

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD TECNICA
CARRERA: ELECTRONICA Y TELECOMUNICACIONES



NIVEL: LICENCIATURA
EXAMEN DE GRADO
TRABAJO DE APLICACIÓN

**“CONTROL AUTOMATIZADO PARA EL
ALINEAMIENTO DE ANTENAS PARABOLICAS
MEDIANTE EL PUERTO USB”**

Postulante: Rivelino Flores Molina

La Paz- Bolivia

Agosto - 2012

DEDICATORIA

A Dios nuestro creador, por darme salud y vida.

A mi madre Asunta que siempre fue la persona que me impulso para culminar mis estudios.

A mi papa Evaristo por la paciencia y el apoyo que siempre me brindo.

A mi hermano Jesús por orientarme y corregir los tropiezos durante mi formación académica.

A mi querido hermano, que desde el cielo me guía y me cuida, el presente trabajo es una dedicatoria muy especial para el.

A mis amistades que siempre confiaron en mí y me brindaron un apoyo y aliento.

AGRADECIMIENTO

*Zuiero agradecer a nuestra querida y prestigiosa
Universidad mayor de san Andrés por acogerme en sus
aulas.*

*A mis docentes que durante toda mi carrera
universitaria me transmitieron todos sus conocimientos y
por brindarme toda su paciencia.*

ÍNDICE GENERAL

Pág.

CAPITULO I.....	1
INTRODUCCION.....	1
1.1 RESUMEN.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	2
1.2.1 Identificacion del problema	2
1.2.2 Formulacion delm problema.....	2
1.3 JUSTIFICACION DEL TRABAJO	2
1.3.1 Justificacion Tecnica.....	2
1.3.2 Justificacion Economica.....	3
1.4 OBJETIVOS.....	3
1.4.1 Objetivo General	3
1.4.2 Objetivos Especificos.....	3
CAPITULO II.....	4
FUNDAMENTO TEORICO	4
2.1 SISTEMA Y CONTROL	4
2.1.1 Sistema	4
2.1.2 Sistema de Control	5
2.2 ANTENA PARABOLICA	5
2.2.1 Parametros de alineacion de una antena parabolica	5
2.2.1.1 Acimut.....	6
2.2.1.2 Elevacion	6
2.3 MICROCONTROLADORES	12
2.3.1 Introduccion a los Microcontroladores	12
2.3.2 Diferencia entre Microprocesador y Microcontrolador.....	12
2.3.3 Aplicación de los Microcontroladores.....	14
2.3.4 Arquitecturas Harvard y Von Neumann	15
2.3.5 Set de Instrucciones	16
2.3.6 Microcontroladores de Gama Alta PIC18F4550	17

2.3.7 Características fundamentales del PIC de Gama Alta	19
2.3.8 Diagrama de pines Disposición de 40 pines	20
2.3.9 Diagrama en Bloques del PIC18F4550	21
2.3.10 Organización de la Memoria	22
2.4 USB (Bus Universal Serial).....	22
2.4.1 Nociones Básicas de la Norma USB 2.0.....	22
2.4.2 Como Funciona.....	23
2.4.3 Controlador	24
2.4.4 periféricos	24
2.4.5 Diagrama de Capas.....	26
2.4.6 Cables y Conectores USB	27
2.5 PROGRAMA VISUAL BASIC 6.0.....	28
2.6 SERVOMOTORES UTILIZADOS EN EL PROYECTO	29
2.6.1 Componentes del Servomotor	30
2.6.2 Funcionamiento del Servomotor	31
2.6.3 Aplicaciones del Servomotor.....	33
2.6.4 Conexión de voltaje a un Servomotor	34
2.6.5 Control del Servomotor	35
CAPITULO III.....	36
INGENIERIA DEL PROYECTO	36
3.1 DESARROLLO DE LA INGENIERIA DEL PROYECTO.....	36
3.1.1 Movimiento del Servomotor con el programa Visual Basic	37
3.1.2 Armado del circuito mecánico del proyecto.....	37
3.1.3 Control del movimiento Horizontal	39
3.1.4 Control del movimiento Vertical	39
3.1.5 Diseño del software para alineamiento y control de la antena	39
3.1.6 Desarrollo software para el entorno Visual Basic.....	40
3.1.7 Desarrollo software para el Microcontrolador PIC18F4550	45
CAPITULO IV	50
COSTOS Y PRESUPUESTOS.....	50

4.1	Concepto de costos	50
4.2	¿Qué es el presupuesto.....	50
4.3	Elementos necesarios para la elaboracion del proyecto	51
4.4	Costo de componentes Hardware.....	52
4.5	Presupuesto para la implementacion del proyecto	53
4.6	Tiempo para la implementacion del proyecto.....	53
4.7	Costo de mano de obra	53
4.8	Costo total de implementacion.....	54
	CONCLUSIONES.....	55
	BIBLIOGRAFIA	

ÍNDICE DE FIGURAS

Pág.

Figuras del capítulo II

Figura N° 2.1 Sistema	4
Figura N° 2.2 Elevación y Acimut de una Antena Parabólica	7
Figura N° 2.3 Estructura de un sistema abierto basado en un microprocesador	13
Figura N° 2.4 El Microcontrolador	14
Figura N° 2.5 Arquitectura Von Neumann	15
Figura N° 2.6 Arquitectura harvar	16
Figura N° 2.7 Gama de Microcontroladores de microchip	18
Figura N° 2.8 Diagrama de pines del PIC 18F4550.....	20
Figura N° 2.9 Diagrama en Bloques del PIC18F4550	21
Figura N° 2.10 Estructura de Capas del bus USB	24
Figura N° 2.11 Posible esquema de conexión del bus USB	25
Figura N° 2.12 Dispositivo USB conectado a un PC.....	25
Figura N° 2.13 Capas del sistema de comunicaciones USB	26
Figura N° 2.14 Esquema del cable para USB.....	27
Figura N° 2.15 Forma física de un servomotor	29
Figura N° 2.16 partes de un servomotor.....	30
Figura N° 2.17 Tiempos de duración de los impulsos y dirección obtenida del actuador.....	33
Figura N° 2.18 Conexión de Voltaje a un servomotor	34

Figuras del Capítulo III

Figura N° 3.1 Diagrama en bloques del proyecto	36
Figura N° 3.2 Armado del circuito mecánico del proyecto	37
Figura N° 3.3 Sistema de engranaje.....	38
Figura N° 3.4 Plataforma de Control Visual	44
Figura N° 3.5 simulación en Isis de proteus	49

CAPITULO I

INTRODUCCION

1.1 RESUMEN

En el actual mundo de la tecnología, la tendencia a la innovación es cada vez más frecuente, convirtiéndose en una característica del hombre postmoderno el buscar métodos de trabajo cada vez más simples.

Es por este motivo que este trabajo de aplicación se basa en automatizar los mecanismos de alineamiento de antenas parabólicas como ser el acimut y la elevación, que generalmente se lo opera manualmente.

Para el desarrollo de este propósito ha sido necesario diseñar una aplicación en Visual Basic, la cual nos servirá de plataforma para controlar los parámetros de alineación de una antena parabólica.

El centro de control de la antena parabólica esta implementado en base a un microcontrolador de gama alta PIC 18F4550, que forma parte del proceso principal de todos los eventos de control, tal proceso se lo realiza utilizando la interfaz USB (Bus Serial Universal), que estará conectado al computador personal, del cual realizaremos nuestras acciones de alineamiento.

Para el accionamiento mecánico automatizado de los parámetros de alineamiento se han utilizado motores de corriente continua que harán posible el movimiento circular de los ejes de ajuste de acimut y elevación.

El conjunto de estos subsistemas nos permitirá poder controlar los parámetros de alineamiento de una antena parabólica para conseguir un óptimo nivel de recepción de señal.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 IDENTIFICACIÓN DEL PROBLEMA

Debido al esfuerzo que requiere la manipulación manual de una antena parabólica por el peso que presenta la estructura física además por la ubicación externa que requiere su instalación, esto da lugar a un problema para la operación del personal técnico.

De esta manera, con mucha frecuencia se evidencian que para el simple ajuste de elevación y acimut que requiere una antena parabólica, es necesario realizar mucho esfuerzo en la hora de la migración satelital. A esto se suma el hecho de que para el ajuste de alineamiento de la antena el personal técnico debe dedicar un tiempo extra en este trabajo.

1.2.2 FORMULACIÓN DEL PROBLEMA

La situación descrita configura un *problema de investigación* que debe ser atendido desde un punto de vista técnico a fin de identificar las medidas que permitan revertir la situación problemática detectada, para cuyo efecto se formula la interrogante que, de hecho, se constituye en el problema de investigación: ¿Cómo contribuir a un funcionamiento más versátil y eficiente del ajuste y alineamiento de una antena parabólica?

1.3 JUSTIFICACIÓN DEL TRABAJO

1.3.1 JUSTIFICACIÓN TÉCNICA

En la actualidad muchas empresas de telecomunicaciones requieren que su personal técnico se dirija al lugar para dar solución a sus problemas de alineamiento de antenas parabólicas, en muchos casos teniendo que intervenir más de un técnico en la operación, por cuya razón, se considera urgente efectuar un análisis técnico del funcionamiento automatizado del sistema de alineamiento de antenas a fin de obtener una adecuada instalación y utilización del sistema.

1.3.2 JUSTIFICACIÓN ECONÓMICA

Las pérdidas económicas originadas no sólo por las demoras en el trabajo por efecto de ajustes manuales sino también por el requerimiento de más de un personal capacitado, dan lugar a una pérdida económica directa para una empresa de telecomunicaciones.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e implementar un sistema de control automatizado para el alineamiento de antenas parabólicas mediante el puerto USB.

1.4.2 OBJETIVOS ESPECÍFICOS

- Diseñar el software para la ejecución de este proyecto.
- Desarrollar un mecanismo que permita modificar los parámetros de alineamiento de las antenas parabólicas.
- Desarrollar una aplicación en Visual Basic para acceder al mecanismo de alineamiento desde una PC.

CAPITULO II

FUNDAMENTO TEORICO

2.1 SISTEMA Y CONTROL

2.1.1 SISTEMA

Es un ensamblaje de componentes que proporcionan acciones interrelacionadas entre sí, los cuales se caracterizan por poseer parámetros inherentes que los definen y por mostrar condiciones físicas asociadas. A los parámetros de cada elemento se les denomina parámetros del sistema y las condiciones físicas de cada componente cambiantes con el tiempo determinan el estado del sistema en cada momento y se les denominan variables del sistema.

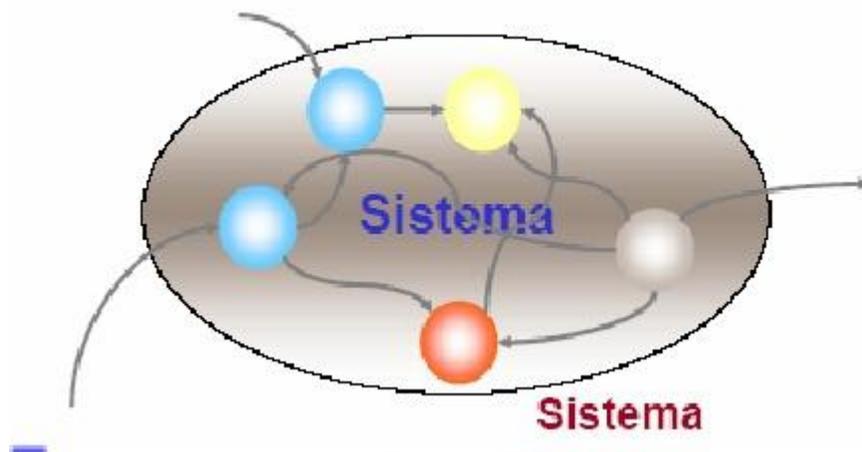


Figura N° 2.1 Sistema

Es potencialmente aplicable a un conjunto diverso de fenómenos. Los sistemas se definen en todas las áreas. En control lo analizaremos en el contexto de sistemas físicos que se describen por leyes de las ciencias físicas.

2.1.2 SISTEMA DE CONTROL

Los sistemas de control según la Teoría Cibernética se aplican en esencia para los organismos vivos, las máquinas, los procesos y las organizaciones. Estos sistemas fueron relacionados por primera vez en 1948 por Norbert Wiener en su obra *Cibernética y sociedad con aplicación en la teoría de los mecanismos de control*. Un sistema de control está definido como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado. Los controladores son sistemas eléctricos o electrónicos que están permanentemente capturando señales de estado del sistema bajo su control y que al detectar una desviación de los parámetros pre-establecidos del funcionamiento normal del sistema, actúan mediante sensores controladores y actuadores.

2.2 ANTENA PARABÓLICA

La antena parabólica es un tipo de antena que se caracteriza por llevar un reflector parabólico. Su nombre proviene de la similitud a la *parábola* generada al cortar un cono recto con un plano paralelo a la directriz.

Las antenas de reflector parabólico proporcionan ganancias y directividades extremadamente altas, y son muy utilizadas en los enlaces de comunicaciones por radio y satélite. Una antena parabólica consiste en dos partes principales: un reflector parabólico y el elemento activo llamado mecanismo de alimentación, la antena irradia ondas electromagnéticas hacia el reflector. El reflector es un dispositivo pasivo, que tan solo refleja la energía que le llega del mecanismo de alimentación.

2.2.1 PARÁMETROS DE ALINEACIÓN DE UNA ANTENA PARABÓLICA

Los parámetros de alineación de una antena parabólica son el acimut y la elevación.

2.2.1.1 ACIMUT

En acimut es el ángulo o longitud de arco medido sobre el horizonte celeste que forman el punto cardinal Norte y la proyección vertical del astro sobre el horizonte del observador situado en alguna latitud. Se mide en grados desde el punto cardinal Norte en el sentido de las agujas del reloj, o sea Norte-Este-Sur-Oeste.

Es una de las dos coordenadas horizontales, siendo la otra la altura. La altura y el acimut son coordenadas que dependen de la posición del observador. Es decir que en un mismo momento, un astro es visto bajo diferentes coordenadas horizontales por diferentes observadores situados en puntos diferentes de la Tierra. Esto significa que dichas coordenadas son locales.

- El valor del Acimut indicará el punto exacto en el que debemos fijar la antena en el plano horizontal.
- Este ángulo se mide desde el norte geográfico en sentido de las agujas del reloj.
- Hay que tener en cuenta que el polo norte geográfico, utilizado como referencia en todos los mapas, es consecuencia de la división imaginaria del globo terráqueo en diferentes gajos (husos) a través de los meridianos. El punto de intersección de todos ellos da lugar a los polos Norte y Sur, por los que pasa el eje de giro de la Tierra.
- El polo norte magnético es el punto de la superficie terrestre que atrae el extremo rojo de la aguja de la brújula.

2.2.1.2 ELEVACIÓN

El ángulo de elevación nos indicará la inclinación que le debemos dar a la antena con respecto al plano vertical para orientarla hacia el satélite.

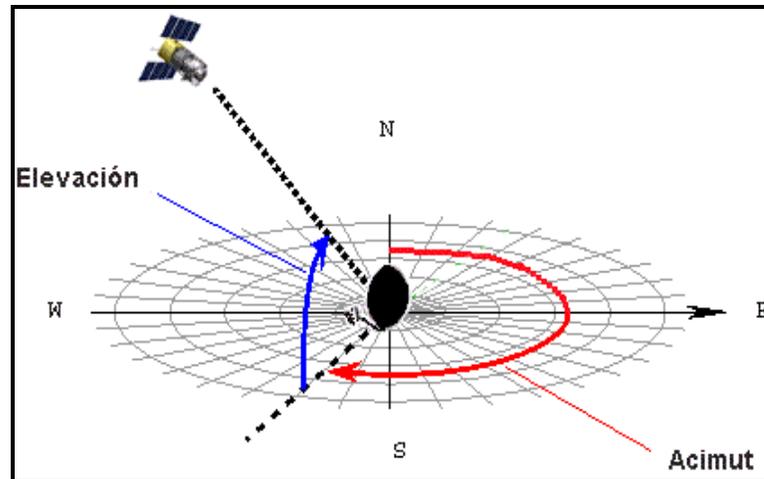


Figura N° 2.2 Elevación y Acimut de una Antena Parabólica

Ganancia de la antena.- La ganancia de la antena es un factor muy importante en cualquier tipo de comunicaciones de radio, e indispensable para el cálculo del enlace satelital. La definición de ganancia de la antena, se refiere a la relación que existe entre la potencia radiada por una antena isotrópica, a una distancia y la densidad de potencia que radiaría la misma antena pero en una sola dirección, con igual cantidad de potencia entregada.

$$A_t(dB) = 10 \log \left(n \left(\frac{\pi D f}{c} \right)^2 \right)$$

A_t = Ganancia de la antena (dB)

n = Eficiencia de la antena (a dimensional)

D = Diámetro de la antena (metros)

f = Frecuencia de transmisión (GHz)

c = Velocidad de la luz (3×10^8 Km/s)

Temperatura equivalente de ruido.- En los sistemas de comunicaciones existe un parámetro que se conoce como índice de ruido, éste es producido por todos los objetos cuya temperatura esté por encima del cero absoluto.

Densidad de ruido.- La densidad de ruido se conoce como la cantidad de potencia de ruido normalizado a un ancho de banda de 1Hz, siendo ésta la relación entre la potencia de ruido generada por un amplificador.

Perdidas por reducción.- Los amplificadores de potencia que se utilizan en las estaciones terrestres, así como los tubos de onda viajera, que se usan en los satélites, son dispositivos no lineales; la ganancia de éstos depende de la potencia de entrada de la señal. Esto nos indica la característica de la potencia de salida en función de la potencia de entrada para un amplificador de potencia (HPA).

Potencia isotrópica radiada efectiva (PIRE).- La potencia isotrópica radiada efectiva PIRE, es una medida que indica la fuerza con que una señal es transmitida hacia un satélite o hacia una estación terrestre. Esto nos indica que la potencia isotrópica radiada efectiva es la combinación de la potencia del transmisor con la ganancia de la antena.

$$PIRE(dB) = 10\text{Log}(PtAt)$$

PIRE = Potencia isotrópica radiada efectiva (dB)

Pt = Potencia de entrada a la antena (Watts)

At = Ganancia de la antena de transmisión (dB)

Perdidas por propagación.- El cálculo en el enlace de comunicaciones satelitales es necesario para determinar el balance de pérdidas y ganancias en potencia de la señal radiada. El diseño correcto de un enlace de radio asegura la recepción de una señal de buena calidad, evitando así el desperdicio de recursos.

Es por esto que se necesitan conocer todos los aspectos que afectan a las señales de radio, desde que se transmiten hasta que se reciben. Por este motivo las pérdidas de propagación juegan un papel muy importante en el diseño de un enlace satelital. Las pérdidas por propagación en el espacio libre, se refieren a

que la energía se reparte mientras la señal se propaga alejándose de la fuente, por lo que se produce una menor densidad de potencia a mayor distancia, la siguiente ecuación muestra las pérdidas por propagación en el espacio libre.

$$L_p = 20 \text{Log} \frac{4\pi Df}{c}$$

Densidad de potencia.- Para determinar la densidad de flujo a la distancia del satélite se aplica en la siguiente ecuación, tomando en cuenta que r representa el rango o la distancia del radio enlace.

$$C = 10 \text{Log} \left(\frac{P_{tx} A_{tx}}{4r^2 \pi} \right)$$

C = Densidad de flujo (dBW/m²)

P_{tx} = Potencia de transmisión

A_{tx} = Ganancia de la antena transmisora

r = Rango del radioenlace (Km)

Potencia en el receptor.- Para obtener la potencia en el receptor de un enlace de radio, se requiere de la ganancia de la antena receptora, esto es con el fin de que la densidad de flujo de la potencia se convierta en potencia eléctrica.

Relación de ganancia a temperatura equivalente de ruido.- La relación de ganancia a temperatura equivalente de ruido G/T_e , es una cifra de mérito que sirve para demostrar la calidad de recepción de un satélite o una estación terrena. En la siguiente ecuación se considera como la relación entre la ganancia de la antena receptora y la temperatura equivalente de ruido.

$$\frac{G}{T_e} (dB) = 10 \text{Log} \left(\frac{A_{rx}}{T_e} \right)$$

G/T_e = Ganancia a temperatura equivalente de ruido (dBK^{-1}).

A_{rx} = Ganancia de la antena receptora (a dimensional).

T_e = Temperatura equivalente de ruido ($^{\circ}\text{K}$).

Relación de portadora a densidad de ruido.- La relación de portadora a densidad de ruido C/N_0 es la relación de la potencia de portadora de banda ancha (potencia combinada de la portadora y sus bandas laterales asociadas) entre la densidad de ruido presente en un ancho de banda de 1Hz.

Relación de la portadora a señal a ruido.- Para realizar el diseño correcto de un enlace satelital se requieren de todos los factores antes mencionados y, para poder calcular la cantidad de potencia que se transmite en una comunicación satelital, se utiliza la relación de la portadora a la señal de ruido C/N . Esta relación se encarga de reunir todos los tipos de pérdidas y ganancias mostrando la eficiencia de un enlace.

Resistencia de radiación.- No toda la potencia suministrada a la antena se irradia. Parte de ella se convierte en calor y se disipa. La resistencia de radiación es un poco "irreal", en cuanto a que no puede ser medida directamente. La resistencia de radiación es una resistencia de la antena en ca y es igual a la relación de la potencia radiada por la antena al cuadrado de la corriente en su punto de alimentación. Matemáticamente, la resistencia de radiación es

$$R_r = \frac{P}{i^2}$$

Donde:

R_r = Resistencia de radiación (ohms)

P = Potencia radiada por la antena (Watts)

i = Corriente de la antena en el punto de alimentación (Amperes)

La resistencia de radiación es la resistencia que, si reemplazara la antena, disiparía exactamente la misma cantidad de potencia de la que irradia la antena

La directividad.- es la razón entre la densidad de potencia en la dirección de máxima radiación (S) y la que radiaría un radiador isotrópico (S_1) con la misma potencia radiada.

$$D = \frac{S}{S_1}$$

D = directividad de la antena

S = densidad de potencia en la dirección de máxima radiación [Wm^2]

S_1 = densidad de potencia isotrópica [Wm^2]

Eficiencia.- Relación entre la potencia radiada y la potencia entregada a la antena.

También se puede definir como la relación entre ganancia y directividad.

$$\eta = \frac{P(r)}{P(in)} = \frac{G}{D}$$

Resistencia de pérdidas.- Representa la potencia disipada por efecto Joule, ya sea en la superficie de los conductores o en elementos disipativos propios del diseño de la antena.

Impedancia de entrada.- La resistencia de antena es la suma de la resistencia de radiación y la resistencia de pérdidas, la resistencia de la antena R_A tiene dos componentes: una, causante de las pérdidas por calentamiento, designada como *resistencia de pérdidas*, R_p y otra, asociada con el proceso de radiación de la energía electromagnética, designada como *resistencia de radiación*, R_r . Es en esta última en la que se considera que se absorbe la potencia cuando la antena se usa como receptora y la responsable de la radiación cuando la antena es transmisora. La resistencia de radiación es una propiedad de la antena y no una

resistencia convencional que pueda ser medida con un óhmetro. De acuerdo a esto:

$$RA = Rp + Rr$$

En el segmento terrestre podemos dividirla de la siguiente manera: lado maestro (unidad interior, unidad exterior) y el lado remoto:

2.3 MICROCONTROLADORES

2.3.1 INTRODUCCIÓN A LOS MICROCONTROLADORES

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricaremos y usamos los seres humanos.

2.3.2 DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador, de un computador. La UCP está formada por la Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta.

Las patitas de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la memoria y los módulos de entrada/salida y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine.

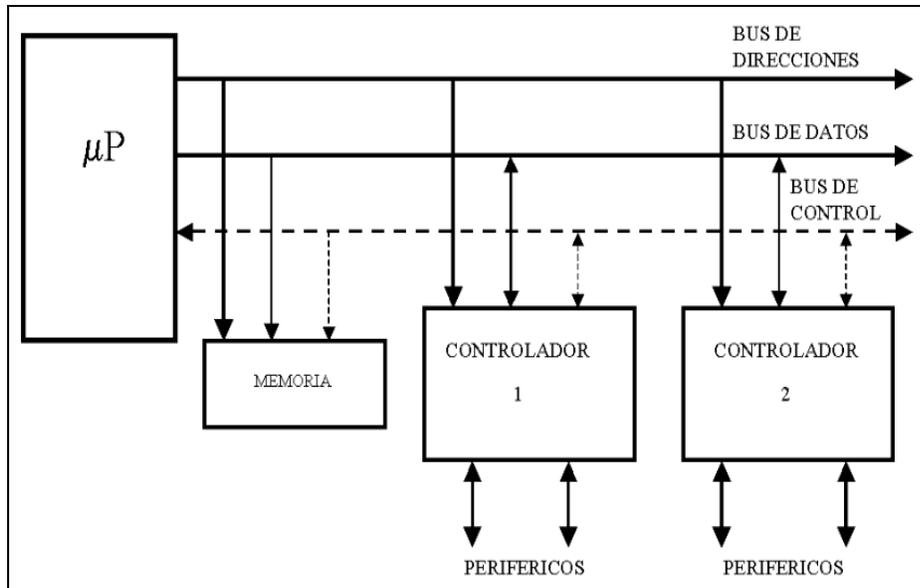


Figura N° 2.3 Estructura de un sistema abierto basado en un microprocesador

La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación. Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de entrada/salida, la cantidad y potencia de los elementos auxiliares y la velocidad de funcionamiento. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

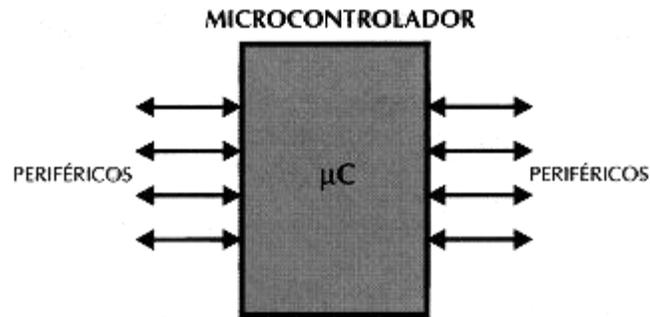


Figura Nº 2.4 El Microcontrolador

El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

2.3.3 APLICACIONES DE LOS MICROCONTROLADORES

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

2.3.4 ARQUITECTURAS HARVARD Y VON NEUMANN

La arquitectura de Von Neumann es caracterizada por los procesadores que poseen el mismo dispositivo de almacenamiento en donde coexisten tanto datos como instrucciones.

Las instrucciones y los datos son almacenados en el mismo formato dentro de la misma memoria, lo que quiere decir que las instrucciones son datos y comparten el mismo bus de datos en su transmisión. En principio un programa en ejecución se puede modificar a sí mismo y en práctica usualmente esto significa que un error ocurrió. La ventaja de este tipo de arquitectura es que es eficiente en la utilización de la memoria pero su desventaja es que los tipos de datos no pueden ser reconocidos sin ambigüedad.

La arquitectura se compone de las siguientes unidades o bloques:

Unidad de memoria principal

Unidad aritmético-lógica.

Unidad de control.

Unidad de entrada y salida.

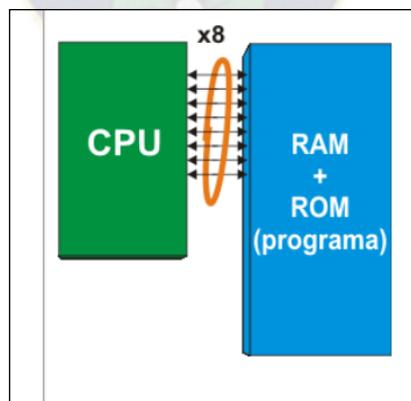


Figura Nº 2.5 Arquitectura Von Neumann

La arquitectura Harvard esencialmente posee una diferencia muy marcada con respecto a la arquitectura Von Neumann, la cual es que los procesadores utilizan dispositivos de

almacenamiento físicamente separados para el manejo de instrucciones y datos. El acceso a cada espacio de memoria puede llegar a ser mediante buses distintos, es decir, existe un bus de datos y un bus de instrucciones; pero lo más habitual es que exista un único bus de direcciones, en cuyo caso debe existir alguna señal de control que permita diferenciar a que espacio de memoria hace referencia.

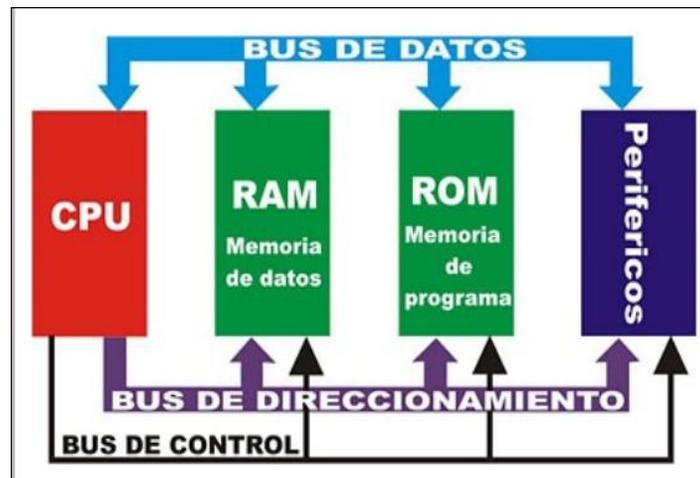


Figura N° 2.6 Arquitectura Harvard

Esta arquitectura también se compone de las unidades o bloques de la arquitectura Von Neumann a diferencia del manejo de memoria con instrucciones y datos. La ventaja de la estructuración de este tipo de arquitectura es que los tipos de datos pueden ser reconocidos sin ningún problema.

2.3.5 SET DE INSTRUCCIONES

CISC

La abreviación se refiere a de Juego de Instrucciones Complejo (Complex Instruction Set) y es una arquitectura que realiza el manejo de instrucciones considerando el manejo de los registros a memoria y con 2 operandos en la Unidad aritmética-lógica. Otra característica de este tipo de arquitectura es que maneja un conjunto muy amplio de instrucciones y necesita una codificación densa que a su vez requerirá muchos ciclos de reloj para ejecutarse. También este tipo de Set de instrucciones posee como desventaja que al realizar una

operación o instrucción, el resultado de esta destruye a uno de los operandos; además el tamaño de sus instrucciones es variable lo que hace que la duración de ejecución de instrucciones sea también variable. En cuanto a su programabilidad, las características antes mencionadas provocan que esta sea difícil de segmentar para su ejecución.

RISC

Esta arquitectura significa Juego de Instrucciones Reducido lo que significa que la ejecución de sus programas requiere pocos ciclos de reloj para su finalización. Estas instrucciones pueden ser utilizadas más eficientemente que la de los procesadores CISC con el diseño de software apropiado, resultando en operaciones más rápidas aprovechando el tamaño fijo de sus instrucciones. También posee una simplificación para la asignación de registros que no necesita acceder a la memoria.

Los tipos de instrucciones más frecuentes para la elaboración de las arquitecturas anteriores son:

- ❖ Instrucciones de carga y almacenamiento desde registros de la CPU
- ❖ Operaciones de transferencia entre registros
- ❖ Operaciones Aritméticas y lógicas
- ❖ Operaciones condicionales e incondicionales
- ❖ Operaciones de desplazamiento y rotación

2.3.6 MICRO CONTROLADORES DE GAMA ALTA PIC18F4550

Los microcontroladores PIC18F4550 se presentaron como una nueva gama alta de microcontroladores con procesador RISC avanzada de 8 bits para la memoria de datos y 16 bits para la memoria de programa. Con una arquitectura Harvard, que incluye los últimos avances como es el puerto USB (Bus Serial Universal) y la mayoría de los periféricos y recursos incluidos. Siendo el objetivo de potenciar

miles de aplicaciones centradas en las áreas de automoción, comunicaciones, ofimáticas, productos de gran consumo y control industrial.

Dentro de la gama de microcontroladores de la compañía microchip, la serie PIC18F que incluye el microcontrolador elegido, corresponde a la zona más alta dentro de la arquitectura de 8 bits de datos, denominada también gama mejorada.

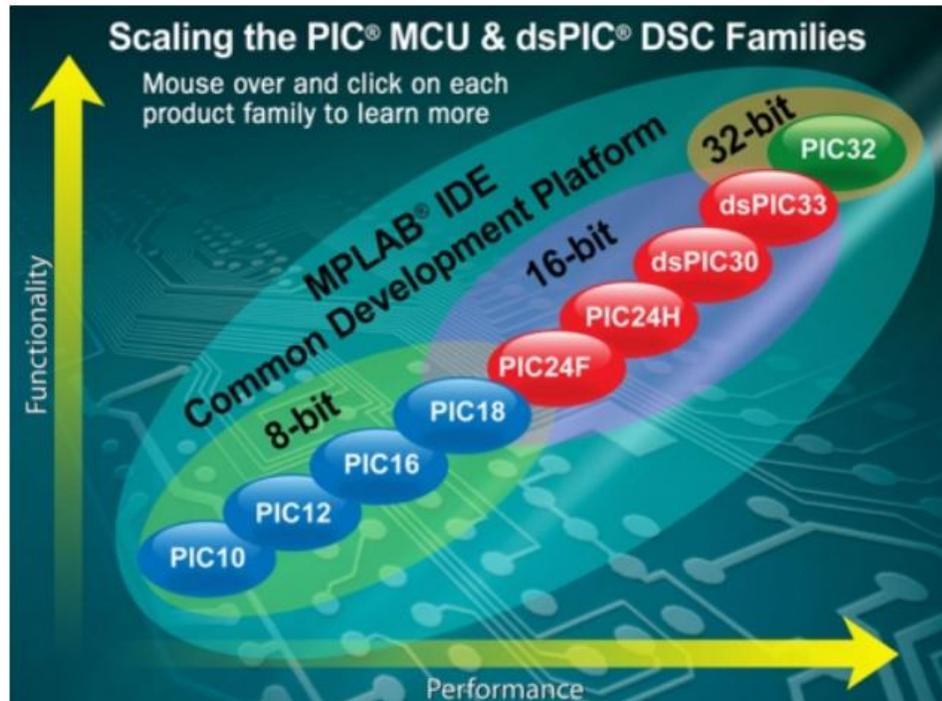


Figura Nº 2.7 Gama de Microcontroladores de microchip

2.3.7 CARACTERÍSTICAS FUNDAMENTALES DE PIC GAMA ALTA

Microcontroladores PIC18F2455, PIC18F2550, PIC18F4455 Y PIC18F4550

CARACTERISTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F4450
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24.576	32.768	24.576	32.768
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de E/S	24	24	35	35
Temporizadores	4	4	4	4
Módulos de Comparación/Captura/PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura/PWM mejorado (ECCP)	0	0	1	1
Canales de Comunicación Serie	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Canales de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de Instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)
Encapsulados	PDIP 28 pines SOIC 28 pines	PDIP 28 pines SOIC 28 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines

Tabla Nº 1 Características Fundamentales de PIC Gama Alta

Dispone de 75 instrucciones generales más 8 instrucciones indexadas, en comparativa a las 35 instrucciones de la gama media, también dispone de tres apuntadores FSR y un apuntador especial para gestión de tablas. Estas características hace óptimo el uso de compiladores C, Basic y Pascal para programar en lenguajes de programación de alto nivel.

Otras de las características fundamentales que hace óptimo a este microcontrolador son:

- ❖ Un multiplicador implementado en hardware.

- ❖ Frecuencia de operación de 48 MHz.
- ❖ Oscilador separado para la CPU y el puerto USB.
- ❖ Oscilador interno y entradas para circuitos de oscilaciones externas
- ❖ Modos de bajo consumo energético seleccionables.

2.3.8 Diagrama de pines: disposición de 40 pines y empaquetadura DIP

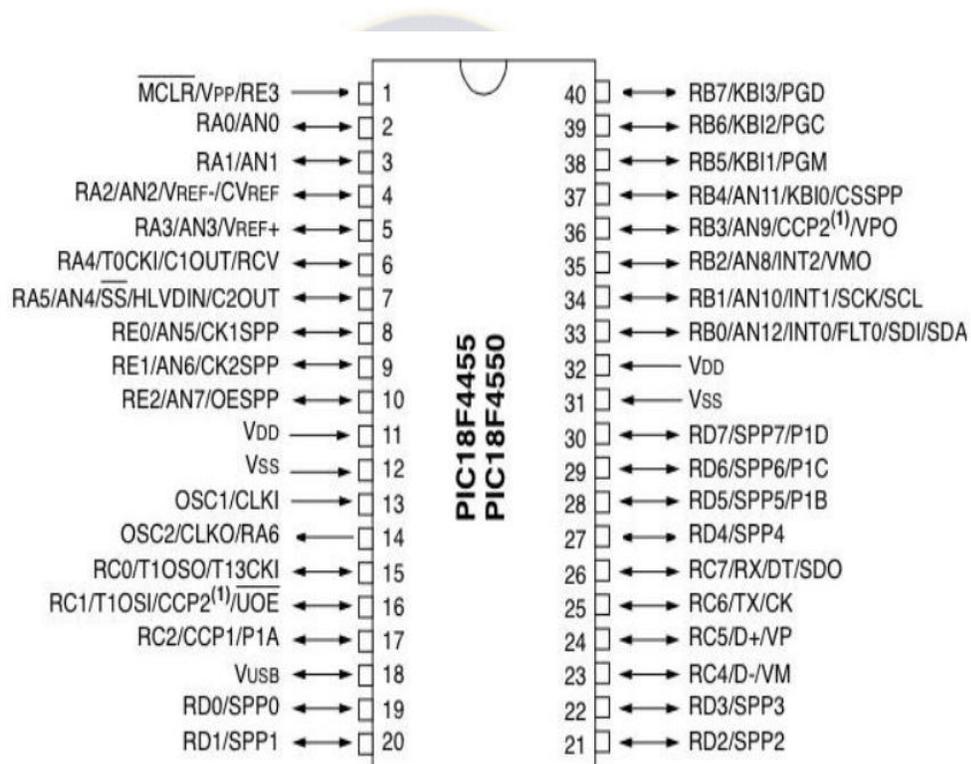


Figura N° 2.8 Diagrama de pines del PIC 18F4550

2.3.9 DIAGRAMA EN BLOQUES DEL PIC18F4550

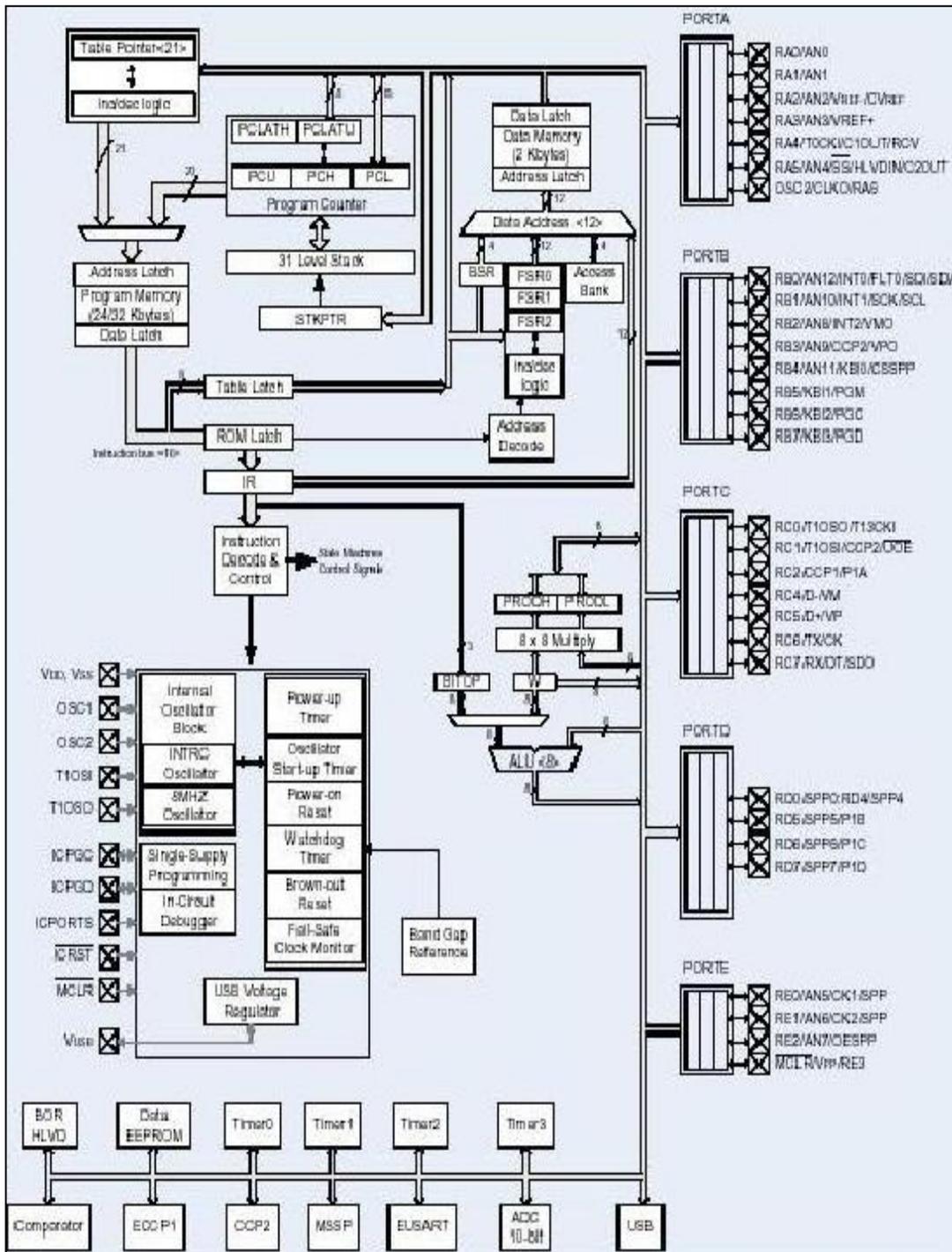


Figura N° 2.9 Diagrama en bloques del PIC18F4550

2.3.10 ORGANIZACIÓN DE LA MEMORIA

El microcontrolador PIC 18F4550 dispone de las siguientes memorias

- ❖ Memoria de programa: memoria Flash interna de 32.768 bytes, almacena instrucciones y constantes / datos, puede ser escrita/leída mediante un programador externo o durante la ejecución de un programa mediante punteros.
- ❖ Memoria RAM de datos: memoria SRAM interna de 2048 bytes en la que están incluidos los registros de función especial, almacena datos de forma temporal durante la ejecución del programa, puede ser escrita /leída en tiempo de ejecución mediante diversas instrucciones.
- ❖ Memoria EEPROM de datos: memoria no volátil de 256 bytes, almacena datos que deben conservar aun en ausencia de tensión de alimentación, puede ser escrita/leída en tiempo de ejecución a través de registros.
- ❖ Pila: bloque de 31 palabras de 21 bits, almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina.

2.4 USB (Bus Universal Serial)

2.4.1 NOCIONES BÁSICAS DE LA NORMA USB 2.0

Para entender el funcionamiento del USB, es necesario estudiar la especificación de este bus. Los fabricantes involucrados en la norma crearon un documento que constituye la referencia. Aunque en estos momentos se está creando la versión 3.0 del bus, en el presente capítulo se va a tratar la versión 2.0 ya que es la más rápida y es la que posee la familia de PICs 18F.

Ya que la norma es muy extensa, aquí se han traducido las partes que se han creído más importantes para una comprensión fácil y rápida de este bus.

Uno de los motivos por los que el USB se haya extendido tan velozmente es que se ha creado mediante un convenio de grandes empresas del sector informático: Compac, Hewlett-Packard, Intel, Lucent, Microsoft; NEC y Philips. Adoptando las mejores características de los diferentes tipos de comunicación de cada empresa

2.4.2 COMO FUNCIONA

Trabaja como interfaz para transmisión de datos y distribución de energía, que ha sido introducida en el mercado de PC's y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Esta interfaz de 4 hilos, 12 Mbps y "plug and play", distribuye 5V para alimentación, transmite datos y está siendo adoptada rápidamente por la industria informática.

Es un bus basado en el paso de un testigo, semejante a otros buses como los de las redes locales en anillo con paso de testigo y las redes FDDI. El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran.

Emplea una topología de estrellas apiladas que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, está el controlador anfitrión o host que controla todo el tráfico que circula por el bus. Esta topología permite a muchos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran más abajo en la pirámide sufran retardo. A diferencia de otras arquitecturas, USB no es un bus de almacenamiento y envío, de forma que no se produce retardo en el envío de un paquete de datos hacia capas inferiores.

El sistema de bus serie universal USB consta de tres componentes:

- Controlador
- Hubs o Concentradores

- Periféricos

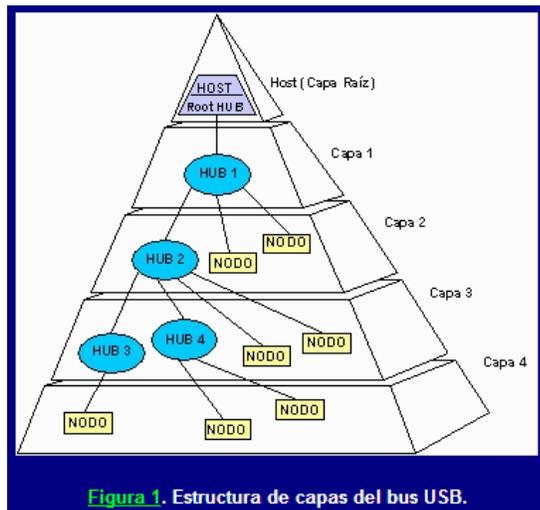


Figura Nº 2.10 Estructura de capas del bus USB

2.4.3 CONTROLADOR

Reside dentro del PC y es responsable de las comunicaciones entre los periféricos USB y la CPU del PC. Es también responsable de la admisión de los periféricos dentro del bus, tanto si se detecta una conexión como una desconexión. Para cada periférico añadido, el controlador determina su tipo y le asigna una dirección lógica para utilizarla siempre en las comunicaciones con el mismo. Si se producen errores durante la conexión, el controlador lo comunica a la CPU, que, a su vez, lo transmite al usuario. Una vez se ha producido la conexión correctamente, el controlador asigna al periférico los recursos del sistema que éste precise para su funcionamiento.

El controlador también es responsable del control de flujo de datos entre el periférico y la CPU.

2.4.4 PERIFÉRICOS

USB soporta periféricos de baja y media velocidad. Empleando dos velocidades para la transmisión de datos de 1.5 y 12 Mbps se consigue una utilización más

eficiente de sus recursos. Los periféricos de baja velocidad tales como teclados, ratones, joysticks, y otros periféricos para juegos, no requieren 12 Mbps. Empleando para ellos 1,5 Mbps, se puede dedicar más recursos del sistema a periféricos tales como monitores, impresoras, módems, scanner, equipos de audio que precisan de velocidades más altas para transmitir mayor volumen de datos o datos cuya dependencia temporal es más estricta .

En las figuras 3 y 4 se puede ver cómo los hubs proporcionan conectividad a toda una serie de dispositivos periféricos

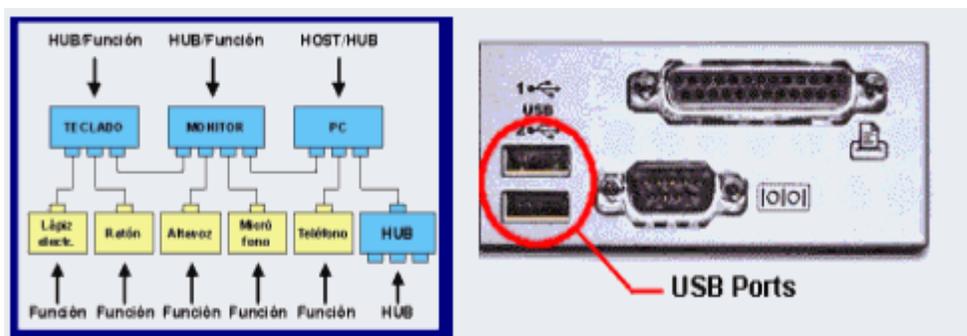


Figura Nº 2.11 Posible esquema de conexión del bus USB

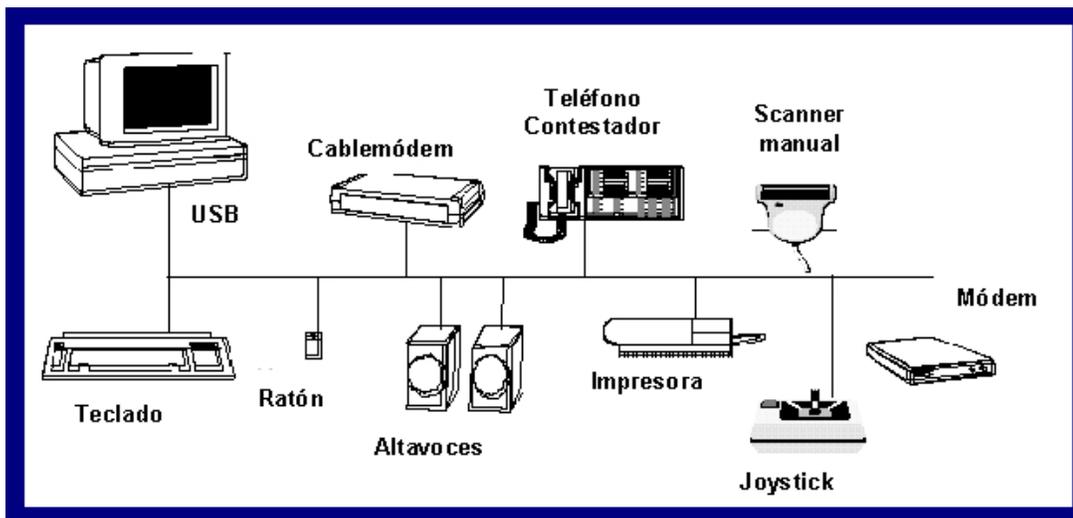


Figura Nº 2.12 Dispositivo USB conectado a un PC

2.4.5 DIAGRAMA DE CAPAS

En el diagrama de capas de la figura 5 podemos ver cómo fluye la información entre las diferentes capas a nivel real y a nivel lógico .

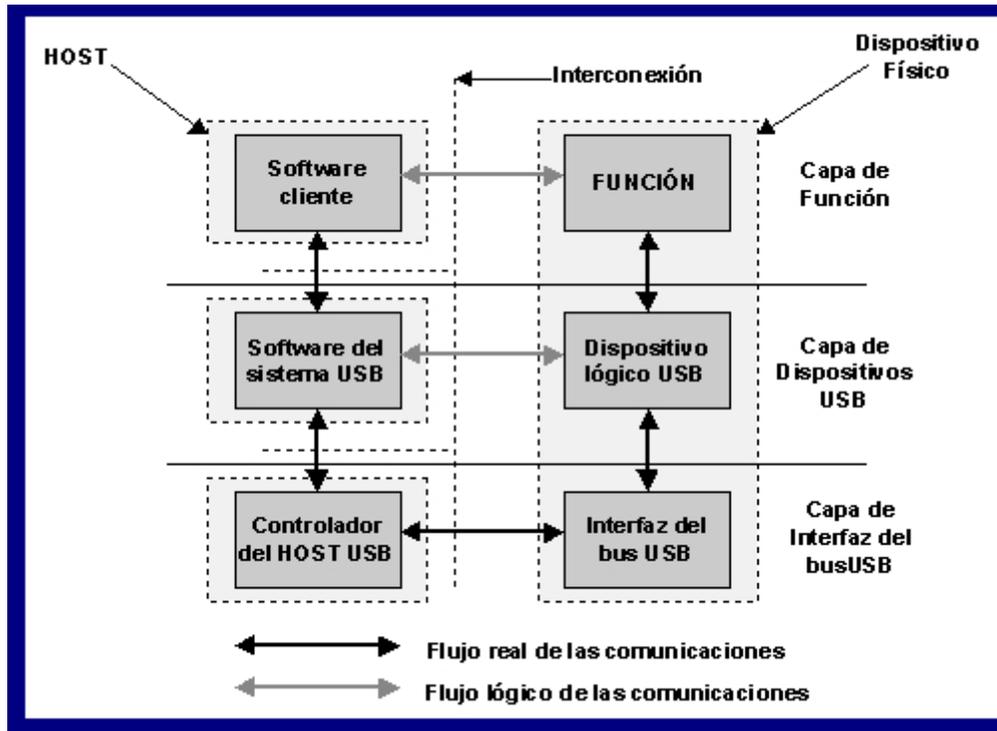


Figura Nº 2.13 Capas del sistema de comunicaciones USB

En dicha figura está materializada la conexión entre el controlador anfitrión o host y un dispositivo o periférico. Este está constituido por hardware al final de un cable USB y realiza alguna función útil para el usuario.

El software cliente se ejecuta en el host y corresponde a un dispositivo USB; se suministra con el sistema operativo o con el dispositivo USB. El software del sistema USB, es el que soporta USB en un determinado sistema operativo y se suministra con el sistema operativo independientemente de los dispositivos USB o del software cliente.

El controlador anfitrión USB está constituido por el hardware y el software que permite a los dispositivos USB ser conectados al anfitrión. Como se muestra en la figura 3, la conexión entre un host y un dispositivo requiere la interacción entre las capas. La capa de interfaz de bus USB proporciona la conexión física entre el host y el dispositivo. La capa de dispositivo USB es la que permite que el software del sistema USB realice operaciones genéricas USB con el dispositivo.

La capa de función proporciona capacidades adicionales al host vía una adecuada capa de software cliente. Las capas de función y dispositivos USB tienen cada una de ellas una visión de la comunicación lógica dentro de su nivel, aunque la comunicación entre ellas se hace realmente por la capa de interfaz de bus USB.

2.4.6 CABLES Y CONECTORES USB

USB transfiere señales y energía a los periféricos utilizando un cable de 4 hilos, apantallado para transmisiones a 12 Mbps y no apantallado para transmisiones a 1.5 Mbps. En la figura 6 se muestra un esquema del cable, con dos conductores para alimentación y los otros dos para señal, debiendo estos últimos ser trenzados o no según la velocidad de transmisión.

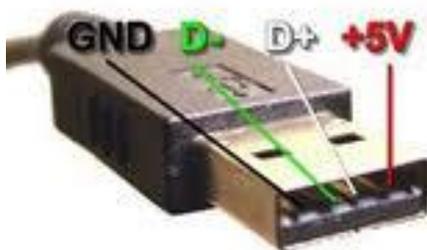


Figura Nº 2.14 Esquema del cable para USB

El calibre de los conductores destinados a alimentación de los periféricos varía desde 20 a 26 AWG, mientras que el de los conductores de señal es de 28 AWG. La longitud máxima de los cables es de 5 metros.

Por lo que respecta a los conectores hay que decir que son del tipo ficha (o conector) y receptáculo, y son de dos tipos: serie A y serie B . Los primeros presentan las cuatro patillas correspondientes a los cuatro conductores alineadas en un plano. El color recomendado es blanco sucio y los receptáculos se presentan en cuatro variantes: vertical, en ángulo recto, panel y apilado en ángulo recto así como para montaje pasa muro. Se emplean en aquellos dispositivos en los que el cable externo, está permanentemente unido a los mismos, tales como teclados, ratones, y hubs o concentradores.

Los conectores de la serie B presentan los contactos distribuidos en dos planos paralelos, dos en cada plano, y se emplean en los dispositivos que deban tener un receptáculo al que poder conectar un cable USB. Por ejemplo impresoras, scanner, y módems.

2.5 PROGRAMA VISUAL BASIC 6.0

Este programa se utilizará para los gráficos y controles es un lenguaje de programación orientado a objetos creado por la Microsoft. Este lenguaje incorpora todas las herramientas necesarias para la creación de cualquier aplicación para Windows. Con este lenguaje se puede crear desde una simple calculadora hasta un procesador de texto de la talla de Word, o una hoja de cálculo como Excel, o bien, cualquier aplicación que se le ocurra al programador. Se origina del lenguaje de programación Basic. La programación en Visual Basic se basa en un ambiente de desarrollo totalmente gráfico, que facilita la creación de interfaces gráficas, y en cierta medida, también la programación misma. Todos los programas que realicemos en Visual Basic serán por defecto, programas con todas las características de una aplicación típica de Windows. Lenguaje de programación visual, también llamado lenguaje de cuarta generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla. es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores.

Utiliza objetos con propiedades y métodos, una excelente herramienta de programación que permite crear aplicaciones propias (programas) para Windows 95/98 o Windows NT. Sus aplicaciones en Ingeniería son casi ilimitadas: representación de movimientos mecánicos o de funciones matemáticas, gráficas termodinámicas, simulación de circuitos, etc. Este programa permite crear ventanas, botones, menús y cualquier otro elemento de Windows de una forma fácil e intuitiva. El lenguaje de programación que se utilizará será el Basic, que se describirá luego.

Para desarrollar un programa en visual Basic se crea un formulario y sobre el se dibujan controles tales como cajas de texto, etiquetas, listas desplegables, gráficos botones de comando .etc. que luego se le asignan las propiedades y finalmente se escriben las instrucciones relacionadas a cada uno de ellos.

2.6 SERVO MOTORES UTILIZADOS EN EL PROYECTO

Un servomotor básicamente es un actuador mecánico basado en un motor y un conjunto de engranajes que permiten multiplicar el torque del sistema final son un tipo especial de motor de corriente continua. Que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su intervalo de operación. Para ello, el servomotor espera un tren de pulsos para realizar su movimiento a la vez posee elementos de control para monitorear de manera constante la posición de un elemento mecánico que será el enlace con el mundo exterior. Es decir, ante una acción inducida electrónicamente a un servomotor, obtendremos por resultado una respuesta mecánica controlada.

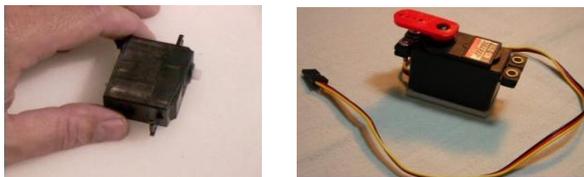


Figura N° 2.15 Forma física de un servomotor

2.6.1 COMPONENTES DEL SERVOMOTOR

Estos dispositivos Están generalmente formados por un amplificador, un motor, un sistema reductor compuesto de ruedas dentadas y un circuito de realimentación, todo en un misma caja de pequeñas dimensiones. El resultado es un servo de posición con un margen de operación de 180° aproximadamente en el caso de nuestro proyectos haremos que gire 360° .

El motor tiene circuitos de control y un potenciómetro (una resistencia variable) esta es conectada al eje central del servo motor. En la figura se puede observar al lado derecho del circuito. Este potenciómetro permite a la circuitería de control, supervisar el ángulo actual del servo motor. Si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito chequea que el ángulo no es el Correcto, el motor girará en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de llegar alrededor de los 360° grados. Normalmente, en pero varía según el fabricante. Un servo normal no es mecánicamente capaz de retornar a su lugar, si hay un mayor peso que el sugerido por las especificaciones del fabricante.

La cantidad de voltaje aplicado al motor es proporcional a la distancia que éste necesita viajar. Así, si el eje necesita regresar una distancia grande, el motor regresará a toda velocidad. Si este necesita regresar sólo una pequeña cantidad, el motor correrá a una velocidad más lenta, a esto se le llama control proporcional.



Figura N° 2.16 partes de un servomotor

2.6.2 FUNCIONAMIENTO DEL SERVOMOTOR

El motor posee la característica de girar a una buena velocidad, la cual disminuye por los juegos de engranajes de la caja reductora que aprovechan esta velocidad para transformarla en fuerza de trabajo. Al girar el último engranaje acoplado al eje de salida obtenemos una velocidad notablemente reducida, a pesar de que, dentro del sistema, el motor está girando a altas velocidades, en esta última rueda de acoplamiento encontraremos topes o límites de recorrido para entregarnos en la salida final un giro de 180° del brazo actuador. En la mayoría de los servomotores, este desplazamiento angular es “copiado” por un potenciómetro incorporado al sistema de control de forma mecánica al eje externo. Este sensor resistivo se encargará de informarle al sistema la posición que posee el actuador exterior para así controlar con exactitud la instrucción de posicionamiento.

El cable de control se usa para comunicar el ángulo. El ángulo está determinado por la duración de un pulso que se aplica al alambre de control. A esto se le llama PCM Modulación codificada de Pulsos. El servo espera ver un pulso cada 20 milisegundos (.02 segundos). La longitud del pulso determinará los giros de motor. Un pulso de 1.5 ms., por ejemplo, hará que el motor se torne a la posición de 90 grados (llamado la posición neutra). Si el pulso es menor de 1.5 ms., entonces el motor se acercará a los 0 grados. Si el pulso es mayor de 1.5ms, el eje se acercará a los 180 grados. Como se observa en la figura, la duración del pulso indica o dictamina el ángulo del eje (mostrado como un círculo verde con flecha). Nótese que las ilustraciones y los tiempos reales dependen del fabricante del Motor. El principio, sin embargo, es el mismo.

Los servomotores se controlan mediante impulsos de ancho variable que deben refrescarse periódicamente. Esto significa que si dejamos de enviar la señal de control en el tiempo en el que el servomotor lo necesita, éste (a pesar de estar energizado) dejará de mantenerse en la posición preestablecida y adoptará cualquier orientación regida por el esfuerzo al que esté sometido. Es decir, si no mantenemos la señal de control en forma efectiva todo el tiempo que sea necesario, el sistema quedará a merced de las fuerzas externas a la que sea

sometido. Por ejemplo, un brazo de palanca dejará de sostener un objeto y se dejará caer todo el trayecto mecánico que pueda recorrer, o un sistema erguido en vertical se caerá hacia atrás o hacia adelante al momento en el que el servomotor deje de “sostener” la aplicación en la posición preestablecida. Para bloquear al servomotor en una posición es necesario, entonces, enviarle continuamente la señal con la posición deseada. De esta forma, el sistema de control seguirá operando y el servo conservará su posición y se resistirá a las fuerzas externas que intenten cambiarlo de posición.

El refresco se realiza habitualmente con una frecuencia de 50 veces por segundo, pero es normal y efectivo trabajar entre los 10 y los 30 milisegundos, tal como muestra el diagrama. Por otro lado, el ancho del impulso, es decir, su tiempo de duración, dará la posición u orientación del actuador mecánico. En este punto en particular, es decir, cuando se está constantemente atento a enviar un impulso de duración exacta para evitar oscilaciones mecánicas indeseadas, se centra la complejidad y la cautela al trabajar con servomotores.

Si para el control de uno de estos sistemas utilizamos un micro controlador que además de actuar sobre el servomotor deba realizar otras actividades, se nos planteará siempre el temor de no lograr actuar en los tiempos requeridos por estos sistemas. Sin embargo, la clave de un funcionamiento satisfactorio y de una realización sin mayores complicaciones se basa en la utilización de un cristal de alta frecuencia (20 Mhz) para poder ejecutar todo el trabajo adicional que sea necesario hasta que llegue el momento de refrescar nuevamente el impulso de control del servomotor. Esto debe ir acompañado, por supuesto, de la observación cuidadosa del diagrama de tiempos de nuestra aplicación.

Tiempos de duración de los impulsos y dirección obtenida del actuador

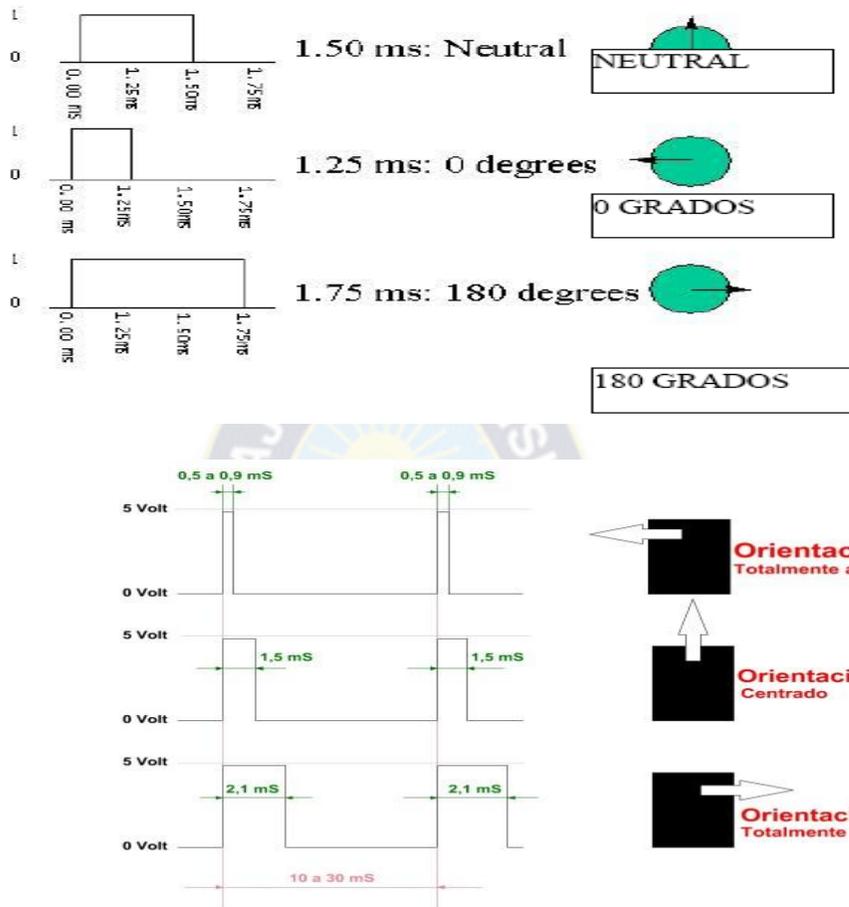


Figura N° 2.17 Tiempos de duración de los impulsos y dirección obtenida del actuador

Además posee un eje de rendimiento controlado ya que puede ser llevado a posiciones angulares específicas al enviar una señal codificada, Con tal de que exista una señal codificada en la línea de entrada, el servo mantendrá la posición angular del engranaje, La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cual es la corriente que consume.

2.6.3 APLICACIONES DEL SERVOMOTOR

Las aplicaciones de estos sistemas esclavos se pueden observar mayormente en aeromodelismo y robótica, pero no son exclusivos de estos usos. Cualquier

sistema que requiera un posicionamiento mecánico preciso y controlado dependerá de un servosistema o servomecanismo, actuado, por supuesto, por un servomotor. El zoom de una cámara, el autoenfoco de un conjunto óptico, un sistema de movilización de cámaras de vigilancia, posicionar elementos de control como palancas, pequeños ascensores y timones y por supuesto, en robots, inclusive las puertas automáticas de un ascensor son sencillos ejemplos de su aplicación.

2.6.4 CONEXIÓN DE VOLTAJE A UN SERVO MOTOR

Las conexiones son muy sencillas y se basan en una normativa de colores muy elementales que involucran al Rojo como positivo de la alimentación principal, junto a otro cable que puede ser de color Negro o Marrón y que, por lógica, podemos deducir que se trata del negativo de alimentación. Un tercer cable, correspondiente al control de posicionamiento del actuador mecánico, es Amarillo o Blanco. La tensión de trabajo de los servomotores suele estar comprendida entre los 3 y los 7 Volts, siendo 5 Voltios la tensión que se utiliza en la mayoría de las aplicaciones fijas donde interviene una fuente de alimentación conectada a la red de energía domiciliaria, y 6 Voltios para los casos de alimentación a baterías cuando se trata de equipos móviles. En todos los casos, siempre se requiere de una señal de control de 5 voltios de amplitud.

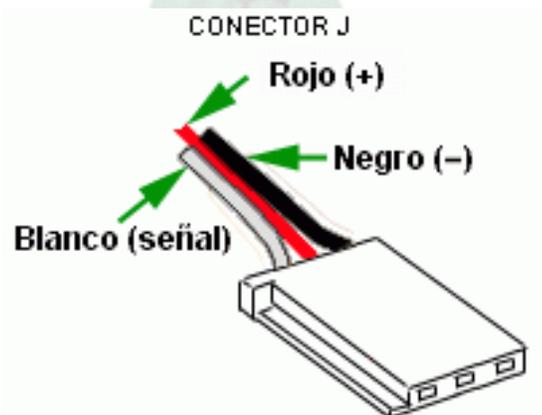


Figura N° 2.18 Conexión de Voltaje a un servomotor

2.6.5 CONTROL DEL SERVOMOTOR

Para controlar un servo, usted le ordena un cierto ángulo, medido desde 0 grados. Usted le envía una serie de pulsos. En un tiempo ON de pulso indica el ángulo al que debe posicionarse; 1ms = 0 grados, 2.0ms = máx. grado (cerca de 120) y algún valor entre ellos da un ángulo de salida proporcional. Generalmente se considera que en 1.5ms está el "centro." Entre límites de 1 ~ 2ms son las recomendaciones de los fabricantes; usted normalmente puede usar un rango mayor de 1.5ms para obtener un ángulo mayor e incluso de 2ms para un ángulo de rendimiento de 180 grados o más. El factor limitante es el tope del potenciómetro y los límites mecánicos construidos en el servo. Un sonido de zumbido normalmente indica que usted está forzando por encima al servo, entonces debe disminuir un poco. El tiempo de OFF en el servo no es crítico; puede estar alrededor de los 20ms. Hemos usado entre 10ms y 30 ms. Esto No tiene que ser de ésta manera, puede variar de un pulso a otro. Los pulsos que ocurren frecuentemente en el tiempo de OFF pueden interferir con el sincronismo interno del servo y podría escucharse un sonido de zumbido o alguna vibración en el eje. Si el espacio del pulso es mayor de 50ms (depende del fabricante), entonces el servo podría estar en modo SLEEP entre los pulsos. Entraría a Funcionar en pasos pequeños y el rendimiento no sería el óptimo.

CAPITULO III

INGENIERIA DEL PROYECTO

3.1 DESARROLLO DE LA INGENIERIA DEL PROYECTO

El análisis y el desarrollo para cumplir con los objetivos del proyecto son tanto en software y hardware, considerando los siguientes puntos de diseño en el proyecto:

Un programa principal que nos permitirá la interconexión de una PC con nuestro sistema de control mediante el puerto USB.

Una aplicación desarrollada en Visual Basic para ejecutar nuestras acciones de alineamiento.

Un sistema mecánico conformado por motores y engranajes (servomotores).

DIAGRAMA GENERAL DEL PROYECTO

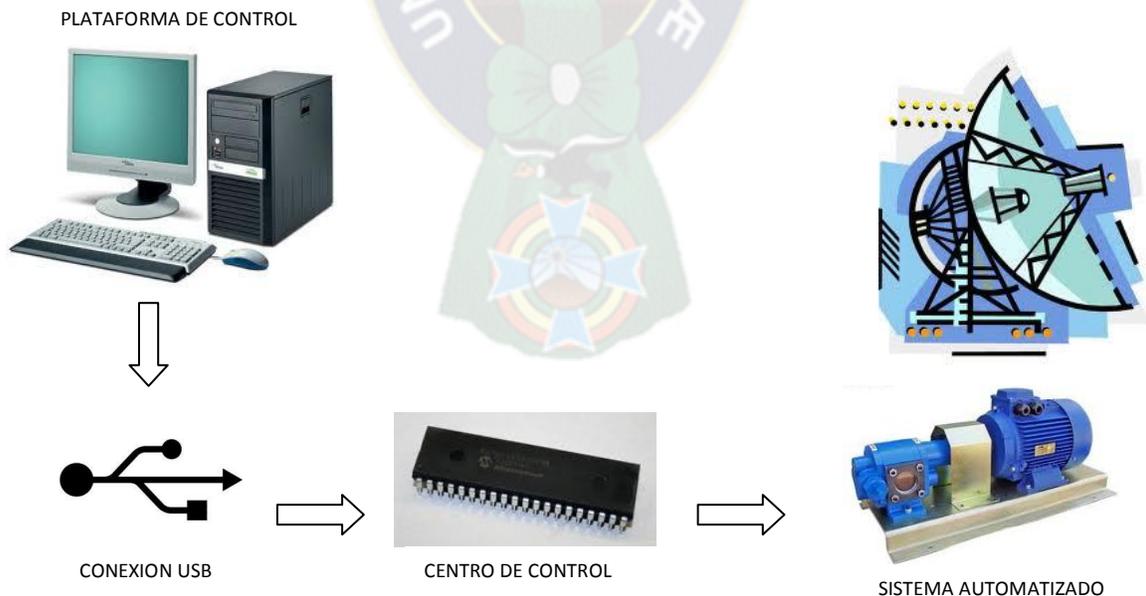


Figura N° 3.1 Diagrama en bloques del proyecto

3.1.1 MOVIMIENTO DEL SERVOMOTOR CON EL PROGRAMA VISUAL BASIC

Este trabajo es extremadamente sencillo de realizar en este lenguaje. Bastará con escribir la instrucción SERVO Pin, Posición y el sistema se posicionará de manera automática con la orientación indicada. Pin corresponderá a la I/O digital del micro controlador donde conectaremos la señal de control requerida por el servomotor, mientras que Posición corresponderá a una variable del tipo WORD, siendo valores habituales para esta variable los siguientes: 1500 para el centro, 500 para el extremo de inicio y 2500 para el extremo de final de carrera del actuador. En el proyecto se utilizara dos servos motores para nuestros movimientos automáticos de la antena tanto horizontal (azimut) como vertical (elevación)

3.1.2 Armado del circuito mecánico del proyecto

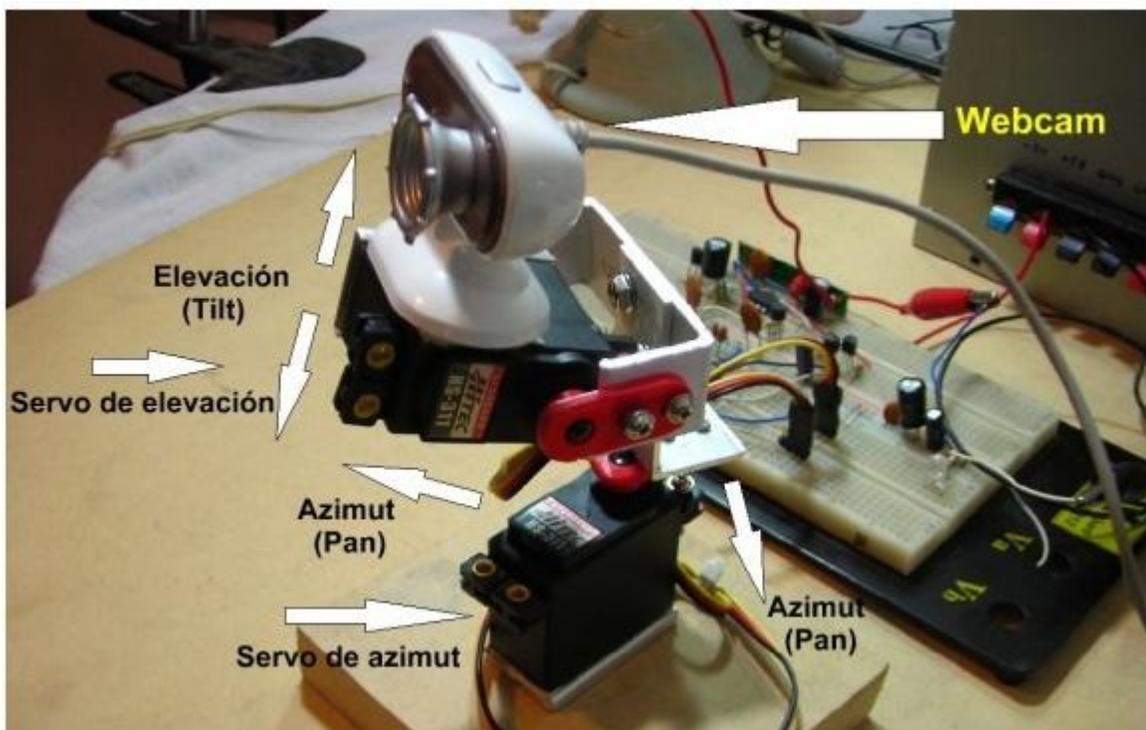


Figura Nº 3.2 Armado del circuito mecánico del proyecto

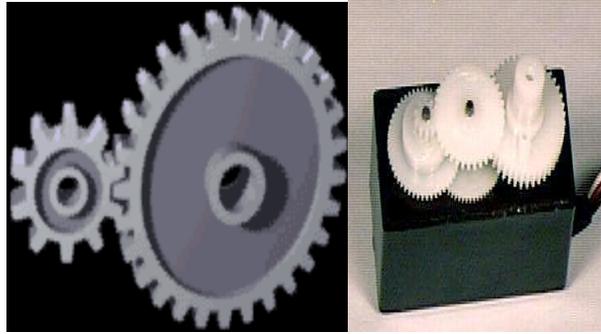


Figura Nº 3.3 Sistema de engranaje

Al hablar de la parte mecánica del armado nos referimos a todo lo que concierne lo que puede ser el hardware del sistema en esta parte mencionaremos que Los engranajes o ruedas dentadas del mecanismo que son usados en el proyecto nos sirven para transmitir potencia, movimiento mediante contacto de ruedas dentadas. La aplicación que realizamos de los engranajes en nuestro caso es la transmisión del movimiento desde el eje de nuestro motor ,eléctrico , hasta el otro eje situado en el soporte de la antena , que realiza un trabajo de manera que una de las ruedas está conectada por el motor conocido como engranaje motor y la otra está conectada al eje que debe recibir el movimiento del eje motor y que se denomina engranaje conducido.

Los engranajes están formados por dos ruedas dentadas, de las cuales la mayor se denomina 'corona' y la menor 'piñón' , La principal ventaja que tenemos en las transmisiones por engranaje respecto de la transmisión por poleas es que no patinan como las poleas, con lo que obtenemos con exactitud la relación de transmisión.

Para el movimiento que se transmite entre nuestros engranajes, se disponen de dos rodillos en contacto, en donde no hay deslizamiento, al diámetro de estos rodillos se les conoce como diámetro primitivo y al círculo que se construye con diámetro primitivo se le conoce como círculo primitivo. Con un diente de engrane se pretende prolongar la acción de los rodillos, en el proyecto usaremos dos

servo motores uno para el movimiento vertical y otro para el movimiento horizontal que a continuación indicaremos con más detalle.

3.1.3 CONTROL DEL MOVIMIENTO HORIZONTAL

En el proyecto La antena realizara dos movimientos tanto horizontal como vertical Para el movimiento horizontal usaremos un servo motor para el movimiento vertical usaremos otro servo motor, en el desplazamiento horizontal se realizara el giro automático que estará controlado por el programa En la imagen del proyecto se puede apreciar que el servo inferior (o de azimut) está fijo y asentado a una base y que moviliza un pequeño trozo de metal en forma de “L”. Este primer servo nos dará la movilidad en el eje cartesiano “x”, entregándonos un giro de 360° de Izquierda a derecha y viceversa.

3.1.4 CONTROL DEL MOVIMIENTO VERTICAL

Del mismo modo Para el movimiento vertical se tendrá un servo motor , que posee un eje en la parte externa saliente del mismo en la cual se hará soportar el soporte de la antena ,puesto que La transmisión ejercida por el motor hará que giren los engranajes sin pérdida de fricción en el sistema , de esta manera nos proporcionara un movimiento vertical automático para la antena ,en función a la señal seleccionada enviada en grados desde las etapas que le anteceden . En nuestro proyecto aseguramos el brazo de la palanca del segundo servo, el cual, gracias a esta disposición, nos dará el movimiento sobre el eje cartesiano “y”, aportando al sistema el ajuste de altura, también conocido como elevación

3.1.5 DISEÑO DEL SOFTWARE PARA ALINEAMIENTO Y CONTROL DE LA ANTENA

El diseño del software implementado para este proyecto está dividido en dos partes:

- ❖ Desarrollo software para el entorno de Visual Basic 6.0
- ❖ Desarrollo software para el Microcontrolador PIC18F4550

3.1.6 DESARROLLO SOFTWARE PARA EL ENTORNO DE VISUAL BASIC

Option Explicit ' declaración de variables como enteros

Dim R As Integer

Dim i As Integer

Dim c As Integer

Dim dato As Integer

Dim dir As Integer

Dim numLed As Integer

Private Sub cmdAceptar1_Click()

cmdAceptar1.Enabled = False (deshabilita el botón aceptar2)

cmdAceptar2.Enabled = False (deshabilita el botón aceptar2)

If Trim(txtDato1.Text) <> "" Then (pregunta si se ha introducido un dato en la caja de texto)

dato = Val(txtDato1.Text) (dato=caja de texto)

If dato <> 0 Then

If dato <= 180 Then (pregunta si el dato es menor o igual 180 grados)

MSComm1.Output = (Chr(0)) (envía el dato para el motor 1 por el puerto USB)

MSComm1.Output = Chr(dato) (envía el dato de las coordenadas por el puerto USB1 Motor 1)

numLed = 0 (valores iníciales posicion inicial para la secuencia de leds motor1)

dir = 1 (giro hacia la derecha motor1)

i = 0 (valor inicial para rotar los leds)

c = 1 (valor inicial para controlar la secuencia de leds hacia la derecha)

tmr1.Enabled = True (habilitar el timer1)

Else (ingresa cuando el dato es mayor a 180 grados)

R = MsgBox("Debe Introducir el Angulo de Elevacion correcto", 1 + 16, "ERROR") (mensaje de error)

txtDato1.Text = "" (limpiar la caja de texto)

txtDato1.SetFocus (poner el cursor en la caja de texto)

tmr1.Enabled = False (deshabilitar el timer1)

End If

Else

R = MsgBox("Debe Introducir un Dato Numerico", 1 + 16, "ERROR") (mensaje de error)

txtDato1.Text = "" (limpiar la caja de texto)

txtDato1.SetFocus (poner el cursor en la caja de texto)

tmr1.Enabled = False (deshabilitar el timer1)

End If

Else (ingresa cuando no se ha introducido ningún valor)

```

R = MsgBox("Introduzca el Angulo de Elevacion", 1 + 16, "ERROR")
(mensaje de error)
txtDato1.Text = "" (limpiar la caja de texto)
txtDato1.SetFocus (poner el cursor en la caja de texto)
tmr1.Enabled = False (deshabilitar el timer1)
End If
End Sub

Private Sub cmdAceptar2_Click()
'cmdAceptar1.Enabled = False (deshabilita el boton aceptar1)
'cmdAceptar2.Enabled = False (deshabilita el boton aceptar2)
If Trim(txtDato2.Text) <> "" Then (pregunta si se ha introducido un dato en la
caja de texto)
dato = Val(txtDato2.Text) (dato=caja de texto)
If dato <> 0 Then
If dato <= 180 Then (pregunta si el dato es menor o igual 180 grados)
numLed = 8 (valor inicial para la secuencia de leds motor2)
If optEste.Value = True Then (pregunta si la opcion pulsada es "ESTE")
dir = 1 (giro hacia la derecha motor1)
c = 1 (valor inicial para controlar la secuencia de leds hacia la
derecha)
dato = 180 + dato (sumarle a 90 grados el dato introducido)
Else (si la opcion es OESTE)
dir = 0 (giro hacia la izquierda motor1)
c = 180 (valor inicial para controlar la secuencia de leds hacia la
izquierda)
dato = 180 - dato (restarle a 180 grados el dato introducido)
End If
If dato <= 255 Then
MSComm1.Output = Chr(1)
MSComm1.Output = Chr(dato)
Else
MSComm1.Output = Chr(255)
MSComm1.Output = Chr(dato - 255)
End If
'MSComm1.Output = (Chr(1)) (envía el dato para el motor2 por el puerto
serial)
'MSComm1.Output = Chr(dato) (envía el dato de las coordenadas por
el puerto) serial Motor2
i = 0 'valor inicial para rotar los leds
tmr1.Enabled = True 'habilitar el timer1
Else
R = MsgBox("Debe Introducir el Angulo de Azimut correcto", 1 + 16,
"ERROR") (mensaje de error)
txtDato2.Text = "" (limpiar la caja de texto)

```

```

        txtDato2.SetFocus (poner el cursor en la caja de texto)
        tmr1.Enabled = False (deshabilitar el timer1)
    End If
Else
    R = MsgBox("Debe introducir un Dato Numerico", 1 + 16,
"ERROR")(mensaje de error)
    txtDato2.Text = "" (limpiar la caja de texto)
    txtDato2.SetFocus (poner el cursor en la caja de texto)
    tmr1.Enabled = False (deshabilitar el timer1)
End If
Else
    R = MsgBox("Introduzca el Angulo de Azimut", 1 + 16, "ERROR") 'mensaje
de error
    txtDato2.Text = "" (limpiar la caja de texto)
    txtDato2.SetFocus (poner el cursor en la caja de texto)
    tmr1.Enabled = False (deshabilitar el timer1)
End If
End Sub

Private Sub cmdNuevo_Click()
i = 0
c = 1
MSComm1.Output = (Chr(0)) (envía el dato inicial para el motor1 por el puerto
USB)
MSComm1.Output = Chr(0) (envía el dato inicial de las coordenadas (dato inicial =
0 grados)por el puerto serial Motor1)
MSComm1.Output = (Chr(1)) (envía el dato inicial para el motor2 por el puerto
USB)
MSComm1.Output = Chr(180) (envía el dato inicial de las coordenadas (dato
inicial = 0 grados) por el puerto USB Motor2)
txtDato1.Text = "" (limpiar la caja de texto)
txtDato2.Text = "" (limpiar la caja de texto)
tmr1.Enabled = False (deshabilitar el timer1)
txtDato1.SetFocus (poner el cursor en la caja de texto)
cmdAceptar1.Enabled = True
cmdAceptar2.Enabled = True
End Sub

Private Sub cmdSalir_Click()
End (salir)
End Sub

Private Sub Form_Load()
i = 0 'valor inicial para rotar los leds
MSComm1.CommPort = 1

```

```

MSComm1.PortOpen = True (habilitar el puerto USB)
MSComm1.Settings = "9600,n,8,1" (configurar el puerto serial
"velocidad=9600,bits de paridad,bits de datos,bit de paridad)
MSComm1.Output = (Chr(0)) (envía el dato inicial para el motor1 por el puerto
USB)
MSComm1.Output = Chr(0) (envía el dato inicial de las coordenadas (dato inicial
= 0 grados) por el puerto USB Motor1)
MSComm1.Output = (Chr(1)) (envía el dato inicial para el motor2 por el puerto
serial)
MSComm1.Output = Chr(180) (envía el dato inicial de las coordenadas (dato
inicial = 0 grados) por el puerto USB Motor2)
Call ApagaLed (salto al procedimiento apagaLed)
tmr1.Enabled = False (desbilitar el timer1)
End Sub

```

```

Public Function ApagaLed() (procedimiento apagaLed)
Dim j As Integer
For j = 0 To 15
    Led(j).BackColor = vbWhite (pintar de blanco el led seleccionado)
Next
End Function
Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False (deshabilita el puerto USB)
End Sub

```

```

Private Sub tmr1_Timer() (procedimiento para el timer1)
ApagaLed (salto al procedimiento apagaLed)
Led(i + numLed).BackColor = &HFF (pinta de color rojo el led seleccionado)
If dir = 1 Then (pregunta si el motor va a girar a la derecha)
    If c = Val(dato) Then (pregunta si la variable c es igual al dato introducido)
        tmr1.Enabled = False (deshabilita el timer1)
        Text7.Text = c
    End If
    i = i + 1 (incrementa la variable i para el encendido del siguiente led)
    If i = 8 Then (pregunta si la variable i ha recorrido los 8 leds)
        i = 0 (inicializa la variable i)
    End If
    c = c + 1 (incrementa la variable c)
Else (ingresa si el motor va a girar a la izquierda)
    If c = Val(dato) Then (pregunta si la variable c es igual al dato introducido)
        tmr1.Enabled = False (deshabilita el timer1)
        Text7.Text = c
    End If
    i = i - 1 (decrementa la variable i para el encendido del siguiente led)
    If i = -1 Then (pregunta si la variable i ha recorrido los 8 leds)

```

```

i = 7 (inicializa la variable i)
End If
c = c - 1 (decrementa la variable c)
End If
End Sub
Private Sub txtDato1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    cmdAceptar1_Click
End If
End Sub

Private Sub txtDato2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    cmdAceptar2_Click
End If
End Sub

```

PRESENTACION DEL TRABAJO EN EJECUCION

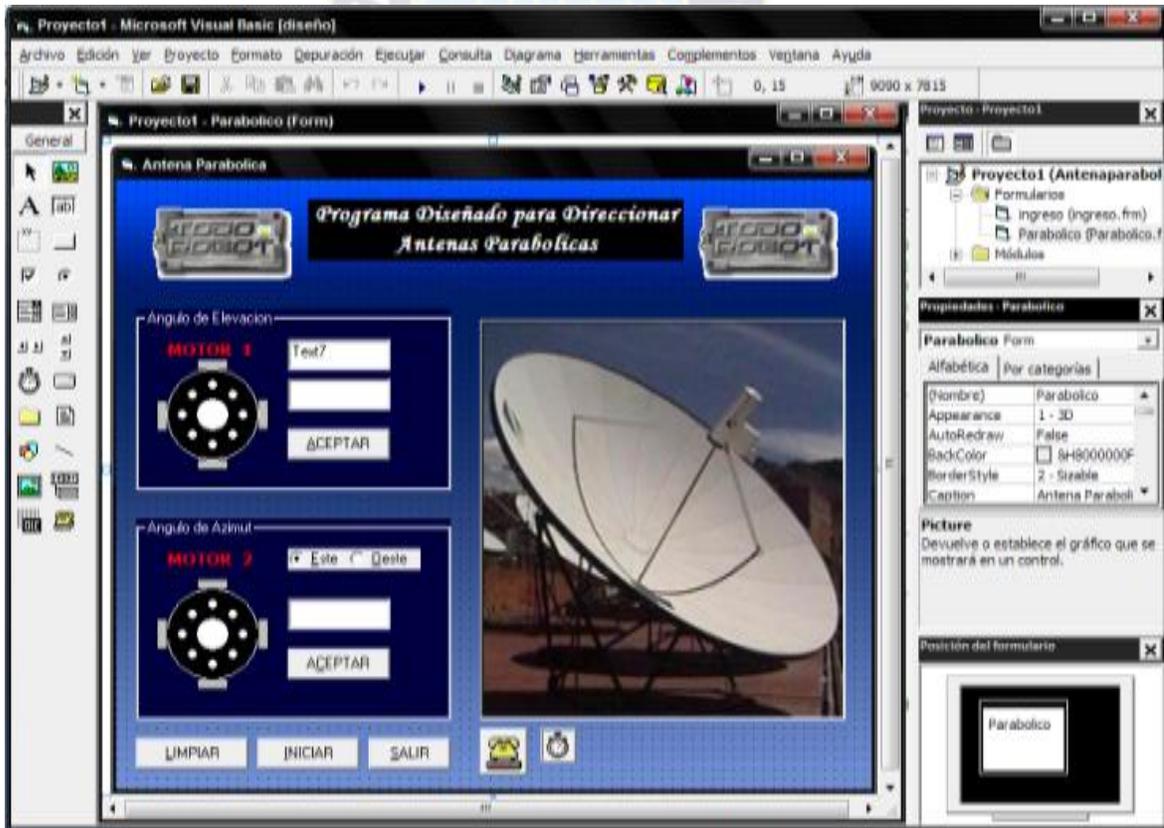


Figura N° 3.4 Plataforma de Control Visual

3.1.7 DESARROLLO SOFTWARE PARA EL MICROCONTROLADOR PIC18F4550

El programa se desarrollo en el lenguaje PIC C, donde las características de diseño e implementación fueron especialmente para comunicar el PIC18F4550 a través del puerto USB.

Código fuente del proyecto

```
#include<18F4550.c> //cargar el mapa de registros del PIC
#fuses XT,NOWDT,NOLVP,NOPROTECT //configurar los fusibles (XT trabajar
con un cristal,codigo no esta protegido,no existe retardo de tiempo en el power
timer,el perro guardian esta desactivado
#use delay(clock=2000000) //frecuencia maxima de oscilacion
#use (BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)
//velocidad,Nro de bits,bit de paridad,pin de transmision,pin de recepcion
#use fast_io(B) //configuracion rapida del puerto B
#priority timer1,rda //prioridad para el timer1
unsigned long Valor_Carga; //valor entero largo "Nro que se a hallado de forma
matematica"
unsigned int Servo=0; //valor entero "para gobernar los motores"
unsigned int Puerto_B=6; //asignar una etiqueta del registro 6
short Band_TH_TL=1; //bandera de tiempo alto y bajo
unsigned long Posicion,n,NServo; //valor recepcionado por el puerto USB para el
calculo
//unsigned int NServo; //valor recepcionado por el puerto serial del Nro de
motores
short Bandera=0; //valor encargado de cambiar de la interrupcion temporizada a
la interrupcion serial o viceversa
unsigned long HIGH[]={61036,61036}; //valor de carga para el tiempo alto
unsigned long LOW[] ={20036,20036}; //valor de carga para el tiempo bajo
```

```

float H,L; //variables con punto decimal
char cadena;
void config(void) //procedimiento "configurar valores iniciales"
{
set_tris_b(0x00); //configurar al puerto B como salida
set_tris_c(0xF0); //configurar al puerto C (4 bits de salida,4 bits de entrada)
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1); //(temporizador
interno,prescaler=1)
set_timer1(Valor_Carga); //enviar al timer1 la variable Valor_carga
enable_interrupts(INT_TIMER1); //habilitar la interrupcion del timer1
enable_interrupts(INT_RDA); //habilitar la interrupcion
enable_interrupts(GLOBAL); //habilitar las interrupciones de forma global
}
#INT_TIMER1 //interrupcion del timer1
void Interrupcion_timer1(void) //procedimiento
{
if(Band_TH_TL==1) //bandera para el tiempo alto
{
bit_set(*Puerto_B,Servo); //envia un pulso alto (registro 6 del puerto B,Nro del
servo motor)
Valor_Carga=HIGH[Servo]; //valor_carga=tiempo alto calculado
Band_TH_TL=0; //cambiar la bandera a 0 para tiempo bajo (registro 6 del
puerto B,Nro del servo motor)
}
else if (Band_TH_TL==0) //bandera para el tiempo bajo
{
bit_clear(*Puerto_B,Servo); //envia un pulso bajo
Servo++; //incremento de la variable servo
if(SERVO>=2)
{

```

```

    Servo=0;
}
Valor_Carga=LOW[SERVO]; //valor_carga=tiempo bajo calculado
Band_TH_TL=1; //cambiar el valor de la bandera
}
set_timer1(Valor_Carga); //envia al timer1 el valor de la carga
//printf(lcd_putc, "\fcarga = %6ld", LOW[SERVO]);
}
#INT_RDA
void Interruccion_X_recepcion(void) //procedimiento
{
    disable_interrupts(INT_TIMER1); //deshabilitar la interrupcion del timer1
    if(kbhit()) //pregunta si ha llegado algun dato por el puerto USB
    {
        NServo=getc(); //guardar en la variable el dato del buffer de recepcion serial
        Posicion=getc(); //guardar en la variable el dato del buffer de recepcion serial
    }
    Bandera=1; //bandera para el control de cambio de interrupciones
    set_timer1(get_timer1()); //
    enable_interrupts(INT_TIMER1); //habilitar la interrupcion del timer1
}
void main(void)
{
    posicion=0;
    //lcd_init();
    config(); //llamar al procedimiento para configurar valores iniciales
    while(TRUE) //bucle infinito
    {
        if(BANDERA==1) //controla bandera para el cambio de interrupciones
        {

```

```

disable_interrupts(INT_RDA); //deshabilitar la interrupcion
//posicion=posicion*10+(cadena-48);
if(NServo<=1)
{
    n=Posicion;
}
else
{
    /*if(NServo==254)
    {
        n=Posicion+255;
        NServo=0;
    }*/
    if(NServo==255)
    {
        n=Posicion+255;
        NServo=1;
    }
}
H=0.0009+(n*0.000006666666667); //pulso alto calculado
L=0.01-H; //pulso bajo calculado
HIGH[NSERVO]=65536-(H/(1*0.0000002)); //valor para el valor_carga alto
LOW[NSERVO]=65536-(L/(1*0.0000002)); //valor para el valor_carga bajo
BANDERA=0;
enable_interrupts(INT_RDA); //habilitar la interrupcion
}
}
}

```

DIAGRAMA ESQUEMATICO DEL PROYECTO

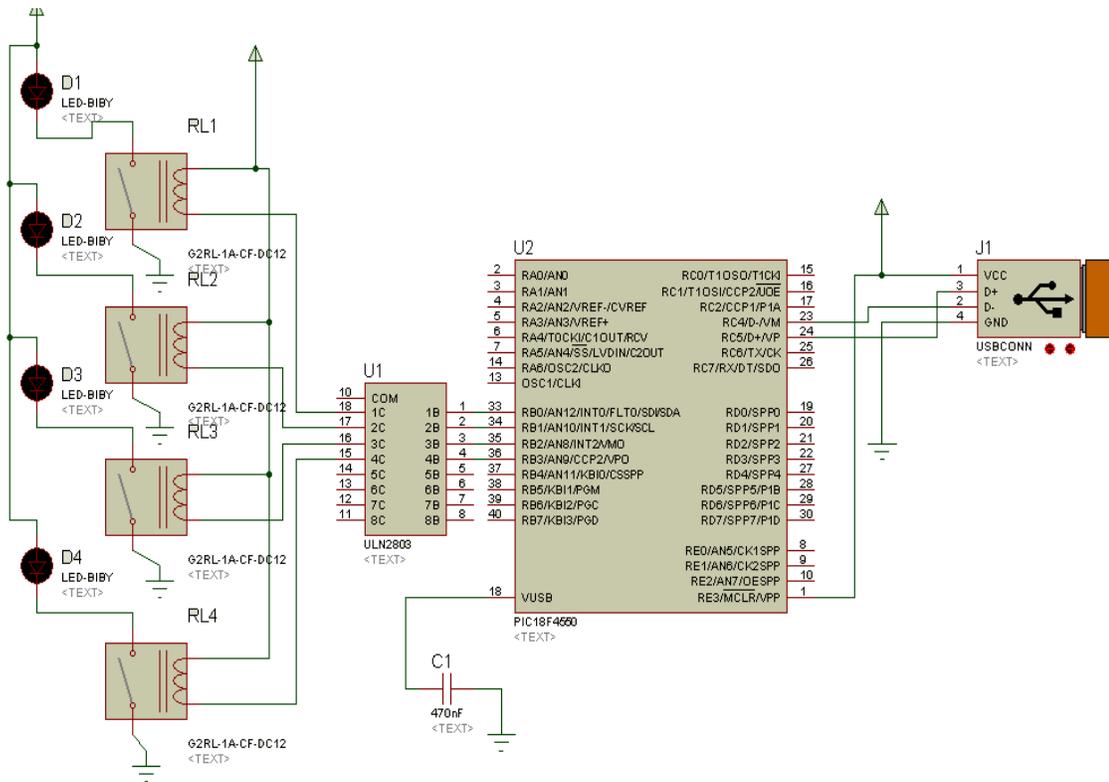


Figura N° 3.5 simulación en Isis de proteus

CAPITULO IV

COSTOS Y PRESUPUESTOS

4.1 CONCEPTO DE COSTOS

El costo es el gasto económico que representa la fabricación de un producto o la prestación de un servicio. Al determinar el costo de producción, se puede establecer el precio de venta al público del bien en cuestión (el precio al público es la suma del costo más el beneficio).

El costo de un producto está formado por el precio de la materia prima, el precio de los elementos que conforma el proyecto, el precio de la implementación para el funcionamiento de un proyecto y el costo de los estudios para el diseño de un proyecto.

Los especialistas afirman que muchos empresarios suelen establecer sus precios de venta en base a los precios de los competidores, sin antes determinar si éstos alcanzan a cubrir sus propios costos. Por eso, una gran cantidad de negocios no prosperan ya que no obtienen la rentabilidad necesaria para su funcionamiento. Esto refleja que el cálculo de los costos es indispensable para un correcto funcionamiento del proyecto.

4.2 ¿QUE ES EL PRESUPUESTO?

Un presupuesto es un plan integrador y coordinador que expresa en términos financieros con respecto a las operaciones y recursos que forman parte de un proyecto para un periodo determinado, con el fin de lograr los objetivos fijados por el proyectista.

Normalmente se realiza este procedimiento en la mayoría de los proyectos, ya que cuentan con un propósito de implementación.

4.3 ELEMENTOS NECESARIOS PARA LA ELABORACION DEL PROYECTO

Elementos Hardware

- ❖ Microcontrolador PIC18F4550 Gama Alta.
- ❖ Servomotores de corriente continua y engranajes reductores.
- ❖ Antena parabólica.
- ❖ Xtal 20MHz.
- ❖ Resistencias de diferentes valores.
- ❖ Capacitores de diferentes valores.
- ❖ Conector USB.
- ❖ Led's.
- ❖ Transistores DB135.
- ❖ Relés de 5V.
- ❖ Diodos IN4000
- ❖ Rodamientos.
- ❖ Ejes de rotación.
- ❖ Pernos de ajuste.

Elementos Software

- ❖ Visual Basic 6.0.
- ❖ Mikroc for compiler.



❖ ISIS de proteus.

❖ Software Grabador PicKid.

4.4 COSTO DE COMPONENTES HARDWARE

DESCRIPCION	UNIDADES	COSTO UNITARIO(BS)	TOTAL (BS)
Microcontrolador PIC18F4550	1	80	80
Cristal 4MHz	1	5	5
Transistor DB135	3	2	6
Relés 5 voltios	4	10	40
Conector USB	1	15	15
Resistencias	16	0.25	4
Diodos	4	0.5	2
Led's	5	0,2	1
Capacitores	3	1	3
Servomotores	2	50	100
Pernos de ajuste	5	2	10
Ejes de rotación	2	10	20
Rodamientos	4	5	20
Antena Parabólica	1	Variable	Variable
		TOTAL	306 Bs.

4.5 PRESUPUESTO PARA LA IMPLEMENTACION DEL PROYECTO

De acuerdo a las necesidades para la implementación del proyecto llegamos a obtener el costo estimado total de los componentes a nivel hardware.

Una vez obtenido el precio de cada uno de los componentes podemos predecir que la implementación de este proyecto tendrá un costo total a nivel Hardware de 306 Bs. Teniendo en cuenta que la mano de obra del personal que lo implementara tendrá un costo variable, a esto será necesario adicionar el costo de la mano de obra para la instalación.

4.6 TIEMPO PARA LA IMPLEMENTACION DEL PROYECTO

TRABAJO REALIZADO	Elaboración del programa(Software)	Montaje del circuito	Montaje del circuito a la antena
TIEMPO (días)	3	1	2
TIEMPO TOTAL DE IMPLEMENTACION	6 DIAS		

4.7 COSTO DE MANO DE OBRA

El costo de la mano de obra está en función de los días de trabajo, en este caso para la implementación de este proyecto consideramos tomar tres días en la elaboración del software, un día para el montaje del circuito y dos días para el montaje del circuito a la antena. Por consiguiente para la implementación de este proyecto se tomaran seis días de trabajo. Teniendo en cuenta que el personal

técnico desea ganar 100 Bs por día, el costo total de la mano de obra será de 600 Bs.

4.8 COSTO TOTAL DE IMPLEMENTACION

DESCRIPCION	UNIDADES	COSTO UNITARIO(BS)	TOTAL (BS)
Microcontrolador PIC18F4550	1	80	80
Cristal 4MHz	1	5	5
Transistor DB135	3	2	6
Relés 5 voltios	4	10	40
Conector USB	1	15	15
Resistencias	16	0.25	4
Diodos	4	0.5	2
Led's	5	0,2	1
Capacitores	3	1	3
Servomotores	2	50	100
Pernos de ajuste	5	2	10
Ejes de rotación	2	10	20
Rodamientos	4	5	20
Antena Parabólica	1	Variable	Variable
Mano de obra	1	600	600
		TOTAL	906 Bs.

CONCLUSIONES

Al finalizar el proyecto de aplicación se llegó a las siguientes conclusiones:

La implementación de hardware y software fue de gran éxito porque responde de manera satisfactoria nuestro objetivo.

La implementación a largo plazo del siguiente proyecto tiene muchas perspectivas y grandes beneficios en cuanto al funcionamiento automatizado del alineamiento de antenas parabólicas.

El uso de la interfaz USB, prácticamente obtuvo grandes ventajas, ya que hoy en día las computadoras portátiles ya no disponen de puertos seriales y puertos paralelos.



BIBLIOGRAFIA

- USB Complete: Everything You Need to Develop USB Peripherals, Third Edition.
- Advanced PIC Microcontroller projects in C From USB to RTOS whit the PIC18F Series Autor: Dogan Ibrahim.
- Microcontroladores PIC, Diseño practico de Aplicaciones 2da parte 16F87x. Autores Jose Angulo Usategui, Ignacio Angulo Martines.
- Curso de Microcontroladores PIC16F87x por Fernando Remiro Dominguez.

PAGINAS WEB CONSULTADAS

- www.microchip.com
- www.labcenter.com
- www.monografias.com/.visual-basic/tutorial.shtml