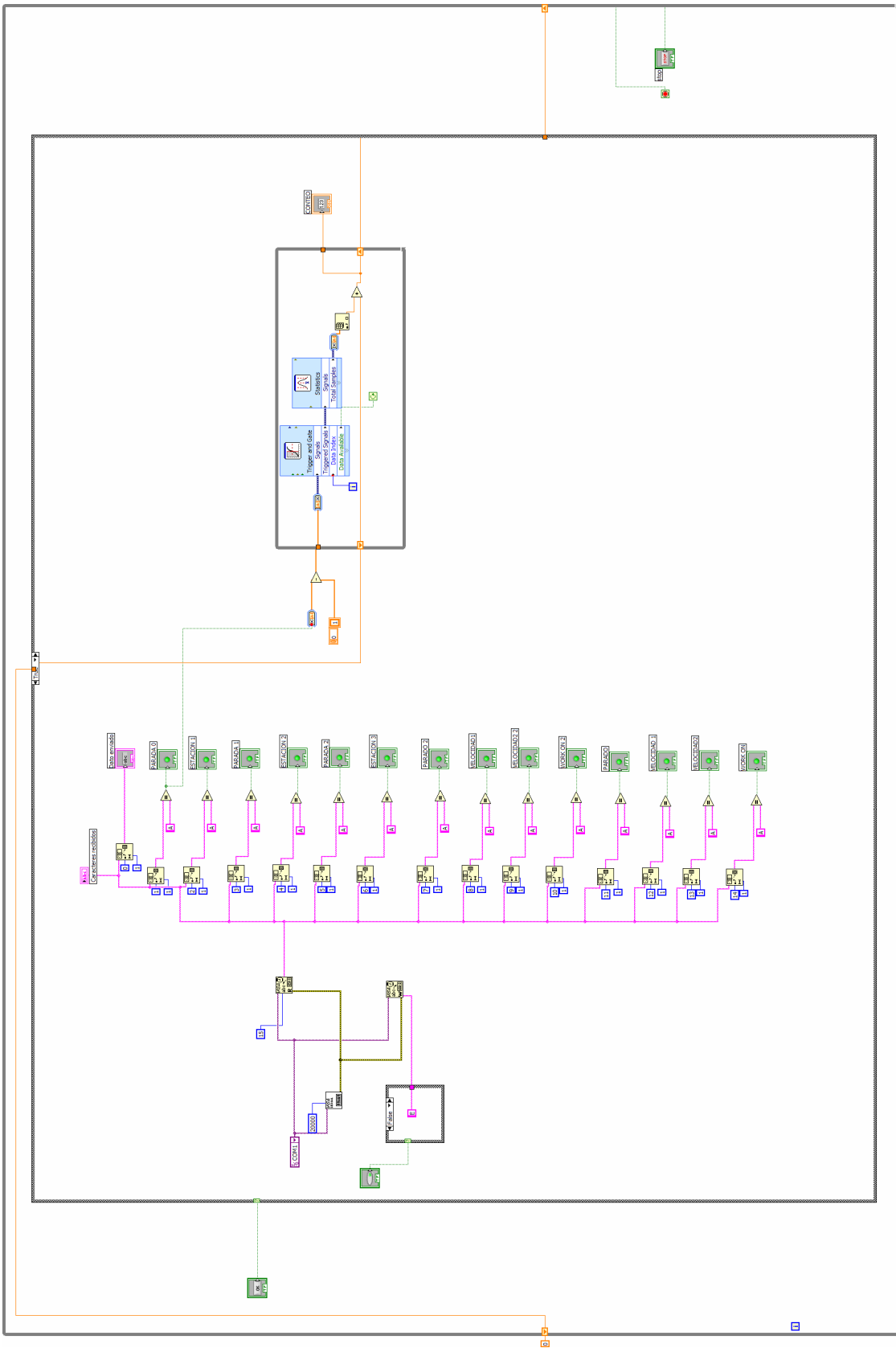


Figura 5.6 Diagrama de Bloque

### 5.3 El Panel de control

En el panel de control se puede visualizar la velocidad que lleva cada cargobus dentro de la línea y el estado de cada parada y estación. Los seis leds ordenados en secuencia en la parte superior del panel representan las tres paradas y las tres estaciones del sistema. Si un cargobus se detiene en una de ellas, el led que la representa se enciende automáticamente, y si el cargobus parte se apaga. De igual forma, los seis leds de la parte inferior representan la velocidad que lleva cada cargobus, si los cargobuses están parados, o tomando una velocidad media o alta es la información que se registra en el panel.

Se incluyó en el diseño un contador que lleva el balance de las unidades que salen desde la primera parada. Al lado izquierdo del panel se incluyó un pulsador para arrancar el sistema de comunicación de forma remota y dos botones para comenzar y detener la adquisición de datos. Dos indicadores de texto presentan los caracteres recibidos de la transmisión y el primer carácter por separado respectivamente.

Se diseñó el panel de control de esta manera algo sencilla con el objetivo de que cualquier operario o personal encargado de monitorear el sistema tenga la facilidad de distinguir los diferentes procesos que se llevan en la línea de la manera más clara posible.

En la **figura 5.7**, se puede observar el panel de control donde se distingue el ciclo de movimiento de los cargobuses a través de las paradas y estaciones, así como también la velocidad que lleva cada uno de ellos.



Figura 5.7 Panel de control programado para el arranque y monitoreo del sistema.

## Capítulo 6

### Conclusiones

El principal objetivo de ésta tesis fue implementar un sistema de comunicación, control y monitoreo de cargobuses o grúas suspendidas automatizadas utilizando la electrónica de control. Se diseñó un sistema distribuido en módulos, comunicados por cable bajo el estándar de comunicación industrial EIA-485, luz infrarroja y radiofrecuencia. El proyecto se enfocó en el uso de microcontroladores para controlar cada uno de los módulos del sistema.

Previo a la realización del diseño de construcción del sistema de comunicación, se analizó el funcionamiento de un sistema overhead real de cargobuses. Se visitó una planta de ensamblaje para constatar los detalles de implementación necesarios para el esquema. La arquitectura utilizada para lograr la comunicación entre módulos fue el resultado de un análisis minucioso del sistema real.

Como primera parte de ésta tesis, se diseñaron los circuitos que forman parte de las tarjetas electrónicas que integran los módulos de comunicación del sistema. Se probó su funcionamiento y rendimiento antes de su construcción. Por su bajo costo y accesibilidad en el mercado se utilizaron microcontroladores PIC para la implementación del control. Los resultados de realizar un diseño previo y de utilizar microcontroladores PIC fueron:

- Un mejor rendimiento de los módulos sin presencia de fallas.
- Una fabricación profesional de los módulos sin ningún tipo de arreglos imprevistos luego de su finalización.
- Se abarataron costos de construcción.

En el capítulo 2 se presentó un resumen con los tópicos de mayor importancia conceptual dentro de la tesis. Se incluyó en éste resumen los diferentes medios de transmisión de datos para tener una idea clara de las facilidades y dificultades de implementar un sistema de comunicación cuando el dispositivo a comunicar se encuentra en constante movimiento.

Se topó el tema de la comunicación serial y estándar EIA-485 con el objetivo de tener una idea concisa de cómo se transportarían los datos entre los diferentes módulos de manera alámbrica o inalámbrica.

Como parte fundamental al momento de programar los microcontroladores principales de cada módulo, se tomó muy en cuenta la velocidad de respuesta en los puertos de comunicación, así como en las salidas análogas.

Se puso mucho énfasis en lograr una respuesta rápida al momento de habilitar un cargobus desde el módulo central de control.

Para comunicar el módulo de comunicación central con los módulos de paradas y estaciones se consideró en gran medida la confiabilidad y la velocidad en la transmisión de los datos. La respuesta que presenta el módulo central a la solicitud de habilitación por parte de los módulos de paradas y estaciones es casi inmediata, casi no existe demoras por procesamiento de datos en los microcontroladores.

Con el uso del estándar EIA-485 para intercomunicar los módulos se logra:

- Separar las paradas y estaciones una distancia mayor a mil metros de la central de control.
- Interconectar los módulos utilizando cable UTP de fácil manipulación.
- Disminuir el ruido eléctrico en la transmisión.



Al momento de programar el microcontrolador principal del módulo de cargobuses se analizaron minuciosamente todas las variables que podrían afectar el funcionamiento del cargobus. Se consideraron factores externos que podrían presentar malfuncionamientos al momento de poner en marcha el sistema, factores como cortes de energía en la alimentación. Se manejó un control local de los motores de traslación y elevación para evitar posibles accidentes producto de fallas en la comunicación con un módulo de control externo.

Se buscó una manera eficiente de monitorear el sistema en tiempo real que fuese eficiente y de fácil acceso. La herramienta LabVIEW facilitó el monitoreo del sistema con una comunicación amigable y una transferencia de datos aceptable.

### **Trabajo futuro y perspectivas**

Si bien el sistema de comunicación implementado cumple con los objetivos propuestos, aún existen posibles mejoras que ayudarían a maximizar su desempeño, por ejemplo:

Crear una red multipunto entre el módulo central y los módulos de paradas y estaciones en lugar de una comunicación punto a punto, de ésta forma se facilita la expansión del sistema en el futuro.

Mejorar el diseño de construcción de los módulos. Asegurar de mejor manera los conectores DB9 a los módulos para evitar fallas de conexión en las tarjetas.

Optimizar el algoritmo de programación de ciertos programas. Esto disminuiría el número de líneas por programa y mejoraría el tiempo de procesamiento.

Utilizar un diseño más completo para implementar la comunicación IR entre las paradas, estaciones y cargobuses. Se podría incluir módulos prediseñados de transmisión.

Incluir fuentes de voltaje DC en cada en módulo, de ésta forma se evita conectar fuentes externas al momento de poner en marcha el sistema.

Mejorar el protocolo utilizado para comunicar el módulo central del sistema con el programa LabVIEW en el computador principal.

# **ANEXO A Programas**

Para programar los microcontroladores se utilizó el editor de texto MicroCode Studio Plus conjuntamente con el compilador PICBasic Pro 2.47 y el programador de prototipos IC-Prog version 1.05D; las tres herramientas son de distribución gratuita.

Se presenta a continuación los programas creados para cada uno de los microcontroladores, organizados según el módulo al cual pertenecen. Se incluye además los diagramas de flujo correspondientes a cada programa.

## A.1 Central de Control

### A.1.1 Programa del microcontrolador principal PIC16F877A, encargado de controlar el movimiento de los cargobuses y de recibir y transmitir la información a LabVIEW.

```
'*****
'* Name      : CENTRAL5.BAS                      *
'* Author    : Andrés Paladines Andrade          *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved                *
'* Date      : 15/07/2008                        *
'* Version   : 1.0                               *
'* Notes     :                                    *
'*           :                                    *
'*****

include "modedefs.bas"

ADCON1 = 7

'-----

'DEFINICION DE VARIABLES:
```

```

'-----
-----1-----
encendido var portd.5           'Luz de encendido
selector var portb.6           'Selector de tipo de arranque
arranque var portb.7           'Botón de arranque
dato var byte

'Datos enviados al micro LCD
parado_1 var byte
parado_2 var byte
vell_1 var byte
vell_2 var byte
vel2_1 var byte
vel2_2 var byte
dato4_1 var byte
dato4_2 var byte

'Datos enviados a la PC
g var byte
h var byte
i var byte
j var byte
k var byte
l var byte
m var byte
n var byte

'-----ENTRADAS:-----

'Parada 1:
presencial var porta.0

```

presencial\_1 var byte

**'Estación 1:**

subpresencial var porta.1

subpresencial\_1 var byte

**'Parada 2:**

presencia2 var porta.2

presencia2\_2 var byte

**'Estación 2:**

subpresencia2 var porta.3

subpresencia2\_2 var byte

**'Parada 3:**

presencia3 var porta.4

presencia3\_3 var byte

**'Estación 3:**

subpresencia3 var porta.5

subpresencia3\_3 var byte

'-----**SALIDAS:**-----

**'Parada1:**

habilitar1 var portb.0

a var byte

**'Estación 1:**

subhabilitar1 var portb.1

b var byte

**'Parada 2:**

habilitar2 var portb.2

c var byte

**'Estación 2:**

subhabilitar2 var portb.3

d var byte

**'Parada 3:**

habilitar3 var portb.4

e var byte

**'Estación 3:**

subhabilitar3 var portb.5

f var byte

-----

**'PROGRAMA**

-----

-----2-----

dato = "E" 'Inicializo dato

-----3-----

seleccion:

high encendido

IF selector = 0 then

-----

**'Envío de datos a las estaciones para que no arranquen**

low habilitar1

low subhabilitar1

```

low habilitar2

low subhabilitar2

low habilitar3

low subhabilitar3

'-----

'Envío hacia la PC para mantener LabView corriendo

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serin portd.7,N9600,200,sigal,dato

sigal:

'-----

'arranque manual

if arranque = 1 then

gosub pulso

goto continua

endif

else

```



'-----

**'Envío de datos a las estaciones para que no arranquen**

low habilitar1

low subhabilitar1

low habilitar2

low subhabilitar2

low habilitar3

low subhabilitar3

'-----

**'Envío hacia PC para mantener LabVIEW corriendo**

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serout portd.6,N9600,["B"]

serin portd.7,N9600,200,siga2,dato

siga2:

'-----

**'arranque from PC**

IF dato = "A" then

goto continua

endif

endif

goto seleccion

'-----

-----4-----

continua:

**'arranque:**

high encendido

pause 200

low encendido

pause 200

high encendido

pause 200

low encendido

pause 200

high encendido

-----5-----

proceso:

serout portd.4,N9600,["A"] 'habilitación al PIC LCD para arrancar

'también

-----6-----

**'Lógica del movimiento de los cargobuses**

if presencial = 0 then high subhabilitar3 'de estación 3 a parada 1

if presencial = 1 then low subhabilitar3

```
if subpresencial = 0 then high habilitar1 'de parada 1 a estación 1
if subpresencial = 1 then low habilitar1
if presencia2 = 0 then high subhabilitar1 'de estación 1 a parada 2
if presencia2 = 1 then low subhabilitar1
if subpresencia2 = 0 then high habilitar2 'de parada 2 a estación 2
if subpresencia2 = 1 then low habilitar2
if presencia3 = 0 then high subhabilitar2 'de estación 2 a parada 3
if presencia3 = 1 then low subhabilitar2
if subpresencia3 = 0 then high habilitar3 'de parada 3 a estación 3
if subpresencia3 = 1 then low habilitar3
```

-----7-----

#### **'Asignación de caracteres para el envío de datos serial**

```
if presencial = 1 then presencial_1 = "A" : a = "A"
if presencial = 0 then presencial_1 = "E" : a = "E"
if subpresencial = 1 then subpresencial_1 = "A" : b = "A"
if subpresencial = 0 then subpresencial_1 = "E" : b = "E"
if presencia2 = 1 then presencia2_2 = "A" : c = "A"
if presencia2 = 0 then presencia2_2 = "E" : c = "E"
if subpresencia2 = 1 then subpresencia2_2 = "A" : d = "A"
if subpresencia2 = 0 then subpresencia2_2 = "E" : d = "E"
if presencia3 = 1 then presencia3_3 = "A" : e = "A"
if presencia3 = 0 then presencia3_3 = "E" : e = "E"
if subpresencia3 = 1 then subpresencia3_3 = "A" : f = "A"
if subpresencia3 = 0 then subpresencia3_3 = "E" : f = "E"
```

-----8-----

#### **'Conversión de datos desde RF**

```
if portc.0 = 1 then parado_1 = "A" : g = "A"
if portc.0 = 0 then parado_1 = "E" : g = "E"
if portc.1 = 1 then vell_1 = "A" : h = "A"
if portc.1 = 0 then vell_1 = "E" : h = "E"
if portc.2 = 1 then vel2_1 = "A" : i = "A"
```

```
if portc.2 = 0 then vel2_1 = "E" : i = "E"
if portc.3 = 1 then dato4_1 = "A" : j = "A"
if portc.3 = 0 then dato4_1 = "E" : j = "E"
if portc.4 = 1 then parado_2 = "A" : k = "A"
if portc.4 = 0 then parado_2 = "E" : k = "E"
if portc.5 = 1 then vel1_2 = "A" : l = "A"
if portc.5 = 0 then vel1_2 = "E" : l = "E"
if portc.6 = 1 then vel2_2 = "A" : m = "A"
if portc.6 = 0 then vel2_2 = "E" : m = "E"
if portc.7 = 1 then dato4_2 = "A" : n = "A"
if portc.7 = 0 then dato4_2 = "E" : n = "E"
```

-----9-----

#### **'Envío de datos al PIC LCD**

```
serout porte.0,T2400,[presencial_1,subpresencial_1]
serout porte.1,T2400,[presencia2_2,subpresencia2_2]
serout porte.2,T2400,[presencia3_3,subpresencia3_3]
serout portd.0,T2400,[parado_1,vel1_1]
serout portd.1,T2400,[vel2_1,dato4_1]
serout portd.2,T2400,[parado_2,vel1_2]
serout portd.3,T2400,[vel2_2,dato4_2]
```

-----10-----

#### **'Envío de datos a la PC**

```
serout portd.6,N9600,["B"]
serout portd.6,N9600,[a]
serout portd.6,N9600,[b]
serout portd.6,N9600,[c]
serout portd.6,N9600,[d]
serout portd.6,N9600,[e]
serout portd.6,N9600,[f]
serout portd.6,N9600,[g]
serout portd.6,N9600,[h]
```

```

serout portd.6,N9600,[i]
serout portd.6,N9600,[j]
serout portd.6,N9600,[k]
serout portd.6,N9600,[l]
serout portd.6,N9600,[m]
serout portd.6,N9600,[n]

goto proceso
-----11-----
pulso:                ' Subrutina para estabilizar el pulsador
if arranque = 1 then goto pulso
pause 200
return

'-----

End

```

### **A1.1.1 Algoritmo global del programa**

1. El programa asigna variables para una mejor programación. Las variables están organizadas de acuerdo a la función del puerto asignado, el tipo de datos que manipulan, y si corresponden a entradas o salidas.
2. Se inicializa la variable “dato” con el carácter E para mantener la secuencia del programa. La variable “dato” representa la información recibida desde el computador personal relacionada al arranque remoto del sistema. El carácter E dentro de la programación significa arranque remoto desactivado.

3. Se mantiene al microcontrolador corriendo dentro de un lazo cerrado llamado "selección". El objetivo de este lazo es permitir la selección del tipo de arranque del sistema.

Dentro del lazo "selección" se mantiene encendido el indicador de arranque del módulo central. El indicador es representado por la variable "encendido". El indicador debe parpadear dos veces y luego permanecer encendido al momento de arrancar el sistema.

La variable "selector" representa en si al selector de arranque manual/automático del módulo central. Si la variable es 0, el programa asume un arranque manual si no lo es considera un arranque automático.

Dentro del arranque manual se mantiene en nivel bajo a todas las salidas asignadas a cada uno de los módulos de paradas y estaciones, de esta forma los microcontroladores PIC12F675 que controlan el envío de datos a cada parada y estación mantendrán a cada uno de los módulos sin arrancar.

Conjuntamente se envía el caracter "B" al computador personal para mantener activa la comunicación con LabVIEW y se reciben los caracteres que envía el computador.

Una vez que la variable "selector" es 0, la variable "arranque", asignada al pulsador de arranque manual, permite encender el sistema completo. Si la variable "arranque" es 1, el programa salta a una subrutina programada para evitar el rebote del pulsador. Subsecuentemente el programa provoca un salto a otra línea del programa llamada "continua" donde otro lazo cerrado llamado "proceso" controlará la comunicación con el resto de módulos.

Si “selector” es 1, se mantiene en nivel bajo a las salidas de paradas y estaciones y se envía el carácter “B” al computador personal, todo esto para arrancar simultáneamente todos los módulos.

Conjuntamente con el dato enviado, el microcontrolador principal abre un puerto para recibir los caracteres desde el computador, si recibe un carácter “A”, inmediatamente el programa salta a “continua” para continuar con el control de la comunicación.

4. Lo primero que hace el programa luego del salto a “continua” es hacer parpadear el indicador de arranque e inmediatamente entrar al lazo cerrado “proceso”.
5. En “proceso” se habilita el PIC16F877A que controla la pantalla LCD para comenzar a presentar los datos del sistema.
6. Se configura la lógica de movimiento de los cargobuses a través de la habilitación de paradas y estaciones. Un estado alto representa la habilitación de una parada o estación.
7. Para enviar la información al computador personal y al PIC que controla la pantalla LCD es necesario asignar caracteres a cada variable que represente el estado de paradas y estaciones.

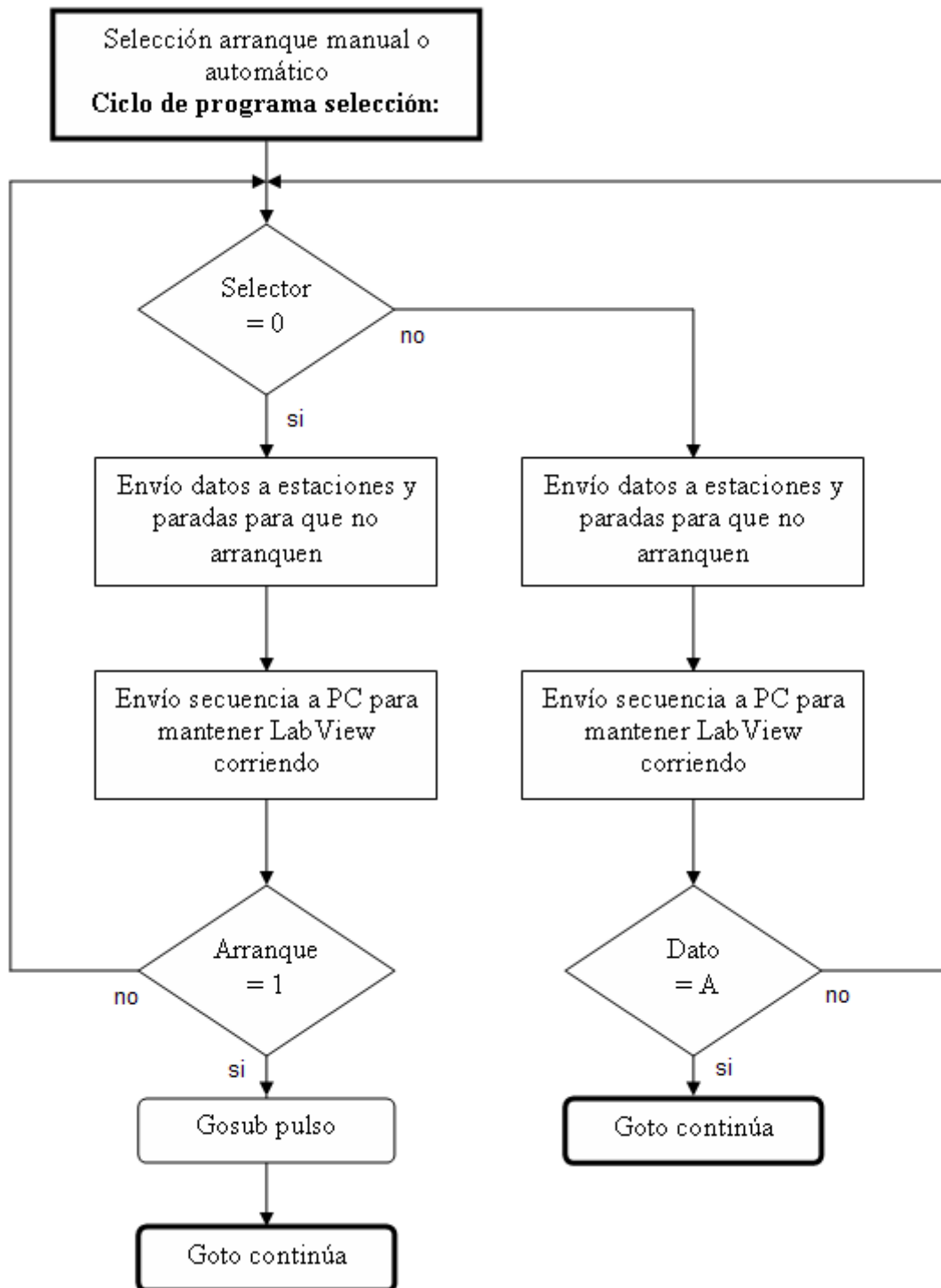
Si el carácter asignado a las variables es “A”, significa que la parada o estación está habilitada para que un cargobus parta de ella, si el carácter asignado es “E” la parada o estación está deshabilitada.

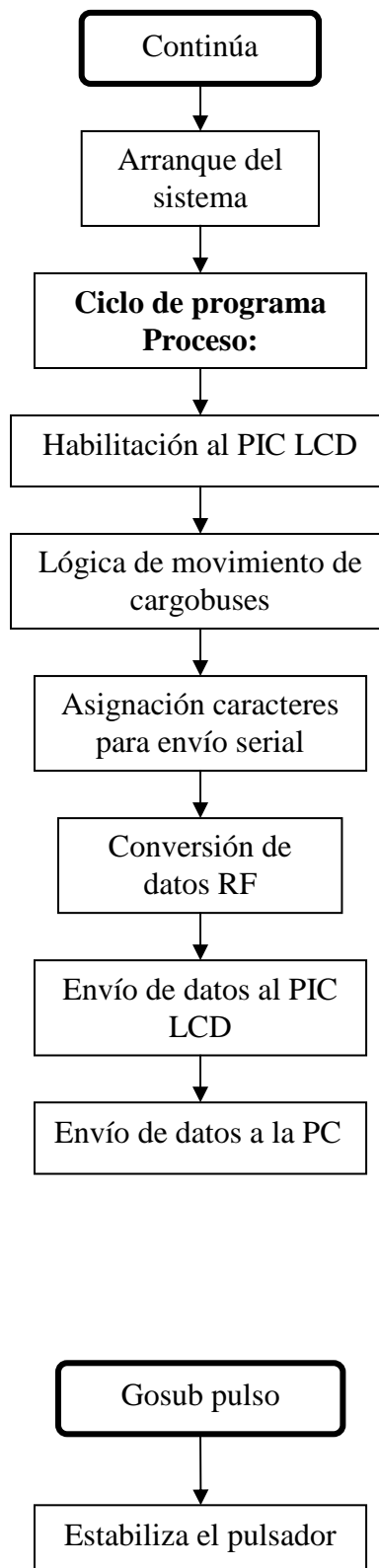
8. Se recibe del puerto C la información proveniente de la comunicación RF con los dos cargobuses y se asigna caracteres a variables respectivas para el envío de la información al computador y al PIC del LCD respectivamente.

9. Se envía al PIC que controla la pantalla LCD la información de cada parada y estación conjuntamente con la información de los cargobuses.
10. Se envía la cadena de bytes que representa el estado de las paradas, estaciones y cargobuses del sistema.
11. Subrutina "pulso" para estabilizar el pulsador.



### A1.1.2 Diagrama de Flujo





## A.1.2 Programa del PIC16F877A, encargado de presentar información del sistema en la pantalla LCD.

```
'*****
'* Name      : LCD_3.BAS                               *
'* Author    : Andrés Paladines Andrade                *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved                     *
'* Date      : 15/07/2008                               *
'* Version   : 1.0                                       *
'* Notes     :                                           *
'*           :                                           *
'*****

ADCON1 = 7

include "modedefs.bas"

define LCD_DREG PORTC

DEFINE LCD_DBIT 4

DEFINE LCD_RSREG PORTD

DEFINE LCD_RSBIT 2

DEFINE LCD_EREG PORTD

DEFINE LCD_EBIT 3

-----1-----

encendido var portd.0

'Dato de arranque desde la central

encienda var byte

'Datos desde la central

estacion1 var byte

parada1 var byte

estacion2 var byte
```

parada2 var byte  
estacion3 var byte  
parada0 var byte  
parado\_1 var byte  
parado\_2 var byte  
vell1\_1 var byte  
vell1\_2 var byte  
vel2\_1 var byte  
vel2\_2 var byte  
dato4\_1 var byte  
dato4\_2 var byte

p0\_1 var byte  
p0\_2 var byte  
p0\_3 var byte  
p1\_1 var byte  
p1\_2 var byte  
p1\_3 var byte  
p2\_1 var byte  
p2\_2 var byte  
p2\_3 var byte  
p3\_1 var byte  
p3\_2 var byte  
p3\_3 var byte  
p4\_1 var byte  
p4\_2 var byte  
p4\_3 var byte  
p5\_1 var byte  
p5\_2 var byte  
p5\_3 var byte

'-----

-----2-----

'arranque:

high encendido

pause 200

low encendido

pause 200

high encendido

pause 200

low encendido

pause 200

high encendido

'-----

-----3-----

pause 200

LCDout \$FE,1

lcdout \$FE,\$80," -"

PAUSE 400

LCDout \$FE,1

lcdout \$FE,\$80," --"

PAUSE 400

LCDout \$FE,1

lcdout \$FE,\$80," --C"

PAUSE 400

LCDout \$FE,1

lcdout \$FE,\$80," --CA"

PAUSE 400

LCDout \$FE,1

lcdout \$FE,\$80," --CAR"

PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARG"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGO"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOB"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBU"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBUS"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBUS "  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBUS I"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBUS IN"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBUS INC"  
PAUSE 400  
LCDout \$FE,1  
lcdout \$FE,\$80," --CARGOBUS INC-"  
PAUSE 400

```
LCDout $FE,1
lcdout $FE,$80,"--CARGOBUS INC--"
pause 600
LCDout $FE,1
lcdout $FE,$80," "
pause 400
LCDout $FE,1
lcdout $FE,$80,"--CARGOBUS INC--"
pause 400
LCDout $FE,1
lcdout $FE,$80," "
pause 400
LCDout $FE,1
Lcdout $FE,$80,"--CARGOBUS INC--"
pause 400
LCDout $FE,1
lcdout $FE,$82," "
pause 400
LCDout $FE,1
Lcdout $FE,$80,"--CARGOBUS INC--"
pause 500
LCDout $FE,1
lcdout $FE,$82," "
pause 400
-----4-----
'Condición de arranque
condicion:

serin portb.7,N9600,encienda      'dato desde el micro principal
IF encienda = "A" then
goto inicio
```

```

endif

goto condicion

-----5-----

inicio:
PAUSE 1000

'Recibir datos desde el PIC principal

serin portb.0,T2400,parada0,estacion1
'siga1:
serin portb.1,T2400,parada1,estacion2
'siga2:
serin portb.2,T2400,parada2,estacion3
'siga3:
serin portb.3,T2400,parado_1,vell_1
'siga4:
serin portb.4,T2400,vel2_1,dato4_1
'siga5:
serin portb.5,T2400,parado_2,vell_2
'siga6:
serin portb.6,T2400,vel2_2,dato4_2
'siga7:
-----6-----

LCDout $FE,1

IF parado_1 = "A" and vell_1 = "E" and vel2_1 = "E" then
LCDOUT $FE,$80,"CARGO1: PARADO"
endif

IF parado_1 = "E" and vell_1 = "A" and vel2_1 = "E" then

```



```
LCDOUT $FE,$80,"CARGO1: VELCD 1"
```

```
endif
```

```
IF parado_1 = "E" and vell_1 = "E" and vel2_1 = "A" then
```

```
LCDOUT $FE,$80,"CARGO1: VELCD 2"
```

```
endif
```

```
IF parado_2 = "A" and vell_2 = "E" and vel2_2 = "E" then
```

```
LCDOUT $FE,$C0,"CARGO2: PARADO"
```

```
endif
```

```
IF parado_2 = "E" and vell_2 = "A" and vel2_2 = "E" then
```

```
LCDOUT $FE,$C0,"CARGO2: VELCD 1"
```

```
endif
```

```
IF parado_2 = "E" and vell_2 = "E" and vel2_2 = "A" then
```

```
LCDOUT $FE,$C0,"CARGO2: VELCD 2"
```

```
endif
```

```
-----7-----
```

```
'Realiza equivalencias para mostrar en el LCD
```

```
if parada0 = "A" then p0_1 = 79 : p0_2 = 78 : p0_3 = 32
```

```
if parada0 = "E" then p0_1 = 79 : p0_2 = 70 : p0_3 = 70
```

```
if estacion1 = "A" then p1_1 = 79 : p1_2 = 78 : p1_3 = 32
```

```
if estacion1 = "E" then p1_1 = 79 : p1_2 = 70 : p1_3 = 70
```

```
if parada1 = "A" then p2_1 = 79 : p2_2 = 78 : p2_3 = 32
```

```
if parada1 = "E" then p2_1 = 79 : p2_2 = 70 : p2_3 = 70
```

```
if estacion2 = "A" then p3_1 = 79 : p3_2 = 78 : p3_3 = 32
```

```
if estacion2 = "E" then p3_1 = 79 : p3_2 = 70 : p3_3 = 70
```

```
if parada2 = "A" then p4_1 = 79 : p4_2 = 78 : p4_3 = 32
```

```
if parada2 = "E" then p4_1 = 79 : p4_2 = 70 : p4_3 = 70
```

```

if estacion3 = "A" then p5_1 = 79 : p5_2 = 78 : p5_3 = 32
if estacion3 = "E" then p5_1 = 79 : p5_2 = 70 : p5_3 = 70
-----8-----
'Muestra datos de las estaciones y paradas en el LCD

pause 1000
LCDout $FE,1

LCDOUT    $FE,$80,"      E:      ",p1_1,p1_2,p1_3,"      ",p3_1,p3_2,p3_3,"
",p5_1,p5_2,p5_3
LCDOUT    $FE,$C0,"      P:      ",p0_1,p0_2,p0_3,"      ",p2_1,p2_2,p2_3,"
",p4_1,p4_2,p4_3

goto inicio

end

```

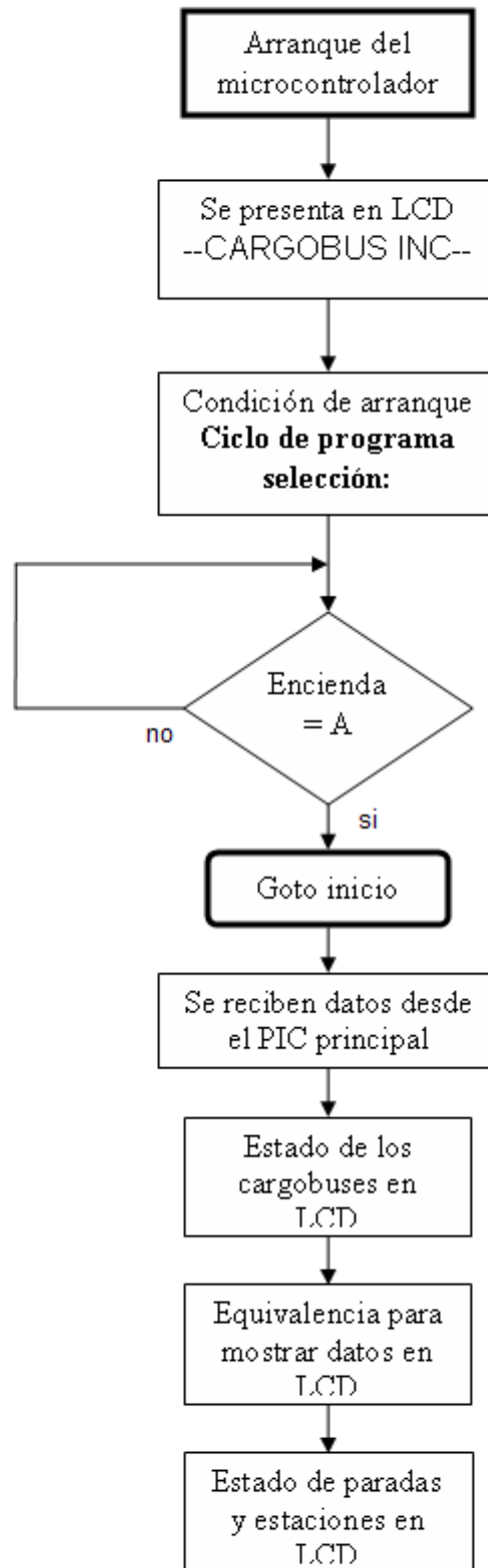
### **A1.2.1 Algoritmo global del programa**

1. Asignación de variables del programa.
2. Se arranca el sistema. El LED de encendido parpadea dos veces y luego permanece encendido.
3. Con una breve pausa para arrancar la LCD, el programa presenta en la pantalla el texto --CARGOBUS INC-- con desplazamientos desde la derecha.
4. El programa entra en un lazo cerrado esperando por su habilitación proveniente del microcontrolador principal. En ésta etapa, la LCD no muestra ningún dato. Si el carácter recibido es "A", el programa salta del

lazo cerrado y entra en el lazo "inicio" donde se llevará a cabo todo el procesamiento y presentación de información.

5. En el lazo cerrado "inicio" en primer lugar se recibe todos los datos de paradas, estaciones y cargobuses desde el microcontrolador principal. Se enviaron desde el PIC central dos caracteres por canal para garantizar una comunicación confiable, más de dos caracteres desordena la secuencia de los datos.
6. Dependiendo de la información recibida de los cargobuses, se presenta en la pantalla LCD su estado, si están parados, con velocidad 1 o 2.
7. Se realizan equivalencias para presentar las letras "ON" u "OFF" en la LCD. On significa parada ocupada y Off parada desocupada. Dependiendo de la información recibida de las paradas y estaciones, se asignan números determinados que corresponden al código ASCII de "ON" y "OFF" a una serie de variables que posteriormente serán colocadas en orden para presentar el estado de las paradas y estaciones en la pantalla.
8. Luego de una pausa de un segundo luego de presentar el estado de los cargobuses, se presenta el estado de las paradas y estaciones en la pantalla LCD.

### A1.2.2 Diagrama de Flujo



### A.1.3 Programa del PIC12F675, encargado de recibir información desde las estaciones y paradas.

```
'*****  
'* Name      : RECIVCENTRAL.BAS      *  
'* Author    : Andrés Paladines Andrade *  
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *  
'*           : All Rights Reserved   *  
'* Date      : 29/05/2008            *  
'* Version   : 1.0                   *  
'* Notes     :                        *  
'*           :                        *  
'*****
```

```
include "modedefs.bas"
```

```
CMCON = %111
```

```
ANSEL = %0000
```

```
-----1-----
```

```
presencia var byte
```

```
to_central var gpio.1
```

```
'-----
```

```
' PROGRAMA
```

```
'-----
```

```
'-----
```

```
-----2-----
```

```
'arranque:
```

```
pause 1000
```

```

'-----
-----3-----
inicio:

DEFINE CHAR_PACING 500

serin gpio.0,T9600,presencia      'recibo la presencia desde la estación
                                  'o parada

if presencia = "A" then high to_central      'si esta presente el
                                             'cargobus luego la salida es alta
if presencia = "E" then low to_central      'si no esta presente el
                                             'cargobus luego la salida es baja

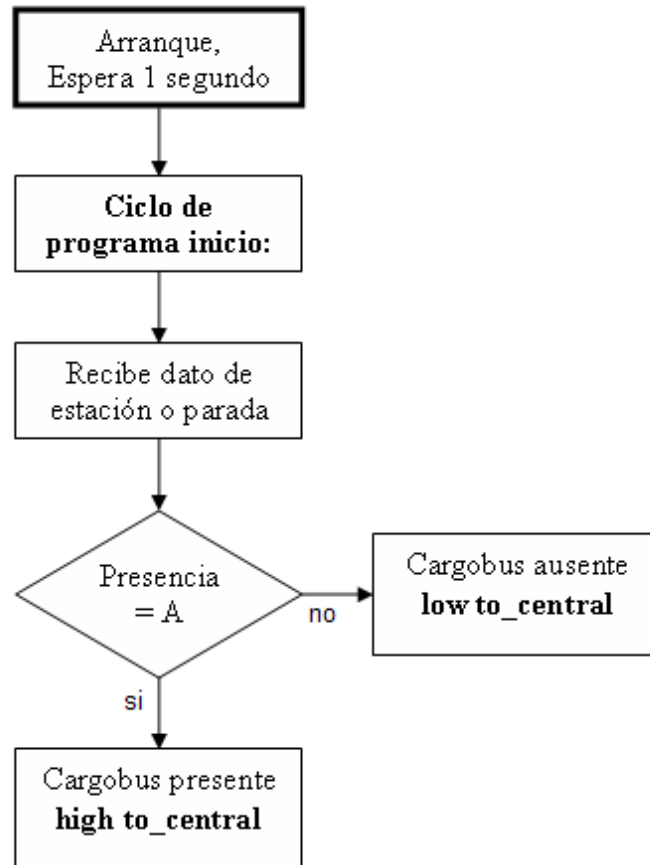
goto inicio

```

### **A1.3.1 Algoritmo global del programa**

1. Asignación de variables del programa.
2. El sistema arranca luego de un segundo con el objetivo de que el microcontrolador, al momento de comenzar a recibir la información serial proveniente del módulo de paradas y estaciones, tenga disponible los datos en el puerto y no se generen fallas.
3. El programa entra en un lazo cerrado donde el microcontrolador recibe el estado de la parada o estación. Si el carácter que llega es "A" se pone en alto el pin de salida "to\_central" que está conectado al microcontrolador principal. Si llega "E" el pin permanece en bajo.

## Diagrama de Flujo



#### A.1.4 Programa del PIC12F675, encargado de enviar información hacia las estaciones y paradas.

```
*****
'* Name      : ENVIOCENTRAL.BAS          *
'* Author    : Andrés Paladines Andrade *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved       *
'* Date      : 29/05/2008                *
'* Version   : 1.0                       *
'* Notes     :                            *
'*           :                            *
*****
```

```
include "modedefs.bas"
```

```
CMCON = %111
```

```
ANSEL = %0000
```

```
-----1-----
```

```
habilitar var byte
```

```
from_central var gpio.1
```

```
'-----
'                                     ' PROGRAMA
'-----
'-----
```

```
-----2-----
```

```
'arranque:
```

```
pause 1000
```



```

'-----
-----3-----
inicio:
DEFINE CHAR_PACING 500

if from_central = 1 then habilitar = "A"      'si la central habilita
                                           'luego envía A
if from_central = 0 then habilitar = "E"      'si la central no habilita
                                           'luego envía E

serout gpio.0,T9600,[habilitar]              'Envío de la señal

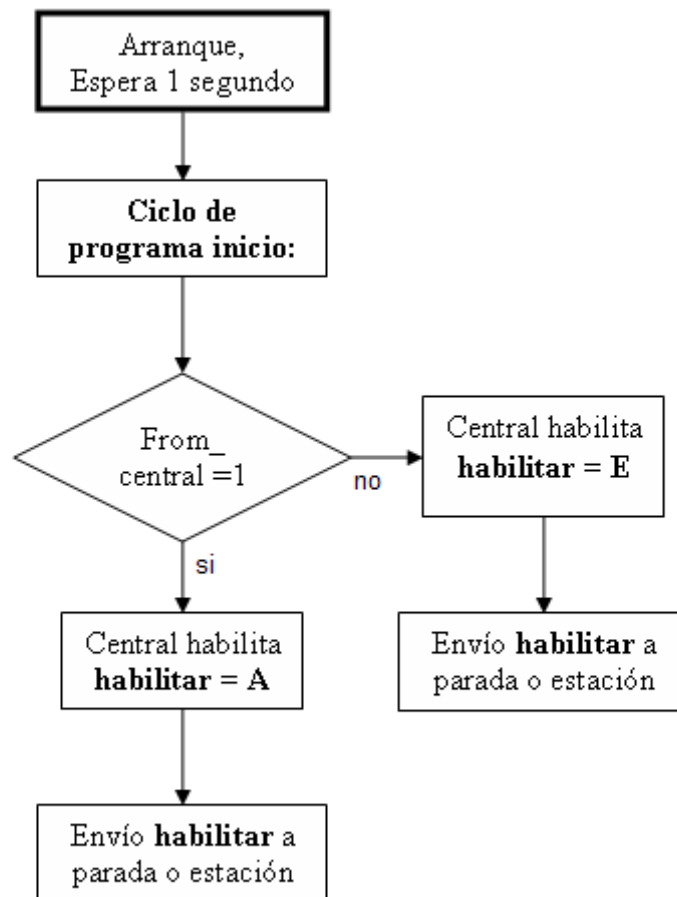
goto inicio

```

### **Algoritmo global del programa**

1. Asignación de variables del programa.
2. Arranque del sistema. El microcontrolador espera un segundo.
3. Recibe la información del microcontrolador central y la envía serialmente al módulo de la parada o estación.

## Diagrama de Flujo



## A.2 Estaciones

### A.2.1 Programa del PIC16F628A, encargado de procesar toda las entradas y salidas del módulo.

```
*****
'* Name      : estacionprueba.BAS          *
'* Author    : Andrés Paladines Andrade   *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved        *
'* Date      : 01/06/2008                  *
'* Version   : 1.0                          *
'* Notes     :                               *
'*           :                               *
*****
```

```
include "modedefs.bas"
```

```
cmcon = 7
```

```
-----1-----
```

```
encendido var porta.1
```

```
habilitar var byte
```

```
operario var byte
```

```
'-----
```

```
-----
```

```
-----2-----
```

```
'arranque:
```

```
high encendido
```

```
pause 200
```

```
low encendido
```

```
pause 200
```

```
high encendido
```

pause 200

low encendido

pause 200

high encendido

pause 200

'-----

-----

-----3-----

inicio:

DEFINE CHAR\_PACING 1000

'-----

-----4-----

if portb.0 = 1 then

'Si recibe habilitación:

high portb.1

'Enciende señal de aviso

habilitar = "A"

serout portb.2,N2400,[habilitar]

'Envía habilitación de forma

'infrarroja al cargobus

ENDIF

IF portb.0 = 0 then low portb.1

'Si no recibe habilitación apaga

'señal de aviso

'-----

-----5-----

if porta.0 = 0 then low portb.5

'Si no hay presencia de cargobus

'apago señal de aviso

if porta.0 = 1 then high portb.5

'Si hay presencia de cargobus

'enciendo la señal

'-----

-----6-----

if portb.4 = 1 then

'Señal operario

operario = "E"

```

serout portb.3,N2400,[operario]      'Envío de la señal del operario
                                     'de forma infrarroja

endif

'-----

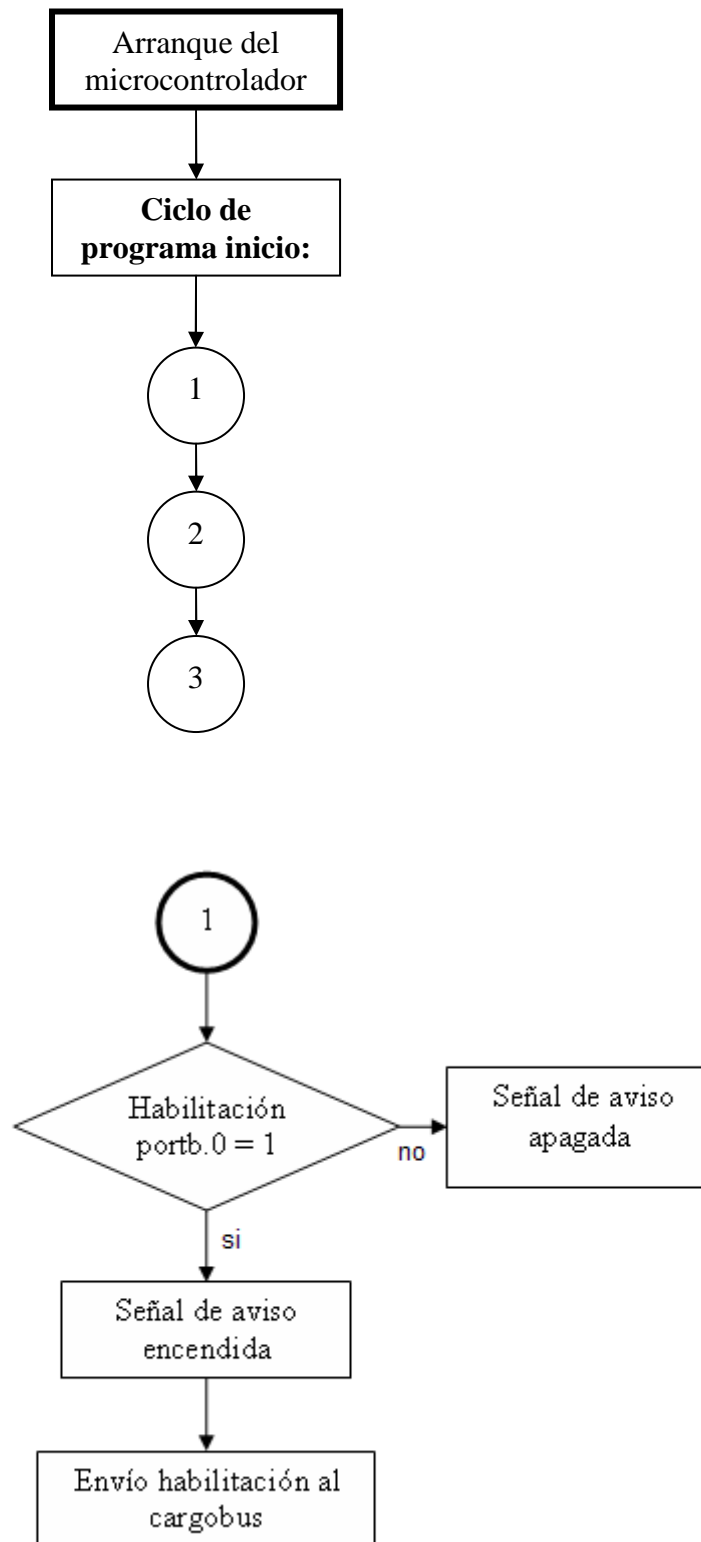
goto inicio

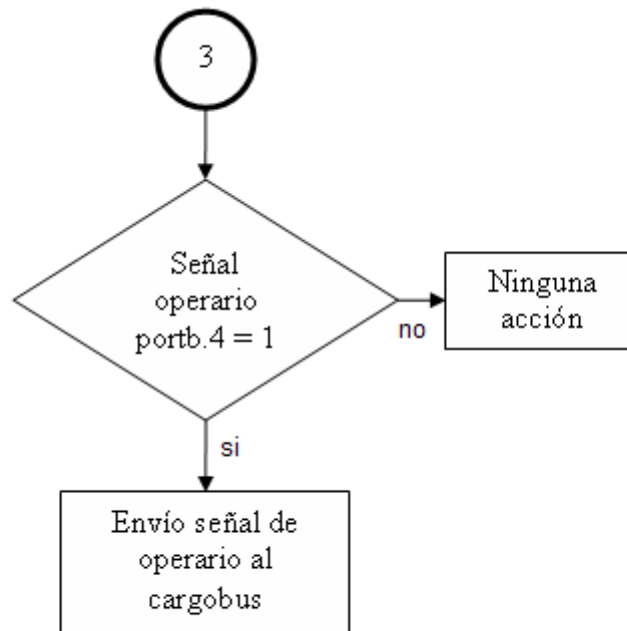
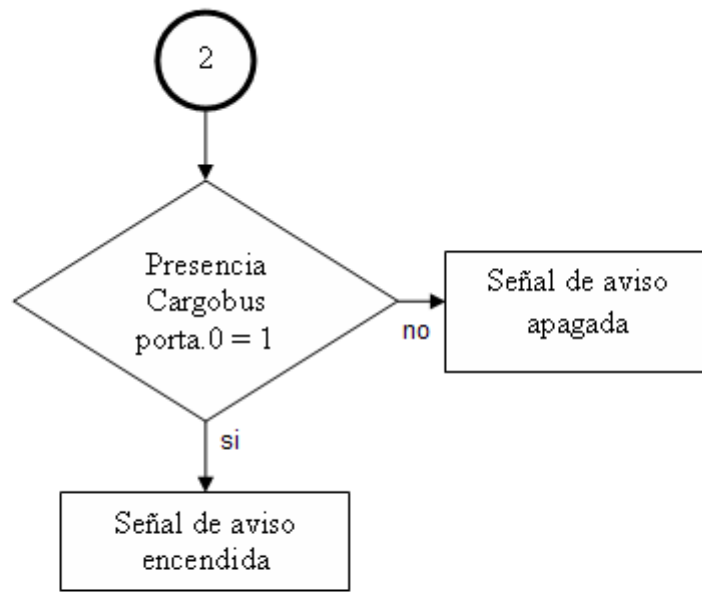
```

### **Algoritmo global del programa**

1. Asignación de variables del programa.
2. Arranque del sistema. El LED de encendido del módulo parpadea dos veces y luego permanece encendido.
3. Ingreso al lazo cerrado de programa.
4. Si el microcontrolador principal PIC16F628A de la estación recibe la respectiva habilitación de partida de cargobus desde el módulo central, enciende un LED de aviso de habilitación y a continuación envía seriamente vía infrarroja el carácter "A" hacia el cargobus. Si el microcontrolador principal no recibe habilitación, apaga el LED de aviso y no envía ningún carácter al cargobus.
5. Si un cargobus llega a la estación se enciende un aviso de presencia, si ningún cargobus llega, el aviso permanece apagado.
6. Si el operario acciona su pulsador, el microcontrolador principal envía serialmente vía radiofrecuencia el carácter "E" hacia el cargobus, si el operario no acciona su pulsador el microcontrolador no envía ningún carácter.

## Diagrama de Flujo





## A.3 Paradas

### A.3.1 Programa del PIC16F628A, encargado de procesar toda las entradas y salidas del módulo.

```
*****
'* Name      : paradaprueba.BAS          *
'* Author    : Andrés Paladines Andrade *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved      *
'* Date      : 11/07/2008                *
'* Version   : 1.0                       *
'* Notes     :                           *
'*           :                           *
*****

include "modedefs.bas"

cmcon = 7

-----1-----
encendido var porta.1
habilitar var byte
operario var byte

'-----
-----2-----

'arranque:
high encendido
pause 200
low encendido
pause 200
high encendido
pause 200
low encendido
```



```

pause 200

high encendido

pause 200

'-----
-----3-----

inicio:

DEFINE CHAR_PACING 1000

'-----

-----4-----

if portb.0 = 1 then                'Si recibe habilitación:
high portb.1                       'Envía señal de aviso
habilitar = "A"
SEROUT portb.2,N2400,[habilitar]    'Envía habilitación de forma
                                     'infrarroja al cargobus

ENDIF

IF portb.0 = 0 then low portb.1     'Si no recibe habilitación apaga
                                     'señal de aviso

'-----

-----5-----

if porta.0 = 0 then low portb.3     'Si no hay presencia de cargobus
                                     'apaga señal de aviso

if porta.0 = 1 then high portb.3    'Si hay presencia de cargobus
                                     'enciende la señal

'-----

goto inicio

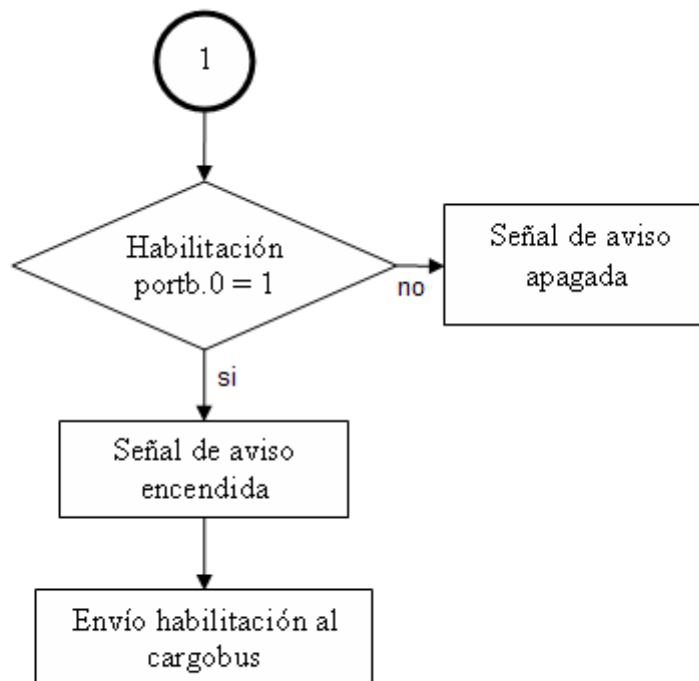
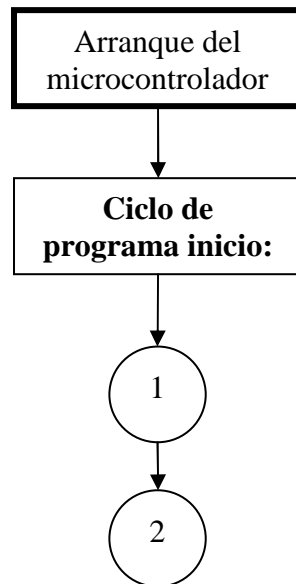
```

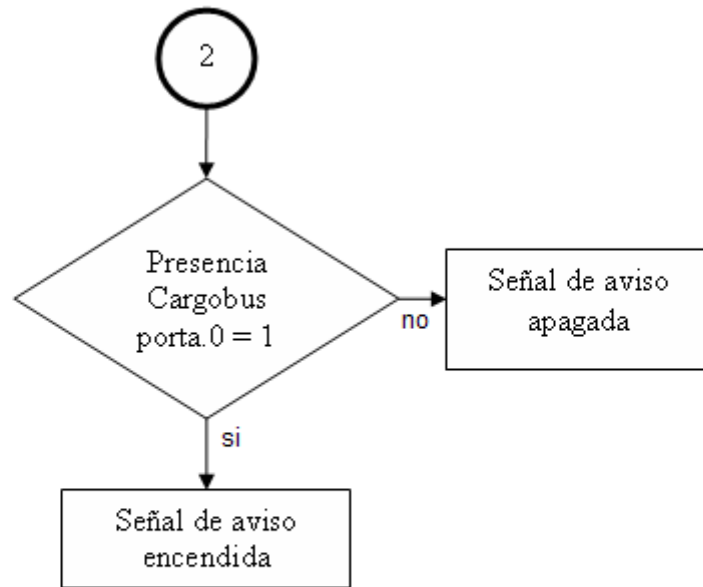
## Algoritmo global del programa

1. Asignación de variables del programa.

2. Arranque del sistema. El LED de encendido parpadea dos veces y luego permanece encendido.
3. Ingreso al lazo cerrado de programa.
4. Si el microcontrolador principal PIC16F628A de la parada recibe la respectiva habilitación de partida de cargobus desde el módulo central, enciende un LED de aviso de habilitación y a continuación envía seriamente vía infrarroja el caracter "A" hacia el cargobus. Si el microcontrolador principal no recibe habilitación, apaga el LED de aviso y no envía ningún caracter al cargobus.
5. Si un cargobus llega a la parada se enciende un aviso de presencia, si ningún cargobus llega, el aviso permanece apagado.

## Diagrama de Flujo





## A.4 Paradas y Estaciones

Los siguientes programas se utilizaron en ambos módulos.

### A.4.1 Programa del PIC12F675, encargado de recibir información desde la central.

```
'*****  
'* Name      : RECIVESTACION.BAS          *  
'* Author    : Andrés Paladines Andrade  *  
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *  
'*           : All Rights Reserved       *  
'* Date      : 29/05/2008                 *  
'* Version   : 1.0                        *  
'* Notes     :                            *  
'*           :                            *  
'*****
```

```
include "modedefs.bas"
```

```
CMCON = %111
```

```
ANSEL = %0000
```

```
-----1-----
```

```
habilitar var byte
```

```
to_estacion var gpio.1
```

```
'-----
```

```
' PROGRAMA
```

```
'-----
```

```
'-----
```

```
-----2-----
```

```
'arranque:
```

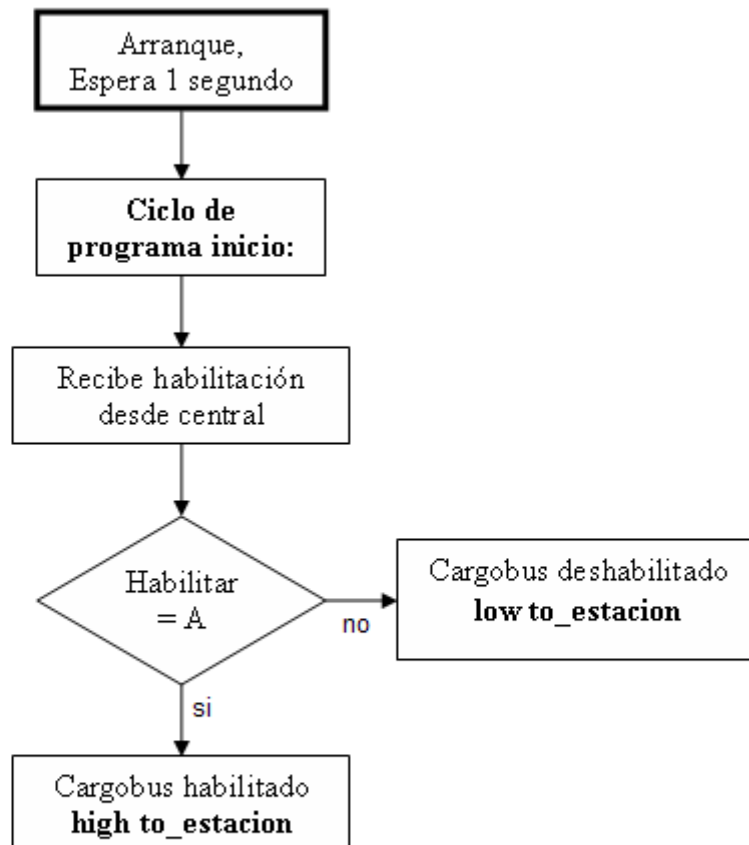
```
pause 1000
```

```
'-----  
-----3-----  
inicio:  
DEFINE CHAR_PACING 500  
-----4-----  
serin gpio.0,T9600,habilitar          'recibo la habilitación desde  
                                       'la central  
  
if habilitar = "A" then high to_estacion      'si esta habilitado el  
                                       'cargobus, luego la salida es alta  
if habilitar = "E" then low to_estacion      'si no esta habilitado  
                                       'el cargobus, luego la salida es baja  
  
goto inicio
```

### **Algoritmo global del programa**

1. Asignación de variables del programa.
2. Arranque del sistema. El microcontrolador espera un segundo.
3. Ingreso al lazo cerrado de programa.
4. El PIC12F675 espera continuamente por el byte proveniente de la comunicación con el módulo central para asignarlo a la variable "habilitar". Si "habilitar" es igual a "A" el PIC pone en alto el pin de salida conectado al microcontrolador principal de la parada o estación. Si "habilitar" es igual a "E", el pin de salida permanece en nivel bajo.

## Diagrama de Flujo



## A.4.2 Programa del PIC12F675, encargado de enviar información hacia la central.

```
'*****
'* Name      : ENVIOESTACION.BAS          *
'* Author    : Andrés Paladines Andrade  *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved       *
'* Date      : 29/05/2008                 *
'* Version   : 1.0                       *
'* Notes     :                            *
'*           :                            *
'*****
```

```
include "modedefs.bas"
```

```
CMCON = %111
```

```
ANSEL = %0000
```

```
-----1-----
```

```
presencia var byte
```

```
from_estacion var gpio.1
```

```
aviso var gpio.2
```

```
'-----
```

```
'PROGRAMA
```

```
'-----
```

```
'-----
```

```
-----2-----
```

```
'arranque:
```

```
pause 1000
```



```

'-----
-----3-----
inicio:

DEFINE CHAR_PACING 500
-----4-----
if from_estacion = 0 then presencia = "A" : high aviso 'si la
                                'estación detecta presencia luego envía A
if from_estacion = 1 then presencia = "E" : Low aviso 'si la
                                'estación no detecta presencia luego envía E
-----5-----
serout gpio.0,T9600,[presencia]

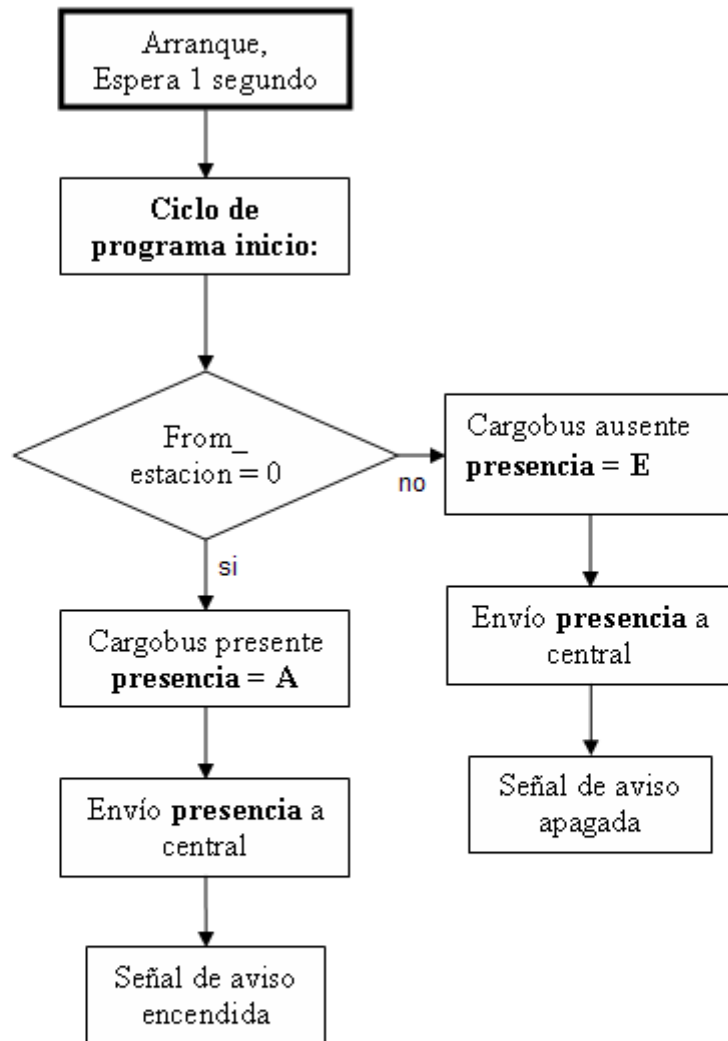
goto inicio

```

### **Algoritmo global del programa**

1. Asignación de variables del programa.
2. Arranque del sistema. El microcontrolador espera un segundo.
3. Ingreso al lazo cerrado de programa.
4. Si el pin de entrada conectado al microcontrolador principal de la parada o estación está en bajo, se asigna a la variable “presencia” el caracter “A”. Si el pin está en alto se asigna a “presencia” el carácter “E”.
5. El PIC12F675 envía serialmente la variable “presencia” al módulo central.

## Diagrama de Flujo



## A.5 Cargobus

### A.5.1 Programa del microcontrolador principal PIC16F877A.

```
'*****
'* Name      : CARGOPRUEBV.3.BAS                               *
'* Author    : Andrés Paladines Andrade                       *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved                             *
'* Date      : 29/05/2008                                       *
'* Version   : 1.0                                              *
'* Notes     :                                                 *
'*           :                                                 *
'*****

include "modedefs.bas"

ADCON1 = 7 ; apaga los convertidores del puerto A
-----1-----
habilitar var byte
operario var byte
encendido var portb.7
'-----

'ENTRADAS

sensor var portb.0 ; etiqueta pin 33, puerto RB0
switch1 var portb.1 ; etiqueta pin 34, puerto RB1
switch2 var portb.2 ; etiqueta pin 35, puerto RB2
siestacion var portb.3 ; etiqueta pin 36, puerto RB3
'.....

motorliup var portd.0 ; etiqueta pin 19, puerto RD0
motorlidown var portd.1 ; etiqueta pin 20, puerto RD1
motor2iup var portd.2 ; etiqueta pin 21, puerto RD2
```

```

motor2idown var portd.3      ; etiqueta pin 22, puerto RD3
'.....

'SALIDAS
velocidad0 var portc.0      ; etiqueta pin 15, puerto RC0
velocidad1 var portc.1      ; etiqueta pin 16, puerto RC1
velocidad2 var portc.2      ; etiqueta pin 17, puerto RC2
velocidad00 var porta.2     ; etiqueta pin 4, puerto RA2
velocidad11 var porta.3     ; etiqueta pin 5, puerto RA3
velocidad22 var porta.4     ; etiqueta pin 6, puerto RA4
velocidad0_0 var portc.3    ; etiqueta pin 18, puerto RC3
velocidad1_1 var portc.4    ; etiqueta pin 23, puerto RC4
velocidad2_2 var portc.5    ; etiqueta pin 24, puerto RC5

'.....

motor1oup var portd.4       ; etiqueta pin 27, puerto RD4
motor11oup var porta.5      ; etiqueta pin 7, puerto RA5
motor1odown var portd.5     ; etiqueta pin 28, puerto RD5
motor11odown var porte.0    ; etiqueta pin 8, puerto RE0
motor2oup var portd.6       ; etiqueta pin 29, puerto RD6
motor22oup var porte.1      ; etiqueta pin 9, puerto RE1
motor2odown var portd.7     ; etiqueta pin 30, puerto RD7
motor22odown var porte.2    ; etiqueta pin 10, puerto RE2

'.....

'-----

arranque var byte
arranque2 var byte
arranque3 var byte
arranque4 var byte

```

```
arranque5 var byte
arranque6 var byte
arranque7 var byte
segunda var byte
primera var byte
freno var byte
```

```
'-----
```

```
-----2-----
```

```
'arranque:
```

```
high encendido
```

```
pause 200
```

```
low encendido
```

```
pause 200
```

```
high encendido
```

```
pause 200
```

```
low encendido
```

```
pause 200
```

```
high encendido
```

```
-----3-----
```

```
'En caso de apagón
```

```
'-----
```

```
'-----
```

```
read 5,arranque
```

```
if arranque = 1 then goto veloz1
```

```
read 6,arranque2
```

```
if arranque2 = 1 then goto veloz2
```

```
read 7,arranque3
```

```
if arranque3 = 1 then goto veloz3
```

```
read 8,arranque4
```

```
if arranque4 = 1 then goto veloz4
```

```

read 9,arranque5
if arranque5 = 1 then goto veloz5
read 10,arranque6
if arranque6 = 1 then goto veloz6
read 11,arranque7
if arranque7 = 1 then goto veloz7
-----4-----
high velocidad0 : low velocidad1 : Low velocidad2
high velocidad00 : low velocidad11 : Low velocidad22
high velocidad0_0 : low velocidad1_1 : Low velocidad2_2

'-----
'-----
-----5-----
inicio:

DEFINE CHAR_PACING 1000
-----6-----
IF sensor = 1 then

serin porta.0,N2400,habilitar      'Recibo dato de IR desde la estación

'-----
-----7-----
'Si se habilita, arranca con una velocidad 1

if habilitar = "A" and velocidad2 = 0 THEN
write 6,0
write 7,0
write 8,0

```

```

write 9,0

write 10,0

write 11,0

write 5,1

read 5,arranque

if arranque = 1 then

veloz1:                                     'Salto en caso de apagón
-----8-----

high velocidad1 : low velocidad0 : low velocidad2
high velocidad11 : low velocidad00 : low velocidad22
high velocidad1_1 : low velocidad0_0 : low velocidad2_2
PAUSE 200
endif
endif

endif

'-----
-----9-----
'Toma una velocidad 2

if switch2 = 1 and velocidad1 = 1 then
gosub contar
write 5,0
write 7,0
write 8,0
write 9,0
write 10,0
write 11,0

```





```

high velocidad1 : low velocidad0 : low velocidad2
high velocidad11 : low velocidad00 : low velocidad22
high velocidad1_1 : low velocidad0_0 : low velocidad2_2
pause 200
endif
endif

'-----
-----11-----
'Frena totalmente

if switch1 = 1 and velocidad1 = 1 then
gosub contar
write 5,0
write 6,0
write 7,0
write 9,0
write 10,0
write 11,0
write 8,1
read 8, arranque4

if arranque4 = 1 then

veloz4:                                'Salto en caso de apagón

high velocidad0 : low velocidad1 : low velocidad2
high velocidad00 : low velocidad11 : low velocidad22
high velocidad0_0 : low velocidad1_1 : low velocidad2_2
pause 200

```

```

-----12-----
if siestacion = 1 then GOTO elevacion
endif
endif

goto inicio

'-----
'-----

IF sensor = 1 and velocidad0 = 1 then 'si llega el cargo bus, luego :
-----13-----
elevacion:                ' lazo cerrado de elevación del cargo bus

write 5,0
write 6,0
write 7,0
write 8,0
write 9,0
write 10,0
write 11,1
read 11, arranque7

if arranque7 = 1 then

veloz7:                    'Salto en caso de apagón
-----14-----
high velocidad0           'mantienen el cargobus parado
high velocidad00
high velocidad0_0

DEFINE CHAR_PACING 1000

```

```
serin porta.1,N2400,operario      'Recibo información del operario para
                                   'subir o bajar
```

```
if operario = "E" and motorliup = 1 and motor2iup = 1 then goto
cargabaja
```

```
-----18-----
```

```
if operario = "E" and motorlidown = 1 and motor2idown = 1 then goto
cargasube
```

```
endif
```

```
goto elevacion
```

```
'-----
```

```
-----15-----
```

```
cargabaja:                          'subrutina para bajar la carga
```

```
write 5,0
```

```
write 6,0
```

```
write 7,0
```

```
write 8,0
```

```
write 10,0
```

```
write 11,0
```

```
write 9,1
```

```
read 9, arranque5
```

```
if arranque5 = 1 then
```

```
veloz5:                              'Salto en caso de apagón
```

```
high velocidad0                       'mantienen el cargobus parado
```

```

high velocidad00
high velocidad0_0
-----16-----
high motor1odown : high motor2odown
high motor1ltdown : high motor22odown

if motor1ltdown = 1 and motor2ltdown = 1 then
low motor1odown : low motor2odown
low motor1ltdown : low motor22odown
-----17-----
goto elevacion                'cuando la carga llega abajo regresa
                                'al lazo de elevacion

endif
endif

goto cargabaja

'.....
-----19-----
cargasube:                    'subrutina para subir la carga

write 5,0
write 6,0
write 7,0
write 8,0
write 9,0
write 11,0
write 10,1
read 10, arranque6

if arranque6 = 1 then

```

```

veloz6:                                     'Salto en caso de apagón

high velocidad0                             'mantienen el cargobus parado
high velocidad00
high velocidad0_0
-----20-----
high motor1oup : high motor2oup
high motor1loup : high motor22oup

if motor1iup = 1 and motor2iup = 1 then
low motor1oup : low motor2oup
low motor1loup : low motor22oup
write 10,0
PAUSE 3000
-----21-----
goto inicio                                 'cuando la carga llega nuevamente
                                           'arriba salta a inicio

endif
endif
goto cargasube

endif
-----22-----
contar:
if switch1 = 1 then contar
if switch2 = 1 then contar
pause 200
return

end

```

## **Algoritmo global del programa**

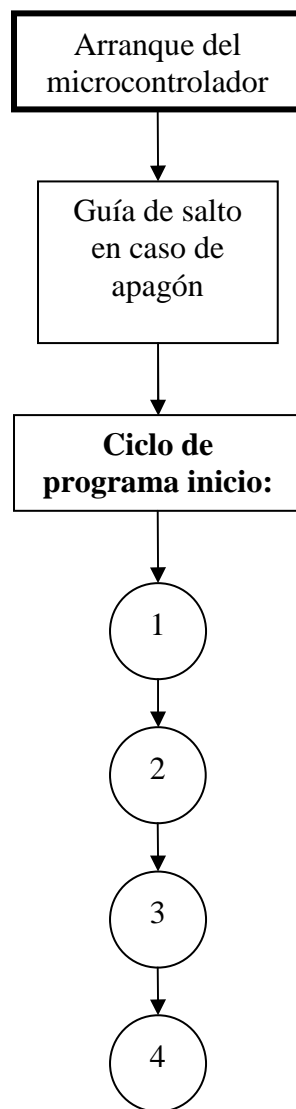
1. Se asigna las variables del programa.
2. Se arranca el sistema. El LED de encendido parpadea dos veces y luego permanece encendido.
3. En caso de ocurrir un apagaron, ésta parte del programa permite continuar con proceso que se estaba realizando exactamente en el mismo lugar donde se encontraba antes de que éste ocurra. Por ejemplo, si el cargobus está descendiendo la carga y ocurre un apagón, permanece en memoria la última acción realizada; al momento de regresar la energía, el cargobus continuará descendiendo la carga hasta que el programa ordene detener la acción.
4. En caso de que el cargobus no se encontrase realizando ningún proceso, se lo mantiene parado.
5. Lazo cerrado de programa.
6. Si se detecta que el cargobus ha llegado a una parada o estación, se abre el puerto de comunicación serial infrarrojo.
7. Si llega el carácter "A", se carga un valor en memoria para que en caso de ocurrir un apagón el programa detecte el punto de retorno.
8. Se activa la velocidad 1 del cargobus, se encienden la salida que activará el motor de traslación en dicha velocidad, representada por la variable "velocidad1", se activa el indicador de velocidad 1 ("velocidad11") y se coloca en uno la variable "velocidad1\_1" que será enviada vía radiofrecuencia al módulo central.

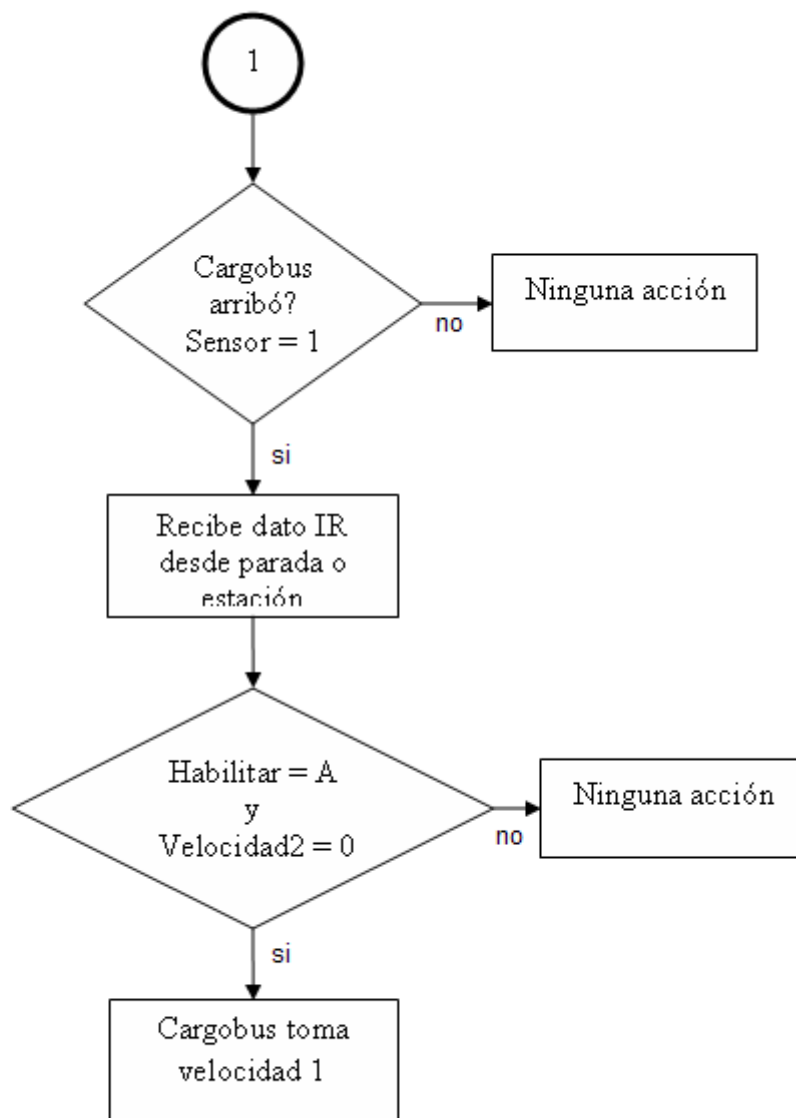
9. Si el microswitch 2 se activa y el cargobus se encuentra trasladándose con velocidad 1, entonces su velocidad cambia a velocidad 2 guardando en memoria el proceso y asignando el estado alto a las variables respectivas.
10. Si el microswitch 2 se activa y el cargobus lleva velocidad 2, luego éste cambia a velocidad 1 nuevamente, guardando en memoria el proceso, y asignando el estado alto a las variables respectivas.
11. Si el microswitch 1 se activa y el cargobus lleva velocidad 1, inmediatamente el cargobus enciende la salida que podrá activar el freno del motor representada por la variable “velocidad0”. Además se activa el indicador de velocidad 0 (“velocidad00”) y se coloca en uno la variable “velocidad0\_0” que será enviada vía radiofrecuencia al módulo central.
12. Luego de que el cargobus se encuentra detenido, “velocidad0” igual a 1, el programa analiza la variable “siestacion” que representa el switch magnético que diferencia una estación de una parada. Si “siestacion” es igual a 1 el programa salta a la etapa de elevación de carga.
13. En el lazo cerrado “elevación”, se guarda primeramente en memoria el punto del programa en caso de apagón.
14. Se mantiene al cargobus parado, y se abre el puerto serial de recepción de datos de operario. Si “operario” es igual a “E” y se detecta que la carga está arriba ( $\text{motor1iup} = 1$ ,  $\text{motor2iup} = 1$ ), se procede a descender la carga, el programa salta a “cargabaja”. Si “operario” es igual a “E” y se detecta que la carga está abajo ( $\text{motor1idown} = 1$ ,  $\text{motor2idown} = 1$ ), se procede a ascender la carga, el programa salta a “cargasube”.
15. En el lazo cerrado cargabaja, primeramente se guarda en memoria el punto del programa en caso de apagón.

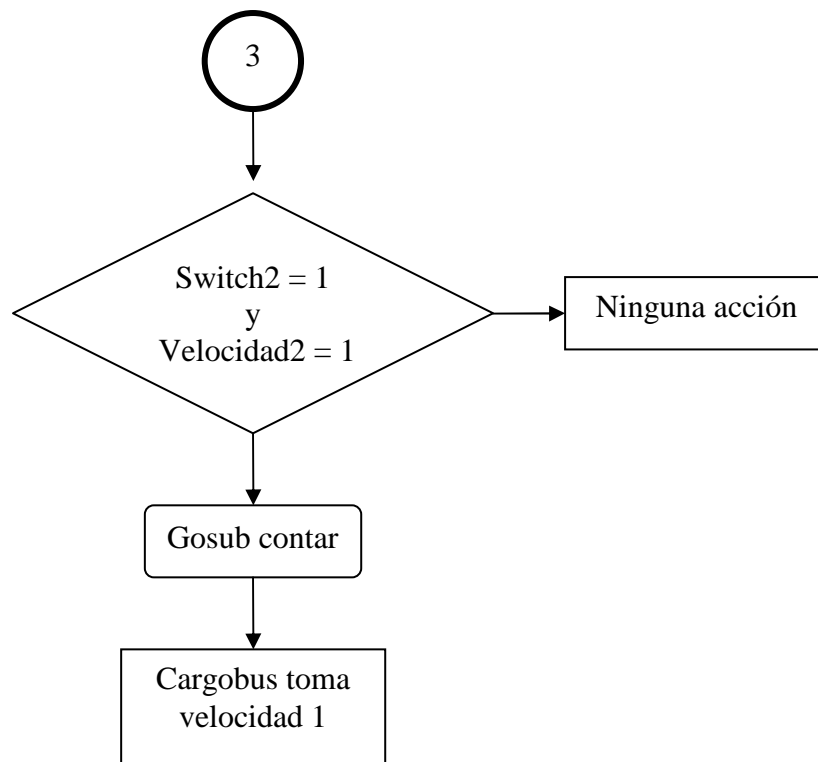
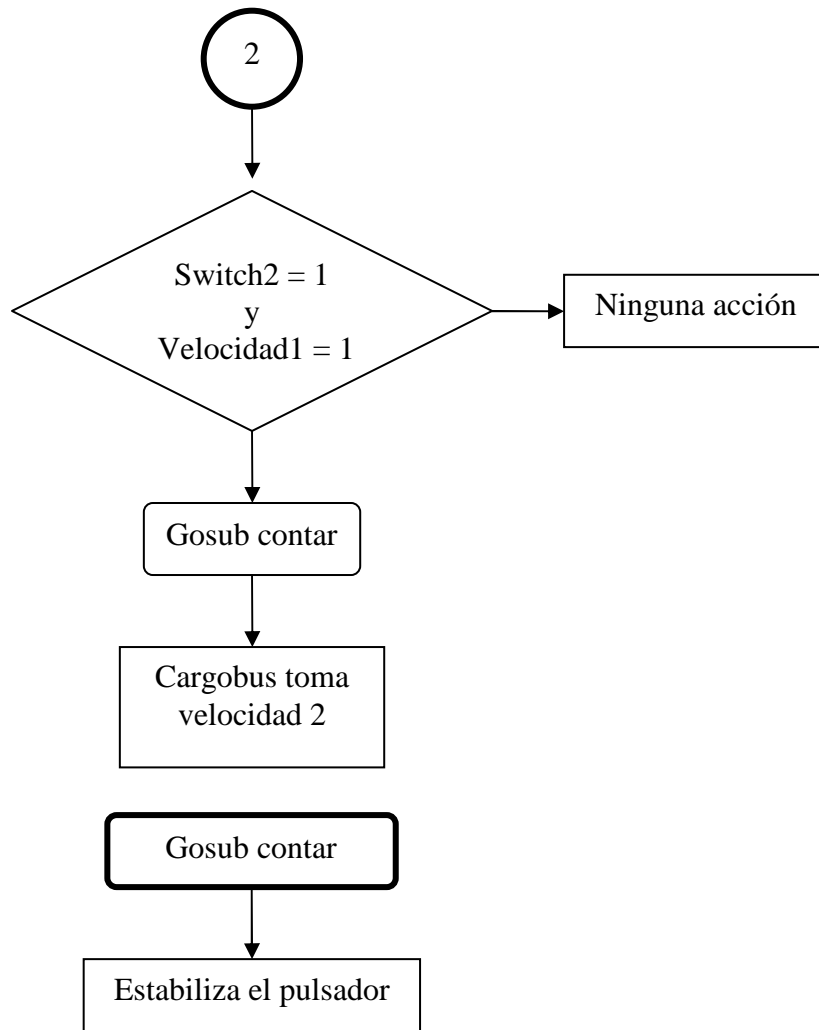
16. Se activan las salidas que permiten que los motores de elevación/descenso bajen la carga (asignadas a las variables "motor1odown" y "motor2odown" ) hasta que los sensores detecten que la carga bajó completamente (asignados a las variables "motor1idown" y "motor2idown").
17. Una vez que la carga descendió, el programa salta nuevamente al lazo cerrado "elevación".
18. Si "operario" es igual a "E" y se detecta que la carga está abajo (motor1idown = 1, motor2idown = 1), se procede a ascender la carga, el programa salta a "cargasube".
19. En el lazo cerrado cargasube, primeramente se guarda en memoria el punto del programa en caso de apagón.
20. Se activan las salidas que permiten que los motores de elevación/descenso eleven la carga (asignadas a las variables "motor1oup" y "motor2oup" ) hasta que los sensores detecten que la carga subió completamente (asignados a las variables "motor1iup" y "motor2iup").
21. Una vez que la carga ascendió completamente, el programa salta nuevamente al lazo cerrado "inicio".
22. Lazo cerrado para evitar rebote.

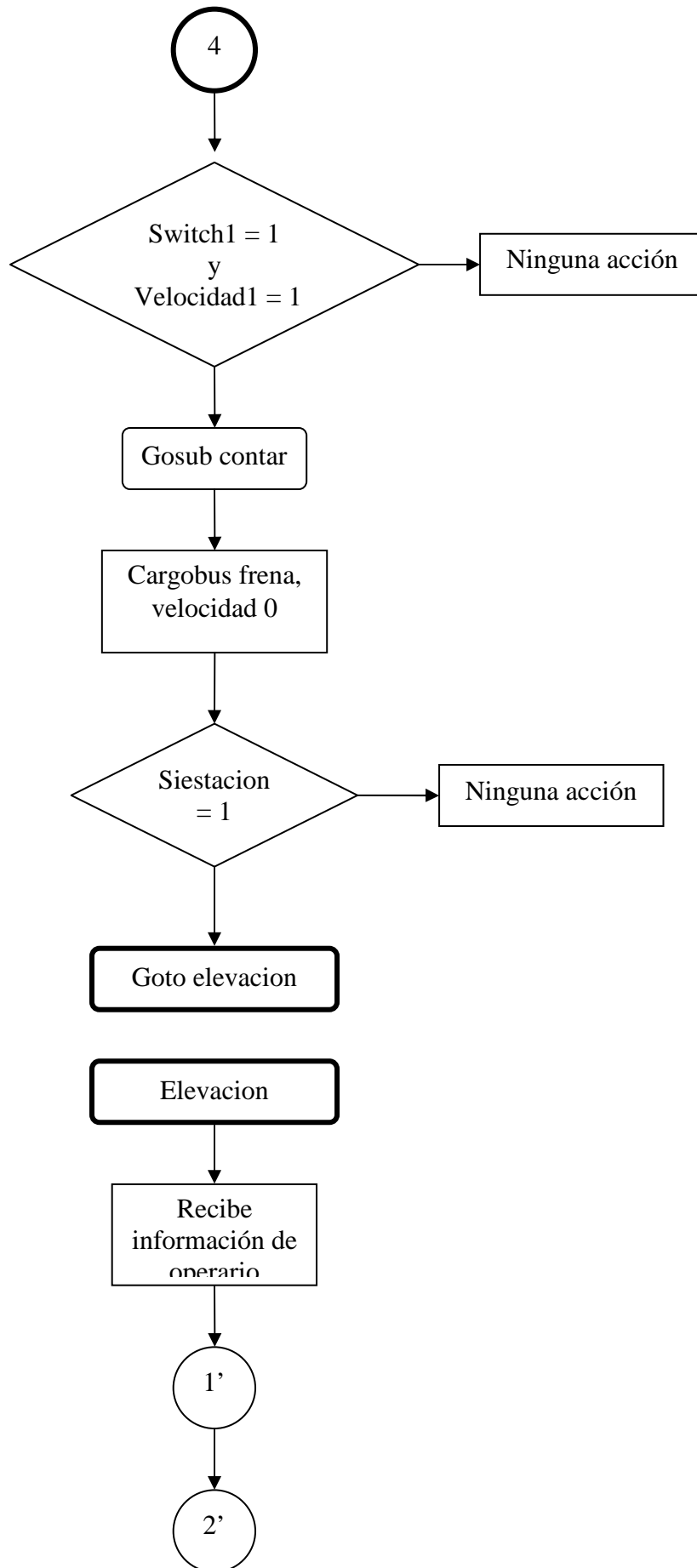


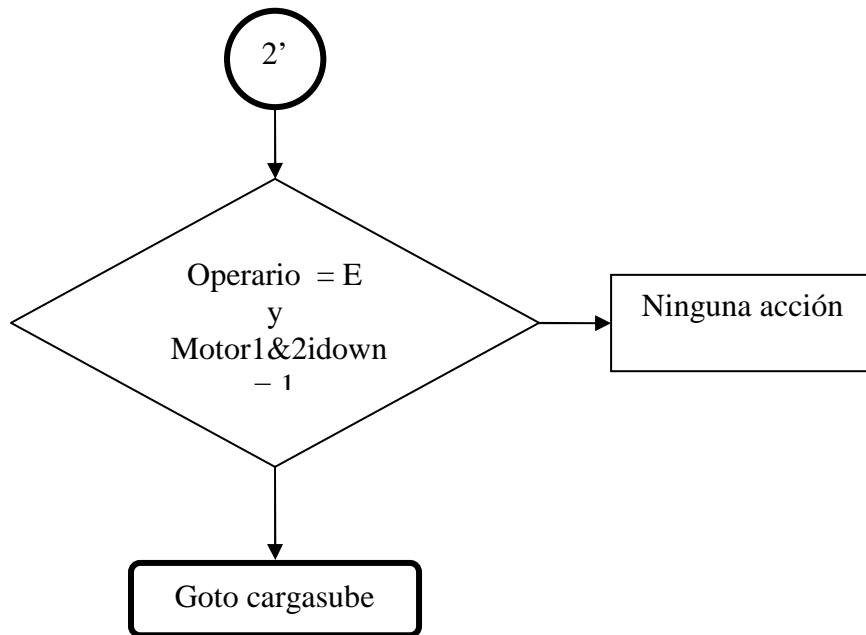
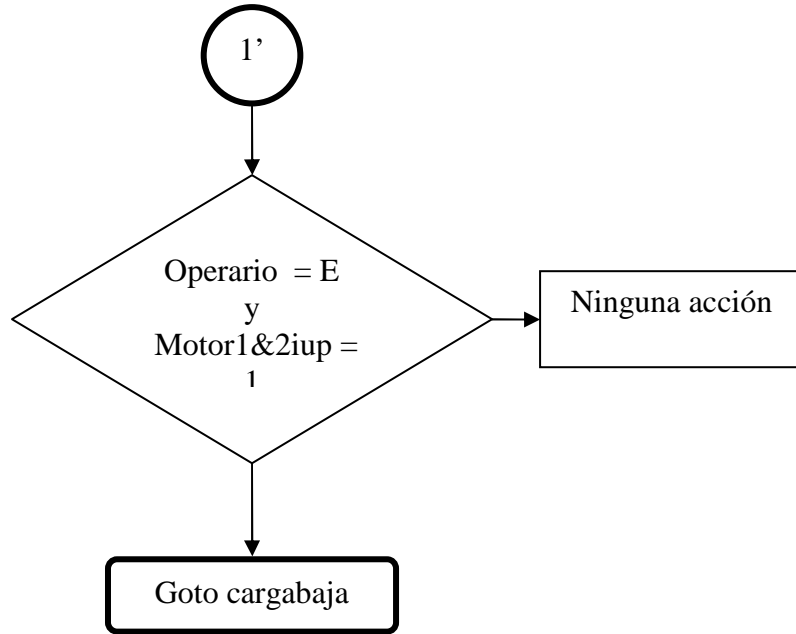
## Diagrama de Flujo

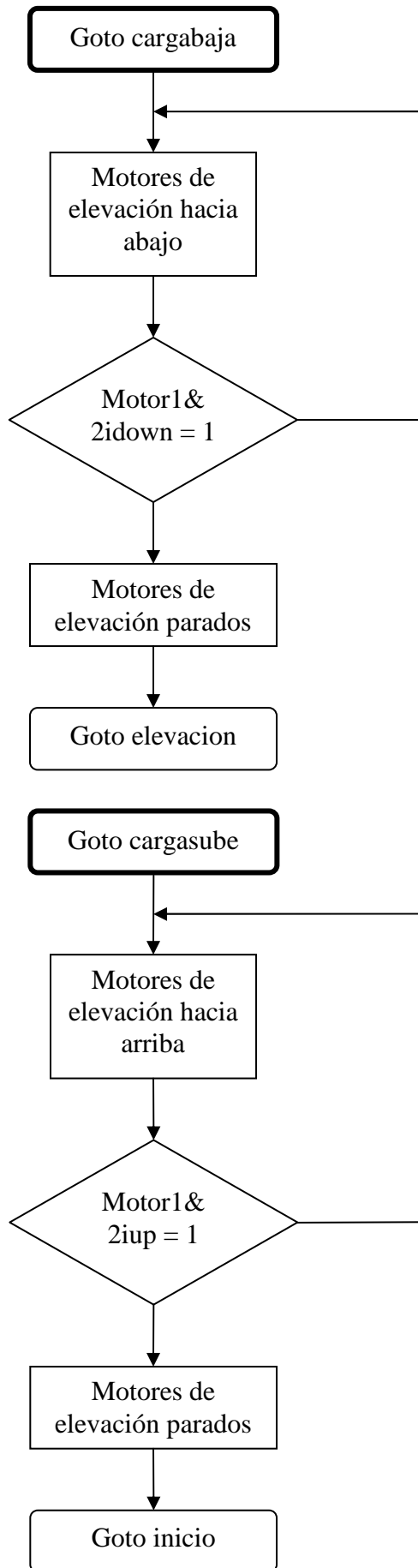












## A.5.2 Programa del PIC12F675 del primer cargobus encargado de intercalar el uso de la frecuencia de transmisión RF.

```
'*****
'* Name      : demorador.bas                *
'* Author    : Andrés Paladines Andrade     *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved          *
'* Date      : 17/07/2008                   *
'* Version   : 1.0                          *
'* Notes     :                               *
'*           :                               *
'*****

include "modedefs.bas"

CMCON = %111

ANSEL = %0000

-----1-----
i var word
-----2-----

inicio:
-----3-----

for i = 1 to 10000
high gpio.2
input gpio.0
output gpio.1
gpio.1 = gpio.0
next i

gosub pausa

goto inicio
```

-----4-----

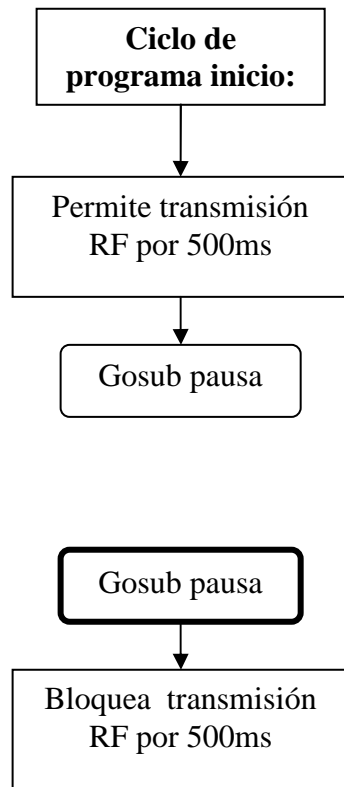
```
pausa:  
low gpio.1  
low gpio.2  
pause 500  
return
```

### **Algoritmo global del programa**

1. Asignación de variables del programa.
2. Ingreso al lazo cerrado de programa.
3. El programa mantiene encendido el pin “gpio.2”, asignado al indicador de transmisión RF (LED rojo instalado cerca al módulo FST-3). Conjuntamente envía la señal RF recibida del pin “gpio.0” por el pin “gpio.1”. Esto se da por un periodo de tiempo determinado por el lazo FOR equivalente a medio segundo.  
  
Luego que el ciclo de repetición termina, el programa salta a la subrutina pausa.
4. La subrutina pausa apaga el indicador de transmisión RF y desconecta la transmisión colocando en nivel bajo el pin “gpio1”. Esto ocurre por un lapso de medio segundo.



## Diagrama de Flujo



### A.5.3 Programa del PIC12F675 del segundo cargobus encargado de intercalar el uso de la frecuencia de transmisión RF.

```
'*****
'* Name      : Demorador2.bas          *
'* Author    : Andrés Paladines Andrade *
'* Notice    : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved     *
'* Date      : 17/07/2008              *
'* Version   : 1.0                     *
'* Notes     :                          *
'*           :                          *
'*****

include "modedefs.bas"

CMCON = %111

ANSEL = %0000

-----1-----
i var word

-----2-----

inicio:

-----3-----

gosub pausa

-----4-----

for i = 1 to 10000

high gpio.2

input gpio.0

output gpio.1

gpio.1 = gpio.0

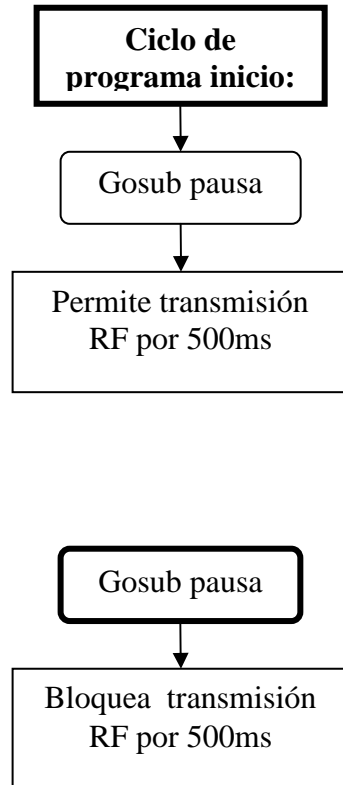
next i
```

```
goto inicio
-----5-----
pausa:
low gpio.1
low gpio.2
pause 500
return
```

### **Algoritmo global del programa**

1. Asignación de variables del programa.
2. Ingreso al lazo cerrado de programa.
3. El programa salta en primer lugar a la subrutina pausa para mantener desconectada la transmisión RF mientras el otro cargobus transmite.
4. El programa mantiene encendido el pin “gpio.2”, asignado al indicador de transmisión RF (LED rojo instalado cerca al módulo FST-3). Conjuntamente envía la señal RF recibida del pin “gpio.0” por el pin “gpio.1”. Esto se da por un periodo de tiempo determinado por el lazo FOR equivalente a medio segundo.
5. La subrutina pausa apaga el indicador de transmisión RF y desconecta la transmisión colocando en nivel bajo el pin “gpio1”. Esto ocurre por un lapso de medio segundo.

## Diagrama de Flujo



## **ANEXO B Hojas de datos**

# PIC16F87XA

## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

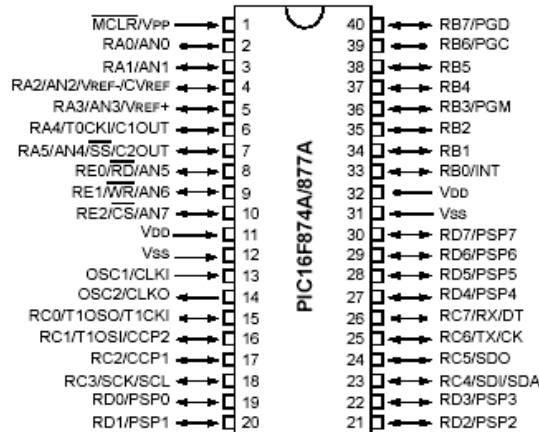
The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

### 40-Pin PDIP



# PIC16F627A/628A/648A

## 1.0 GENERAL DESCRIPTION

The PIC16F627A/628A/648A are 18-pin Flash-based members of the versatile PIC16F627A/628A/648A family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC<sup>®</sup> microcontrollers employ an advanced RISC architecture. The PIC16F627A/628A/648A have enhanced core features, an eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single-cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available, complemented by a large register set.

PIC16F627A/628A/648A microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC16F627A/628A/648A devices have integrated features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption.

The PIC16F627A/628A/648A has 8 oscillator configurations. The single-pin RC oscillator provides a low-cost solution. The LP oscillator minimizes power consumption, XT is a standard crystal, and INTOSC is a self-contained precision two-speed internal oscillator.

The HS mode is for High-Speed crystals. The EC mode is for an external clock source.

The Sleep (Power-down) mode offers power savings. Users can wake-up the chip from Sleep through several external interrupts, internal interrupts and Resets.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

Table 1-1 shows the features of the PIC16F627A/628A/648A mid-range microcontroller family.

A simplified block diagram of the PIC16F627A/628A/648A is shown in Figure 3-1.

The PIC16F627A/628A/648A series fits in applications ranging from battery chargers to low power remote sensors. The Flash technology makes customizing application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages makes this microcontroller series ideal for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16F627A/628A/648A very versatile.

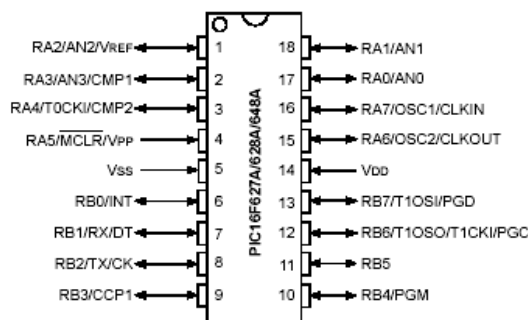
### 1.1 Development Support

The PIC16F627A/628A/648A family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low cost in-circuit debugger, a low cost development programmer and a full-featured programmer. A Third Party "C" compiler support tool is also available.

TABLE 1-1: PIC16F627A/628A/648A FAMILY OF DEVICES

		PIC16F627A	PIC16F628A	PIC16F648A	PIC16LF627A	PIC16LF628A	PIC16LF648A
Clock	Maximum Frequency of Operation (MHz)	20	20	20	20	20	20
	Flash Program Memory (words)	1024	2048	4096	1024	2048	4096
Memory	RAM Data Memory (bytes)	224	224	256	224	224	256
	EEPROM Data Memory (bytes)	128	128	256	128	128	256
	Timer module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
Peripherals	Comparator(s)	2	2	2	2	2	2
	Capture/Compare/PWM modules	1	1	1	1	1	1
	Serial Communications	USART	USART	USART	USART	USART	USART
	Internal Voltage Reference	Yes	Yes	Yes	Yes	Yes	Yes
Features	Interrupt Sources	10	10	10	10	10	10
	I/O Pins	16	16	16	16	16	16
	Voltage Range (Volts)	3.0-5.5	3.0-5.5	3.0-5.5	2.0-5.5	2.0-5.5	2.0-5.5
	Brown-out Reset	Yes	Yes	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN

All PIC<sup>®</sup> family devices have Power-on Reset, selectable Watchdog Timer, selectable code-protect and high I/O current capability. All PIC16F627A/628A/648A family devices use serial programming with clock pin RB6 and data pin RB7.





# PIC12F629/675

## 8-Pin FLASH-Based 8-Bit CMOS Microcontroller

### High Performance RISC CPU:

- Only 35 instructions to learn
  - All single cycle instructions except branches
- Operating speed:
  - DC - 20 MHz oscillator/clock input
  - DC - 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect, and Relative Addressing modes

### Special Microcontroller Features:

- Internal and external oscillator options
  - Precision Internal 4 MHz oscillator factory calibrated to  $\pm 1\%$
  - External Oscillator support for crystals and resonators
  - 5  $\mu$ s wake-up from SLEEP, 3.0V, typical
- Power saving SLEEP mode
- Wide operating voltage range - 2.0V to 5.5V
- Industrial and Extended temperature range
- Low power Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Detect (BOD)
- Watchdog Timer (WDT) with independent oscillator for reliable operation
- Multiplexed MCLR/Input-pin
- Interrupt-on-pin change
- Individual programmable weak pull-ups
- Programmable code protection
- High Endurance FLASH/EEPROM Cell
  - 100,000 write FLASH endurance
  - 1,000,000 write EEPROM endurance
  - FLASH/Data EEPROM Retention: > 40 years

### Low Power Features:

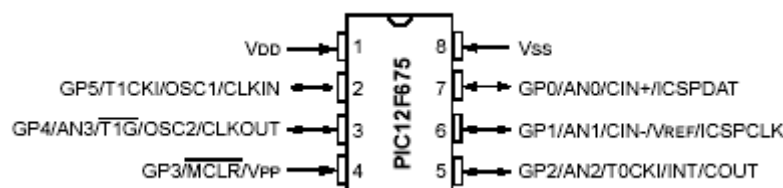
- Standby Current:
  - 1 nA @ 2.0V, typical
- Operating Current:
  - 8.5  $\mu$ A @ 32 kHz, 2.0V, typical
  - 100  $\mu$ A @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 300 nA @ 2.0V, typical
- Timer1 oscillator current:
  - 4  $\mu$ A @ 32 kHz, 2.0V, typical

### Peripheral Features:

- 6 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
  - One analog comparator
  - Programmable on-chip comparator voltage reference (CVREF) module
  - Programmable input multiplexing from device inputs
  - Comparator output is externally accessible
- Analog-to-Digital Converter module (PIC12F675):
  - 10-bit resolution
  - Programmable 4-channel input
  - Voltage reference input
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator, if INTOSC mode selected
- In-Circuit Serial Programming™ (ICSP™) via two pins

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	FLASH (words)	SRAM (bytes)	EEPROM (bytes)				
PIC12F629	1024	64	128	6	-	1	1/1
PIC12F675	1024	64	128	6	4	1	1/1

\* 8-bit, 8-pin devices protected by Microchip's Low Pin Count Patent: U.S. Patent No. 5,847,450. Additional U.S. and foreign patents and applications may be issued or pending.



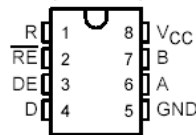


# SN65176B, SN75176B DIFFERENTIAL BUS TRANSCEIVERS

SLLS101D – JULY 1985 – REVISED APRIL 2003

- Bidirectional Transceivers
- Meet or Exceed the Requirements of ANSI Standards TIA/EIA-422-B and TIA/EIA-485-A and ITU Recommendations V.11 and X.27
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability . . .  $\pm 60$  mA Max
- Thermal Shutdown Protection
- Driver Positive and Negative Current Limiting
- Receiver Input Impedance . . .  $12\text{ k}\Omega$  Min
- Receiver Input Sensitivity . . .  $\pm 200$  mV
- Receiver Input Hysteresis . . .  $50$  mV Typ
- Operate From Single 5-V Supply

SN65176B . . . D OR P PACKAGE  
SN75176B . . . D, P, OR PS PACKAGE  
(TOP VIEW)



## description/ordering information

The SN65176B and SN75176B differential bus transceivers are integrated circuits designed for bidirectional data communication on multipoint bus transmission lines. They are designed for balanced transmission lines and meet ANSI Standards TIA/EIA-422-B and TIA/EIA-485-A and ITU Recommendations V.11 and X.27.

The SN65176B and SN75176B combine a 3-state differential line driver and a differential input line receiver, both of which operate from a single 5-V power supply. The driver and receiver have active-high and active-low enables, respectively, that can be connected together externally to function as a direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O) bus ports that are designed to offer minimum loading to the bus when the driver is disabled or  $V_{CC} = 0$ . These ports feature wide positive and negative common-mode voltage ranges, making the device suitable for party-line applications.

## ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (P)	Tube of 50	SN75176BP	SN75176BP
	SOIC (D)	Tube of 75	SN75176BD	75176B
		Reel of 2500	SN75176BDR	
	SOP (PS)	Reel of 2000	SN75176BPSR	A176B
-40°C to 105°C	PDIP (P)	Tube of 50	SN65176BP	SN65176BP
	SOIC (D)	Tube of 75	SN65176BD	65176B
		Reel of 2500	SN65176BDR	

Ht12e

## Pin Assignment

standard packing quantities, thermal data, symbolization, and PCB design guidelines are available in the data sheet for each package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2003, Texas Instruments Incorporated  
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

# SN65176B, SN75176B DIFFERENTIAL BUS TRANSCEIVERS

SLLS101D – JULY 1985 – REVISED APRIL 2003

## description/ordering information (continued)

The driver is designed for up to 60 mA of sink or source current. The driver features positive and negative current limiting and thermal shutdown for protection from line-fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150°C. The receiver features a minimum input impedance of 12 kΩ, an input sensitivity of ±200 mV, and a typical input hysteresis of 50 mV.

The SN65176B and SN75176B can be used in transmission-line applications employing the SN75172 and SN75174 quadruple differential line drivers and SN75173 and SN75175 quadruple differential line receivers.

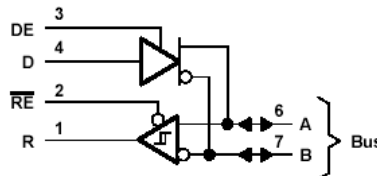
### Function Tables

DRIVER			
INPUT D	ENABLE DE	OUTPUTS	
		A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

RECEIVER		
DIFFERENTIAL INPUTS A-B	ENABLE $\overline{RE}$	OUTPUT R
$V_{ID} \geq 0.2 \text{ V}$	L	H
$-0.2 \text{ V} < V_{ID} < 0.2 \text{ V}$	L	?
$V_{ID} \leq -0.2 \text{ V}$	L	L
X	H	Z
Open	L	?

H = high level, L = low level, ? = indeterminate,  
X = irrelevant, Z = high impedance (off)

### logic diagram (positive logic)



### Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V<sub>DD</sub>=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
  - Four words for the HT12E
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- Pair with Holtek's 2<sup>12</sup> series of decoders
- 18-pin DIP, 20-pin SOP package

### Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

### General Description

The 2<sup>12</sup> encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12-N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header

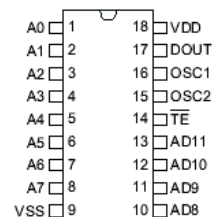
bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a  $\overline{TE}$  trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2<sup>12</sup> series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

### Selection Table

Function Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Carrier Output	Negative Polarity	Package
HT12A	8	0	4	455kHz resonator	D8~D11	38kHz	No	18DIP, 20SOP
HT12E	8	4	0	RC oscillator	$\overline{TE}$	No	No	18DIP, 20SOP

Note: Address/Data represents pins that can be either address or data according to the application requirement.

#### 8-Address 4-Address/Data



**HT12E**  
**-18 DIP-A**

### Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
  - HT12D: 8 address bits and 4 data bits
  - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components
- Pair with Holtek's 2<sup>12</sup> series of encoders
- 18-pin DIP, 20-pin SOP package

### Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

### General Description

The 2<sup>12</sup> decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2<sup>12</sup> series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2<sup>12</sup> series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continu-

ously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2<sup>12</sup> series of decoders are capable of decoding informations that consist of N bits of address and 12~N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

### Selection Table

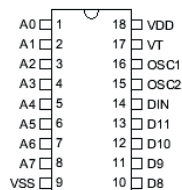
Part No.	Function	Address No.	Data		VT	Oscillator	Trigger	Package
			No.	Type				
HT12D		8	4	L	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP
HT12F		12	0	—	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP

Notes: Data type: L stands for latch type data output.

VT can be used as a momentary data output.

### Pin Assignment

8-Address  
4-Data



HT12D  
– 18 DIP-A

## Bibliografía

- [1] Neoteo.com. Ariel Palazzesi. 07 de junio de 2006. MCUmall Electronics Inc. <<http://www.neoteo.com/microcontroladores.neo>>.
- [2] Reyes, Carlos. Microcontroladores PIC Programación en Basic. Quito: RISPERGRAF, 2006, pp. 17 – 19.
- [3] Reyes, Carlos. Microcontroladores PIC Programación en Basic. Quito: RISPERGRAF, 2006, pp. 23.
- [4] Arqhys.com. 2004. Alexa. <http://www.arqhys.com/arquitectura/cable-coaxial.html>.
- [5] Cib.espol.edu.ec Escuela Superior Politécnica del Litoral.<[www.cib.espol.edu.ec](http://www.cib.espol.edu.ec)>.
- [6] Kioskea.net.Anúncios Google.<<http://es.kioskea.net/contents/transmission/transcabl.php3>>.
- [7] Kioskea.net.Anúncios Google.<<http://es.kioskea.net/contents/transmission/transcabl.php3>>.
- [8] Hispazone.com. Foro. 15 de noviembre del 2002. Alexa. <<http://www.hispazone.com/Articulo/54/Cable-de-par-trenzado.html>>.
- [9] Arqhys.com. 2004. Alexa. <<http://www.arqhys.com/arquitectura/cable-ftp.html>>.
- [10] Es.wikipedia.org. MediaWiki. < [http://es.wikipedia.org/wiki/Cable\\_de\\_fibra\\_%C3%B3ptica#Cables\\_de\\_fibra\\_.C3.B3ptica](http://es.wikipedia.org/wiki/Cable_de_fibra_%C3%B3ptica#Cables_de_fibra_.C3.B3ptica)>.
- [11] Wikipedia.org. 26 de agosto de 2008. MediaWiki. <[http://es.wikipedia.org/wiki/Cable\\_de\\_fibra\\_%C3%B3ptica](http://es.wikipedia.org/wiki/Cable_de_fibra_%C3%B3ptica)>.
- [12] Ni.com. National Instruments Corporation. 2008.<<http://digital.ni.com/worldwide/latam.nsf/web/all/D275FD9CA656670286256F930061491D>>.
- [13] Ni.com. National Instruments Corporation. 2008.<<http://digital.ni.com/worldwide/latam.nsf/web/all/D275FD9CA656670286256F930061491D>>.
- [14] Reyes, Carlos. Microcontroladores PIC Programación en Basic. Quito: RISPERGRAF, 2006, pp. 128 – 129.
- [15] Ni.com. National Instruments Corporation. 2008.<<http://digital.ni.com/worldwide/latam.nsf/web/all/D275FD9CA656670286256F930061491D>>.

- [16] Wikipedia.org. 26 de agosto de 2008. MediaWiki. <[http://es.wikipedia.org/wiki/Puerto\\_serial](http://es.wikipedia.org/wiki/Puerto_serial)>.
- [17] Wut.de. Wiesemann & Theis GMBH. < <http://www.wut.de/e-6www-11-apes-000.php>>.
- [18] Wut.de. Wiesemann & Theis GMBH. < <http://www.wut.de/e-6www-11-apes-000.php>.
- [19] Machay Steave, Wright, Edwin. Industrial Data Networks.Oxford: Newnes, 2004.
- [22] Leuze electronic. Parallel optical data transmisión DLSP 160S,1997, pp. 5.
- [23] Apleiningenieros.com.ApleiningenierosS.A.<<http://www.apleiningenieros.com/viaradio.pdf> >.
- [28] Holtek.com. Holtek Semiconductor Inc. <[http://www.holtek.com/english/docum/consumer/2\\_12d.htm](http://www.holtek.com/english/docum/consumer/2_12d.htm)>.
- [30] Holtek.com. Holtek Semiconductor Inc. <[http://www.holtek.com.tw/english/docum/consumer/2\\_12e.htm](http://www.holtek.com.tw/english/docum/consumer/2_12e.htm)>.
- [31] Szsaw.com.<[http://www.szsaw.com/yw\\_version\\_en/www/xmb/product/AGP\\_147\\_href.html](http://www.szsaw.com/yw_version_en/www/xmb/product/AGP_147_href.html)>.
- [32] Szsaw.com. < [http://www.szsaw.com/yw\\_version\\_en/www/xmb/product/AGP\\_133\\_href.html](http://www.szsaw.com/yw_version_en/www/xmb/product/AGP_133_href.html)>.
- [33] Ni.com. National Instruments Corporation. 2008.< <http://digital.ni.com/public.nsf/allkb/8C5F1FDC45A30155862570E500711955>>.
- [34] 34t.com. 34Telecom SL. 30 de enero de 2002. Axis Mobotix. < <http://www.34t.com/box-news.asp?area=76&suba=04&IDN=263>>.
- [35] Robots-argentina.com.ar. Eduardo J. Carletti. 25 de julio de 2008. <[http://robots-argentina.com.ar/Comunicacion\\_IR.htm](http://robots-argentina.com.ar/Comunicacion_IR.htm)>.
- [37] Chipkin.com. Steven Smethurst. 2007. Chipkin Automation Systems. <<http://www.chipkin.com/articles/what-is-rs485-eia-485>>.
- [38] Microchip.com. 2008. <<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>>.
- [39] Microchip.com. 2008. <<http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>>.
- [40] Microchip.com. 2008. <<http://ww1.microchip.com/downloads/en/DeviceDoc/41190E.pdf>>.

[41] Datasheetcatalog.com. 2008. <[http://www.datasheetcatalog.com/datasheets\\_pdf/S/N/7/5/SN75176B.shtml](http://www.datasheetcatalog.com/datasheets_pdf/S/N/7/5/SN75176B.shtml)>.

[42] Datasheetcatalog.com. 2008. <[http://www.datasheetcatalog.net/es/datasheets\\_pdf/H/T/1/2/HT12E.shtml](http://www.datasheetcatalog.net/es/datasheets_pdf/H/T/1/2/HT12E.shtml)>.

[43] Datasheetcatalog.com. 2008. <[http://www.datasheetcatalog.net/es/datasheets\\_pdf/H/T/1/2/HT12D-18.shtml](http://www.datasheetcatalog.net/es/datasheets_pdf/H/T/1/2/HT12D-18.shtml)>.

Upcommons.upc.edu. Gerrit Faerber, Anna Maria Coves Moreno. Junio de 2004. Universitat Politecnica de Catalunya. <<https://upcommons.upc.edu/eprints/bitstream/2117/573/1/IOC-DT-P-2004-10.pdf>>.