UNIVERSIDAD MAYOR DE SAN ANDRES

FACULTAD DE TECNOLOGIA

CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

TRABAJO DE APLICACION

“DISEÑO DE UN PROTOTIPO DE SISTEMA DE CONTROL DE POSICIONAMIENTO Y CONTROL DE TIEMPOS PARA UN ESTACIONAMIENTO VERTICAL EN LA CIUDAD DE LA PAZ”

Examen de Grado presentado para obtener el Grado de Licenciado en Electrónica y Telecomunicaciones

POSTULANTE: Marco Iván Quispe Yavi

La Paz – Bolivia

Noviembre, 2017

RESUMEN

Para la realización del presente proyecto se observó que en las ciudades céntricas de nuestro país la falta de estacionamientos es la causa de problemas como congestionamiento del tráfico, robo de vehículos y otros perjudicando tanto al sector del transporte como a los transeúntes. El objetivo de este proyecto es el de diseñar un prototipo de sistema de control de posicionamiento y control de tiempospara un estacionamiento vertical; este tipo de estacionamientos nos ofrece las ventajas de ocupar menos espacio que un estacionamiento de suelo o uno subterráneo, el costo del estacionamiento resulta más económico por contar con mayor capacidad por espacio utilizado con lo cual se lograría un retorno de la inversión en menor tiempo.

Para el diseño del prototipo se empleó como elemento principal al microcontrolador de gama media PIC16F877A el cual está encargado de controlar el posicionamiento del vehículo en un determinado espacio del estacionamiento mediante motores que se encargan del desplazamiento vertical y horizontal y también de la bandeja transportadora;para el control de tiempos y costo por el uso del estacionamiento se volvió a emplear un microcontrolador idéntico acompañado de otro microcontrolador también de gama media PIC16F628A logrando visualizar ambos valores en un display LCD. El lenguaje de programación empleado para este proyecto es mikroBasic.

En países como Alemania la empresa Volkswagen implementó un estacionamiento vertical con una capacidad para 400 vehículos, así también lo hizo Samsung en el continente asiático y otras empresas en el continente Latinoamericano, llegando a la conclusión que este tipo de estacionamiento nos brindaría muchas ventajas a las ciudades céntricas de nuestro país.

INDICE

CAPITULO 1

1.1 PLANTEAMIENTO DEL PROBLEMA

Para la realización del presente proyecto seobservó que el estacionamiento temporal de vehículos en las calles céntricas de las ciudades de nuestro país es un problema que complica tanto al sector del transporte como también a los transeúntes debido a que debemos cumplir con las tareas u obligaciones cotidianas como visitar oficinas públicas, banca, comercio, salud u otros servicios. Viendo que no existen los parqueos suficientes para todos los motorizados que transitan por sectores céntricos de la cuidad, se plantea como solución a este problema, la implementación de un estacionamiento de forma vertical con el fin de tener una mayor capacidad y optimizar el espacio ocupado para un parqueo.

A causa de la falta de parqueos muchos proceden estacionar los vehículos en las aceras poniendo en riesgo la seguridad de los peatones y dando paso al robo de vehículos y autopartes, también se ocasiona la lenta circulación de las movilidades lo que provoca una congestión de tráfico en horas pico. Habiendo una prohibición de estacionar en los espacios públicos por parte de la alcaldía los parqueos incrementaron el precio del servicio cobrando entre 4 a 5 bolivianos por hora en lugares céntricos y entre 3 a 4 bolivianos por las laderas de la ciudad.

Las avenidas y hasta las estrechas calles de la ciudad de La Paz se encuentran con vehículos estacionados, las autoridades deberían preocuparse por habilitar espacios apropiados para esta necesidad de dejar una movilidad por tiempo limitado, sin que policías de tránsito o funcionarios municipales apliquen boletas de multa o coloquen inmovilizadores cuyo retiro cuesta dinero. La alcaldía de La Paz inició el uso de parqueos ediles ubicados en 19 víasde la zona de Sopocachi, donde el costo mínimo es de tres bolivianos por media hora, siendo un costo elevado.

Se pretende hacer la simulación mediante un prototipo de estacionamiento vertical, para que sea una alternativa de solución al problema de congestionamiento por falta de parqueos en la ciudad de La Paz.

1.2 JUSTIFICACION DEL TRABAJO

Con la implementación de estacionamientos de forma vertical se beneficiarían los usuarios de transporte privado, evitando que den vueltas por la ciudad en busca de parqueos, transporte público por la ayuda en la descongestión de las vías y transeúntes en general.

Para el presente proyecto se emplean conocimientos obtenidos en la carrera de Electrónica y Telecomunicaciones como microprocesadores, laboratorio de microprocesadores, electrónica industrial, electrónica digital.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un prototipo de sistema de controlde posicionamiento y control de tiempospara un estacionamiento vertical en la ciudad de La Paz, mediante el uso de microcontroladores.

1.3.2 OBJETIVOS ESPECIFICOS

1.3.3 Diseñar un prototipo de sistema de control para el posicionamiento de un vehículo en un determinado espacio del estacionamiento vertical.

1.3.4 Diseñar un prototipo de sistema de control de tiempo y cálculo de costo por el uso del estacionamiento vertical.

1.3.5 Probar el funcionamiento ambos sistemas de control, empleando una maqueta de estacionamiento vertical.

CAPITULO 2

2.1 FUNDAMENTACION TEORICA

2.2 LOS MICROCONTROLADORES

La situación actual en el campo de los microcontroladores se ha producido gracias al desarrollo de la tecnología de fabricación de los circuitos integrados. Este desarrollo ha permitido construir las centenas de miles de transistores en un chip. Esto fue una condición previa para la fabricación de un microprocesador. Las primeras microcomputadoras se fabricaron al añadirles periféricos externos, tales como memoria, líneas de entrada/salida, temporizadores u otros. El incremento posterior de la densidad de integración permitió crear un circuito integrado que contenía tanto al procesador como periféricos. Así es cómo fue desarrollada la primera microcomputadora en un solo chip, denominada más tarde microcontrolador.

Los principiantes en electrónica creen que un microcontrolador es igual a un microprocesador. Esto no es cierto. Difieren uno del otro en muchos sentidos. La primera y la más importante diferencia es su funcionalidad. Para utilizar al microprocesador en una aplicación real, se debe de conectar con otros componentes, en primer lugar con la memoria. Aunque el microprocesador se considera una máquina de computación poderosa, no está preparado para la comunicación con los dispositivos periféricos que se le conectan. Para que el microprocesador se comunique con algún periférico, se deben utilizar los circuitos especiales. Así era en el principio y esta práctica sigue vigente en la actualidad.

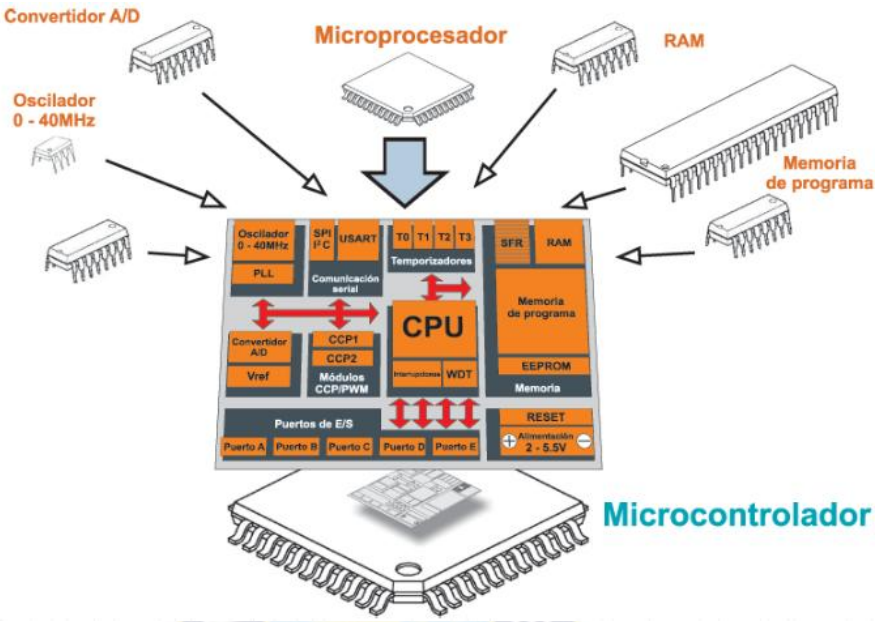


FIGURA 1 Componentes de un microcontrolador

FUENTE: Milan Verle, 2010, PIC Microcontrollers – Programming in Basic

Por otro lado, al microcontrolador se le diseña de tal manera que tenga todas las componentes integradas en el mismo chip. No necesita de otros componentes especializados para su aplicación, porque todos los circuitos necesarios, que de otra manera correspondan a los periféricos, ya se encuentran incorporados. Así se ahorra tiempo y espacio necesario para construir un dispositivo.

Los microcontroladores PIC.- El nombre verdadero de este microcontrolador es PICmicro - controlador de interfaz periférico (Peripheral Interface Controller), conocido bajo el nombre PIC. Su primer antecesor fue creado en 1975 por la compañía General Instruments. Este chip denominado PIC1650 fue diseñado para propósitos completamente diferentes. Aproximadamente diez años más tarde, al añadir una memoria EEPROM, este circuito seconvirtió en un verdadero microcontrolador PIC.

Todos los microcontroladores PIC utilizan una arquitectura Harvard, lo que quiere decir que su memoria de programa está conectada a la CPU por más de 8 líneas. Hay microcontroladores de 12, 14 y 16 bits, dependiendo de la anchura del bus.

2.2.1 MICROCONTOLADOR PIC 16F877A

El PIC 16F877A es un producto conocido de la compañía Microchip. Dispone de todos los componentes disponibles en la mayoría de los microcontroladores modernos. Por su bajo precio, un rango amplio de aplicaciones, alta calidad y disponibilidad, es una solución perfecta aplicarlo para controlar diferentes procesos en la industria, en dispositivos de control de máquinas, para medir variables de procesos etc. Algunas de sus características principales se enumeran a continuación.

- Arquitectura RISC.-El microcontrolador cuenta con solo 35 instrucciones diferentes. Todas las instrucciones son uni-ciclo excepto por las de ramificación
- Frecuencia de operación 0-20 MHz
- Oscilador interno de alta precisión.-Calibrado de fábrica. Rango de frecuencia de 8MHz a 31KHz seleccionado por software.
- Voltaje de la fuente de alimentación de 2.0V a 5.5V.-Consumo: 220uA (2.0V, 4MHz), 11uA (2.0 V, 32 KHz), 50nA (en modo de espera)
- Ahorro de energía en el Modo de reposo.
- Brown-out Reset (BOR) con opción para controlar por software.
- 35 pines de entrada/salida.-Alta corriente de fuente y de drenador para manejo de LED. Resistencias pull-up programablesindividualmente por software. Interrupción al cambiar el estado del pin.
- Memoria ROM de 8K con tecnología FLASH.-El chip se puede re-programar hasta 100.000 veces.
- Opción de programación serial en el circuito.-El chip se puede programar incluso incorporado en el circuito.
- 256 bytes de memoria EEPROM.- Los datos se pueden grabar más de 1.000.000 veces.
- 368 bytes de memoria RAM.
- Convertidor A/D.-14 canales con una resolución de 10 bits.
- 3 temporizadores/contadores independientes.
- Temporizador perro guardián.

- Módulo comparador analógico.-Dos comparadores analógicos, referencia de voltaje fija (0.6V), referencia de voltaje programable en el chip.
- Módulo PWM incorporado.
- Módulo USART mejorado.-Soporta las comunicaciones seriales RS-485, RS-232 y LIN2.0, auto detección de baudios.
- Puerto Serie Síncrono Maestro (MSSP).-Soporta los modos SPI e I2C

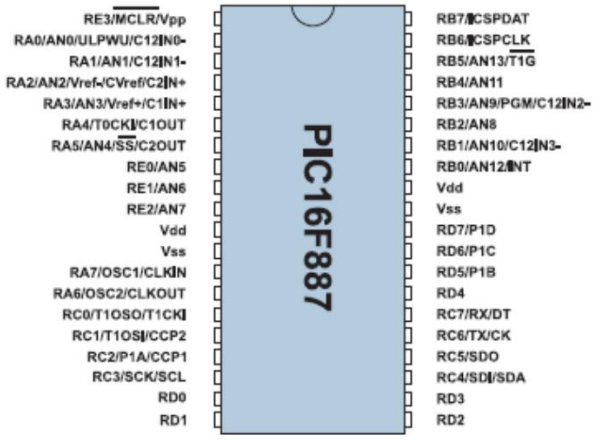


FIGURA 2 Disposición de pines en el encapsulado DIP40 del P16F877

FUENTE: Milan Verle, 2010, PIC Microcontrollers – Programming in Basic

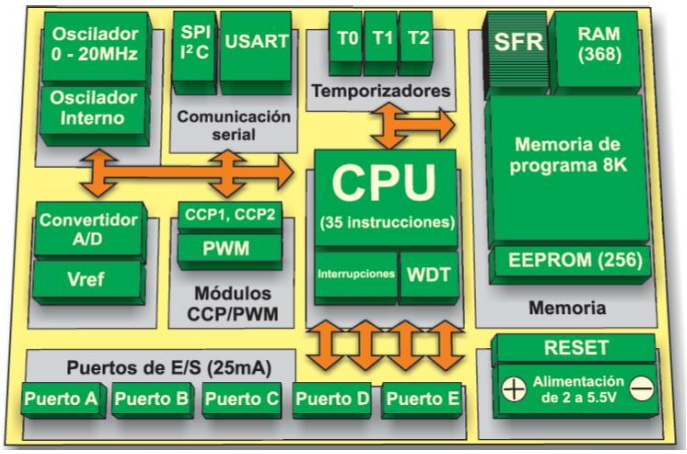


FIGURA 3 Esquema de módulos del P16F877

FUENTE: Milan Verle, 2010, PIC Microcontrollers – Programming in Basic

Registros de Funciones Especiales(SFR).-Los registros de funciones especiales son también parte de la memoria RAM. A diferencia de los registros de propósito general, su propósito es predeterminado durante el proceso de fabricación y no se pueden cambiar. Como los bits están conectados a los circuitos particulares en el chip (convertidor A/D, módulo de comunicación serial, etc.), cualquier cambio de su contenido afecta directamente al funcionamiento del microcontrolador o de alguno de sus módulos. Por ejemplo, el registro ADCON0 controla el funcionamiento del convertidor A/D. Al cambiar los bits se determina qué pin del puerto se configurará como la entrada del convertidor, el momento del inicio de la conversión así como la velocidad de la conversión.

Otra característica de estas localidades de memoria es que tienen nombres (tanto los registros como sus bits), lo que simplifica considerablemente el proceso de escribir un programa. Como el lenguaje de programación de alto nivel puede utilizar la lista de todos los registros con sus direcciones exactas, basta con especificar el nombre de registro para leer o cambiar su contenido.

La memoria RAM está dividida en cuatro bancos. Antes de acceder a un registro al escribir un programa (para leer o cambiar su contenido), es necesario seleccionar el banco que contiene ese registro. Más tarde vamos a tratar dos bits del registro STATUS utilizados para selección del banco. Para simplificar el funcionamiento, los SFR utilizados con más frecuencia tienen la misma dirección en todos los bancos, lo que permite accederlos con facilidad.

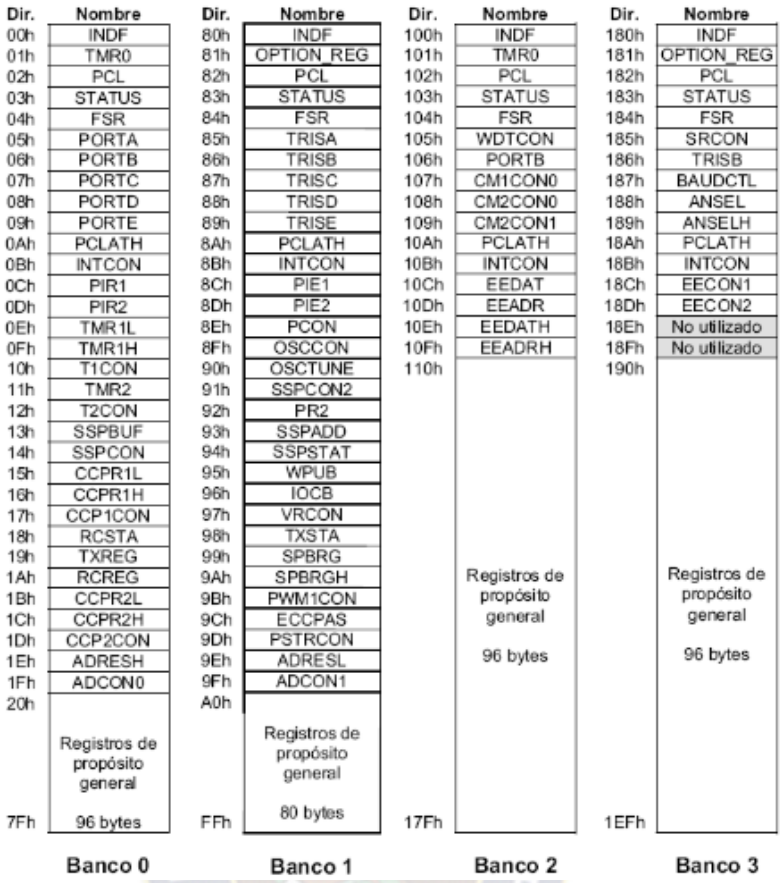


FIGURA 4 Bancos de la memoria RAM

FUENTE: Milan Verle, 2010, PIC Microcontrollers – Programming in Basic

Sistemas de Interrupciones.-Al aparecer una petición de interrupción lo primero que hace el microcontrolador es ejecutar la instrucción actual después de que se detiene el proceso de ejecución de programa. Como resultado, la dirección de memoria de programa actual se apila automáticamente y la dirección por defecto (predefinida por el fabricante) se escribe en el contador de programa. La localidad en la que el programa continúa con la ejecución se le denomina vector de interrupción. En el caso del microcontrolador PIC16F887 esta dirección es 0x0004h. La localidad que contiene el vector de interrupción se omite durante la ejecución de programa regular.

Una parte de programa que se ejecutará al hacer una petición de interrupción se le denomina rutina de interrupción. Su primera instrucción se encuentra en el vector de

interrupción. Cuánto tiempo tardará en ejecutar esta subrutina y cómo será depende de la destreza del programador así como de la fuente de interrupción. Algunos microcontroladores tienen más de un vector de interrupción (cada petición de interrupción tiene su vector), pero en este caso sólo hay uno. En consecuencia, la primera parte da la rutina de interrupción consiste en detectar la fuente de interrupción.Por fin, al reconocer la fuente de interrupción y al terminar de ejecutar la rutina de interrupción el microcontrolador alcanza la instrucción RETFIE, toma la dirección de la pila y continúa con la ejecución de programa desde donde se interrumpió.

El microcontrolador PIC16F877A cuenta con 14 fuentes de interrupción. Para el presente proyecto se emplea la fuente de interrupción por cambio de nivel en los pines del puerto B (RB4 al RB7), para tal efecto se debe ajustar los bits del registro INTCON

Registro INTCON

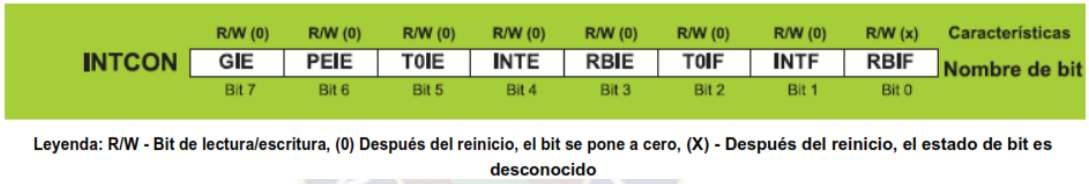


Figura 5 Bits del registro INTCON

FUENTE:Milan Verle, 2010, PIC Microcontrollers – Programming in Basic

- **GIE - Global InterruptEnablebit** (bit de habilitación de interrupciones globales) controla simultáneamente todas las fuentes de interrupciones posibles.
  1 - Habilita las interrupciones no enmascaradas.
  0 - Deshabilita las interrupciones no enmascaradas.
- **PEIE - PeripheralInterruptEnable bit** (bit de habilitación de interrupciones periféricas) es similar al bit GIE, sin embargo controla interrupciones habilitadas por los periféricos. Eso significa que no influye en interrupciones causadas por el temporizador Timer0 o por el cambio del estado en el puerto PORTB o por el cambio en el pin RB0/INT.
  1 -Habilita las interrupciones periféricas no enmascaradas.

0 -Deshabilita las interrupciones periféricas no enmascaradas

- **T0IE - TMR0 OverflowInterruptEnable bit** (bit de habilitación de interrupciones por el desbordamiento del registro Timer0) controla interrupciones causadas por el desbordamiento del Timer0.

  1 - Habilita interrupciones por Timer0

  0 - Deshabilita interrupciones por Timer0

- **INTE - RB0/INT ExternalInterruptEnable bit** (bit de habilitación de la interrupción externa en RB0) controla interrupciones causadas por el cambio del estado lógico en el pin de entrada RB0/INT (interrupción externa).

  1 - Habilita interrupciones externas INT

  0 - Deshabilita interrupciones externas INT

- **RBIE - RB Port ChangeInterruptEnable bit** (bit de habilitación de interrupciones por cambios en el puerto PORTB). Cuando se configuran como entradas, los pines en el puerto PORTB pueden causar una interrupción al cambiar el estado lógico (no importa si se produce bajada o subida de tensión, lo que importa es que se produce un cambio). Este bit determina si una interrupción va a ocurrir.

  1 - Habilita interrupciones por cambio en el puerto PORTB

  0 - Deshabilita interrupciones por cambio en el puerto PORTB

- **T0IF - TMR0 OverflowInterruptFlag bit** (bit de bandera de interrupción por el desbordamiento del Timer0) detecta el desbordamiento en el registro del temporizador Timer0, o sea el contador se pone a cero.

  1 - En el registro TMR0 ha ocurrido desbordamiento (esta bandera debe volverse a 0 por software)

  0 - En el registro TMR0 no ha ocurrido desbordamiento

- **INTF - RB0/INT ExternalInterruptFlag bit** (bit de bandera de interrupción externa en INT) detecta el cambio en el estado lógico en el pin INT.

  1 - Ha ocurrido una interrupción externa por INT (esta bandera debe volverse a 0 por software)

  0 - No ha ocurrido una interrupción externa por INT

- **RBIF – RB Port ChangeInterruptFlagbit**(bit de bandera de interrupción por cambio en el puerto RB) detecta cualquier cambio del estado lógico de alguno de los pines de entrada en el puerto PORTB.

  1 - Al menos uno de los pines de E/S de propósito general en el puerto PORTB ha cambido de valor. Después de leer el puerto PORTB, el bit RBIF debe volverse a 0 por software)

  0 - Ninguno de los pines de E/S de propósito general en el puerto PORTB ha cambiado de valor.

* Los puntos 4.1 MICROCONTROLADORES y 4.1.1 MICROCONTROLADOR PIC 16F877A, son de la fuente: Milan Verle, 2010, PIC Microcontrollers – Programming in Basic

2.2.2 MICROCONTOLADOR PIC 16F628A

El pic16F628A es un microcontrolador de 8 bit, posee una arquitectura RISC avanzada así como un juego reducido de 35 instrucciones. Este microcontrolador es el remplazo del obsoleto pic16F84A, los pines del pic16F628A son compatibles con el pic16F84A, así se podrían actualizar proyectos que hemos utilizado con el pic16F84A.

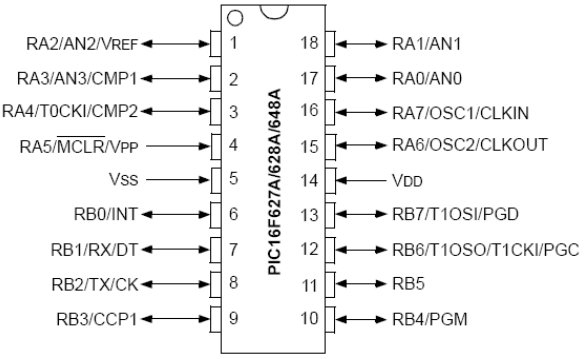


FIGURA 6 Disposición de pines en el encapsulado DIP18 del P16F628A

FUENTE: www.google.com/site/proyectospic2010/PIC18F452/introduccion-pic16f628a-1

Características del PIC16F628A:

- CPU De alto rendimiento RISC.
- Velocidades de operación de DC - 20 MHz.
- Capacidad de interrupción.
- Pila de 8 niveles.
- Modos de direccionamiento directos, indirectos y relativo.
- 35 instrucciones.
- Todas las instrucciones de ciclo único, excepto las de salto.
- Opciones de oscilador externo e interno.
- Precisión de fábrica del oscilador interno de 4 MHz calibrada a ± 1%.
- Oscilador de 48 kHz De bajo consumo interno.
- Modo de ahorro de energía en modo sueño.
- Resistencias programable pul-ups del PORTB.
- Multiplexado del pin reset / Entrada-pin.
- Temporizador Watchdog con oscilador independiente para un funcionamiento fiable.
- Baja tensión de programación In-Circuit Serial (a través de dos pines).
- Protección de código programable.
- Brown-out Reset.
- Power-on Reset.
- Power-up Timer y el oscilador de puesta en marcha del temporizador.
- Amplio rango de funcionamiento de tensión (2.0-5.5V).
- 40 años de retención de datos..
- Corriente en espera: 100 Na a 2.0V, típico.
- Corriente de funcionamiento: 12μA a 32 kHz, 2,0 V, típica;120μA a 1 MHz, 2,0 V, típica.
- Temporizador Watchdog actual: 1μA a 2.0V, típico.
- Timer1 oscilador actual: 1.2μA a 32 kHz, 2,0 V, típica.
- Doble velocidad del oscilador interno.
- Tiempo de ejecución seleccionable entre 4 MHz y de 48 kHz.
- 4μs despertar de un sueño, 3.0V, típico.

Mapa de memoria del pic16F628A

Como se observa en la siguiente figura el mapa de la memoria RAM que se encuentra dividida en 4 bancos en estos bancos se encuentra los registros de control y registros de propósito general.

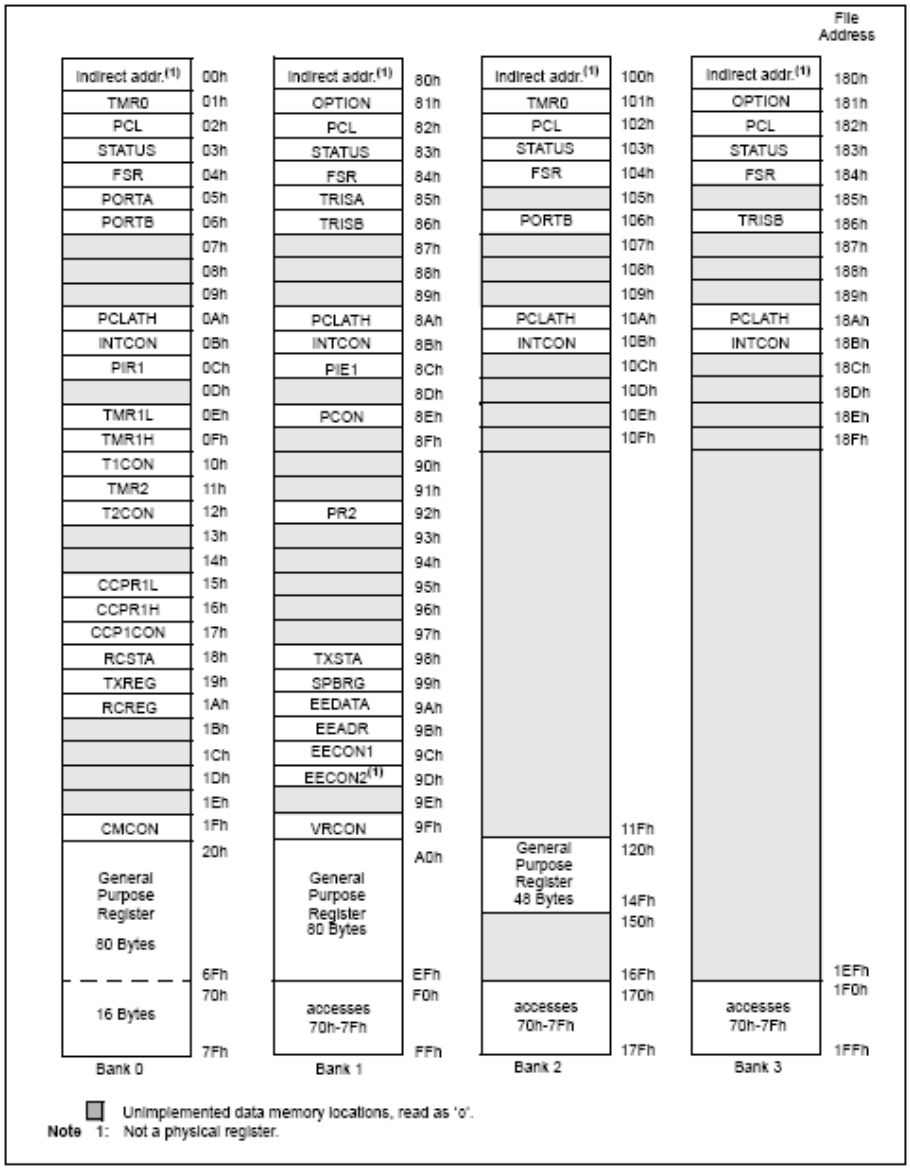


FIGURA 7 Bancos de la memoria RAM

FUENTE: www.google.com/site/proyectospic2010/PIC18F452/introduccion-pic16f628a-1

2.3 PUENTE H

Un Puente en H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avance y retroceso. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.

El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o mediante transistores). Cuando los interruptores S1 y S4 (ver primera figura) están cerrados (y S2 y S3 abiertos) se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 (y cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

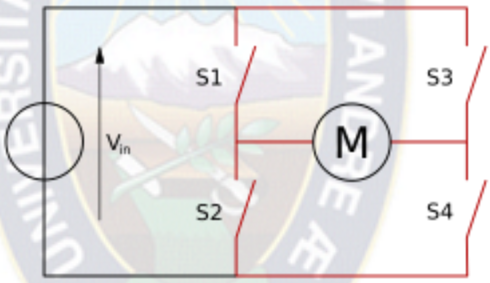


FIGURA 8 Estructura de un puente H

FUENTE: Wikipedia - Puente H

Con la nomenclatura que estamos usando, los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, porque esto cortocircuitaría la fuente de tensión. Lo mismo sucede con S3 y S4.

El puente H se usa para invertir el giro de un motor, pero también puede usarse para frenarlo (de manera brusca), al hacer un corto entre las bornas del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, cuando desconectamos el motor de la fuente que lo alimenta. En el siguiente cuadro se resumen las diferentes acciones.

Circuito Integrado **L293D**

El Circuito Integrado L293D es de gran utilidad para controlar pequeños motores y actuadores de corriente directa. Este circuito es bastante utilizado en robótica para controlar motores a pasos y de corriente directa.

Incluye cuatro circuitos para manejar cargas de potencia media, en especial pequeños motores y cargas inductivas, con la capacidad de controlar corriente hasta 600 mA en cada circuito y una tensión entre 4,5 V a 36 V.

Los circuitos individuales se pueden usar de manera independiente para controlar cargas de todo tipo y, en el caso de ser motores, manejar un único sentido de giro. Pero además, cualquiera de estos cuatro circuitos sirve para configurar la mitad de un puente H.

El integrado permite formar, entonces, dos puentes H completos, con los que se puede realizar el manejo de dos motores. En este caso el manejo será bidireccional, con frenado rápido y con posibilidad de implementar fácilmente el control de velocidad.

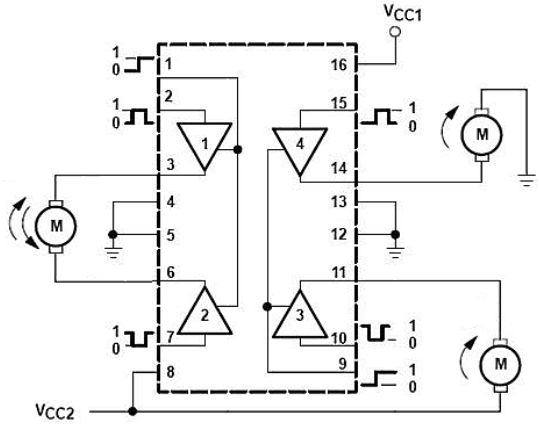


FIGURA 9 Conexionado para un motor con giro en ambos sentidos (lado izquierdo) y con motores con giro en sentido único en dos salidas (lado derecho)

FUENTE: www.robots-argentina.com.ar/MotorCC_L293D.htm

Las salidas tienen un diseño que permite el manejo directo de cargas inductivas tales como relés, solenoides, motores de corriente continua y motores por pasos, ya que

incorpora internamente los diodos de protección de contracorriente para cargas inductivas.

Las entradas son compatibles con niveles de lógica TTL. Para lograr esto, incluso cuando se manejen motores de voltajes no compatibles con los niveles TTL, el chip tiene patas de alimentación separadas para la lógica (VCC1, que debe ser de 5V) y para la alimentación de la carga (VCC2, que puede ser entre 4,5V y 36V).

2.4 AMPLIFICADOR DE CORRIENTE

El transistor Darlington (comúnmente llamado par Darlington por aquellos en la industria electrónica) es una estructura compuesta de un diseño particular hecho por dos transistores bipolares conectados de tal manera que la corriente amplificada por el primer transistor se amplifica aún más por el segundo. Esta configuración proporciona una ganancia de corriente mucho más alta que cada transistor tomado por separado.

FIGURA 10 Diagrama de circuito de un par Darlington utilizando transistores NPN

FUENTE: Wikipedia - Transistor Darlington

El ULN2003A es un conjunto de siete transistores NPN Darlington capaces de generar 500 mA, 50 V de salida. Presenta diodos de retorno de cátodo común para conmutar cargas inductivas. Puede venir en paquetes PDIP , SOIC , SOP o TSSOP . En la misma familia están ULN2002A, ULN2004A, así como ULQ2003A y ULQ2004A, diseñados para diferentes niveles de entrada lógica.

Especificaciones principales del ULN2003A:

- Corriente de colector nominal de 500 mA (salida única)
- Salida de 50 V (hay una versión que admite salida de 100 V)
- Incluye diodos flyback de salida
- Entradas compatibles con TTL y lógica CMOS de 5 V

Aplicaciones

El uso típico del ULN2003A está en los circuitos del controlador para relés, lámparas y pantallas LED, motores paso a paso, buffers lógicos y controladores de línea.

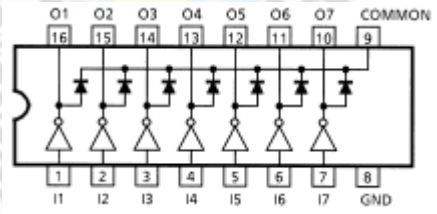


FIGURA 11 Esquema interno del integrado ULN2003A

FUENTE: Wikipedia - ULN2003A

2.5 MOTORESELECTRICOS

El motor eléctrico es un dispositivo que convierte la energía eléctrica en energía mecánica por medio de la acción de los campos magnéticos generados en sus bobinas. Son máquinas eléctricas rotatorias compuestas por un estator y un rotor

Son utilizados en infinidad de sectores tales como instalaciones industriales, comerciales y particulares. Su uso está generalizado en ventiladores, vibradores para teléfonos móviles, bombas, medios de transporte eléctricos, electrodomésticos, esmeriles angulares y otras herramientas eléctricas, unidades de disco, etc. Los motores eléctricos pueden ser impulsados por fuentes de corriente continua (CC), y por fuentes de corriente alterna (AC).

La corriente directa o corriente continua proviene de las baterías, los paneles solares, dínamos, fuentes de alimentación instaladas en el interior de los aparatos que operan con estos motores y con rectificadores. La corriente alterna puede tomarse para su uso en motores eléctricos bien sea directamente de la red eléctrica, alternadores de las plantas eléctricas de emergencia y otras fuentes de corriente alterna bifásica o trifásica como los inversores de potencia.

2.5.1 MOTORES DE CORRIENTE CONTINUA

2.5.1.1 Motores de corriente continua de imán permanente

Los motores de imán permanente (IP) son motores eléctricos que utilizan la combinación de campos magnéticos de naturaleza permanente (Imanes) y campos magnéticos inducidos producidos por la corriente de excitación externa que fluye a través de los devanados del estator.

2.5.1.2 Motor paso a paso

El motor paso a paso conocido también como motor de pasos es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de girar una cantidad de grados (paso o medio paso) dependiendo de sus entradas de control. El motor paso a paso se comporta de la misma manera que un conversor digital-analógico (D/A) y puede ser gobernado por impulsos procedentes de sistemas digitales. Este motor presenta las ventajas de tener precisión y repetitividad en cuanto al posicionamiento. Entre sus principales aplicaciones destacan los robots, drones, radiocontrol, impresoras digitales, automatización, fotocomponedoras, preprensa, etc.

2.5.1.3 Servomotor

Un servomotor es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.1

Un servomotor es un motor eléctrico que puede ser controlado tanto en velocidad como en posición.

Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

Está conformado por un motor, una caja reductora y un circuito de control. También potencia proporcional para cargas mecánicas. Un servo, por consiguiente, tiene un consumo de energía reducido.

La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cuál es la corriente que consume. La corriente depende principalmente del par, y puede exceder un amperio si el servo está enclavado.

En otras palabras, un servomotor es un motor especial al que se ha añadido un sistema de control (tarjeta electrónica), un potenciómetro y un conjunto de engranajes. Con anterioridad los servomotores no permitían que el motor girara 360 grados, solo aproximadamente 180; sin embargo, hoy en día existen servomotores en los que puede ser controlada su posición y velocidad en los 360 grados. Los servomotores son comúnmente usados en modelismo como aviones, barcos, helicópteros y trenes para controlar de manera eficaz los sistemas motores y los de dirección.

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los 50 hertz, así las señales PWM tendrán un periodo de veinte milisegundos. La electrónica dentro del servomotor responderá al ancho de la señal modulada. Si los circuitos dentro del servomotor reciben una señal de entre 1 a 1,4 milisegundos, éste se moverá en sentido horario; entre 1,6 a 2 milisegundos moverá el servomotor en sentido antihorario; 1,5 milisegundos representa un estado neutro para los servomotores estándares.

2.5.2 MOTORES DE CORRIENTE ALTERNA

2.5.2.1 Motor Monofásico Síncrono

Es un motor idéntico al motor de corriente contiua con excitación en serie.Pero en corriente alterna, el funcionamiento del motor se basa en el acoplamiento de campos magnéticos que giran al unísono.

Para que se produzca este acoplamiento, el rótor tiene unas bobinas unidas a un colector formado por delgas, en serie con las bobinas del inductor. Un par de escobillas aplican la corriente al rótor.

Por lo tanto, el mismo motor puede funcionar tanto con corriente continua como con alterna. Pero, además, puede funcionar como dínamo. Por ello se le denomina motor universal, y es ampliamente utilizado en pequeños electrodomésticos.También son motores típicos de los ferrocarriles eléctricos, especialmente suburbanos.

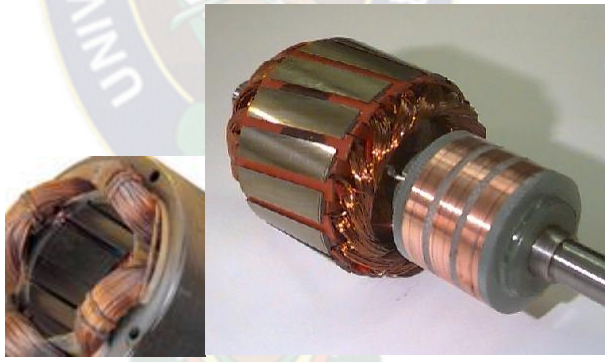


FIGURA 12 Inductor e Inducido de un motor síncrono

FUENTE: http://www.juntadeandalucia.es/averroes/centros-tic/21700290/helvia/aula/archivos/repositorio/0/29/html/Motores_de_corriente_alterna.htm

2.5.2.2 Motor Monofásico Asíncrono

En este tipo de motores, el estátor genera un campo magnético giratorio. Para ello, se dispone de dos pares de bobinas perpendiculares. Una de ellas se conecta directamente a la corriente alterna, generando un campo magnético oscilante. En la otra bobina se intercala un condensador cuya misión es desfasar la corriente que llega a la bobina 90° (eléctricos) respecto a la corriente de la bobina anterior, con lo cual, el campo magnético que genera esta segunda bobina estará también desfasado respecto al anterior. La composición de ambos campos es una suma de vectores y la resultante gira en el espacio, como se puede comprobar en la animación:

El campo magnético giratorio induce una corriente en los conductores del rótor (razón por la que al rótor se le llama también inducido) siempre que exista una variación de flujo magnético. Ésto ocurre siempre, ya que el rótor gira a menor velocidad que la velocidad de sincronismo a la que gira el campo.

Para que se produzca corriente que circule libremente por el rotor, los conductores deben formar un cortocircuito, que se consigue con llamados rotores de jaula de ardilla.

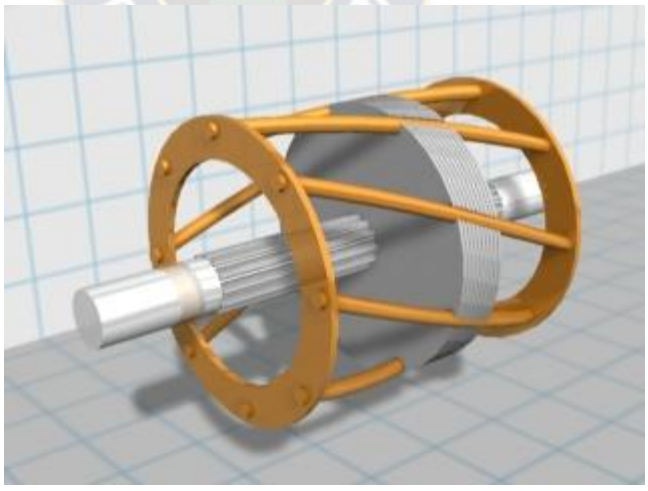


FIGURA 13 Rotor jaula de ardilla

FUENTE: http://www.juntadeandalucia.es/averroes/centros-tic/21700290/helvia/aula/archivos/repositorio/0/29/html/Motores_de_corriente_alterna.htm

2.5.2.3 Motor Trifásico Síncrono

De forma similar a los motores monofásicos, los motores trifásicos consiguen un campo magnético giratorio. El motor trifásico síncrono tiene un rótor constitutido por un electroimán. No es un motor muy corriente por la complicación que supone alimentar el inductor con corriente alterna y el inducido con corriente continua, pero su velocidad de giro es fija e igual a la de sincronismo.

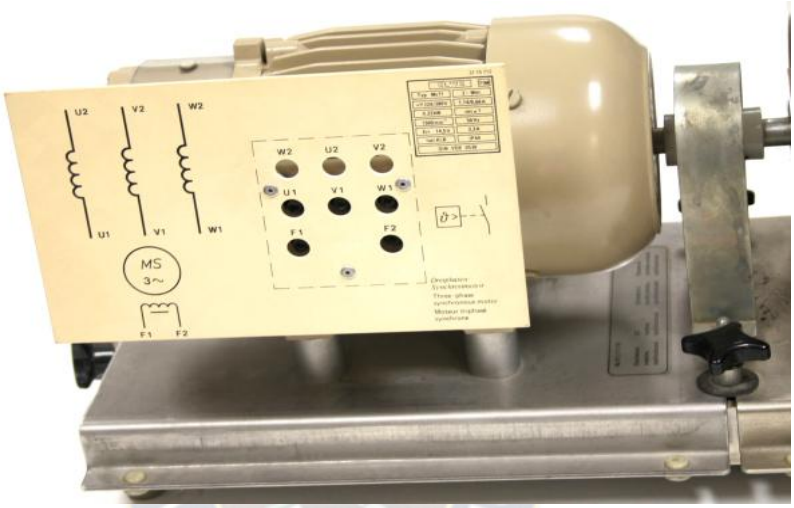


FIGURA 14 Motor Trifásico Síncrono

FUENTE: http://www.juntadeandalucia.es/averroes/centros-tic/21700290/helvia/aula/archivos/repositorio/0/29/html/Motores_de_corriente_alterna.htm

2.5.2.4 Motor Trifásico Asíncrono

El funcionamiento de estos motores es totalmente análogo al de los motores monofásicos de inducción:

- Un campo magnético giratorio.
- Inducción de corriente en el rótor por causa del campo que gira a mayor velocidad que el propio rótor
- Fuerza de Lorentz y fuerza de atracción magnética

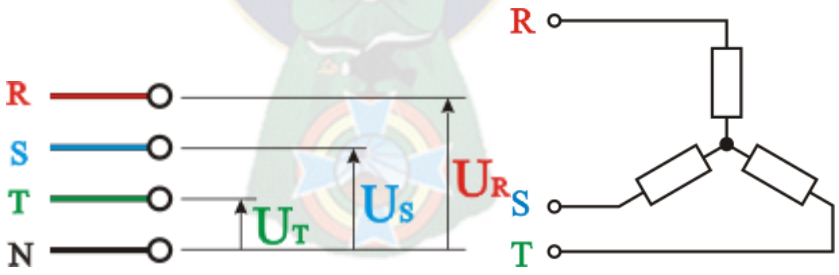
Y el campo magnético giratorio se consigue conectando cada una de las bobinas a una línea de corriente trifásica:

El rótor o inducido suele ser de jaula de ardilla, pero también puede ser de tipo bobinado, con la ventaja de poder regular la corriente de cortocircuito mediante potenciómetros, con lo cual se regula la velocidad de giro y el par desarrollado por el motor.

Los motores trifásicos presentan unas características especiales de utilización, ya que con los mismos tres cables de corriente se pueden realizar dos tipos de conexiones en el inductor.

Conexión en estrella.- Un extremo de las tres bobinas se junta y cada uno de los extremos libres se conecta a cada uno de los cables (si las tres bobinas son idénticas, las corrientes se compensan y no es necesario el conductor neutro). En este caso cada bobina del motor está sometida a la tensión UR, US y UT, que suele ser de 220 V, y por cada una circula una intensidad igual a la que circula por cada conductor:

La tensión a la que está conectada cada fase del motor es la tensión de línea entre √3 (220 V)



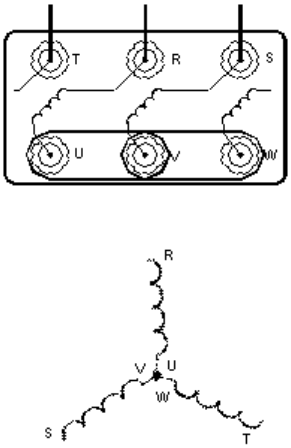
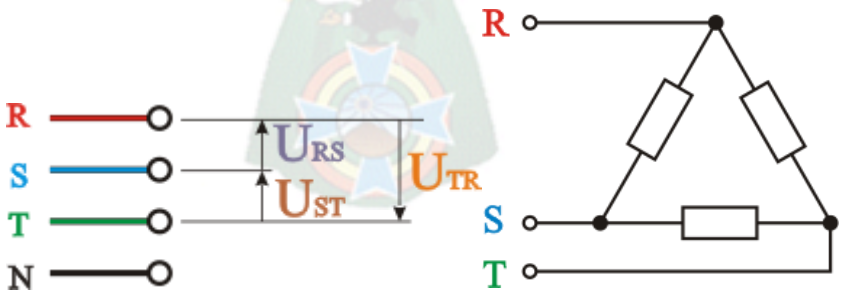


FIGURA 15 Conexión en estrella

FUENTE: http://www.juntadeandalucia.es/averroes/centros-tic/21700290/helvia/aula/archivos/repositorio/0/29/html/Motores_de_corriente_alterna.htm

Conexión en triángulo.-Cada extremo de las tres bobinas se une al extremo de la bobina siguiente no siendo necesario el conductor neutro. En este caso cada bobina está sometida a tensión de línea, URS, UST y UTR, que suele ser de 380 V. La intensidad que circula por cada bobina es inferior a la que circula por cada conductor precisamente √3 veces.



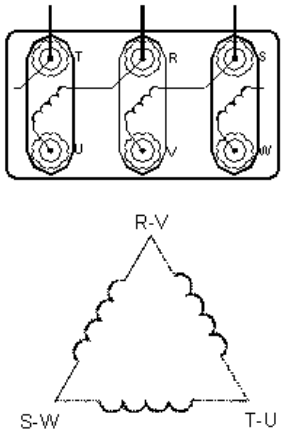


FIGURA 16 Conexión en triangulo

FUENTE: http://www.juntadeandalucia.es/averroes/centros-tic/21700290/helvia/aula/archivos/repositorio/0/29/html/Motores_de_corriente_alterna.htm

2.6 EL POLIPASTO

En un sistema de poleas hay tres clases:

- Aparejo potencial o trocla
- Aparejo factorial o montón
- Aparejo diferencial o tecle

2.6.1 Aparejo potencial o trocla

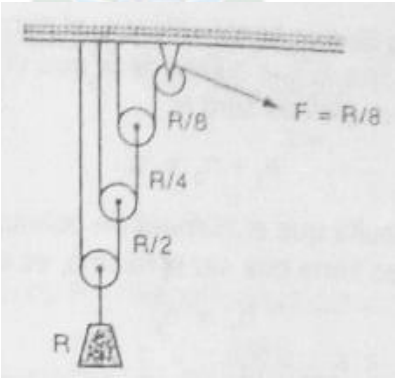


FIGURA 17 Aparejo potencial o trocla

FUENTE Goñi Galarza – FISICA GENERAL

Es el conjunto de una polea fija y varias poleas móviles. La primera polea móvil de abajo reduce a la mitad la fuerza necesaria para levantar la resistencia; la segunda de abajo reduce a la cuarta parte, la tercera a la octava, etc. es decir: en general, según el número de poleas móviles, la fuerza necesaria para levantar un peso se reduce a la resistencia dividida entre 2 elevado a una potencia igual al número de poleas móviles:

$$F = \frac{R}{2^n}$$

F: Fuerza Aplicada.

R: Resistencia a vencer o peso que levantar.

n: Numero de poleas móviles.

2.6.2 Aparejo factorial o montón

Es un conjunto de poleas móviles y un conjunto de poleas fijas. Puede ser n1 el número de poleas móviles y n2 el número de poleas fijas lo que quiere decir que el número total de poleas será n:

$$n_1 + n_2 = n$$

Pero resulta que el número de poleas móviles y fijas tiene que ser el mismo, es decir:

$$n_1 = n_2$$

Si la fuerza “F” se desplaza una distancia d1, la resistencia “R” sube una distancia d2. El trabajo realizado por “F” ha sido transmitido a la resistencia “R”, luego igualando los trabajos:

$$F.d_1 = R.d_2$$

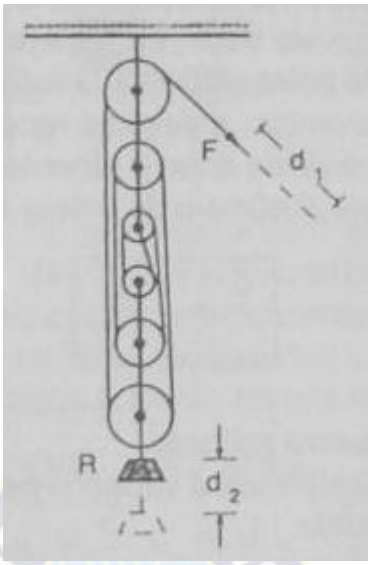


FIGURA 18Aparejo factorial o montón

FUENTE Goñi Galarza – FISICA GENERAL

Pero:

$$d_1 = n \cdot d_2$$

Por tanto:

$$F \cdot n \cdot d_2 = R \cdot d_2$$

$$F = \frac{R}{n}$$

F: Fuerza requerida para equilibrar R.

R: Resistencia, o peso que se quiere levantar

n: Númerototal de poleas entre fijas y móviles.

2.6.3 Aparejo diferencial o tecle

Consta de una polea fija con dos radios distintos (R y r) y con perímetros engranados; en realidad se trata de dos poleas soldadas en sus caras laterales; además, consta de una polea móvil, también con perímetro engranado, esta polea es la que soporta la carga “P”.

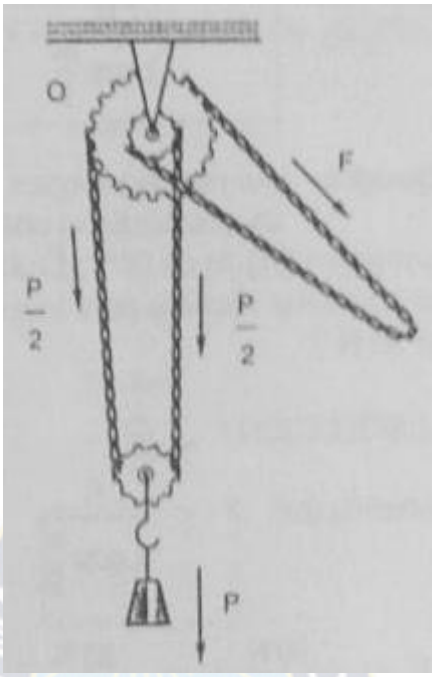


FIGURA 19 Aparejo diferencial o tecle

FUENTE Goñi Galarza – FISICA GENERAL

La condición de equilibrio ideal se obtiene tomando momentos con respecto al eje de giro "O" de la polea fija.

$$\Sigma M_O = 0$$

$$FR + \frac{P}{2} r - \frac{P}{2} R = 0$$

$$F = \frac{P(R - r)}{2R}$$

2.7 POTENCIA MECANICA

Es una magnitud escalar que mide la rapidez con que se transfiere energía; es decir la rapidez con que se realiza un trabajo mecánico.

Por ejemplo dos personas A y B elevan bloques idénticos, las mismas alturas "h". "A" lo hace en 4 segundos, "B" en 2 segundos.

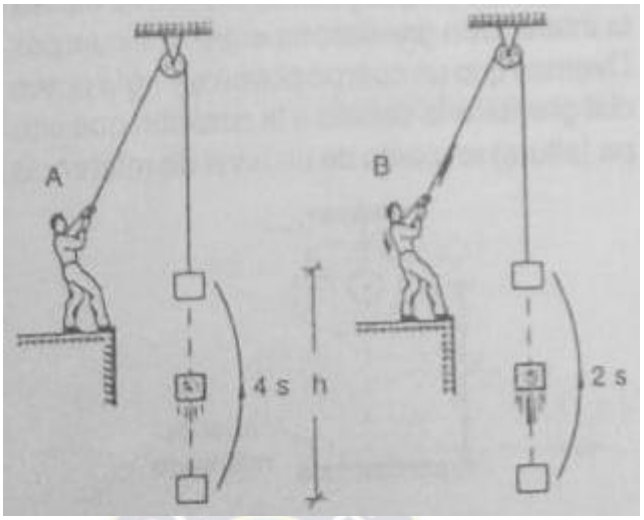


FIGURA 20 Transferencia de energía

FUENTE Goñi Galarza – FISICA GENERAL

Las personas A y B transfieren energía a los bloques. Si los trabajos son iguales:

$$T_A = T_B$$

Y se cumple que B realiza el trabajo en menos tiempo que A. Entonces se tiene que la potencia desarrollada por B es mayor que la potencia desarrollada por A.

$$P_B > P_A$$

Se define:

$$\text{POTENCIA} = \frac{\text{ENERGÍA TRANSFERIDA}}{\text{TIEMPO}}$$

$$P = \frac{T^F_{A \to B}}{t}$$

$T^{F}_{A \to B}$: Es el trabajo desarrollado, en joule “J”

t: Es el tiempo empleado, en segundos “s”

P: es la potencia desarrollada, medida en watt $W = \frac{J}{S}$

* Los puntos 4.5 EL POLIPASTO y 4.6 POTENCIA MECANICA son copia del libro FISICA GENERAL del autor Goñi Galarza.

CAPITULO 3

3.1 DESARROLLO DEL TRABAJO

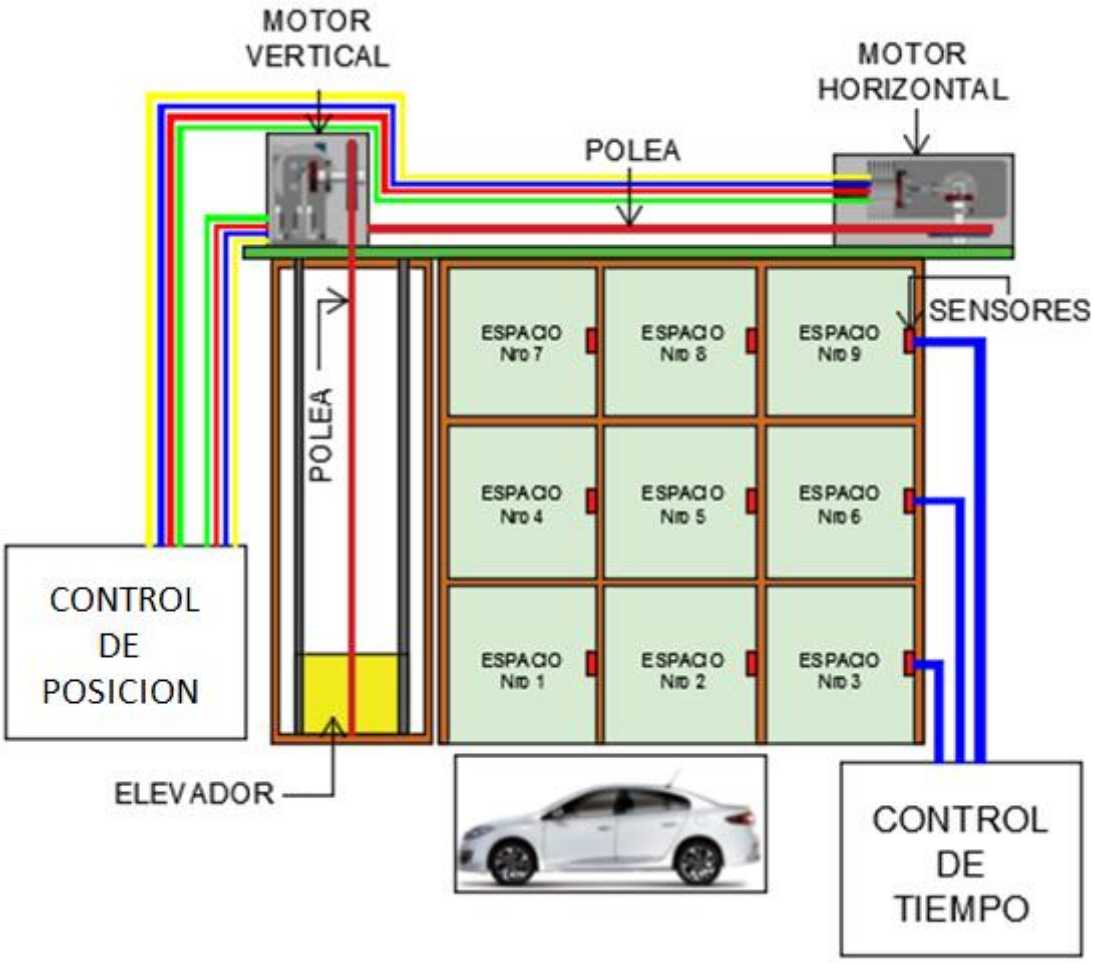
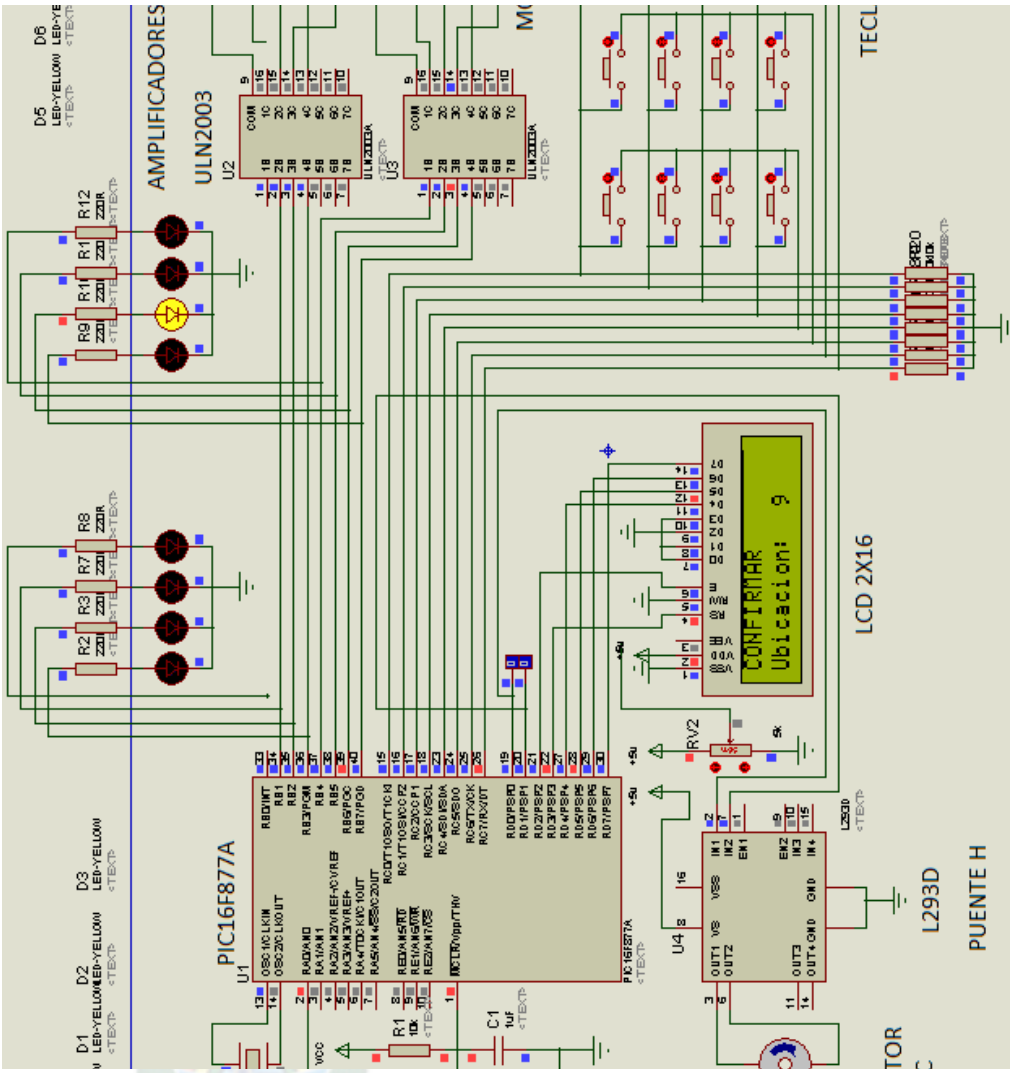


FIGURA 21 ESQUEMA DEL ESTACIONAMIENTO

3.2 CONTROL DE POSICIONAMIENTO

3.2.1 DIAGRAMA DEL CIRCUITO



El sistema tiene como elemento principal al PIC16F877A el cual controla a dos motores paso a paso para el desplazamiento vertical y horizontal mediante amplificadores de corriente conectados al puerto B y un motor DC controlando la inversión de giro mediante un circuito puente H conectado al puerto D que desplaza una bandeja para posicionar el vehículo en un determinado espacio.

El control de ingreso del vehículo o salida del vehículo se hace mediante una palanca de selección conectada al puerto A del microcontrolador.

El control de la posición en la que se desea estacionar el vehículo se hace mediante un teclado que está conectado al puerto C del microcontrolador.

El sistema cuenta con un display LCD para que el control del estacionamiento sea comprensible.

3.2.2 LISTA DE MATERIALES

ITEM	CANTIDAD	P. UNITARIO (Bs)	COSTO (Bs)
PIC16F877A	1	40	40
LCD 2X16	1	80	80
L293D	1	20	20
TECLADO 4X4	1	40	40
ULN2003	2	5	10
MOTOR PASO	2	10	20
MOTOR DC	1	5	5
RESISTENCIA 10K	18	0,20	3,60
SWITCH/PULSADOR	2	1	2
CRISTAL 4MHz	1	5	5
CAPACITOR 22nF	2	0,50	1
TOTAL			226,60

3.2.3 CODIGO DE PROGRAMACION

Lenguaje de programación: mikroBasic

```
program CONTROL_DE_POSICION

const motor as byte[4]=(1,2,4,8)  'vector con valores para girar el motor paso
dim paso as byte                   'variable para dar un paso
const motor2 as byte[4]=(16,32,64,128) 'vector con valores para girar el motor paso 2
dim paso2 as byte                      'variable para dar un paso
dimaux as byte                   'variable auxiliar

dim tecla as byte                 'variable para teclado
dimteclaStr as string[3]         'cadena para mostra tecla en display
dim confirmar as byte             'variable para confirmar
```

```
'declaracion de banderas para verificar si el espacio esta Ocupado o Desocupado
dim band1,band2,band3,band4,band5,band6,band7,band8,band9 as bit

dimkeypadPort as byte at PORTC   'conectar teclado al puerto C

'conectar display al puerto D
dim
  LCD_RS as sbit at RD3_bit
  LCD_EN as sbit at RD2_bit
  LCD_D7 as sbit at RD7_bit
  LCD_D6 as sbit at RD6_bit
  LCD_D5 as sbit at RD5_bit
  LCD_D4 as sbit at RD4_bit

dim
LCD_RS_Direction as sbit at TRISD3_bit
LCD_EN_Direction as sbit at TRISD2_bit
  LCD_D7_Direction as sbit at TRISD7_bit
  LCD_D6_Direction as sbit at TRISD6_bit
  LCD_D5_Direction as sbit at TRISD5_bit
  LCD_D4_Direction as sbit at TRISD4_bit

Sub procedurelimpiar_LCD 'procedimiento para Limpiar LCD
Lcd_Cmd(_LCD_CLEAR)
Lcd_Out(2,1,"Ubicacion:     ")
end sub

subprocedure Subir 'procedimiento para subir
paso=paso+1
if paso>3 then
paso=0
end if
    PORTB=motor[paso]
delay_ms(20)
end sub

subprocedure Bajar  'procedimiento para bajar
paso=paso-1
if paso=255 then
paso=3
endif
if paso=254 then   'solo pasa la primera vez
paso=0
end if
    PORTB=motor[paso]
delay_ms(20)
end sub

subprocedure Izquierda  'procedimiento para desplazar a Izquierda
    paso2=paso2+1
if paso2>3 then
       paso2=0
end if
    PORTB=motor2[paso2]
delay_ms(20)
end sub

subprocedure Derecha    'procedimiento para desplazar a Derecha
    paso2=paso2-1
if paso2=255 then
       paso2=3
endif
if paso2=254 then   'solo pasa la primera vez
paso2=0
end if
    PORTB=motor2[paso2]
delay_ms(20)
end sub

subprocedure Ocupado  'procedimiento para mostrar espacio ocupado
```

```
for aux=0 to 3
Lcd_Out(1,1,"ESTA OCUPADO")
delay_ms(150)
Lcd_Cmd(_LCD_CLEAR)
delay_ms(150)
next aux
end sub

subprocedure Desocupado  'procedimiento para mostrar espacio desocupado
for aux=0 to 3
Lcd_Out(1,1,"ESTA DESOCUPADO")
delay_ms(150)
Lcd_Cmd(_LCD_CLEAR)
delay_ms(150)
next aux
end sub

subprocedure Hora  'procedimiento para verificar hora
dimverificar as byte
verificar=0
for aux=0 to 3
Lcd_Out(1,1,"                ")
delay_ms(150)
Lcd_Out(1,1,"VERIFIQUE HORA")
delay_ms(150)
next aux
while (verificar=0)
verificar=Keypad_Key_Click()
ifverificar=16 then
goto final
else
verificar=0
end if
wend
final:
limpiar_LCD
end sub

sub procedure Subir_30 'sube 30 pasos
for aux=1 to 30
Subir
next aux
end sub

sub procedure Bajar_30 'baja 30 pasos
for aux=1 to 30
Bajar
next aux
end sub

sub procedure Subir_180  'sube 180 pasos
for aux=1 to 180
Subir
next aux
end sub

sub procedure Bajar_180  'baja 180 pasos
for aux=1 to 180
Bajar
next aux
end sub

sub procedure Subir_348   'sube 348 pasos
for aux=1 to 200
Subir
next aux
for aux=1 to 148
Subir
next aux
end sub
```

```
sub procedure Bajar_348   'baja 348 pasos
for aux=1 to 200
Bajar
next aux
for aux=1 to 148
Bajar
nextaux
end sub

subprocedure Derecha_150   'desplaza a Derecha 150 pasos
for aux=1 to 150
Derecha
nextaux
end sub

subprocedure Izquierda_150 'desplaza a Izquierda 150 pasos
for aux=1 to 150
Izquierda
nextaux
end sub

subprocedure ingresa    'ingresar bandeja porta auto
PORTD.0=1
    PORTD.1=0
delay_ms(2000)
    PORTD.0=0
    PORTD.1=0
end sub

subprocedure sale       'sacar bandeja porta auto
PORTD.0=0
    PORTD.1=1
delay_ms(2000)
    PORTD.0=0
    PORTD.1=0
end sub


main:
     ADCON1=7      'entradas analogicas deshabilitadas

     PORTA=0       'limpiar puerto A
     TRISA.0=1     'bit 0 del puerto A como entrada
     PORTB=0       'limpiar puerto B
     TRISB=0       'puerto B como salida
     PORTD.0=0     'limpiar bit para el motor DC
     PORTD.1=0     'limpiar bit para el motor DC
     TRISD.0=0    'bit 0 del puerto D como salida
     TRISD.1=0    'bit 1 del puerto D como salida
keypad_Init()
Lcd_Init()
Lcd_Cmd(_LCD_CURSOR_OFF)

band1=0  'banderas que indican posicion desocupada
band2=0
     band3=0
     band4=0
     band5=0
     band6=0
band7=0
     band8=0
     band9=0

paso=255    'valores iniciales de variable paso
     paso2=255

     'Mostrar un mensaje inicial en display
Lcd_Out(1,1," ESTACIONAMIENTO")
Lcd_Out(2,1,"     VERTICAL   ")
delay_ms(2000)
Lcd_Cmd(_LCD_CLEAR)
```

```
Lcd_Out(1,1,"Ingrese porfavor")
Lcd_Out(2,1,"  la ubicacion  ")
delay_ms(2000)
limpiar_LCD

bucle:
tecla=0
while (tecla=0) 'bucle para determinar ubicacion de los espacios
tecla=keypad_key_click()
                Select case tecla
case 1
tecla=7
case 2
tecla=4
case 3
tecla=1
case 5
tecla=8
case 6
tecla=5
case 7
tecla=2
case 9
tecla=9
case 10
tecla=6
case 11
tecla=3
case else
tecla=0
end select
wend
ByteToStr(tecla,teclaStr)
Lcd_Out(2,11,teclaStr)   'mostrar ubicacion en display

Lcd_Out(1,1,"CONFIRMAR")
confirmar=0
while (confirmar=0)    'bucle para confirmar o denegar ubicacion
confirmar=Keypad_Key_Click()
                Select case confirmar
case 16
goto inicio
case 12
limpiar_LCD
gotobucle
case else
confirmar=0
endselect
wend

inicio:     'inicio del posicionamiento
select case tecla
case 1   'caso en que se selecciona espacio 1
if PORTA.0=1 then   'pregunta si ingresa el auto
if band1=0 then  'pregunta si la posicionesta desocupada
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            band1=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band1=1 then  'pregunta si la posicionesta ocupada
                             Hora      'ir a la rutina para verificar hora
                             Ingresa
                             Subir_30
                             Sale
```

```
                              Bajar_30
                              band1=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                               Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif
case 2  'caso en que se selecciona espacio 2
if PORTA.0=1 then   'pregunta si ingresa el auto
if band2=0 then  'pregunta si la posicionesta desocupada
                             Derecha_150
                             Subir_30
                             Ingresa
                             Bajar_30
                             Sale
                             Izquierda_150
                             band2=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                              Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band2=1 then  'pregunta si la posicionesta ocupada
                              Hora      'ir a la rutina para verificar hora
                              Derecha_150
                              Ingresa
                              Subir_30
                              Sale
                              Bajar_30
                              Izquierda_150
                              band2=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                               Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif
case 3  'caso en que se selecciona espacio 3
if PORTA.0=1 then   'pregunta si ingresa el auto
if band3=0 then  'pregunta si la posicionesta desocupada
                             Derecha_150
                             Derecha_150
                             Subir_30
                             Ingresa
                             Bajar_30
                             Sale
                             Izquierda_150
                             Izquierda_150
                             band3=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                              Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band3=1 then  'pregunta si la posicionesta ocupada
                              Hora      'ir a la rutina para verificar hora
                              Derecha_150
                              Derecha_150
                              Ingresa
                              Subir_30
                              Sale
                              Bajar_30
                              Izquierda_150
                              Izquierda_150
                              band3=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                               Desocupado 'ir a la rutina de posicion desocupada
```

```
limpiar_LCD 'limpiar display
endif
endif
case 4  'caso en que se selecciona espacio 4
if PORTA.0=1 then   'pregunta si ingresa el auto
if band4=0 then  'pregunta si la posicionesta desocupada
                            Subir_180
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            Bajar_180
                            band4=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band4=1 then  'pregunta si la posicionesta ocupada
                            Hora      'ir a la rutina para verificar hora
                            Subir_180
                            Ingresa
                            Subir_30
                            Sale
                            Bajar_30
                            Bajar_180
                            band4=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                             Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif

case 5  'caso en que se selecciona espacio 5
if PORTA.0=1 then   'pregunta si ingresa el auto
if band5=0 then  'pregunta si la posicionesta desocupada
                            Derecha_150
                            Subir_180
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            Bajar_180
                            Izquierda_150
                            band5=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band5=1 then  'pregunta si la posicionesta ocupada
                             Hora      'ir a la rutina para verificar hora
                             Derecha_150
                             Subir_180
                             Ingresa
                             Subir_30
                             Sale
                             Bajar_30
                             Bajar_180
                             Izquierda_150
                             band5=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                              Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif
case 6  'caso en que se selecciona espacio 6
```

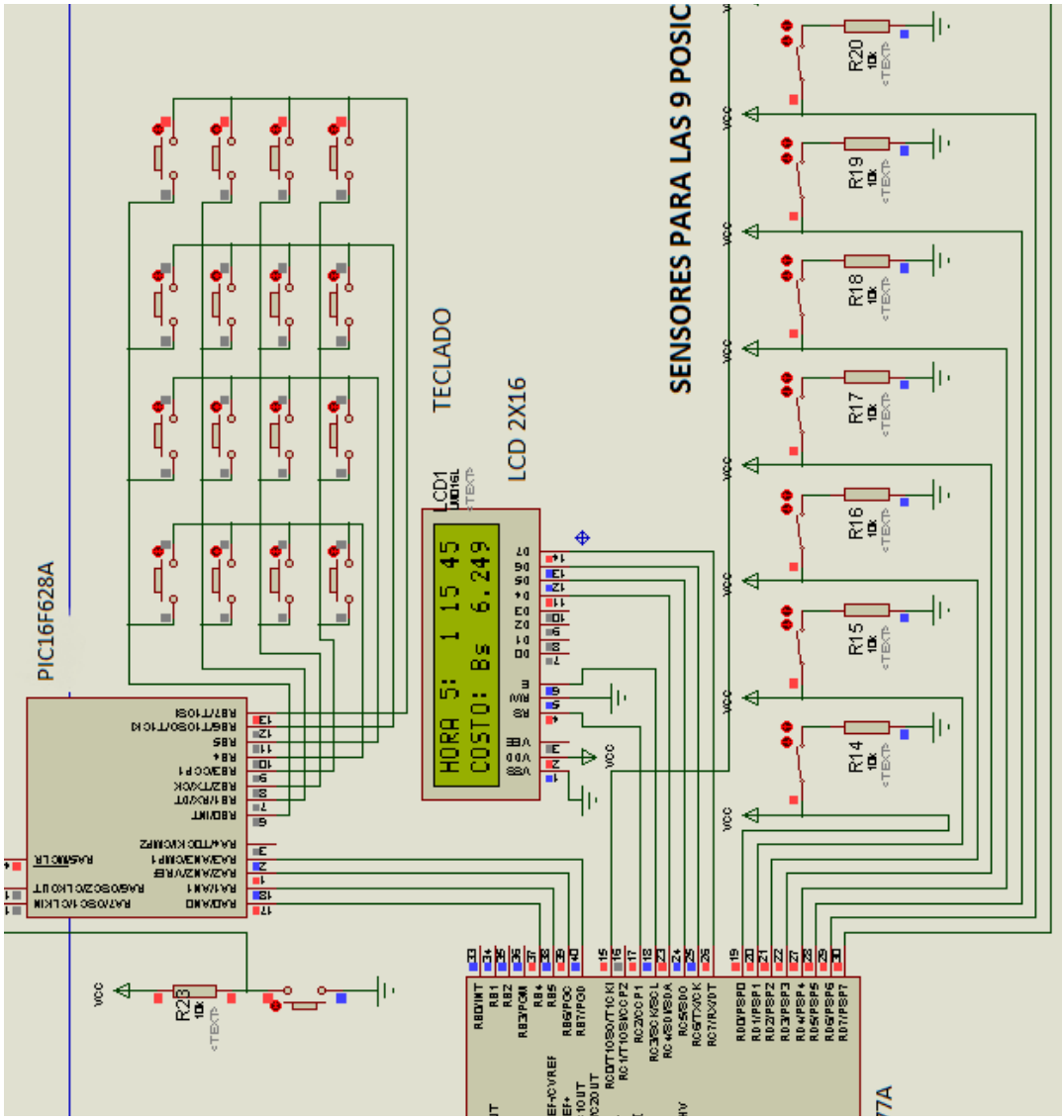
```
if PORTA.0=1 then   'pregunta si ingresa el auto
if band6=0 then  'pregunta si la posicionesta desocupada
                            Derecha_150
                            Derecha_150
                            Subir_180
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            Bajar_180
                            Izquierda_150
                            Izquierda_150
                            band6=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band6=1 then  'pregunta si la posicionesta ocupada
                            Hora      'ir a la rutina para verificar hora
                            Derecha_150
                            Derecha_150
                            Subir_180
                            Ingresa
                            Subir_30
                            Sale
                            Bajar_30
                            Bajar_180
                            Izquierda_150
                            Izquierda_150
                            band6=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                             Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif
case 7  'caso en que se selecciona espacio 7
if PORTA.0=1 then   'pregunta si ingresa el auto
if band7=0 then  'pregunta si la posicionesta desocupada
                            Subir_348
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            Bajar_348
                            band7=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band7=1 then  'pregunta si la posicionesta ocupada
                             Hora      'ir a la rutina para verificar hora
                             Subir_348
                             Ingresa
                             Subir_30
                             Sale
                             Bajar_30
                             Bajar_348
                             band7=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                              Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif
case 8  'caso en que se selecciona espacio 8
if PORTA.0=1 then   'pregunta si ingresa el auto
```

```
if band8=0 then  'pregunta si la posicionesta desocupada
                            Derecha_150
                            Subir_348
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            Bajar_348
                            Izquierda_150
                            band8=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band8=1 then  'pregunta si la posicionesta ocupada
                             Hora      'ir a la rutina para verificar hora
                             Derecha_150
                             Subir_348
                             Ingresa
                             Subir_30
                             Sale
                             Bajar_30
                             Bajar_348
                             Izquierda_150
                             band8=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                              Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
endif
endif
case 9  'caso en que se selecciona espacio 9
if PORTA.0=1 then   'pregunta si ingresa el auto
if band9=0 then  'pregunta si la posicionesta desocupada
                            Derecha_150
                            Derecha_150
                            Subir_348
                            Subir_30
                            Ingresa
                            Bajar_30
                            Sale
                            Bajar_348
                            Izquierda_150
                            Izquierda_150
                            band9=1     'indica que la posicionesta ocupada
limpiar_LCD 'limpiar display
else    'posicion ocupada
                             Ocupado 'ir a la rutina de posicion ocupada
limpiar_LCD 'limpiar display
end if
else        'sacar el auto
if band9=1 then  'pregunta si la posicionesta ocupada
                             Hora      'ir a la rutina para verificar hora
                             Derecha_150
                             Derecha_150
                             Subir_348
                             Ingresa
                             Subir_30
                             Sale
                             Bajar_30
                             Bajar_348
                             Izquierda_150
                             Izquierda_150
                             band9=0    'indica que la posicionesta desocupada
limpiar_LCD 'limpiar display
else
                              Desocupado 'ir a la rutina de posicion desocupada
limpiar_LCD 'limpiar display
end if
```

```
end if
end select
gotobucle
end.
```

3.3 CONTROL DE TIEMPO Y COSTO

3.3.1 DIAGRAMA DEL CIRCUITO



El sistema está basado en dos microcontroladores: PIC16F877A y PIC16F628A.

El PIC16F877A que se encarga de verificar si las posiciones están ocupadas o desocupadas mediante sensores que están simulados por pulsadores conectados al puerto D y C; en el caso que la posición estuviera ocupada, el microcontrolador se encarga de iniciar un contador cuya función es de cronometrar el tiempo de uso, con tales valores de

tiempo cronometrado se hace un cálculo del costo cuyos datos son desplegados en un display LCD conectado al puerto C, para tal efecto el microcontrolador está en un constante proceso de verificación de posiciones ocupadas o desocupadas estando en un bucle infinito.

Para romper el bucle se emplea un microcontrolador PIC16F628A cuya función es de transmitir el dato ingresado por un teclado conectado al puerto B; la función del teclado es de seleccionar la posición del estacionamiento del cual se desea ver el tiempo de uso y el costo; una vez introducido el dato por el teclado el PIC16F628A se encarga de transmitir este dato a través del puerto A (RA0-RA3) al puerto B (RB4-RB7) del PIC16F877A provocando una interrupción por cambio de nivel en el puerto B, el cual guarda los datos de tiempo y costo de la posición seleccionada y los muestra en el display LCD.

3.3.2 LISTA DE MATERIALES

ITEM	CANTIDAD	P. UNITARIO (Bs)	COSTO (Bs)
PIC16F877A	1	40	40
PIC16F628A	1	20	20
LCD 2X16	1	80	80
TECLADO 4X4	1	40	40
RESISTENCIA 10K	18	0,20	3,60
SWITCH/PULSADOR	2	1	2
CRISTAL 4MHz	1	5	5
CAPACITOR 22nF	2	0,50	1
SENSORES (PULSADORES)	9	1	9
TOTAL			200,60

3.3.3 CODIGO DE PROGRAMACION

PARA EL PIC 16F877A

```
program PRUEBA_INTERRUPCION
dim Pulsador as byte
dim Bandera as bit

' Declarando segundos, minutos y horas
```

```
dim seg1,seg2,seg3,seg4,seg5,seg6,seg7,seg8,seg9 as byte
dim min1,min2,min3,min4,min5,min6,min7,min8,min9 as byte
dim hor1,hor2,hor3,hor4,hor5,hor6,hor7,hor8,hor9 as byte

'Declarando variables para la pantalla
dimsegu as byte
dimminu as byte
dimhora as byte
dimsegStr as string[3]
dimminStr as string[3]
dimhorStr as string[3]

dimespacio as byte

dim centavos as float
dim bolivianos as float
dim total as float
dimtotalStr as string[5]

dim
  LCD_RS as sbit at RC2_bit
  LCD_EN as sbit at RC3_bit
  LCD_D7 as sbit at RC7_bit
  LCD_D6 as sbit at RC6_bit
  LCD_D5 as sbit at RC5_bit
  LCD_D4 as sbit at RC4_bit
dim
LCD_RS_Direction as sbit at TRISC2_bit
LCD_EN_Direction as sbit at TRISC3_bit
  LCD_D7_Direction as sbit at TRISC7_bit
  LCD_D6_Direction as sbit at TRISC6_bit
  LCD_D5_Direction as sbit at TRISC5_bit
  LCD_D4_Direction as sbit at TRISC4_bit

subprocedureinterrupt
    Pulsador=(PORTB)>>4    'tomar el dato de RB4-RB7
select case Pulsador
case 1
segu=seg1
minu=min1
hora=hor1
espacio=49   '1 en caracter ASCII
case 2
segu=seg2
minu=min2
hora=hor2
espacio=50   '2 en caracter ASCII
case 3
segu=seg3
minu=min3
hora=hor3
espacio=51   '3 en caracter ASCII
case 4
segu=seg4
minu=min4
hora=hor4
espacio=52   '4 en caracter ASCII
case 5
segu=seg5
minu=min5
hora=hor5
espacio=53   '5 en caracter ASCII
case 6
segu=seg6
minu=min6
hora=hor6
espacio=54   '6 en caracter ASCII
case 7
segu=seg7
minu=min7
hora=hor7
```

```
espacio=55   '7 en caracter ASCII
case 8
segu=seg8
minu=min8
hora=hor8
espacio=56   '8 en caracter ASCII
case 9
segu=seg9
minu=min9
hora=hor9
espacio=57   '8 en caracter ASCII
endselect
    Bandera=1
    INTCON.RBIF=0        'apagar la Bandera de interrupcion
end sub

main:

     PORTD=0    'limpiarpuerto D
TRISD=$FF  'puerto D como entrada para pulsadores
     PORTB=$00  'limpiar puerto B
     TRISB=%11110000  'RB4 - RB7 entrada para interrupcion
     TRISC.0=1        'bit 0 del puerto C como entrada
     INTCON.RBIF=0    'apagar Bandera de interrupcion por cambio de nivel
     INTCON.RBIE=1    'habilitar interrupcion por cambio de nivel RB4-RB7
     INTCON.GIE=1     'habilitar mascara global de interrupciones

Lcd_Init()       'iniciar libreria del LCD
Lcd_Cmd(_LCD_CURSOR_OFF)  'apagar el cursor del LCD
Lcd_Out(1,1,"HORA")

     seg1=0    'iniciar segundos en 0
seg2=0
     seg3=0
     seg4=0
     seg5=0
     seg6=0
seg7=0
     seg8=0
     seg9=0

     min1=0    'iniciar minutos en 0
     min2=0
     min3=0
     min4=0
     min5=0
     min6=0
     min7=0
     min8=0
     min9=0

     hor1=0    'iniciar horas en 0
hor2=0
     hor3=0
     hor4=0
     hor5=0
     hor6=0
     hor7=0
     hor8=0
     hor9=0

bucle:

Bandera=0
while (not Bandera)
if PORTD.0=1 then 'preguntar si la posicionesta ocupada
if seg1<59 then    'preguta si seg1 es menor a 59
                     seg1=seg1+1     'incrementar seg1
delay_ms(10)    'retardo de segundos
else
                      seg1=0                'vuelve a cero seg1
```

```
if min1<59 then        'pregunta si min1 es menor a 59
                         min1=min1+1         'incrementar min1
else
                          min1=0             'vuelve a cero min1
if hor1<59 then    'pregunta si hor1 es menor a 59
                             hor1=hor1+1     'incrementar hor1
else
                              hor1=0         'vuelve a cero hor1
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg1=0
                   min1=0
hor1=0
endif

if PORTD.1=1 then 'preguntar si la posicionesta ocupada
if seg2<59 then    'preguta si seg2 es menor a 59
                     seg2=seg2+1     'incrementar seg2
delay_ms(10)    'retardo de segundos
else
                      seg2=0                 'vuelve a cero seg2
if min2<59 then        'pregunta si min2 es menor a 59
                         min2=min2+1         'incrementar min2
else
                          min2=0             'vuelve a cero min2
if hor2<59 then    'pregunta si hor2 es menor a 59
                             hor2=hor2+1     'incrementar hor2
else
                              hor2=0         'vuelve a cero hor2
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg2=0
                   min2=0
hor2=0
endif

if PORTD.2=1 then 'preguntar si la posicionesta ocupada
if seg3<59 then    'preguta si seg3 es menor a 59
                     seg3=seg3+1     'incrementar seg3
delay_ms(10)    'retardo de segundos
else
                      seg3=0                 'vuelve a cero seg3
if min3<59 then        'pregunta si min3 es menor a 59
                         min3=min3+1         'incrementar min3
else
                          min3=0             'vuelve a cero min3
if hor3<59 then    'pregunta si hor3 es menor a 59
                             hor3=hor3+1     'incrementar hor3
else
                              hor3=0         'vuelve a cero hor3
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg3=0
                   min3=0
hor3=0
endif

if PORTD.3=1 then 'preguntar si la posicionesta ocupada
if seg4<59 then    'preguta si seg4 es menor a 59
                     seg4=seg4+1     'incrementar seg4
delay_ms(10)    'retardo de segundos
else
```

```
                      seg4=0                'vuelve a cero seg4
if min4<59 then        'pregunta si min4 es menor a 59
                         min4=min4+1         'incrementar min4
else
                          min4=0             'vuelve a cero min4
if hor4<59 then    'pregunta si hor4 es menor a 59
                             hor4=hor4+1     'incrementar hor4
else
                              hor4=0         'vuelve a cero hor4
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg4=0
                   min4=0
hor4=0
endif

if PORTD.4=1 then 'preguntar si la posicionesta ocupada
if seg5<59 then    'preguta si seg5 es menor a 59
                     seg5=seg5+1     'incrementar seg5
delay_ms(10)    'retardo de segundos
else
                      seg5=0                'vuelve a cero seg5
if min5<59 then        'pregunta si min5 es menor a 59
                         min5=min5+1         'incrementar min5
else
                          min5=0             'vuelve a cero min5
if hor5<59 then    'pregunta si hor5 es menor a 59
                             hor5=hor5+1     'incrementar hor5
else
                              hor5=0         'vuelve a cero hor5
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg5=0
                   min5=0
hor5=0
endif

if PORTD.5=1 then 'preguntar si la posicionesta ocupada
if seg6<59 then    'preguta si seg6 es menor a 59
                     seg6=seg6+1     'incrementar seg6
delay_ms(10)    'retardo de segundos
else
                      seg6=0                'vuelve a cero seg6
if min6<59 then        'pregunta si min6 es menor a 59
                         min6=min6+1         'incrementar min6
else
                          min6=0             'vuelve a cero min6
if hor6<59 then    'pregunta si hor6 es menor a 59
                             hor6=hor6+1     'incrementar hor6
else
                              hor6=0         'vuelve a cero hor6
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg6=0
                   min6=0
hor6=0
endif

if PORTD.6=1 then 'preguntar si la posicionesta ocupada
if seg7<59 then    'preguta si seg7 es menor a 59
                     seg7=seg7+1     'incrementar seg7
delay_ms(10)    'retardo de segundos
```

```
else
                      seg7=0                'vuelve a cero seg7
if min7<59 then        'pregunta si min7 es menor a 59
                         min7=min7+1         'incrementar min7
else
                           min7=0            'vuelve a cero min7
if hor7<59 then    'pregunta si hor7 es menor a 59
                             hor7=hor7+1     'incrementar hor7
else
                              hor7=0         'vuelve a cero hor7
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg7=0
                   min7=0
hor7=0
endif

if PORTD.7=1 then 'preguntar si la posicionesta ocupada
if seg8<59 then    'preguta si seg8 es menor a 59
                     seg8=seg8+1     'incrementar seg8
delay_ms(10)    'retardo de segundos
else
                      seg8=0                'vuelve a cero seg8
if min8<59 then        'pregunta si min8 es menor a 59
                         min8=min8+1         'incrementar min8
else
                           min8=0            'vuelve a cero min8
if hor8<59 then    'pregunta si hor8 es menor a 59
                             hor8=hor8+1     'incrementar hor8
else
                              hor8=0         'vuelve a cero hor8
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg8=0
                   min8=0
hor8=0
endif

if PORTC.0=1 then 'preguntar si la posicionesta ocupada
if seg9<59 then    'preguta si seg9 es menor a 59
                     seg9=seg9+1     'incrementar seg9
delay_ms(10)    'retardo de segundos
else
                      seg9=0                'vuelve a cero seg9
if min9<59 then        'pregunta si min9 es menor a 59
                         min9=min9+1         'incrementar min9
else
                           min9=0            'vuelve a cero min9
if hor9<59 then    'pregunta si hor9 es menor a 59
                             hor9=hor9+1     'incrementar hor9
else
                              hor9=0         'vuelve a cero hor9
end if
end if
delay_ms(10)           'retardo de segundos
end if
else
                   seg9=0
                   min9=0
hor9=0
endif
wend


ByteToStr(segu,segStr)  'convertir segu a cadena
```

```
ByteToStr(minu,minStr)  'convertir minu a cadena
ByteToStr(hora,horStr)  'convertir hora a cadena

Lcd_Out(1,1,"HORA  :")
Lcd_Chr(1,6,espacio)    'mostrar numero del espacio (caracter)
Lcd_Out(1,14,segStr)
Lcd_Out(1,11,minStr)
Lcd_Out(1,8,horStr)

centavos=minu*0.0833
bolivianos=hora*5
total=bolivianos+centavos
FloatToStr(total,totalStr)
Lcd_Out(2,1,"COSTO: Bs")
Lcd_Out(2,12,totalStr)

gotobucle
end.
```

PARA EL PIC 16F628A

```
program TECLA_INTERRUPCION

dimtecla as byte
dimkeypadPort as byte at PORTB

main:
     CMCON=$07    'Apagar comparadores
     PORTA=0
TRISA=%11110000
bucle:
Keypad_Init()
tecla=0
while (tecla=0)
tecla=Keypad_Key_Click()
wend
select case tecla
case 1
PORTA=%00000001
case 2
                        PORTA=%00000101
case 3
                        PORTA=%00001001
case 4
                        PORTA=%00001101
case 5
                        PORTA=%00000010
case 6
                        PORTA=%00000110
case 7
                        PORTA=%00001010
case 8
                        PORTA=%00001110
case 9
                        PORTA=%00000011
case 10
                        PORTA=%00000111
case 11
                        PORTA=%00001011
case 12
                        PORTA=%00001111
case 13
                        PORTA=%00000100
case 14
                        PORTA=%00001000
case 15
                        PORTA=%00001100
case 16
                        PORTA=%00000000
```

```
end select
gotobucle
end.
```

3.4 MAQUETA DEL ESTACIONAMIENTO VERTICAL

FIGURA 22 MAQUETA DEL ESTACIONAMIENTO VERTICAL

3.5 CALCULOS

Para el sistema emplearemos el Aparejo potencial o trocla

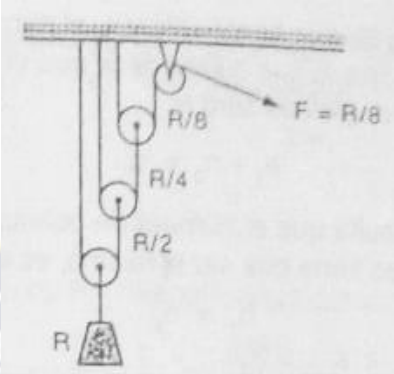


FIGURA 23 Aparejo potencial o trocla

FUENTE Goñi Galarza – FISICA GENERAL

Es el conjunto de una polea fija y varias poleas móviles. La primera polea móvil de abajo reduce a la mitad la fuerza necesaria para levantar la resistencia; la segunda de abajo reduce a la cuarta parte, la tercera a la octava, etc. es decir: en general, según el número de poleas móviles, la fuerza necesaria para levantar un peso se reduce a la resistencia dividida entre 2 elevado a una potencia igual al número de poleas móviles:

$$F = \frac{R}{2^n}$$

F: Fuerza Aplicada.

R: Resistencia a vencer o peso que levantar.

n: Numero de poleas móviles.

El peso de un auto promedio es de 2 toneladas o 2000 kilos convirtiendo en Newton:

R=2000Kg*9,81 m/s2=19620 Newton

Empleando 3 poleas móviles como en la figura reemplazando los datos en la ecuación tenemos que:

F=2452,5 Newton

Para vencer el peso de un auto con peso promedio 2 toneladas se debe aplicar esa fuerza.

Relaciones entre movimiento lineal y movimiento angular.

Velocidad lineal $v = \frac{dx}{\mathrm{dt}}$ ecuación 1

Donde x es la distancia recorrida en m.

Velocidad angular $w = \frac{d\Theta}{\mathrm{dt}}$ ecuación 2

Donde Θ es el ángulo recorrido en radianes

Aceleración lineal $a = \frac{dv}{\mathrm{dt}}$ ecuación 3

Aceleración angular $\alpha = \frac{dw}{\mathrm{dt}}$ ecuación 4

Se define torque (τ en Nm) como la fuerza de torsión que se aplica a un eje, depende de la magnitud de la fuerza aplicada y de la distancia al eje de rotación.

τ= Fuerza aplicada* distancia perpendicular (entre la línea de la fuerza y el eje de rotación)

$$\tau = r * F * sen\theta$$ Ecuación 5

Donde τ: torque en el eje en Nm

r: radio en m

En la Figura 24 se observa una fuerza aplicada en un punto de la circunferencia de radio r. Solo la componente de la fuerza que es perpendicular a la dirección del radio tiene efecto sobre el torque que puede hacer girar la circunferencia. Observe que si la fuerza es aplicada en la dirección del radio no daría lugar a torque, no hay movimiento.

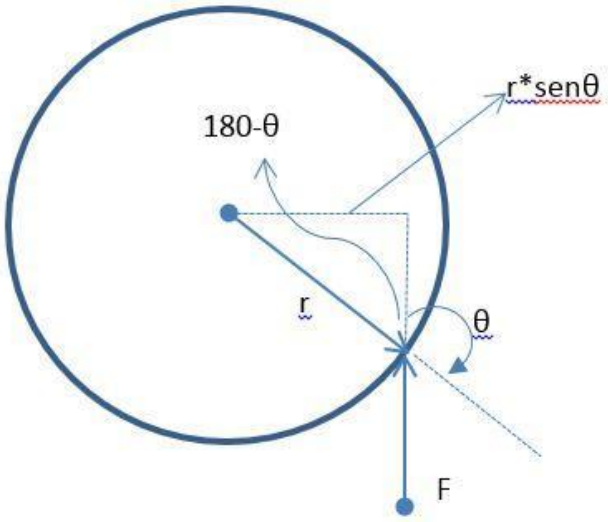


Figura 24. Efecto de la fuerza aplicada en una circunferencia que puede girar sobre su eje Fuente: (Mejía Arango, 2015)

Si el ángulo es de 90, entonces el torque se calcula como:

$$\tau = r * F * 1 \qquad \text{Ecuación 6}$$

Dónde: $F = m * a$

Siendo a la aceleración y m la masa en Kg

Como el recorrido es circular

$$a = \frac{dv}{\mathrm{dt}}$$

Donde $v = \frac{dL}{\mathrm{dt}}$

Siendo L el arco recorrido por la circunferencia.

$$L = r * \theta \qquad \text{Ecuación 7}$$

Entonces

$$v = \frac{d(r*\theta)}{\mathrm{dt}} = r * \frac{d\theta}{\mathrm{dt}} = r * w$$ Ecuación 8

$$a = \frac{d(r * w)}{\mathrm{dt}} = r * \frac{dw}{\mathrm{dt}} = r * \alpha$$

Entonces $\tau = r*F$ Ecuación 9

Trabajo y potencia

En un movimiento lineal el trabajo está dado por:

$$W = \int F*dx = F*x$$ Ecuación 10

Mientras que en el movimiento angular

$$W = \int \tau * d\theta = \tau * \theta$$ Ecuación 11

La potencia es la derivada con respecto al tiempo

Para movimiento lineal y angular se tienen las siguientes relaciones

$$P = \frac{dW}{\mathrm{dt}}$$ Ecuación 12

$$P = \frac{d(F*x)}{\mathrm{dt}} = F * \frac{dx}{\mathrm{dt}} = F * v$$ Ecuación 13

$$P = \frac{d(\tau*\theta)}{\mathrm{dt}} = \tau * \frac{d\theta}{\mathrm{dt}} = \tau * w$$ Ecuación 14

Aplicando estas ecuaciones al sistema a implementar se tiene:

Fuerza (peso a vencer) = 2452,5 Newton

El torque para un radio de 20 cm será:

τ = F*r = 2452,5 Newton* 0.2 m = 490,5 Nm

Velocidad angular de 35 rpm.

Con esto la potencia requerida por el motor será:

$$P = \tau * w$$

Donde τ es el torque en Nm y w es la velocidad angular en rad/s

$$P = 490{,}5 \text{ Nm} * 35 \frac{rev * 2 * \pi rad * 1min}{\text{min} * \text{rev} * 60\text{seg}}$$

$$P = 1795{,}94 \text{ W}$$

Redondeando:

$$P = 2000 \text{ W}$$

*Para una potencia estable se debe trabajar con una tensión trifásica.

CAPITULO 4

4.1 CONCLUSIONES

Al realizar el código de programación para el microcontrolador del sistema de control de posición, se obtuvieron más de 800 líneas de código, por tal situación se observó el mensaje de memoria ROM llena imposibilitando continuar la compilación del código, para solucionar esa situación, se tenía dos opciones: cambiar de microcontrolador con más memoria u optimizar el código, se optó por optimizar el código empleando sub rutinas logrando reducir a 583 líneas de código.

Para la realización del sistema de control de tiempo se emplearon microcontroladores, lo cual no sería factible si se tendrían que controlar muchas más que 9 posiciones de un estacionamiento, en tal caso se sugiere hacer ese control mediante un software para computadora.

Al realizar una investigación sobre los estacionamientos verticales se observó que en Uruguay existe una empresa que se dedica a la instalación de estacionamientos verticales denominada "Vertical Elevadores" la cual indica que un estacionamiento de este tipo resulta 3 veces más económico que un estacionamiento de suelo o subterráneo, ocupa menos espacio, y se puede generar un retorno de inversión de casi 800% al año, siendo un sistema factible en ciudades céntricas de nuestro país.

4.2 BIBLIOGRAFIA

Autor: Opinión, 29 de abril de 2012, Parqueos el gran problema, disponible en: www.opinion.com.bo

Autor: La Patria, 10 de diciembre de 2012, El problema del parqueo de vehículos, disponible en: www.lapatria.com.bo

Autor: Pagina 7, 12 de septiembre de 2017, Alcaldía inicia simulacro sobre uso de parqueos en Sopocachi, disponible en: www.paginasiete.com.bo

Autor: Milan Verle, 2010, PIC Microcontrollers – Programming in Basic, 1ra edicion, disponible en: www.mikroe.com

Autor: Ing. Roger Guachalla Narvaez, 2013, Guías de laboratorio de microprocesadores 1, disponible en: xoroger@yahoo.es

Autor: Proyectos PIC2010, Introducción PIC16F628A disponible en:www.google.com/site/proyectospic2010/PIC18F452/introduccion-pic16f628a-1

Autor: Wikipedia, la enciclopedia libre, 25 de nov 2016, Puente H (electrónica) disponible en: www.es.wikipedia.org/wiki/Puente_H_(electr%C3%B3nica)

Autor: Eduardo J. Carletti, 2007, Manejo de potencia para motores con el integrado L293D, disponible en:www.robots-argentina.com.ar/MotorCC_L293D.htm

Autor: Wikipedia la enciclopedia libre, 4 de octubre de 2017,Transistor Darlington, disponible en:www.en.wikipedia.org/wiki/ULN2003A

Autor: Wikipedia la enciclopedia libre, 14 de agosto de 2017, ULN2003A, disponible en:www.en.wikipedia.org/wiki/ULN2003A

Autor: Wikipedia, la enciclopedia libre, 22 octubre 2017, Motor eléctrico, disponible en:www.es.wikipedia.org/wiki/Motor_electrico

Autor: Gerard Moisés García, 2005, Motores con imanes permanentes, disponible en: www.monografias.com/trabajos100/motores-imanes-permanentes/motores-imanes-permanentes.shtml

Autor: Wikipedia la enciclopedia libre, 27 septiembre 2017. Motor paso a paso, disponible en:www.es.wikipedia.org/wiki/Motor_paso_a_paso

Autor: Wikipedia la enciclopedia libre, 26 septiembre 2017, Servomotor, disponible en:www.es.wikipedia.org/wiki/Servomotor

Autor: Junta de Andalucía, 2007, Motores de corriente alterna, disponible en:http://www.juntadeandalucia.es/averroes/centros-

tic/21700290/helvia/aula/archivos/repositorio/0/29/html/Motores_de_corriente_alterna.htm

Autor: Goñi Galarza, 2016, Novena edicion - FISICA GENERAL, paginas 153, 154, 228 y 229