

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO PARA LA
APERTURA DE LA PUERTA DE UN GARAJE, CONTROLADO
DESDE EL VEHÍCULO, MEDIANTE UN MÓDULO ELECTRÓNICO
UTILIZANDO MICROCONTROLADORES RF, SENSORES
INFRARROJOS Y TECLADO HEXADECIMAL**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

MERCY ALEXANDRA LOACHAMIN CHANATAXI

megcy585@hotmail.com

MARÍA JOSÉ CHÁVEZ CHAQUINGA

majos-ch@hotmail.com

DIRECTOR: ING. ALCÍVAR EDUARDO COSTALES G.

eduardo.costales@epn.edu.ec

Quito, Diciembre 2010

DECLARACIÓN

Nosotras, Mercy Alexandra Loachamin Chanataxi y María José Chávez Chaquinga, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Mercy Alexandra Loachamin Chanataxi

María José Chávez Chaquinga

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Mercy Alexandra Loachamin Chanataxi y María José Chávez Chaquina, bajo mi supervisión.

Ing. Alcívar Costales

DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradezco a Dios por ser el centro de mi vida y por haberme dado una familia que me ha sabido apoyar incondicionalmente y guiarme en forma espiritual y personal durante toda mi vida; en cada paso que he dado, especialmente en lo referente a mis estudios.

A mi HEART que me ha apoyado y comprendido durante todo este tiempo de trabajo y que ha estado siempre a mi lado en las buenas y en las malas.

Y en especial al Ing. Alcívar Costales y a mi querida Escuela Politécnica Nacional por permitir cristalizar con éxito mi proyecto y anhelos.

Mercy Loachamin

AGRADECIMIENTO

Mi eterna gratitud a Dios por cuidarme y darme una familia tan maravillosa, a mis padres que han sido testigos de mis triunfos y fracasos, me han brindado su apoyo incondicional en mi vida estudiantil.

A mi niño travieso que forma parte esencial en mi vida, me enseñó que siempre puedo cumplir las metas que me proponga.

De manera especial al Ing. Alcívar Costales y a mi querida Institución de la cual llevo las mejores enseñanzas.

María José Chávez Chaquinga

DEDICATORIA

Dedico este trabajo a Dios que siempre está conmigo, protegiéndome e iluminándome, a mi mamá que con sus consejos ha sabido guiarme por la luz del bien en el caminar de la vida, apoyándome en todo lo que me proponga.

A mi papi que con su ejemplo me ha enseñado a salir adelante trabajando con esfuerzo, dedicación y honestidad.

A mi hermanita que es el angelito que ilumina mi casa y que con su alegría e inocencia, siempre con una sonrisa y abrazo me ha sacado de los momentos tristes que he atravesado.

A mi HEART que con su amor, confianza y paciencia me ha enseñado que siendo humilde y perseverante siempre se consiguen grandes cosas.

Atentamente, Mercy

DEDICATORIA

Dedico el proyecto a Dios que está a mi lado, a mi mami que con su cariño, trabajo y dedicación me ha guiado por el sendero del bien dedicándose a trabajar día a día por mi educación, a mi papi que por su ejemplo de esfuerzo y perseverancia.

A mis abuelitos, que me cuidaron de niña y me enseñaron muchos valores.

A mi Ricky, que está en las buenas y malas brindándome su amor, confianza y paciencia.

María José Chávez Chaquinga

CONTENIDO

DECLARACIÓN	I
CERTIFICACIÓN	II
AGRADECIMIENTO	III
AGRADECIMIENTO	IV
DEDICATORIA	V
DEDICATORIA	VI
CONTENIDO	VII
RESUMEN	XII
PRESENTACIÓN	XIII
CAPÍTULO 1	1
CONCEPTOS GENERALES	1
1.1 SISTEMAS DE CONTROL DE ACCESO	1
1.1.1 MÉTODOS DE IDENTIFICACIÓN DE USUARIOS.....	1
1.1.1.1 Sistemas biométricos	1
1.1.1.2 Control de contraseñas.....	2
1.1.1.3 Acceso con objetos físicos (tokens)	2
1.1.2 TECNOLOGÍAS EMPLEADAS.....	2
1.1.2.1 Sistema de acceso por teclado.....	2
1.1.2.2 Tarjetas magnéticas	3
1.1.2.3 Tarjetas de proximidad o tarjetas RFID.....	4
1.1.2.4 Tarjetas de código de barras.....	4
1.1.2.5 Tarjetas de chip o tarjetas inteligentes.....	5
1.1.2.6 Lectores biométricos.....	6
1.2 TRANSMISIÓN INALÁMBRICA.....	7
1.2.1 COMUNICACIÓN INALÁMBRICA.....	7
1.2.2 ESPECTRO ELECTROMAGNÉTICO	8
1.2.3 BANDAS DEL ESPECTRO ELECTROMAGNÉTICO.....	9
1.2.4 MEDIOS DE TRANSMISIÓN.....	9
1.2.4.1 Medios de transmisión guiados	10
1.2.4.1.1 Guía de onda.....	10
1.2.4.1.2 Cable par trenzado	10
1.2.4.1.3 Cable coaxial	10
1.2.4.1.4 Fibra óptica.....	11
1.2.4.2 Medios de transmisión no guiados.....	11

1.2.4.2.1 Infrarrojos.....	12
1.2.4.2.2 Microondas.....	12
1.2.4.2.3 Satélite.....	12
1.2.4.2.4 Radio enlaces de VHF y UHF.....	12
1.2.4.2.5 Ondas cortas.....	13
1.3 RADIO FRECUENCIA.....	13
1.3.1 CLASIFICACIÓN DE RADIO FRECUENCIA.....	13
1.3.2 COMUNICACIÓN POR RADIO FRECUENCIA.....	14
1.3.2.1 Tipos de comunicaciones inalámbricas por RF.....	15
1.3.2.2 Bandas ISM (Industriales, Científicas y Médicas).....	15
1.3.2.3 ZigBee (Estándar de comunicaciones Inalámbricas IEEE 802.15.4).....	16
1.3.2.3.1 Características generales ZigBee/IEEE 802.15.4.....	17
1.3.2.3.2 Estructura de la trama.....	17
1.3.2.3.3 Topologías de Redes ZigBee.....	17
1.3.3 CONCEPTOS GENERALES PARA LA COMUNICACIÓN.....	18
1.3.3.1 Comunicación de datos.....	18
1.3.3.2 Formas de transmisión.....	19
1.3.3.2.1 Transmisión en Serie.....	19
1.3.3.2.2 Transmisión en Paralelo.....	19
1.3.3.3 Tipos de transmisión.....	19
1.3.3.3.1 Transmisión Asíncrona.....	19
1.3.3.3.2 Transmisión Síncrona.....	19
1.4 MICROCONTROLADORES.....	20
1.4.1 INTRODUCCIÓN.....	20
1.4.2 DEFINICIÓN.....	20
1.4.3 ARQUITECTURA DEL MICROCONTROLADOR.....	21
1.4.4 PARTES DEL MICROCONTROLADOR.....	23
1.4.4.1 La CPU o unidad central de proceso.....	23
1.4.4.2 Memorias.....	23
1.4.4.3 Unidades de entrada/salida.....	24
1.4.4.4 Recursos auxiliares.....	25
1.5 MOTORES.....	26
1.5.1 INTRODUCCIÓN.....	26
1.5.2 PRINCIPIO DE FUNCIONAMIENTO.....	27
1.5.3 MOTORES DE CORRIENTE CONTINUA.....	27
1.6 SENSORES.....	28
1.6.1 CARACTERÍSTICAS DE UN SENSOR.....	28
1.6.2 TIPOS DE SENSORES.....	29
1.6.2.1 Interruptores final de carrera.....	29
1.6.2.2 Productos infrarrojos.....	29

1.6.2.2.1 Sensores infrarrojos.....	30
1.7 COMUNICACIÓN USB.....	32
1.7.1 INTRODUCCIÓN AL USB.....	32
1.7.2 CARACTERÍSTICAS GENERALES DEL USB.....	32
1.7.3 LIMITACIONES.....	35
1.7.4 CABLES Y CONECTORES USB.....	35
CAPÍTULO 2.....	37
DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO.....	37
2.1 PRINCIPIO DE FUNCIONAMIENTO DEL PROTOTIPO.....	37
2.2 COMPONENTES DEL HARDWARE.....	38
2.2.1 MÓDULOS DE RADIO FRECUENCIA.....	39
2.2.1.1 Módulos de Comunicación XBee.....	39
2.2.1.2 Circuito básico y Distribución de pines del XBee.....	40
2.2.1.3 Comunicación del módulo XBee.....	42
2.2.1.4 Modos de operación.....	43
2.2.2 MICROCONTROLADORES.....	43
2.2.2.1 Microcontrolador ATmega8.....	44
2.2.2.1.1 Características generales del ATmega8.....	44
2.2.2.1.2 Distribución de pines del ATmega8.....	45
2.2.2.1.3 Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART).....	47
2.2.2.2 Microcontrolador ATmega164P.....	48
2.2.2.2.1 Características principales del ATmega164P.....	48
2.2.2.2.2 Distribución de pines del ATmega164P.....	49
2.2.2.3 Interrupciones del ATmega.....	52
2.2.2.4 ICSP (In Circuit Serial Programming).....	52
2.2.2.4.1 Comunicación SPI (Serial Peripheral Interface).....	53
2.2.2.4.2 Descripción de líneas de conexión.....	53
2.2.2.4.3 Modo de conexión del ICSP para cada módulo.....	54
2.2.3 FUENTE DE ALIMENTACIÓN.....	54
2.2.3.1 Distribución y regulación de voltajes.....	55
2.2.4 MOTORES PASO A PASO.....	57
2.2.4.1 Introducción.....	57
2.2.4.2 Principio de funcionamiento de motores paso a paso.....	58
2.2.4.3 Secuencias para manejar motores paso a paso Unipolares.....	59
2.2.4.4 Diagrama circuital del Motor Paso a Paso.....	61
2.2.5 SENSORES INFRARROJOS.....	63
2.2.5.1 Módulo del Diodo Emisor.....	63
2.2.5.1.1 Circuito Integrado LM555.....	63
2.2.5.1.2 Funcionamiento del diodo emisor.....	64

2.2.5.2 Módulo del Fotodiodo Receptor	65
2.2.5.2.1 Amplificador Operacional LM358.....	65
2.2.5.2.2 Funcionamiento del fotodiodo receptor.....	66
2.2.6 <i>RELÉ</i>	66
2.2.6.1 Características generales.....	67
2.3 MÓDULO DEL CONTROL REMOTO.....	68
2.3.1 <i>MANEJO DEL TECLADO PARA EL INGRESO DE CLAVES</i>	68
2.3.1.1 Introducción Teclado Hexadecimal.....	68
2.3.1.2 Diagrama circuital del teclado hexadecimal.....	68
2.3.2 <i>TRANSMISIÓN DE LA CLAVE INGRESADA</i>	69
2.4 MÓDULO RECEPTOR.....	69
2.4.1 <i>RECEPCIÓN DE LA CLAVE INGRESADA</i>	69
2.4.2 <i>COMUNICACIÓN SERIAL</i>	71
2.4.2.1 Interfaz de comunicación serial usando USB.....	71
2.4.2.2 Conexión del microcontrolador con el interfaz USB.....	73
2.5 DESARROLLO DEL SOFTWARE.....	73
2.5.1 <i>PROGRAMACIÓN DEL MÓDULO XBee</i>	74
2.5.1.1 Proceso de transmisión del módulo XBee.....	75
2.5.2 <i>PROGRAMACIÓN DE LOS MICROCONTROLADORES</i>	76
2.5.2.1 Interfaz de trabajo del Bascom AVR - Control Remoto.....	77
2.5.2.1.1 Lectura del teclado hexadecimal.....	77
2.5.2.1.2 Envío de datos por el púrtico DOUT pin 2 del dispositivo XBee.....	78
2.5.2.2 Interfaz de trabajo del Bascom AVR - Módulo Receptor.....	78
2.5.2.2.1 Recepción de los datos provenientes del control remoto.....	80
2.5.2.2.2 Comunicación serial USB entre el módulo receptor y el computador.....	80
2.5.2.2.3 Control del movimiento del eje del motor paso a paso.....	80
2.5.2.2.4 Inicio y control de los sensores infrarrojos.....	81
2.5.3 <i>SOFTWARE PARA EL COMPUTADOR</i>	82
2.5.3.1 Microsoft Visual Basic 6.0.....	82
2.5.3.1.1 <i>Características de Microsoft Visual Basic 6.0</i>	82
2.5.3.2 Interfaz de trabajo del Microsoft Visual Basic 6.0.....	83
2.5.3.3 Diagrama de bloques del Software en el computador.....	83
2.6 CONSTRUCCIÓN DEL PROTOTIPO.....	85
2.6.1 <i>SOFTWARE UTILIZADO</i>	86
2.6.2 <i>DIAGRAMA CIRCUITAL DEL CONTROL REMOTO ELABORADO EN ORCAD</i>	87
2.6.2.1 Circuito impreso y Screen de la tarjeta.....	88
2.6.2.2 Elaboración de la baquelita y tarjeta electrónica ensamblada.....	88
2.6.3 <i>DIAGRAMA CIRCUITAL DEL MÓDULO RECEPTOR ELABORADO EN ORCAD</i>	89
2.6.3.1 Circuito impreso.....	90
2.6.3.2 Screen de tarjeta.....	91

2.6.3.3 Elaboración de la baquelita.....	92
2.6.3.4 Tarjeta electrónica ensamblada.....	93
2.6.4 DIAGRAMA CIRCUITAL DE LOS SENSORES INFRARROJOS ELABORADOS EN ORCAD... 94	
2.6.4.1 Módulo EMISOR.....	94
2.6.4.1.1 Circuito impreso y Screen de la tarjeta.....	94
2.6.4.2 Módulo RECEPTOR.....	95
2.6.4.2.1 Circuito impreso y Screen de la tarjeta.....	95
2.6.4.3 Elaboración de las baquelitas.....	96
2.6.4.4 Tarjetas electrónicas ensambladas.....	96
2.7 COSTO DE LOS EQUIPOS Y MATERIALES UTILIZADOS 97	
2.7.1 MATERIALES DEL MÓDULO CONTROL REMOTO.....	97
2.7.2 MATERIALES DEL MÓDULO RECEPTOR.....	98
2.7.3 MATERIALES PARA LOS SENSORES INFRARROJOS.....	99
2.7.4 OTROS MATERIALES.....	99
CAPÍTULO 3 100	
3.1 PRUEBAS DEL MÓDULO DE CONTROL REMOTO 100	
3.1.1 ANÁLISIS DE LA LECTURA DE COMUNICACIÓN SERIAL.....	101
3.2 PRUEBAS DEL MÓDULO RECEPTOR Y EL MOTOR PASO A PASO 101	
3.3 PRUEBAS DEL CONTROL DE ACCESO 104	
3.3.1 COMUNICACIÓN ENTRE EL SOFTWARE DEL PC Y EL CONTROL DE ACCESO.....	104
3.3.2 ALMACENAMIENTO CORRECTO DE INFORMACIÓN EN EL SISTEMA.....	106
CAPÍTULO 4 110	
CONCLUSIONES.....	110
RECOMENDACIONES.....	111
REFERENCIAS BIBLIOGRÁFICAS.....	112
ANEXOS 115	

RESUMEN

Lo que se busca con el proyecto debido al avance continuo de la tecnología es presentar innumerables herramientas que permitan a las personas facilitar las actividades que realizan, como el entrar o salir de la casa, dando una mejor comodidad, eficiencia y seguridad al usuario.

Capítulo 1. Se ahonda en el tema de los sistemas de control de acceso, microcontroladores, así como las diferentes técnicas inalámbricas que están avanzando continuamente, acerca de los motores, diferentes sensores y la comunicación USB.

Capítulo 2. Se trata sobre el diseño y construcción del prototipo, en el cuál se analizan los requerimientos y funciones de éste, describiendo cada uno de los dispositivos electrónicos que forman parte de los módulos, así como el software que tiene para su comunicación. Y además se detalla el presupuesto utilizado para su construcción.

Capítulo 3. Este capítulo presenta las pruebas realizadas de cada uno de los componentes, mostrando los resultados visuales obtenidos de los módulos y software del computador, además de la interfaz de comunicación USB entre el módulo receptor y el computador.

Capítulo 4. Presentamos conclusiones y recomendaciones del diseño y construcción del prototipo, así como del software y los diferentes aspectos que el usuario debe tomar en cuenta para su construcción.

PRESENTACIÓN

Lo que se busca con la construcción del PROTOTIPO PARA LA APERTURA DE UN GARAJE, es poner al alcance de las personas un control de acceso, haciendo que su vida diaria sea más fácil y cómoda, mediante sistemas no muy complicados y fáciles de usar, permitiéndonos demostrar la potencialidad de los módulos de RF y el avance continuo de la tecnología inalámbrica.

CAPÍTULO 1

CONCEPTOS GENERALES

1.1 SISTEMAS DE CONTROL DE ACCESO

Control de acceso se define como la capacidad de permitir el ingreso y egreso a determinadas áreas, a los usuarios autorizados de acuerdo a horarios y niveles de acceso previamente establecidos.

1.1.1 MÉTODOS DE IDENTIFICACIÓN DE USUARIOS

En un sistema de control de acceso es necesario identificar a la persona o usuario que intenta ingresar a un sector. Esta identificación debe tener métodos que garanticen la independencia con la máquina utilizada y el sistema de comunicación.

Actualmente se conocen tres formas de identificar personas y son las siguientes:

- Por las características físicas: biométricos.
- Por un secreto compartido: contraseñas.
- Por la posesión de un objeto.

1.1.1.1 Sistemas biométricos

Son sistemas que se dedican a la identificación física de individuos a partir de una característica anatómica o un rasgo de su comportamiento. Una característica anatómica tiene la cualidad de ser relativamente estable en el tiempo, como una huella dactilar, la silueta de la mano, patrones de la retina o el iris.

Un rasgo del comportamiento es menos estable, pues depende de la disposición psicológica de la persona, por ejemplo la firma. No cualquier característica anatómica puede ser utilizada con éxito por un sistema biométrico.

1.1.1.2 Control de contraseñas

Una contraseña o clave (en inglés password) es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. La contraseña normalmente debe mantenerse en secreto ante aquellos a quienes no se les permite el acceso. Aquellos que desean acceder a algún lugar se les solicita una clave; si conocen o no la contraseña, se concede o se niega el acceso según sea el caso.

1.1.1.3 Acceso con objetos físicos (tokens¹)

Los tokens son pequeños dispositivos electrónicos utilizados para el control de usuarios y estos pueden ser:

- Memorias que guarden una palabra clave (contraseña).
- Equipos electrónicos donde se crean contraseñas.

El problema de este sistema es el robo o pérdida del token, por lo cual se debe combinar a la contraseña con un sistema biométrico para una mayor seguridad.

1.1.2 TECNOLOGÍAS EMPLEADAS

Los sistemas de control de acceso generalmente emplean tarjetas como elementos de identificación de usuarios y las principales son: sistemas de acceso por teclado, tarjetas magnéticas, tarjetas de proximidad o tarjetas RFID, tarjetas de código de barras, tarjetas de chip o tarjetas inteligentes y lectoras biométricas.

1.1.2.1 Sistema de acceso por teclado

Es un sistema de control que utiliza un teclado para ingresar la clave del usuario al sistema, esta puede ser la misma u otra. Permitiendo controlar los accesos a viviendas y edificios, sin necesidad de utilizar las llaves.

¹ http://es.wikipedia.org/wiki/Token_de_seguridad

Este sistema consiste en ingresar la clave del usuario mediante el teclado, el sistema detecta la clave ingresada la compara con su base de datos interna y determina si el usuario puede ingresar o no a una área determinada.



Figura 1.1 Sistema de acceso con teclado

1.1.2.2 Tarjetas magnéticas

Son unas tarjetas que tienen una banda magnética con un código para identificarlas rápidamente, utilizan señales electromagnéticas de alta y baja energía para registrar y codificar información de la banda, esta puede ser leída por una máquina para identificación instantánea.

Los usos que se les puede dar a estas tarjetas son:

- Tarjeta de crédito y de débito.
- En cerraduras electrónicas.
- Cajas fuertes.
- Vale como pago de un servicio, tiempo de juego en una máquina, hasta para pagar un viaje de colectivo o un pago en línea.



Figura 1.2 Tarjeta de banda magnética

1.1.2.3 Tarjetas de proximidad o tarjetas RFID²

Las tarjetas de proximidad llamadas también RFID son mayormente implementadas en aquellos escenarios donde la fluidez y el ahorro de tiempo son factores importantes en sistemas de control de accesos y de horario en gimnasios, oficinas, almacenes, peajes y autobuses, con una duración de alrededor de 6 ó 7 años pero su costo es mayor al de una tarjeta magnética.

Las tarjetas plásticas de proximidad utilizan microchip con circuitos de radio frecuencia, que poseen internamente una antena y no necesitan ser insertadas en un lector: el chip de la tarjeta se comunica con el lector por radiofrecuencia (RF) e identifica al titular (ID) sin contacto físico.



Figura 1.3 Tarjetas de proximidad o RFID

1.1.2.4 Tarjetas de código de barras

El código de barras es un código basado en la representación mediante un conjunto de líneas paralelas verticales de distinto grosor y espaciado que en su conjunto contienen una determinada información. De este modo, el código de barras permite reconocer rápidamente un artículo en un punto de la cadena logística y así poder realizar inventario o consultar sus características asociadas. Actualmente, el código de barras está implantado masivamente de forma global.

Las bandas magnéticas se utilizan para la identificación de personas, mientras que los códigos de barras se aplican en la identificación de envases, embalajes o etiquetas de los productos, por lo cual los códigos de barras se han transformado

² <http://es.wikipedia.org/wiki/RFID>

en la forma estándar de representar la información en los productos de mercado en un formato accesible para las máquinas, especialmente en los centros comerciales.



Figura 1.4 Tarjeta de código de barras

Entre las ventajas y primeras justificaciones de la implantación del código de barras se encuentran las siguientes:

- Agilidad en etiquetar o poner precios sobre los artículos.
- Rápido control del stock de mercancías.
- Evitar errores de digitación.
- Estadísticas comerciales.
- El consumidor obtiene una relación de artículos en el ticket de compra lo que permite su comprobación y eventual reclamación.

Sus desventajas se atribuyen a la imposibilidad de recordar el precio del producto.

1.1.2.5 Tarjetas de chip o tarjetas inteligentes

Este tipo de tarjeta consiste en un plástico de PVC del tamaño de una tarjeta de crédito, el cual tiene implantado un chip de silicio en uno de sus extremos, sus características físicas y electrónicas están definidas y reguladas a través del estándar ISO 7816³.

³ ISO 7816, es un estándar internacional relacionado con las *tarjetas de identificación* electrónicas, en especial las tarjetas inteligentes, gestionado por la Organización Internacional De Normalización (ISO) y Comisión Electrotécnica Internacional (IEC).

De acuerdo con el tipo de chip que poseen se pueden clasificar en:

- **Tarjetas de memoria**, las cuales simplemente almacenan información y poseen esquemas de seguridad poco sofisticadas.
- **Tarjetas microprocesadoras**, contienen memorias y microprocesadores, que pueden manipular la información almacenada con un sistema operativo propio.

Por su interfaz, las tarjetas inteligentes se clasifican en:

- De contacto, que tiene un chip en la superficie, que entra en contacto con el lector para poder realizar la transacción.
- De inducción, que posee un chip y una antena interna que permite realizar transacciones por medio de RF, con solo acercar la tarjeta al lector.
- Combinada que cuenten con las dos características anteriores.



Figura 1.5 Tarjeta inteligente

1.1.2.6 Lectores biométricos⁴

Este sistema posee un alto grado de seguridad debido a que la información de los usuarios se encuentra codificada, por lo que se requiere de tecnología especial para decodificar una identificación.

La biometría permite, mediante métodos automáticos, el reconocimiento único de humanos basados en uno o más rasgos físicos.

⁴ <http://es.wikipedia.org/wiki/Biometr%C3%ADa>

El término se deriva de las palabras griegas "bios" de vida y "metron" de medida.

La biometría es una herramienta poderosa. Se rige por métodos automatizados de reconocimiento, basados en características fisiológicas o de comportamiento. Las tecnologías aplicadas comúnmente en este campo explotan el reconocimiento de voz, iris, sistemas dactilares y faciales, geometría de manos, olor corporal, reconocimiento del ADN, la forma de la oreja, etc.

Esta tecnología puede ser implementada en instituciones o áreas que necesiten un sistema de alta seguridad.



Figura 1.6 Sistemas de acceso biométricos

1.2 TRANSMISIÓN INALÁMBRICA

1.2.1 COMUNICACIÓN INALÁMBRICA⁵

La comunicación inalámbrica (wireless, sin cables) es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.

La comunicación inalámbrica, que se realiza a través de ondas de radiofrecuencia, facilita la operación en lugares donde la computadora no se

⁵ http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_inal%C3%A1mbrica

encuentra en una ubicación fija, es decir en almacenes, oficinas de varios pisos, etc.

Actualmente, las transmisiones inalámbricas constituyen una eficaz herramienta que permite la transferencia de voz, datos y vídeo sin la necesidad del cableado. Esta transferencia de información se logra a través de la emisión de ondas de radio teniendo dos ventajas: movilidad y flexibilidad del sistema en general.

En general, la tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y una banda específica, de uso libre para transmitir, entre dispositivos.

1.2.2 ESPECTRO ELECTROMAGNÉTICO

Se denomina espectro electromagnético a la distribución energética del conjunto de las ondas electromagnéticas.

Referido a un objeto se denomina espectro electromagnético o simplemente espectro a la radiación electromagnética que emite (espectro de emisión) o absorbe (espectro de absorción) una sustancia. Dicha radiación sirve para identificar la sustancia de manera análoga a una huella dactilar.

Los espectros se pueden observar mediante espectroscopios que, además de permitir observar el espectro, permiten realizar medidas sobre éste, como la longitud de onda, la frecuencia y la intensidad de la radiación.

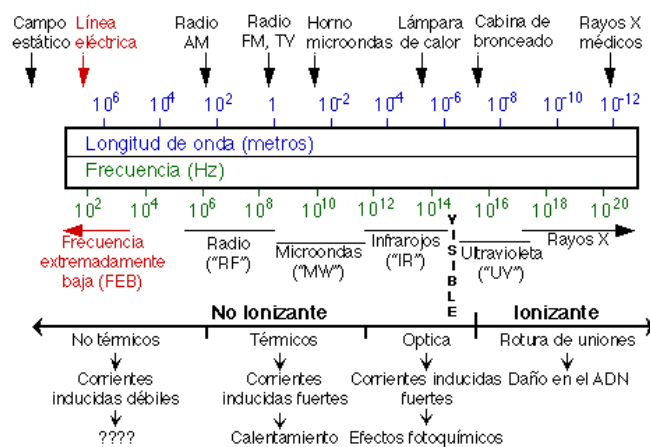


Figura 1.7 Espectro electromagnético

1.2.3 BANDAS DEL ESPECTRO ELECTROMAGNÉTICO

El espectro electromagnético se divide en segmentos o bandas, aunque esta división es inexacta. Existen ondas que tienen una frecuencia, pero varios usos.

Banda	Longitud de onda (m)	Frecuencia (Hz)	Energía (J)
Rayos gamma	< 10 pm	> 30,0 EHz	> $20 \cdot 10^{-15}$ J
Rayos X	< 10 nm	> 30,0 PHz	> $20 \cdot 10^{-18}$ J
Ultravioleta extreme	< 200 nm	> 1,5 PHz	> $993 \cdot 10^{-21}$ J
Ultravioleta cercano	< 380 nm	> 789 THz	> $523 \cdot 10^{-21}$ J
Luz Visible	< 780 nm	> 384 THz	> $255 \cdot 10^{-21}$ J
Infrarrojo cercano	< 2,5 μ m	> 120 THz	> $79 \cdot 10^{-21}$ J
Infrarrojo medio	< 50 μ m	> 6,00 THz	> $4 \cdot 10^{-21}$ J
Infrarrojo lejano/submilimétrico	< 1 mm	> 300 GHz	> $200 \cdot 10^{-24}$ J
Microondas	< 30 cm	> 1 GHz	> $2 \cdot 10^{-24}$ J
Ultra Alta Frecuencia - Radio	< 1 m	> 300 MHz	> $19.8 \cdot 10^{-26}$ J
Muy Alta Frecuencia - Radio	< 10 m	> 30 MHz	> $19.8 \cdot 10^{-28}$ J
Onda Corta - Radio	< 180 m	> 1,7 MHz	> $11.22 \cdot 10^{-28}$ J
Onda Media - Radio	< 650 m	> 650 kHz	> $42.9 \cdot 10^{-29}$ J
Onda Larga - Radio	< 10 km	> 30 kHz	> $19.8 \cdot 10^{-30}$ J
Muy Baja Frecuencia - Radio	> 10 km	< 30 kHz	< $19.8 \cdot 10^{-30}$ J

Tabla 1.1 Bandas del espectro electromagnético

1.2.4 MEDIOS DE TRANSMISIÓN⁶

El medio de transmisión constituye el soporte físico a través del cual emisor y receptor pueden comunicarse en un sistema de transmisión de datos.

⁶ <http://www.monografias.com/trabajos17/medios-de-transmision/medios-de-transmision.shtml>

Distinguimos dos tipos de medios: guiados y no guiados. En ambos casos la transmisión se realiza por medio de ondas electromagnéticas.

1.2.4.1 Medios de transmisión guiados

Los medios guiados conducen (guían) las ondas a través de un camino físico y sólido, ejemplos de estos son el cable coaxial, la fibra óptica y el par trenzado.

1.2.4.1.1 Guía de onda

La guía de onda es un medio de comunicación muy usado, el cual opera en el rango de las frecuencias comúnmente llamadas microondas (en el orden de GHz).

El ancho de banda es extremadamente grande y es usado principalmente cuando se requiere bajas pérdidas en la señal bajo condiciones de muy alta potencia como es el caso desde una antena de microondas al receptor/transmisor de radio frecuencia.

1.2.4.1.2 Cable par trenzado

Consiste en hilos de cobre aislados por una cubierta plástica y torzonada entre sí. Debido a que puede haber acoples entre pares, éstos se trenza con pasos diferentes. La utilización del trenzado tiende a disminuir la interferencia electromagnética, es el medio más utilizado debido a su comodidad de instalación y a su precio.

1.2.4.1.3 Cable coaxial

Consiste en un cable conductor interno (cilíndrico) separado de otro cable conductor externo por anillos aislantes o por un aislante macizo. Todo esto se recubre por otra capa aislante que es la funda del cable.

Este cable, aunque es más caro que el par trenzado, se puede utilizar a más larga distancia, con velocidades de transmisión superiores, menos interferencias y permite conectar más estaciones.

Se suele utilizar para televisión, telefonía a larga distancia, redes de área local, conexión de periféricos a corta distancia, para transmitir señales analógicas o digitales. Sus inconvenientes principales son: atenuación, ruido térmico y ruido de intermodulación.

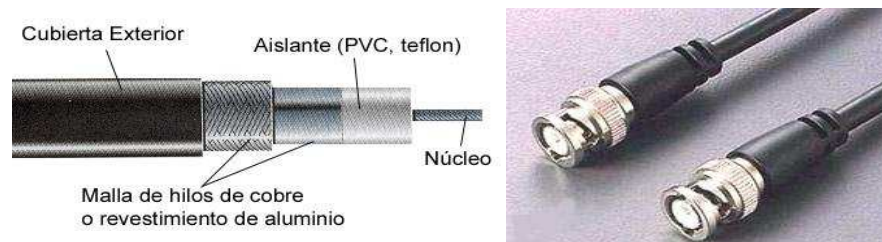


Figura 1.8 Cable coaxial

1.2.4.1.4 Fibra óptica

Es el medio de transmisión más novedoso dentro de los guiados y su uso se está masificando en todo el mundo reemplazando el par trenzado y el cable coaxial en casi todos los campos. En estos días lo podemos encontrar en la televisión por cable y la telefonía. En este medio los datos se transmiten mediante una haz confinado de naturaleza óptica, de ahí su nombre, es mucho más caro y difícil de manejar pero sus ventajas sobre los otros medios lo convierten muchas veces en una muy buena elección al momento de observar rendimiento y calidad de transmisión.

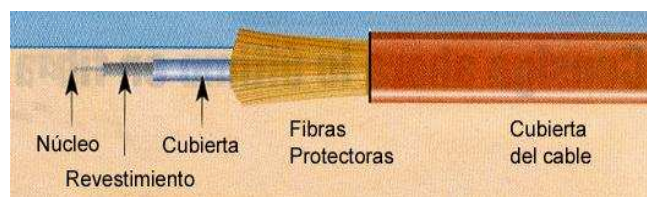


Figura 1.9 Cable de fibra óptica

1.2.4.2 Medios de transmisión no guiados

Los medios no guiados proporcionan un soporte para que las ondas se transmitan, pero no las dirigen; como ejemplo de ello son el aire y el vacío.

1.2.4.2.1 Infrarrojos

Poseen las mismas técnicas que emplean la fibra óptica, usando como medio de transmisión el aire. Y son excelentes para distancias cortas.

No tienen problemas de seguridad ni de interferencias ya que estos rayos no pueden atravesar los objetos. Tampoco es necesario permiso para su utilización (en microondas y ondas de radio si es necesario un permiso para asignar una frecuencia de uso).

1.2.4.2.2 Microondas

En un sistema de microondas se usa el espacio aéreo como medio físico de transmisión. La información se transmite en forma analógica y digital a través de ondas de radio de muy corta longitud teniendo una línea de vista entre los dos puntos.

Las microondas son ondas electromagnéticas cuyas frecuencias se encuentran dentro del espectro de las super altas frecuencias (SHF), utilizándose para las redes inalámbricas la banda de los 18-19GHz. Éstas redes tienen una propagación muy localizada y un ancho de banda que permite alcanzar los 15Mbps.

1.2.4.2.3 Satélite

Tiene la ventaja de la libertad geográfica y su alta velocidad, etc. Pero su desventaja se basa en el retardo de las transmisiones debido a que viajan grandes distancias.

1.2.4.2.4 Radio enlaces de VHF y UHF

Estas bandas cubren aproximadamente desde 55 a 550MHz. Son también omnidireccionales, pero a diferencia de las anteriores la ionósfera es transparente a ellas. Su alcance máximo es de un centenar de kilómetros, y las velocidades que permite es del orden de los 9600bps.

1.2.4.2.5 Ondas cortas

Se llaman radio de alta frecuencia, su ventaja es transmitir a grandes distancias con poca potencia y desventaja es que son menos confiables que otras ondas.

1.3 RADIO FRECUENCIA

El término radio frecuencia, también denominado espectro de radio frecuencia o RF, se aplica a la porción menos energética del espectro electromagnético, situada entre unos 3Hz y unos 300GHz.

1.3.1 CLASIFICACIÓN DE RADIO FRECUENCIA

El espectro de radiofrecuencia se divide en las siguientes bandas:

Nombre	Abreviatura inglesa	Banda ITU	Frecuencias	Longitud de onda
			< 3 Hz	> 100.000 km
Extremely Low Frequency	ELF	1	3-30 Hz	100.000-10.000 km
Super Low Frequency	SLF	2	30-300 Hz	10.000-1.000 km
Ultra Low Frequency	ULF	3	300-3.000 Hz	1.000-100 km
Very Low Frequency	VLF	4	3-30 kHz	100-10 km
Low Frequency	LF	5	30-300 kHz	10-1 km
Medium Frequency	MF	6	300-3.000 kHz	1 km-100 m
High Frequency	HF	7	3-30 MHz	100-10 m
Very High Frequency	VHF	8	30-300 MHz	10-1 m
Ultra High Frequency	UHF	9	300-3.000 MHz	1 m-100 mm
Super High Frequency	SHF	10	3-30 GHz	100-10 mm
Extremely High Frequency	EHF	11	30-300 GHz	10-1 mm
			> 300 GHz	< 1 mm

Tabla 1.2 Clasificación de radio frecuencia

1.3.2 COMUNICACIÓN POR RADIO FRECUENCIA

La transmisión de datos entre equipos electrónicos sin cables se está aplicando cada vez más debido a los medios tecnológicos actuales, permitiendo una facilidad en el desplazamiento y reubicación, lo cual evita la necesidad de establecer un cableado y su instalación es rápida.

Los dispositivos utilizados para las comunicaciones vía radio frecuencia han ido evolucionando con el fin de simplificar la transmisión y recepción de datos, primero se utilizaron módulos de RF con componentes discretos unidireccionales, posteriormente con la aparición de circuitos transmisores completamente integrados con funciones de emisor y receptor en diferentes bandas de frecuencias que gradualmente se fueron estabilizando por regiones, han permitido utilizarlos en diferentes aplicaciones: industrial, comercial y médico como por ejemplo en: controles remotos, transmisión de datos en sensores, sistemas de adquisición de datos, en monitorización médica o salud, etc.

Los sistemas de radiofrecuencia emplean el espacio como medio de transmisión, debido a que tanto en el transmisor como en el receptor la información está contenida como ondas electromagnéticas guiadas y que para poderlas transmitir por el espacio deben ser ondas electromagnéticas no guiadas se necesitan de un transductor como una antena, la que se encargara de realizar la conversión de ondas guiadas a no guiadas en el transmisor y viceversa en el receptor.

Estos sistemas permiten comunicaciones de corto y medio alcance, además pueden atravesar obstáculos y paredes. Hay una serie de parámetros que influyen en estos sistemas por ejemplo:

- El alcance depende de la frecuencia de trabajo, a mayor frecuencia mayor alcance.
- El alcance depende de la potencia de salida, pero también de la sensibilidad de recepción.

- La potencia de salida y la sensibilidad del receptor también dependen de la antena, es decir el tipo de antena (externa, cerámica o del circuito impreso) y de sus características.
- Y finalmente depende del entorno del medio, es decir no es lo mismo en el aire libre en campo abierto o en la ciudad, dentro de un edificio, con paredes sencillas o con muros de cemento.

Concluyendo, el rango de trabajo en RF depende de: la frecuencia, la potencia de salida, la sensibilidad de recepción, las características de la antena y el entorno de trabajo.

1.3.2.1 Tipos de comunicaciones inalámbricas por RF

Las comunicaciones inalámbricas por RF se pueden dividir en las que no cumplen ningún protocolo estándar (llamadas propietarias) y las que cumplen un protocolo estándar y por otro lado en las frecuencias de trabajo (llamadas menores a 1GHz y las de 2.4GHz). Las menores a 1GHz van desde 300 a 900MHz (según normativas en cada zona) y las de 2.4GHz que están normalizadas en todo el mundo, que a la vez definen velocidad de transmisión o ancho de banda y campo de aplicación.

1.3.2.2 Bandas ISM⁷ (Industriales, Científicas y Médicas)

Estas bandas de frecuencia, llamadas bandas ISM, estaban anteriormente limitadas a instrumentos científicos, médicos e industriales, pero la FCC (Comisión Federal de Comunicaciones) permitió la operación sin licencia de dispositivos que utilizan 1 vatio de energía o menos en tres bandas de frecuencia:

- La primera banda de 315MHz en USA, 433MHz y 868MHz en Europa.

⁷ http://es.wikipedia.org/wiki/Banda_ISM

- La segunda banda de 2.4GHz que está normalizada en casi todo el mundo con algunas diferencias, con respecto a la potencia máxima de salida y la banda de 5.6GHz.

Los dispositivos inalámbricos ahora se están diseñando para trabajar en la banda de 2.4GHz por lo que pueden funcionar en cualquier lugar debido a que esta banda está disponible para todo el mundo sin necesidad de licencia, no como las otras dos bandas que varían sus frecuencias dependiendo de la región.

En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones WLAN (Wi-Fi) o WPAN (Bluetooth).

Las bandas ISM fueron definidas por la UIT, en el artículo 5 de las Regulaciones Radio (RR), concretamente los puntos 5.138 y 5.150.

1.3.2.3 ZigBee⁸ (Estándar de comunicaciones Inalámbricas IEEE 802.15.4)

ZigBee es el nombre de la especificación de un conjunto de protocolos de comunicación de alto nivel para su utilización con radios digitales de **bajo consumo**, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías, como la domótica, debido a su bajo consumo, su sistema de comunicaciones vía radio (con topología de red en malla) y su fácil integración (para fabricar nodos con muy poca electrónica).

Sus aplicaciones de propósito general se dan en el control industrial, albergar sensores empotrados, recolectar datos medios, ejercer labores de detección de humo o intrusos y domótica.

⁸ http://es.wikipedia.org/wiki/IEEE_802.15.4

1.3.2.3.1 Características generales ZigBee/IEEE 802.15.4

- Doble capa física PHY (868/915MHz y 2.4GHz).
- Razón de transmisión de 250Kbps (a 2.4GHz), 40Kbps (a 915MHz) y 20Kbps (a 868MHz).
- Acceso de canal mediante CSMA-CA (acceso múltiple por detección de portadora con evasión de colisiones).
- Bajo consumo de potencia (la duración de la batería dura varios meses incluso años).
- Múltiples topologías: estrella, peer to peer, malla, árbol.
- Rango de alcance de 50m típico (de 5m a 500m basado sobre el medio ambiente).

1.3.2.3.2 Estructura de la trama

La estructura de la trama ha sido diseñada para mantener la complejidad al mínimo, al mismo tiempo hacerla lo suficientemente sólida para su transmisión en un canal ruidoso, cada uno de los protocolos de capa añade a la estructura encabezados.

El estándar IEEE 802.15.4 proporciona tres niveles de seguridad:

- Sin seguridad, ejemplo: aplicaciones de publicidad.
- Control de acceso a listas.
- Seguridad con clave simétrica.

1.3.2.3.3 Topologías de Redes ZigBee

La capa de red soporta múltiples configuraciones de red incluyendo estrella, árbol y malla.

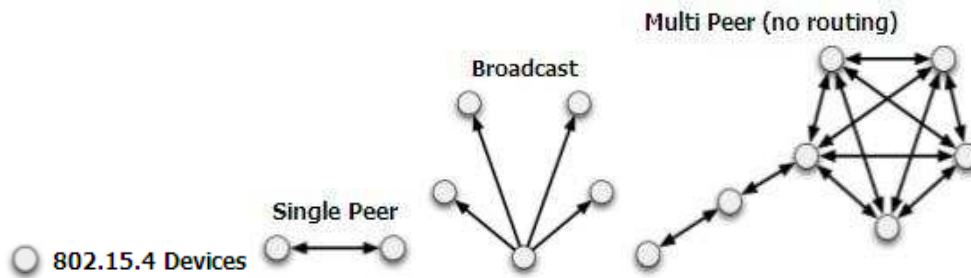


Figura 1.10 Redes del Protocolo IEEE 802.15.4

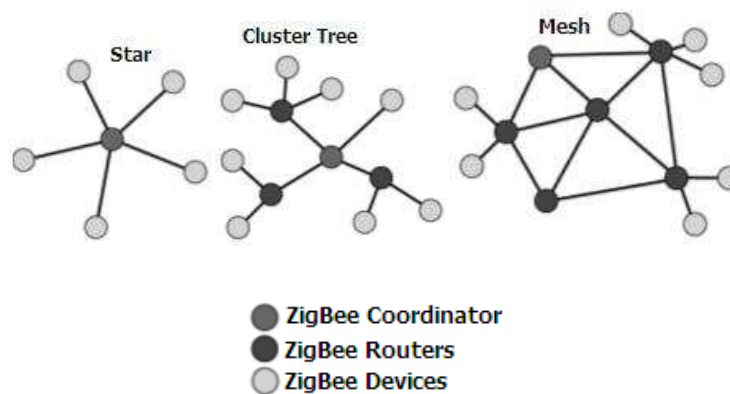


Figura 1.11 Redes del Protocolo ZigBee

1.3.3 CONCEPTOS GENERALES PARA LA COMUNICACIÓN

1.3.3.1 Comunicación de datos⁹

Es el proceso de comunicar información en forma binaria entre dos o más puntos. Y requiere de cuatro elementos básicos que son:

- **Emisor:** dispositivo que transmite los datos.
- **Mensaje:** lo forman los datos a ser transmitidos.
- **Medio:** consiste en el recorrido de los datos desde el origen hasta su destino.

⁹ <http://www.monografias.com/trabajos/redesconcep/redesconcep.shtml>

- **Receptor:** dispositivo de destino de los datos.

1.3.3.2 Formas de transmisión

1.3.3.2.1 Transmisión en Serie.

Los bits se transmiten de uno a uno sobre una línea única. Se utiliza para transmitir a larga distancia.

1.3.3.2.2 Transmisión en Paralelo

Los bits se transmiten en grupo sobre varias líneas al mismo tiempo. Es utilizada dentro del computador.

La transmisión en paralela es más rápida que la transmisión en serie pero en la medida que la distancia entre equipos se incrementa (no debe sobrepasarse la distancia de 100 pies), no solo se encarecen los cables sino que además aumenta la complejidad de los transmisores y los receptores de la línea a causa de la dificultad de transmitir y recibir señales de pulsos a través de cables largos.

1.3.3.3 Tipos de transmisión

1.3.3.3.1 Transmisión Asíncrona: cada byte de datos incluye señales de arranque y parada al principio y al final. La misión de estas señales consiste en:

- Avisar al receptor de que está llegando un dato.
- Darle suficiente tiempo al receptor de realizar funciones de sincronismo antes de que llegue el siguiente byte.

1.3.3.3.2 Transmisión Síncrona: se utilizan canales separados de reloj que administran la recepción y transmisión de los datos. Su misión principal es alertar al receptor de la llegada de los datos. Al inicio de cada transmisión se emplean unas señales preliminares llamadas:

- Bytes de sincronización en los protocolos orientados a byte.
- Flags en los protocolos orientados a bit.

1.4 MICROCONTROLADORES¹⁰

1.4.1 INTRODUCCIÓN

Los microcontroladores están conquistando el mundo, pero la invasión acaba de comenzar y el nacimiento del siglo XXI es testigo de la conquista masiva de éstos diminutos computadores, que gobernarán la mayor parte de los aparatos que se fabrican y utilizan hoy en día.

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

El funcionamiento y aplicaciones de los microcontroladores permiten profundizar en los aspectos tecnológicos de las arquitecturas de las nuevas computadoras, convirtiéndose de esta forma en una herramienta útil para el desarrollo de varias aplicaciones que contribuyen al avance tecnológico y desarrollo integral de la sociedad.

1.4.2 DEFINICIÓN

Un microcontrolador es un circuito integrado programable cuya arquitectura contiene todos los componentes de un microcomputador, es decir posee CPU, memoria RAM, ROM y circuitos de entrada y salida, todos estos contenidos en un único chip. Cuando se programa el microcontrolador solo se emplea para el funcionamiento de la única tarea a la que ha sido programado.

En el microcontrolador todas las partes del computador están contenidas en su interior y solo salen al exterior las líneas que gobiernan los periféricos.

¹⁰ <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

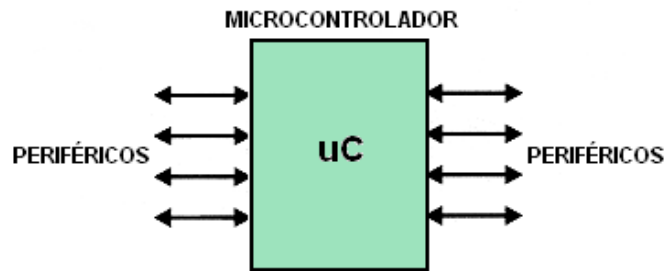


Figura 1.12 Microcontrolador

1.4.3 ARQUITECTURA DEL MICROCONTROLADOR¹¹

Debido a la necesidad de conseguir elevados rendimientos en este proceso, se ha desembocado en el empleo generalizado de procesadores de arquitectura Harvard frente a los tradicionales que seguían la arquitectura de von Neumann.

La arquitectura tradicional de sistemas digitales programables se basa en el esquema propuesto por John Von Neumann.

En este modelo la unidad central de proceso o CPU se conecta a una memoria única, donde coexistían datos e instrucciones del programa, a través de un sistema de buses. Figura 1.13.

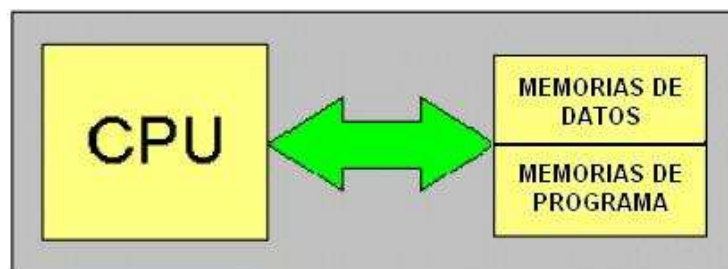


Figura 1.13 Arquitectura de Von Neumann

¹¹ <http://www.mitecnologico.com/Main/ArquitecturaDelMicrocontrolador>

El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus de datos de la memoria exterior utilizada, que es de 8 bits. Un microprocesador con un bus de 8 bits que lo conecta con la memoria deberá manejar datos e instrucciones de una o más unidades de 8 bits de longitud. Cuando deba acceder a una instrucción o dato de más de un byte (8 bits) de longitud, deberá de realizar más de un acceso a la memoria. Por otro lado este bus único limita la velocidad de operación del microprocesador, ya que no se puede buscar en la memoria una nueva instrucción antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior.

Es decir las dos principales limitaciones de esta arquitectura son:

- La longitud de las instrucciones está limitada por la unidad de longitud de los datos, por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones complejas.
- La velocidad de operación (ancho de banda) está limitada por el efecto de cuello de botella que significa un bus único para datos e instrucciones que impiden superponer ambos tiempos de acceso.

En la arquitectura Harvard son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso.

Esta dualidad, además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos.

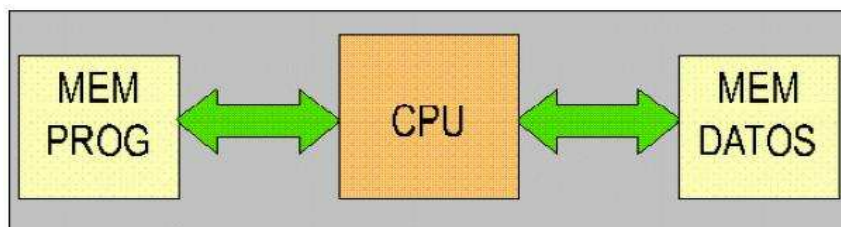


Figura 1.14 Arquitectura Harvard

1.4.4 PARTES DEL MICROCONTROLADOR¹²

Los componentes fundamentales que componen un microcontrolador son: la CPU o unidad central de proceso, la memoria y las unidades de entrada/salida.

1.4.4.1 La CPU o unidad central de proceso

Podemos decir que la CPU, siglas en inglés de unidad central de proceso, es el núcleo del microcontrolador. Que se encarga de ejecutar las instrucciones almacenadas en la memoria. Es lo que habitualmente llamamos procesador o microprocesador, término que a menudo se confunde con el de microcontrolador. En esta línea cabe aclarar que, tal y como estamos viendo, ambos términos no son lo mismo: el microprocesador es una parte de un microcontrolador y sin él no sería útil; un microcontrolador, en cambio, es un sistema completo que puede llevar a cabo de forma autónoma una labor.

Podemos ver al microprocesador como el cerebro de una persona y al microcontrolador como el cuerpo: el cerebro se encarga de procesar toda la información, pero necesita a los demás órganos para funcionar. De la misma forma, el microprocesador, que únicamente se encarga de ejecutar las instrucciones, necesita, por un lado, un lugar donde almacenarlas, es decir, la memoria, y por otro, un medio para interactuar con el exterior, es decir, los dispositivos de entrada/salida.

1.4.4.2 Memorias

Entendemos por memoria los diferentes componentes del microcontrolador que se emplean para almacenar información durante un periodo determinado de tiempo. La información que necesitaremos durante la ejecución del programa será, por un lado, el propio código, y por otro, los diferentes datos que usemos durante la ejecución del mismo.

¹² <http://trecedb.wordpress.com/2009/02/13/partes-del-microcontrolador/>

La diferente naturaleza de información que hay que almacenar hace necesario el uso de diferentes tipos de memorias. Sin hacer especial énfasis en este apartado, sí habrá que tener en cuenta una clasificación básica, que distingue entre memoria volátil y no volátil. La primera es aquella que pierde la información que almacena al desconectarla de la alimentación; la segunda, como resulta obvio, no se pierde la información desconectando la alimentación. Por lo tanto, se hace evidente que al menos la memoria de programa deberá ser no volátil: no sería práctico que el programa grabado en el microcontrolador se borrara cada vez que apagamos el dispositivo. Con respecto a la memoria de datos, según la situación puede interesarnos la una o la otra.

La existencia de dos tipos de información a almacenar, nos lleva también al concepto de arquitectura. Entendemos por arquitectura el conjunto de componentes del microcontrolador y la forma en la que éstos se relacionan. Las más conocidas son la arquitectura Harvard y la arquitectura Von Neumann, como se explicó anteriormente en la arquitectura del microcontrolador.

1.4.4.3 Unidades de entrada/salida

Ya hemos visto qué parte del microcontrolador ejecuta las instrucciones de nuestro programa y de dónde las lee. Ahora sólo nos falta ver qué son las unidades de entrada/salida. Las unidades de entrada/salida son los sistemas que emplea el microcontrolador para comunicarse con el exterior. Imaginemos una televisión: por un lado tiene un dispositivo de salida, como es la pantalla, y por otro lado, de entrada, como son los botones de subir o bajar volumen y de cambio de canal. Así, los dispositivos de entrada nos permitirán introducir información en el microcontrolador y los de salida nos servirán para que éste la saque al exterior.

Estos dispositivos, son fundamentales a la hora de desarrollar una aplicación: ¿qué sentido tendría diseñar un programa que hiciera ciertas operaciones para al final no mostrar ningún resultado al usuario final?.

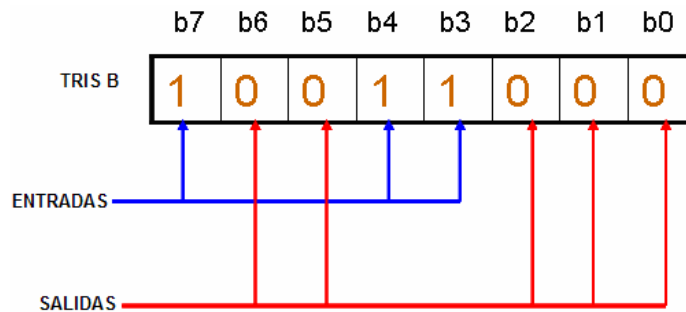


Figura 1.15 Ejemplo de configuración de líneas I/O

1.4.4.4 Recursos auxiliares

Cada fabricante ofrece microcontroladores para funciones específicas, por lo que algunos amplían sus complementos para dar mayor potencia, rendimiento y flexibilidad, por lo tanto minimizarán los costos, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- **Circuito de reloj.**- encargado de generar los pulsos usados en la sincronización para el funcionamiento de todo el sistema.
- **Temporizadores.**- utilizados para controlar los tiempos.
- **Perro guardián o watchdog.**- usado para una reinicialización, cuando el programa queda bloqueado por fallo del software u otra cosa.
- **Protección ante falla de alimentación.**- se trata de un circuito que resetea el microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo.
- **Estado de reposo o de bajo consumo.**- utilizado para minimizar el consumo de energía. En dicho estado se detiene el reloj principal y se congelan sus circuitos asociados, quedando el microcontrolador en un sumido y profundo estado de reposo.
- **Convertor A/D y D/A.**- es el encargado de convertir las señales analógicas en digitales y viceversa.
- **Comparador analógico.**- utilizado para verificar el valor de una señal analógica.

- **Modulador de anchura de pulsos o PWM.-** son circuitos que proporcionan en su salida pulsos de anchura variable, que se ofrecen al exterior a través de los terminales del encapsulado.
- **Puertas de E/S digitales.-** son líneas que toman el nombre de puertos o pórtilos, estas líneas digitales pueden configurarse como entradas o salidas cargando con un 1 o 0 en el bit correspondiente del registro que se quiera configurar.

1.5 MOTORES¹³

1.5.1 INTRODUCCIÓN

Un motor es una máquina eléctrica que transforma energía eléctrica en energía mecánica por medio de interacciones electromagnéticas. Algunos de los motores son reversibles, pueden transformar energía mecánica en energía eléctrica funcionando como generadores.

Son ampliamente utilizados en instalaciones industriales, comerciales y particulares. Pueden funcionar conectados a una red de suministro eléctrico o a baterías. Así, en automóviles se están empezando a utilizar en vehículos híbridos para aprovechar las ventajas de ambos.



Figura 1.16 Diversos motores eléctricos

¹³ http://es.wikipedia.org/wiki/Motor_el%C3%A9ctrico

1.5.2 PRINCIPIO DE FUNCIONAMIENTO

Los motores de corriente alterna y los de corriente continua se basan en el mismo principio de funcionamiento, el cual establece que si un conductor por el que circula una corriente eléctrica se encuentra dentro de la acción de un campo magnético, éste tiende a desplazarse perpendicularmente a las líneas de acción del campo magnético.

El conductor tiende a funcionar como un electroimán debido a la corriente eléctrica que circula por el mismo adquiriendo de esta manera propiedades magnéticas, que provocan, debido a la interacción con los polos ubicados en el estator, el movimiento circular que se observa en el rotor del motor.

Partiendo del hecho de que cuando pasa corriente por un conductor se produce un campo magnético, además si lo ponemos dentro de la acción de un campo magnético potente, el producto de la interacción de ambos campos magnéticos hace que el conductor tienda a desplazarse produciendo así la energía mecánica. Dicha energía es comunicada al exterior mediante un dispositivo llamado flecha.

1.5.3 MOTORES DE CORRIENTE CONTINUA¹⁴

El motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica, principalmente mediante el movimiento rotatorio. Esta máquina de corriente continua es una de las más versátiles en la industria. Su fácil control de posición, paro y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos.

Los motores de corriente continua se clasifican según la forma como estén conectados, en:

- Motor serie.
- Motor compound.

¹⁴ <http://html.rincondelvago.com/motores-de-corriente-continua.html>

- Motor shunt.
- Motor eléctrico sin escobillas.

Además de los anteriores, existen otros tipos que son utilizados en electrónica:

- Motor paso a paso.
- Servomotor.
- Motor sin núcleo.

1.6 SENSORES¹⁵

Un sensor es un dispositivo capaz de medir magnitudes físicas o químicas, llamadas variables de instrumentación, en variables eléctricas. Estas variables de instrumentación dependen del tipo de sensor a utilizar.

1.6.1 CARACTERÍSTICAS DE UN SENSOR

Entre las características técnicas de un sensor destacan las siguientes:

- **Rango de medida:** dominio en la magnitud medida en el que puede aplicarse el sensor.
- **Precisión:** es el error de medida máximo esperado.
- **Offset o desviación de cero:** valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el offset.
- **Linealidad o correlación lineal.**
- **Sensibilidad de un sensor:** relación entre la variación de la magnitud de salida y la variación de la magnitud de entrada.
- **Resolución:** mínima variación de la magnitud de entrada que puede apreciarse a la salida.

¹⁵ <http://es.wikipedia.org/wiki/Sensor>

- **Rapidez de respuesta:** puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- **Derivas:** son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- **Repetitividad:** error esperado al repetir varias veces la misma medida.

1.6.2 TIPOS DE SENSORES¹⁶

1.6.2.1 Interruptores final de carrera

El microswitch es un conmutador de 2 posiciones con retorno a la posición de reposo y viene con un botón o con una palanca de accionamiento, la cual también puede traer una ruedita.

En estado de reposo la patita común (COM) y la de contacto normal cerrado (NC), están en contacto permanente hasta que la presión aplicada a la palanca del microswitch hace saltar la pequeña platina acerada interior y entonces el contacto pasa de la posición de normal cerrado a la de normal abierto (NO), se puede escuchar cuando el microswitch cambia de estado, porque se oye un pequeño clic, esto sucede casi al final del recorrido de la palanca.

1.6.2.2 Productos infrarrojos

La optoelectrónica es la integración de los principios ópticos y la electrónica de semiconductores. Los componentes optoelectrónicos son sensores fiables y económicos. Se incluyen diodos emisores de infrarrojos (IREDs), sensores y montajes.

¹⁶ http://www.profesormolina.com.ar/tecnologia/sens_transduct/tipos.htm

1.6.2.2.1 Sensores infrarrojos

Los diodos infrarrojos son dispositivos semiconductores que transmiten y detectan señales luminosas, la luz emitida está en la banda de los infrarrojos (invisible). Se hace uso de este tipo de emisión de luz con la intención de detectar obstáculos sin que exista contacto físico con el mismo.

En los diodos infrarrojos el emisor y el receptor suelen ser elementos separados el primero suele ser un diodo de luz (LED) y el receptor en fotodiodo, cuando se establece la comunicación entre el emisor y el receptor, es posible realizar una transmisión de datos.

1.6.2.2.1.1 Estructura

Los diodos emisores de luz infrarroja (IRED) están comúnmente encapsulados en una cubierta de plástico de mayor resistencia que las de cristal. El material que compone el diodo es importante ya que el color de la luz emitida por el LED depende únicamente del material y del proceso de fabricación (principalmente de los dopados).



Figura 1.17 LEDs infrarrojos

Los diodos IRED, además tienen geometrías especiales para evitar que la radiación emitida sea reabsorbida por el material circundante del propio diodo, lo que sucede en los convencionales.

El material del que están compuestos los diodos infrarrojos es de Arseniuro de Galio (GaAs) o de Arseniuro de Galio y Aluminio (AlGaAs).

1.6.2.2.1.2 Diodo emisor de luz infrarroja

Es un dispositivo que emite un rayo de flujo radiante cuando se polariza en forma directa, además varía sus propiedades ópticas con la aplicación de un determinado potencial.

Un diodo emisor emite una luz en la banda del infrarrojo, sobre la que se superpone una señal, convenientemente modulada con la información de control.

Estos dispositivos transforman energía eléctrica en energía luminosa.

1.6.2.2.1.3 Fotodiodo receptor

Un fotodiodo receptor es un dispositivo semiconductor de unión PN, tiene la misión de extraer de la señal recibida la información de control, la cantidad de corriente eléctrica es proporcional a la cantidad de luz que lo ilumina. Esta corriente eléctrica fluye en sentido opuesto a la flecha del diodo, es decir, está polarizado en sentido inverso.



Figura 1.18 Sentido de la corriente generada

El fotodiodo se puede utilizar como dispositivo detector de luz, pues convierte la luz en electricidad y esta variación de electricidad es la que se utiliza para informar que hubo un cambio en el nivel de iluminación sobre el fotodiodo.

Se define la sensibilidad del fotodiodo al incremento de intensidad al polarizar el dispositivo en inversa por unidad de intensidad de luz, expresada en luxes o en mW/cm^2 .

1.7 COMUNICACIÓN USB¹⁷

1.7.1 INTRODUCCIÓN AL USB

El USB¹⁸ o Universal Serial Bus es una interfaz para la transmisión serial de datos y distribución de energía desarrollada por empresas líderes del sector de las telecomunicaciones y de los ordenadores que han sido introducidos en el mercado de los computadores, periféricos para mejorar las lentas interfaces serie RS-323 y paralelo.

La tecnología USB se basa en una arquitectura de tipo serial. Sin embargo, es una interfaz de entrada/salida mucho más rápida que los puertos seriales estándar, la arquitectura serial se utilizó para este tipo por dos razones principales:

- Brinda al usuario una velocidad de reloj mucho más alta que la interfaz paralela debido a que este tipo de interfaz no admite frecuencias demasiado altas.
- Los cables seriales, resultan mucho más económicos que los cables paralelos.

1.7.2 CARACTERÍSTICAS GENERALES DEL USB

1. Los cables USB están diseñados para trabajar en 4 hilos, de los cuales dos son de alimentación y dos de datos (data+ y data-).

¹⁷ <http://es.scribd.com/doc/17048589/Comunicacion-Usb>

¹⁸ http://es.wikipedia.org/wiki/Universal_Serial_Bus

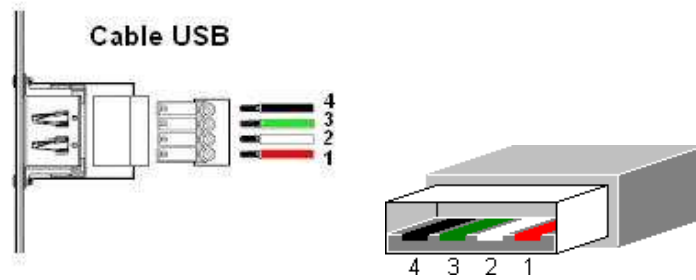


Figura 1.19 Diagrama de pines del USB

Nº de pin	Señal	Cable Color
1	+ VCC	Rojo
2	Datos -	Blanco
3	Datos +	Verde
4	GND	Negro

Tabla 1.3 Conexión USB dispositivo de los contactos

- La distancia máxima del cable que se puede conectar entre el computador y el periférico es de 5m, pero se puede incrementar la distancia utilizando un hub.
- Su conector es más pequeño en comparación con los conectores seriales o paralelos.



Figura 1.20 Conectores USB tipo A y tipo B

- El bus USB puede soportar 3 niveles de transferencias de información.

High Speed a 480Mbps.

Full Speed a 12Mbps.

Low Speed a 1.5Mbps.

Los computadores actuales soportan los tres tipos.

5. Tanto los drivers USB como la recepción de los datos y cables garantizan una interfaz que elimina el ruido que pueda provocar error en los datos.

Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de $90\Omega \pm 15\%$, cuyos hilos se denominan D+ y D-. Estos colectivamente, utilizan señalización diferencial en full dúplex para combatir los efectos de ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples. Los niveles de transmisión de la señal varían de 0 a 0.3v para bajos (ceros) y de 2.8 a 3.6v para altos (unos) en las versiones 1.0 y 1.1 y en $\pm 400\text{mV}$ en altas velocidades (2.0).

En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de 45Ω a tierra o un diferencial de 90Ω para acoplar la impedancia del cable. Este puerto solo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100mA por cada puerto: sin embargo, en caso de que estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de 4 puertos), entonces la energía del USB se asignará en unidades de 100mA hasta un máximo de 500mA por puerto.

Posee un circuito de bajo consumo y un código que automáticamente puede poner el sistema en reposo logrando de esta forma ahorro de energía y un tiempo de recarga de baterías más largo.

El USB originalmente utiliza una interfaz cableada, pero ahora existen dispositivos inalámbricos que usa el USB para conectarse con los computadores.

Para la comunicación con los periféricos, el USB ha definido clases que especifican los requerimientos de los dispositivos. Estos requerimientos son 3 como mínimo que se debe cumplir:

- Detección del dispositivo USB, en la conexión y desconexión.

- Al tener un nuevo dispositivo conectado, este debe ser capaz de descubrir como intercambiar información.
- Provee un mecanismo que habilita el software, para la comunicación entre el hardware del USB y los periféricos.

El periférico es responsable de responder las peticiones para enviar y recibir datos usados en la identificación y configuración del dispositivo, así como su lectura y escritura,

1.7.3 LIMITACIONES

Como todo equipo o tecnología tienen notables ventajas, también tienen limitaciones que restringen al usuario de utilizar dicha tecnología, el USB no es una excepción. A continuación se describe las limitaciones que tiene en Estándar USB para su uso:

- El USB fue diseñado como un bus de expansión fijo con la expectativa que los periféricos estarían cerca, su cable puede llegar a ser hasta 5m. A diferencia de otras interfaces donde se permite mayor alcance, tales como el RS-232, RS485, Ethernet y IEEE-1394. Se puede incrementar la distancia a 30m colocando 5 hubs. Otra opción cuando se necesita transmitir a distancias grandes es cambiar de UBS a RS-485.
- En la comunicación USB, solo se puede comunicar el periférico con el host, pero no el host con el host, tampoco lo hace entre periféricos como lo hacen otras interfaces como IEEE-1394.
- El USB no puede transmitir información simultánea a diferentes dispositivos en el bus, los envía individualmente. Ethernet o IEEE-1394 poseen esta habilidad.

1.7.4 CABLES Y CONECTORES USB

En lo que respecta a conectores hay que decir que son del tipo ficha (o conector) y receptáculo, y son de dos tipos: serie **A** y serie **B**. Ambos son polarizados (solo pueden conectarse en una posición) y utilizan sistemas de presión para sujetarse.

- Los de tipo **A** utilizan la hembra en el sistema anfitrión y suelen usarse en dispositivos en los que la conexión es permanente, por ejemplo: ratones y teclados.
- Los de tipo **B** utilizan la hembra en el dispositivo USB (función), y se utilizan en sistemas móviles, por ejemplo: cámaras fotográficas, impresoras, scanner, módems o altavoces.

En general podemos afirmar que la hembra de los conectores **A** están en el lado del host (computador) o de los concentradores (hubs), mientras que las de tipo **B** están del lado de los periféricos.

Se puede apreciar, en la figura 1.21 los diferentes tipos de conectores existentes en el mercado, los cuales se usan para las diferentes aplicaciones de periféricos.

	Tipo A		Tipo B	
	Macho	Hembra	Macho	Hembra
USB estándar				
Mini USB 5 pines				
Mini USB 8 pines				
Micro USB				
USB 3.0 estándar				
Micro USB 3.0				

Figura 1.21 Tipos de conectores USB

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO

2.1 PRINCIPIO DE FUNCIONAMIENTO DEL PROTOTIPO

El prototipo a desarrollarse consta básicamente de dos partes: un módulo de control remoto (Transmisor) y otro módulo de recepción (Receptor).

Para el funcionamiento del módulo de control remoto, se requiere de un microcontrolador que cumpla con las características requeridas para esta aplicación las cuales son: realizar la lectura y procesamiento de los datos ingresados serialmente utilizando un teclado hexadecimal y enviarlos al dispositivo XBee para transmitirlos vía RF al módulo receptor.

El módulo receptor consta principalmente de un microcontrolador, el cual recibe los datos enviados desde el control remoto vía RF, para procesarlos y enviarlos mediante una comunicación serial USB al computador en forma bidireccional esperando una respuesta positiva o negativa del computador para que el microcontrolador envíe una instrucción al motor paso a paso para la apertura de la puerta de garaje.

Además este módulo receptor consta de otros dispositivos como: un sensor infrarrojo para detectar el paso del vehículo, fines de carrera para parar la puerta, un relé para la iluminación en la noche y un pulsador para la apertura manual de la puerta de garaje.

En la figura 2.1 se puede observar un diagrama de bloques del principio de funcionamiento del sistema.

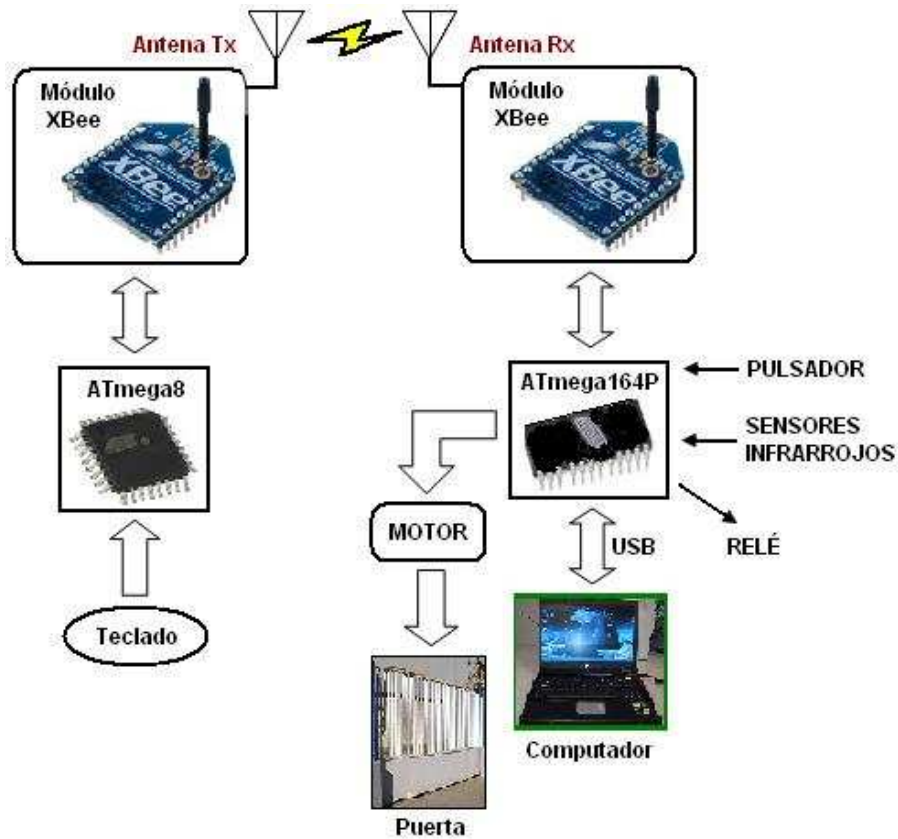


Figura 2.1 Diagrama de bloques del prototipo

2.2 COMPONENTES DEL HARDWARE

Los dispositivos a utilizarse en la implementación del hardware son muy importantes ya que nos ayuda a tener un buen panorama de todos los elementos que se van a utilizar en el diseño y construcción del prototipo, así como el proceso de comunicación de datos y el presupuesto necesario para la implementación.

A continuación se menciona el funcionamiento y las características principales de los dispositivos del sistema que conformarán los módulos de nuestro prototipo.

2.2.1 MÓDULOS DE RADIO FRECUENCIA

Para la comunicación inalámbrica entre el módulo de control remoto y el módulo receptor, se puede utilizar diferentes módulos de RF, pero para nuestro caso se utiliza el módulo XBee¹⁹, que permite una comunicación bidireccional entre los microcontroladores, con un bajo consumo de potencia así como su sensibilidad de recepción y una alimentación mínima de voltaje lo cual permite un transporte confiable de datos entre dispositivos remotos.

2.2.1.1 Módulos de Comunicación XBee

Los módulos XBee presentan una potencia de salida de 1mW (0dBm), la sensibilidad del receptor es de -92dBm y una velocidad de datos de 250Kbps. Esto le permite operar hasta de 30m en interiores y hasta 100m con línea de vista. La característica fundamental que ha destacado es la sensibilidad de sus receptores, para lograr mayor alcance. La corriente de operación de estos dispositivos ronda los 40mA, tanto para la transmisión como para la recepción, mientras que en el modo de bajo consumo se reduce a tan solo 10uA. Su tensión de operación es de 2.8 a 3.4v.

Las opciones de antena son: antena chip integrada en el módulo, antena de cuarto de onda en el módulo, conector U.FL o RPSMA para antena externa al módulo.

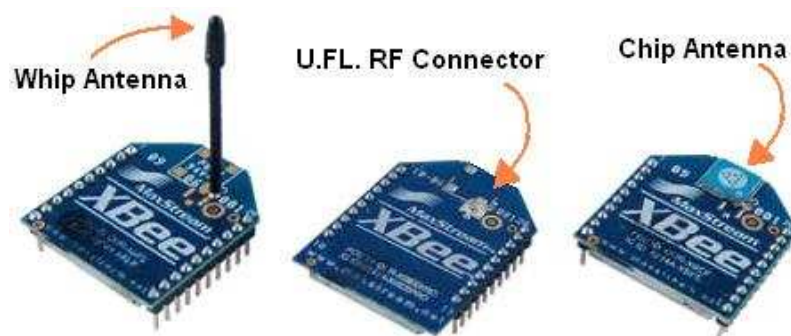


Figura 2.2 Módulos XBee

¹⁹ XBee-Datasheet.pdf

En la siguiente tabla 2.1 se puede observar, otras características de los módulos XBee.

Especificaciones	XBee
Desempeño	
Indoor / Rango Urbano	Hasta 100 ft (30m)
Outdoor RF línea de Vista	Hasta 300ft (100m)
Potencia de Transmisión	1mW (0 dBm)
Vel de Datos RF	250,000 bps
Vel de Datos con la Interfaz (elegible por Software)	1,200-115,200 bps (también soporta velocidades de baudios no estándares)
Sensibilidad del Receptor	-92 dBm (1% de promedio de error)
Requerimientos de Energía	
Voltaje de Alimentación	2.8 ~ 3.4V
Corriente (típica) del Transmisor	45mA (@3.3V)
Corriente (típica) del Receptor	50 mA (@3.3V)
Corriente de Apagado	< 10 μ A
General	
Frecuencia de Operación	ISM 2.4 GHz
Dimensiones	0.960" x 1.087" (2.438cm x 2.761cm)
Temperatura de Operación	-40° a 85 °C (industrial)
Opciones de Antena	Conector U. FL , antena de circuito o antena Whip.
Red y Seguridad	
Topologías de red soportadas	Punto a punto, punto a Multipunto, Peer-to-peer, y Malla (próximamente)
Número de Canales (elegibles por software)	16 Canales de Secuencia Directa
Opciones de Filtrado	PAN ID; Canal y Fuente / Dirección Destino

Tabla 2.1 Especificaciones del módulo XBee

2.2.1.2 Circuito básico y Distribución de pines del XBee

La figura 2.3 muestra las conexiones mínimas que necesita el módulo XBee para ser utilizado. Luego de esto, se debe configurar según el modo de operación adecuado para la aplicación requerida por el usuario.

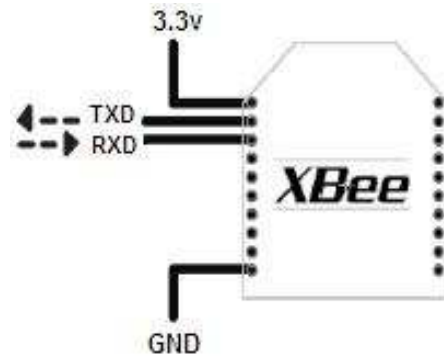


Figura 2.3 Conexiones mínimas requeridas para el XBee

El módulo requiere una alimentación desde 2.8 a 3.4v, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con el microcontrolador, o directamente a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje.

En la tabla 2.2 se presentan las características de los terminales y una breve descripción de ellos, la distribución de los pines físicamente en el módulo XBee se muestra en la figura 2.4.

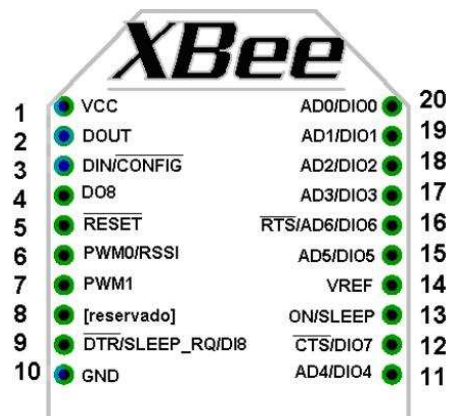


Figura 2.4 Diagrama del módulo XBee

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Tabla 2.2 Distribución de pines del módulo XBee

2.2.1.3 Comunicación del módulo XBee

La interfaz con el computador se realiza vía una UART, de niveles lógicos a través de la cual se puede comunicar con cualquier dispositivo compatible en niveles lógicos y voltajes. Todas las tramas de transmisión cumplen el patrón de: 1bit de inicio, 8bits de datos y 1bit de parada como se muestra en la figura 2.5.

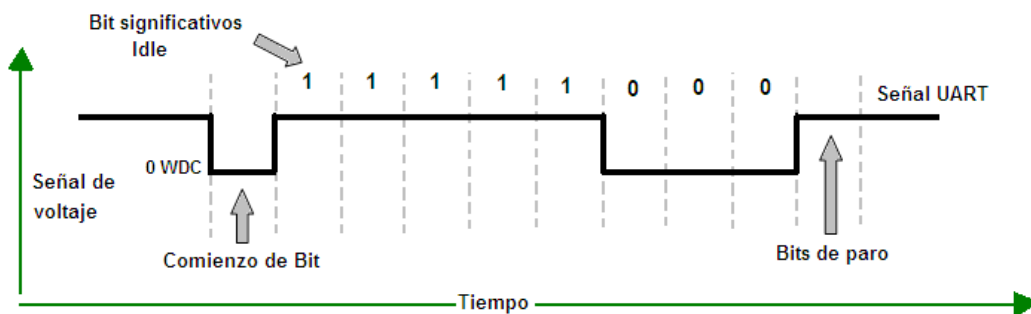


Figura 2.5 Trama de datos serie

2.2.1.4 Modos de operación

Los módulos XBee, pueden operar en 5 modos diferentes, los cuales son: modo de inactividad, de transmisión, de recepción, de comandos y dormido principalmente.

Para nuestro caso se va a utilizar el modo Recibir/Transmitir, que se encuentra en estos modos cuando el XBee recibe algún paquete de RF a través de la antena (modo Receiver) o cuando se envía información serial al buffer del pin 3 (UART Data in) que luego será transmitida.

La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo indirecto la información es retenida por el módulo durante un periodo de tiempo y es enviada solo cuando la dirección de destino la solicita.

Además es posible enviar información por dos formas diferentes. Unicast y Broadcast. En la primera, la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un ACK (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un ACK, en caso de que no llegue, reenviará el paquete hasta 3 veces o hasta que reciba el ACK, Después de los 3 intentos se incrementa el registro de fallas ACK un uno.

En el modo Broadcast la comunicación es entre un nodo y a todos los nodos de la red. En este modo, no hay confirmación por ACK.

2.2.2 MICROCONTROLADORES

Para determinar el mejor microcontrolador a utilizarse en los dos módulos, se debe tener en cuenta tanto el software como el hardware del que dispone, el procesamiento y velocidad de transmisión de datos, el número de puertos de E/S, el consumo, la capacidad de memoria requerida, su ancho de palabra, así como el diseño de la placa y sus aplicaciones.

En nuestro caso se utilizará los microcontroladores AVR, ya que su programación es fácil mediante el programador ICSP y requieren menos líneas de comandos que un microcontrolador normal debido que tiene tecnología RISC, además porque trabaja con una alimentación alrededor de los 3.3v igual al módulo XBee.

2.2.2.1 Microcontrolador ATmega8

En el caso del control remoto se utiliza un microcontrolador, que cuente con un periférico de comunicación serial USART, un puerto para manejar el teclado hexadecimal, puertos para la alimentación y otros para los LEDs indicadores, por lo cual hemos optado usar un AVR ATmega8 CMOS de 8bits con arquitectura avanzada RISC de alto rendimiento y bajo consumo de energía.

2.2.2.1.1 Características generales del ATmega8

Este microcontrolador ofrece un repertorio de 130 instrucciones, 32 registros de uso general, una memoria FLASH de 8Kbytes, una EEPROM de 512bytes y una SRAM de 1Kbytes para los datos.

Voltaje de operación	2.7 - 5.5 V _{DC}
Rangos de velocidad	0 - 16 MHz
Consumo de energía	Activo: 3.6 mA
	Reposo: 1.0 mA
	Modo Power - down= 0.5 uA
Características de periféricos	2 Timmer/counter de 8 bits
	1 Timmer/counter de 16 bits
	3 Canales PWM
	8 Canales análogos - digitales
	Módulo USART
	Interfaz serial SPI
Características especiales	Power on Reset
	Oscilador programable interno
	5 sleep mode
Entradas - Salidas (E/S)	23 E/S programables

Tabla 2.3 Características del microcontrolador ATmega8

2.2.2.1.2 Distribución de pines del ATmega8

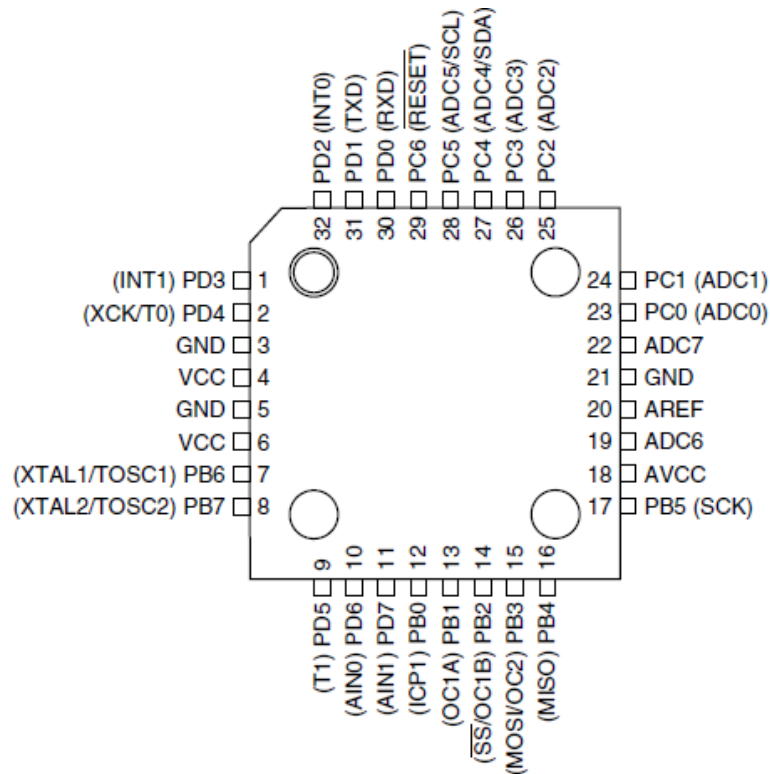


Figura 2.6 Distribución de pines del ATmega8

A continuación se presentan las configuraciones de los pines utilizados del microcontrolador ATmega8 para el presente proyecto, así como los distintos pines de polarización, reset y oscilación externa.

El microcontrolador ATmega8, posee tres puertos bidireccionales (B, C y D) que incluyen líneas digitales configurables como entradas o salidas, dependiendo del uso que se le dé.

a) Pórticos de alimentación

Para nuestro caso el AVR se va alimentar con un voltaje de 3.3v en los pórticos 4 y 6 y conexión a tierra los pórticos 3, 5 y 21.

b) Pórtico B (PB7...PB0) XTAL1/XTAL2/TOSC1/TOSC2

El pórtico B conformado por 8 líneas bidireccionales con resistencias de pull-up internas, de las cuales seis son maniobrables por el usuario. Se van usar dos de estas (PB0 y PB1) para la conexión del teclado hexadecimal. Además cuenta con buffers que tienen características simétricas de capacidad y cederán corriente en bajo si las resistencias de pull-up están activas.

Los pórticos PB6 y PB7 tienen funciones especiales (I/O del oscilador o TOSC del async Timer/Counter2).

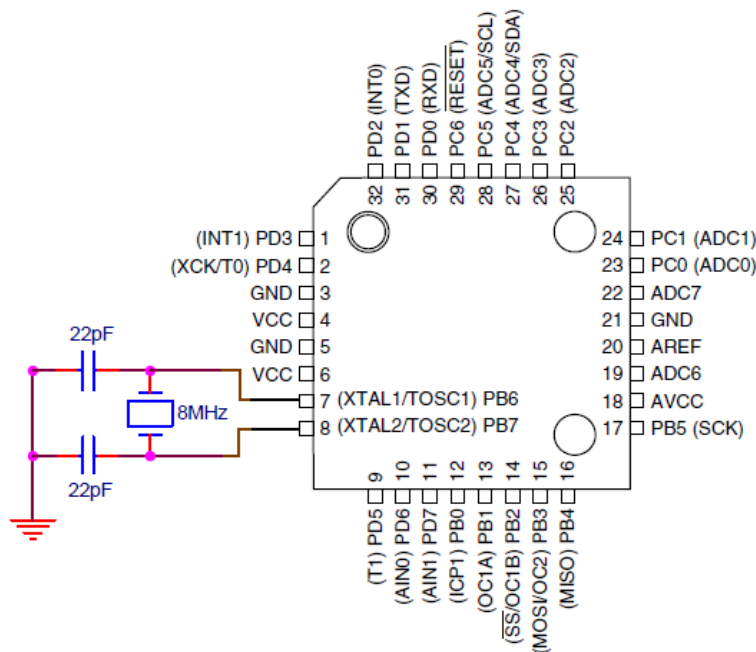


Figura 2.7 Circuito de oscilación

c) Pórtico D (PD7...PD0)

Este pórtico consta de 8 líneas bidireccionales las cuales cumplen con las mismas características que el puerto B, de las cuales se van a utilizar para la conexión del teclado hexadecimal seis líneas (PD2, PD3, PD4, PD5, PD6 y PD7).

Y para la transmisión y recepción de datos se utiliza los pórticos (PD0 y PD1) respectivamente.

d) Pórtico C (PC6...PC0)

Es un pórtico bidireccional de 7 líneas con las mismas características del puerto B. En nuestro caso se utiliza el pórtico PC6 como reset con una tensión en bajo con una duración superior al tiempo de reloj lo cual producirá su activación aunque el reloj no esté activado.

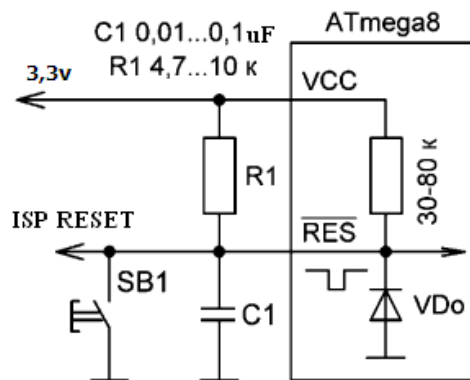


Figura 2.8 Circuito reset del microcontrolador ATmega8

e) AVCC, a este pórtico se le aplica una corriente de alimentación V_{CC} para el convertor A/D, con un filtro de paso bajo.

f) AREF, referencia analógica para el convertor A/D.

g) ADC7...6 (TQFP y QFN/MLF)

En el paquete TQFP y QFN/MLF, sirven como entradas analógicas para el convertor A/D. Estos pines se alimentan de la fuente analógica y sirven como canales ADC de 10 bits.

2.2.2.1.3 Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART).

Es un dispositivo de comunicación en serie full-duplex, asíncrono y síncrono, master y slave, con varios métodos de detección de errores.

2.2.2.2 Microcontrolador ATmega164P

Para el módulo receptor se utiliza un microcontrolador que tenga un puerto para recibir la señal de RF, un puerto para manejar el motor paso a paso, un puerto para el relé, un puerto para los sensores infrarrojos, un puerto serial para la comunicación USB y otros puertos para aplicaciones adicionales. Tomando en cuenta estas características se va utilizar un AVR ATmega164P CMOS de 40 pines, de bajo consumo de potencia basado en la arquitectura RISC mejorada.

2.2.2.2.1 Características principales del ATmega164P

El ATmega164P tiene 16Kbytes de memoria flash de programa, lo suficiente para la programación de nuestro módulo. En la tabla 2.4 se describe las características generales del ATmega164P.

Arquitectura Avanzada RISC	131 instrucciones. La mayoría de un solo ciclo de reloj 32 registros de trabajo de 8 bits para propósito general Capacidad de procesamiento de unos 20 MIPS a 20MHz
Memorias de programa y de datos no volátiles de alta duración	16/32/44 K bytes de FLASH 512B/1K/2K bytes de EEPROM 1/2/4K bytes de SRAM Interna
Retención de Datos	20 años a 85°C / 100 años a 25°C
Características de periféricos	Dos puertos Seriales USART Programables
Encapsulados para Entradas/Salidas (E/S)	32 líneas de E/S programables
Voltajes de Operación	1.8 – 5.5V
Velocidad de Funcionamiento	0-4MHz @ 1.8 – 5.5V - 10MHz @ 2.7 - 5.5V 0-10MHz @ 2.7 – 5.5V - 20MHz @ 4.5 - 5.5V
Consumo de energía a 1MHz, 1.8V, 25°C	Activo: 0.4mA

Tabla 2.4 Características generales del ATmega164P

2.2.2.2.2 Distribución de pines del ATmega164P

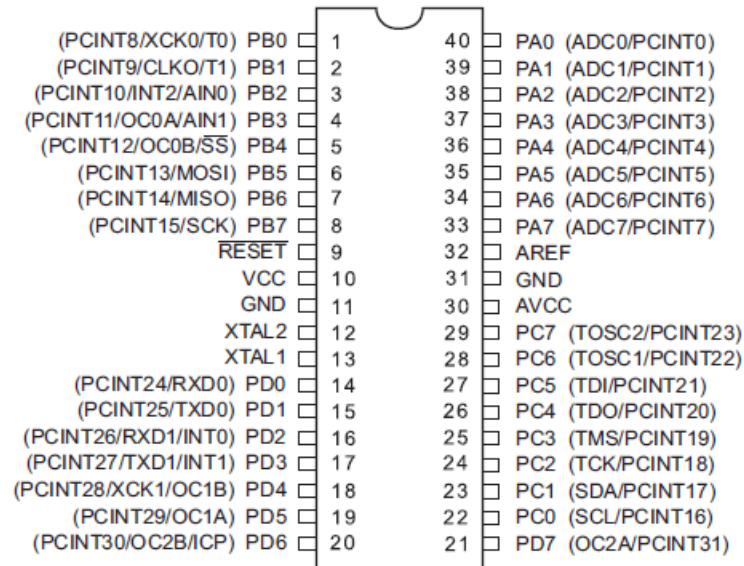


Figura 2.9 Distribución de pines del ATmega164P

En este caso utilizaremos los siguientes pórtricos para las distintas funciones de nuestro módulo receptor, así como sus pines de polarización, reset y oscilación interna.

a) Pórtico A (PA0...PA7)

Es un pórtrico bidireccional de 8 bits con resistencias internas de pull up (seleccionables para cada bit). Los buffers de salida del puerto A tienen características simétricas controladas con fuentes de alta capacidad. Los pines del puerto A están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo.

b) Pórtico B (PB0...PB7)

El pórtrico B es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up. Las salidas de los buffers del puerto B tienen características simétricas controladas con fuentes de alta capacidad. Los pines del puerto B están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté

corriendo. Para nuestro caso utilizaremos tres pines (PB5, PB6 y PB7) para la programación del AVR con la finalidad de no estar extrayendo nuestro micro a cada momento para programarlo.

c) Pórtico C (PC0...PC7)

El pórtico C es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up. Las salidas de los buffers del puerto C tienen características simétricas controladas con fuentes de alta capacidad. Los pines del puerto C están en tri-estado cuando las condiciones de reset están activadas siempre y cuando el reloj no esté corriendo. Para nuestro caso vamos a utilizar los ocho pines de los cuales, cuatro de ellos PC0, PC1, PC2 y PC3 son las salidas para el motor paso a paso, dos de los pines PC4 y PC5 para los sensores infrarrojos, el pino PC6 para el botón de apertura manual y el pino PC7 para el LED indicador del microcontrolador.

d) Pórtico D (PD0...PD7)

El pórtico D es un puerto bidireccional de entradas y salidas con resistencias internas de pull up. Las salidas de los buffers del puerto D tienen características simétricas controladas con sumideros de fuentes de alta capacidad. Los pines del puerto D están en tri-estado cuando llega una condición de reset activa, siempre y cuando el reloj no esté corriendo.

De este pórtico se va utilizar los pines PD0 a PD1 para la recepción y transmisión de datos con el módulo XBee, los pines PD2 y PD3 para la comunicación serial USB, los pines PD5 y PD6 para el control de los fines de carrera y el pino PD7 es utilizado para la activación del relé de nuestro prototipo.

e) RESET

Un pulso de nivel bajo en este pino con períodos de duración mínima va a generar un reset, siempre y cuando el reloj no esté corriendo. Pulsos cortos no garantizan un reset, en la figura 2.10 podemos observar el circuito reset diseñado para nuestro prototipo.

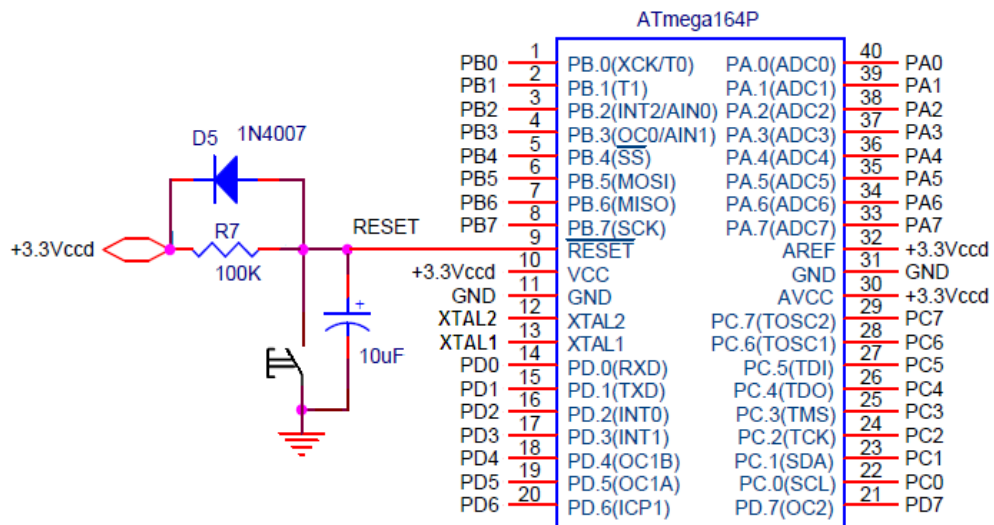


Figura 2.10 Circuito RESET

f) XTAL1 y XTAL2

Entre estos dos p rticos (12 y 13) se ubica un cristal de oscilaci n para utilizarse como reloj externo, figura 2.11.

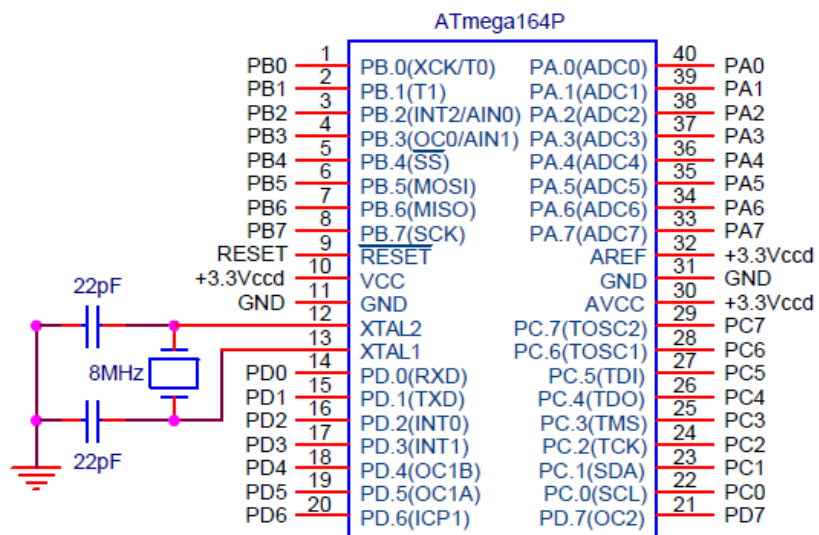


Figura 2.11 Circuito de oscilaci n

g) Pórticos de polarización

El voltaje de polarización del AVR son los pines 10 y 11, además el pin 30 es AV_{CC} el cual representa a la alimentación de voltaje para el pin del puerto F y el Conversor Análogo a Digital. Este debe ser conectado externamente a V_{CC} (3.3v), siempre y cuando el ADC no sea usado. Si el ADC es usado, este deberá ser conectado a V_{CC} a través de un filtro paso bajo.

2.2.2.3 Interrupciones del ATmega

En el ATmega las interrupciones están en un vector ordenado por las prioridades de las interrupciones que pueden colocarse en diferentes posiciones de la memoria:

- El principio de la memoria flash.
- El principio de la sección de boot de la memoria flash, esta viene determinada por el contenido del registro general (GICR).

Las interrupciones externas son capturadas por los puertos INT0 e INT1, aunque estén configurados como salidas. Se detectan con flanco de subida, bajada (síncrona) o con nivel bajo (asíncrona).

2.2.2.4 ICSP (In Circuit Serial Programming)

Este es un método para la programación directa de los AVR's después que son colocados en una placa, lo cual tiene varios beneficios como por ejemplo: actualizar el software de manera rápida, realizar pruebas, control de periféricos y otros propios de cada dispositivo.

Para la programación de nuestros AVR's podemos fijarnos en las funciones adicionales que tiene el pòrtico B, el cual nos permitirá utilizar este método valiéndonos de la comunicación SPI (Serial Peripheral Interface) hacia un dispositivo programador basado en un ATmega8 el cual realiza la conversión USB por software a serial para la transferencia de datos de programación desde el computador a cada uno de nuestros microcontroladores ATmega8 y

ATmega164P. En la figura 2.12 podemos apreciar el circuito para la programación de un AVR utilizando el método ICSP.

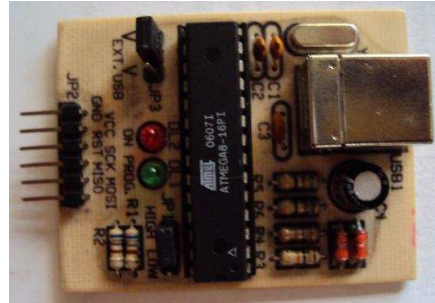


Figura 2.12 Circuito para la programación de AVR's con ICSP

2.2.2.4.1 Comunicación SPI (Serial Peripheral Interface)

El SPI es un estándar de enlace de datos seriales sincronizados por un reloj que operan en modo full-duplex a través de un medio de cuatro líneas, es decir, que se puede enviar y recibir información de manera simultánea, lo cual, eleva la transferencia de datos.

2.2.2.4.2 Descripción de líneas de conexión

Las cuatro líneas requeridas para este tipo de conexión están pensadas para interconectar múltiples dispositivos, las líneas MOSI y MISO son de transferencia, la línea SCK para sincronización y la línea SS que funciona a manera de selector de los diferentes dispositivos.

En la tabla 2.5 podemos observar la descripción de las líneas de la comunicación SPI.

SCK	Señal de reloj del bus de datos, generada por el maestro
MOSI	Salida de datos del maestro y entrada de datos a los esclavos
MISO	Salida de datos de los esclavos y entrada de datos al maestro
SS	Habilitación del esclavo por parte del maestro.

Tabla 2.5 Descripción de las líneas del SPI

2.2.2.4.3 Modo de conexión del ICSP para cada módulo

Para implementar la conexión ICSP en los diferentes módulos de nuestro prototipo se va a utilizar los siguientes pines: en el caso del control remoto se utilizan los pines 15, 16 y 17 y en el caso del módulo receptor los pines utilizados son 6, 7 y 8, que de acuerdo con la distribución de pines de nuestros AVR's representan al MOSI, MISO y SCK respectivamente para la comunicación SPI, además en nuestro caso no se utiliza la línea SS ya que solamente se va a realizar la comunicación para un solo dispositivo en cada módulo.

En la siguiente figura 2.13 podemos observar el conector jumper para la comunicación SPI, utilizando la conexión ICSP para cada uno de nuestros módulos.

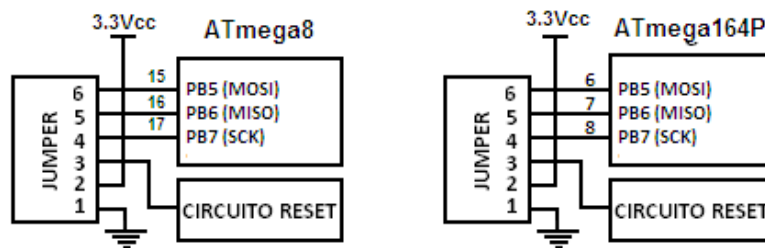


Figura 2.13 Conexión del ATmega8 y ATmega164P hacia un jumper

2.2.3 FUENTE DE ALIMENTACIÓN

La fuente de alimentación (power supply) es aquella que se encarga de suministrar energía eléctrica de corriente continua a los distintos elementos de un circuito, para su respectivo funcionamiento.

Nuestro prototipo requiere de dos fuentes de alimentación de voltaje externo para satisfacer las necesidades de corriente y voltaje que necesita el diseño de cada uno de nuestros módulos.

Para lo cual se ha optado por utilizar una batería de 9v en el caso del módulo de control remoto, figura 2.14.



Figura 2.14 Bateria de 9V_{DC}

Y para el caso de nuestro módulo receptor las características de la fuente son las siguientes:

Marca	INTELLIGENT ADAPTOR
Modelo	AD-1200F
Entrada	AC 90/240V 50/60Hz
Potencia	15w
Salida	3 a 12 V _{DC}
Corriente	1.2 A máxima

Tabla 2.6 Características generales de la fuente de alimentación

Acompañado de un conector Jack hembra adherido al circuito impreso para su alimentación, figura 2.15.

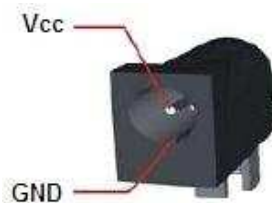


Figura 2.15 Conector hembra para la fuente de alimentación externa

2.2.3.1 Distribución y regulación de voltajes

Una vez alimentado los circuitos, se debe analizar los voltajes necesarios para el funcionamiento de nuestro prototipo.

En el caso del control remoto se recibirá un voltaje de 9V_{DC}, el cual se lo pasa por un circuito rectificador compuesto por un diodo 1N4007 que protege al circuito de regresos de corriente y por un capacitor de 100uF que va a funcionar a manera de filtro para la salida. Y para la alimentación total del control remoto tenemos un

regulador de voltaje constante LM1117T que a su salida nos da un voltaje de 3.3v, acompañado de un LED indicador que nos permite ver si la fuente está entregando voltaje a su salida seguido de un capacitor el cual se lo utiliza como filtro para evitar picos de corriente.

En la siguiente figura 2.16 podemos ver el diagrama para la distribución de voltajes del módulo de control remoto.

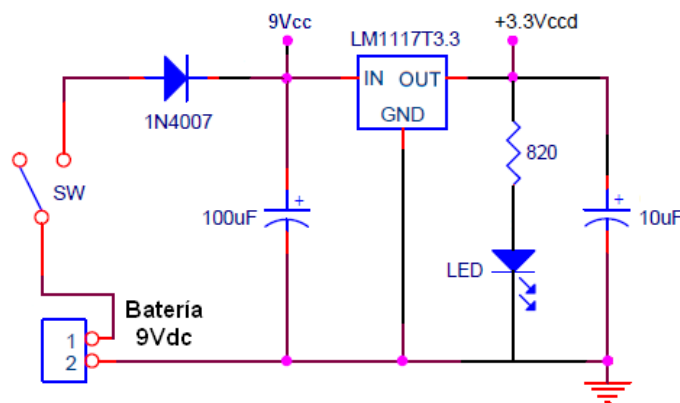


Figura 2.16 Diagrama de regulación de voltaje para el control remoto

Para nuestro módulo receptor nuestra fuente de voltaje es máximo de $12V_{DC}$, la cual nos permitirá realizar el control del relé para la activación de la luz del garaje y además lo utilizaremos para el encendido del motor paso a paso. Este voltaje se lo pasa por un diodo rectificador 1N4007 brindando protección de regresos de corriente y por un capacitor de $1000\mu F$ que funciona a manera de filtro para la salida de la fuente.

En cambio para la polarización del AVR, el módulo XBee y la interfaz USB, tenemos la necesidad de utilizar un regulador de voltaje LM1117T, para obtener un voltaje de 3.3v de alimentación y además a su salida tenemos un LED que nos indica si el regulador está o no brindando voltaje a su salida con un capacitor de $10\mu F$ que funciona a manera de filtro.

En la siguiente figura 2.17, se observa el diagrama para la distribución de voltajes del módulo receptor.

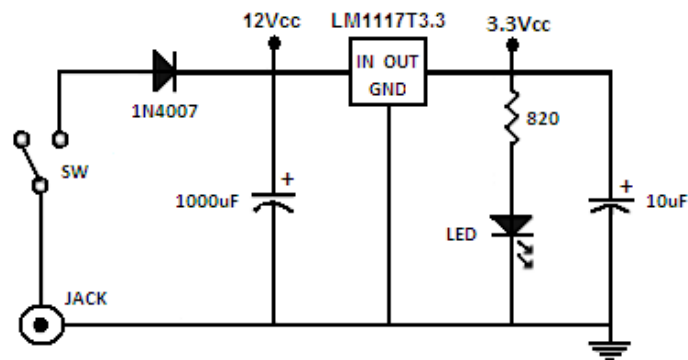


Figura 2.17 Diagrama de regulación de voltaje para el módulo receptor

2.2.4 MOTORES PASO A PASO²⁰

El motor PAP es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de avanzar una serie de grados (pasos) dependiendo de sus entradas de control. El motor paso a paso se comporta de la misma manera que un convertidor digital-analógico y puede ser gobernado por impulsos procedentes de sistemas lógicos.

2.2.4.1 Introducción

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos.

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de tan solo 1.8°, es decir, que se necesitaran 4 pasos en el primer caso (90°) y 200 pasos para el segundo caso (1.8°), para completar un giro de 360°.

²⁰ <http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>

Estos motores poseen la habilidad de quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula ninguna corriente por sus bobinas.

2.2.4.2 Principio de funcionamiento de motores paso a paso

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por cierto número de bobinas excitadoras en su estator figura 2.18. Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deben ser externamente manejadas por un controlador.

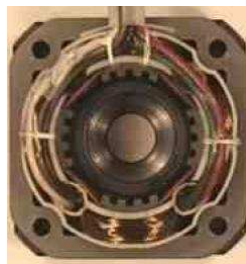


Figura 2.18 Imagen de un estator de 4 bobinas

Existen dos tipos básicos de motores paso a paso de imán permanente:

- Los bipolares.
- Los unipolares.

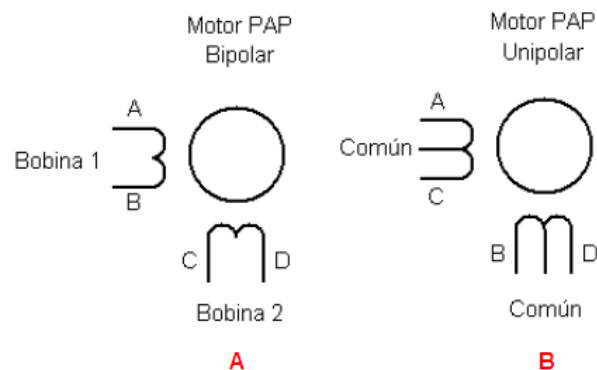


Figura 2.19 Tipos de motores paso a paso

Motores Bipolares: Este tipo de motores, generalmente tienen 4 cables de salida (figura 2.19 A). Necesitan ciertos trucos para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

Motores Unipolares: Estos motores suelen tener cinco ó seis cables de salida, dependiendo de su conexión interna (figura 2.19 B). Este motor se caracteriza por ser más simple de controlar.

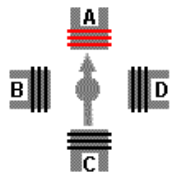
2.2.4.3 Secuencias para manejar motores paso a paso Unipolares

Existen tres secuencias posibles para este tipo de motores, las cuales se detallan a continuación. Todas las secuencias comienzan nuevamente por el paso uno, una vez alcanzado el paso final (4 u 8). Para revertir el sentido de giro, simplemente se deben ejecutar las secuencias en modo inverso.

Secuencia Normal: Esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, hace que el campo magnético sea más potente y se obtenga un alto torque de paso y de retención.

Secuencia del tipo wave drive o secuencia por ola: Esta es la forma más fácil de manejar un motor, consiste en energizar solo una bobina a la vez, A, B, C y por último la D. En algunos motores esto brinda un funcionamiento más suave. La contrapartida es que al estar solo una bobina activada, el torque de paso y retención es menor.

En la tabla 2.7 podemos observar el energizado para conseguir que el motor gire en sentido antihorario, el cual consta de cuatro movimientos exactos.

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D	RECREACIÓN
1	ON 1	OFF 0	OFF 0	OFF 0	

2	OFF 0	ON 1	OFF 0	OFF 0	
3	OFF 0	OFF 0	ON 1	OFF 0	
4	OFF 0	OFF 0	OFF 0	ON 1	

Tabla 2.7 Secuencia tipo ola en sentido antihorario

Secuencia del tipo medio paso: En esta secuencia se activan las bobinas de tal forma de brindar un movimiento igual a la mitad del paso real. Para ello se activan primero dos bobinas y luego solo una y así sucesivamente.

Los motores paso a paso son dispositivos mecánicos y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los pulsos aplicados son puntos muy importantes. Si la frecuencia de pulsos es muy elevada, el motor puede reaccionar de las siguientes formas:

- Puede que no realice ningún movimiento en absoluto.
- Puede comenzar a vibrar pero sin llegar a girar.
- Puede girar erráticamente.
- O puede llegar a girar en sentido opuesto.

Para obtener un arranque suave y preciso, es recomendable comenzar con una frecuencia de pulso baja y gradualmente ir aumentándola hasta la velocidad deseada sin superar la máxima tolerada. El giro en reversa debería también ser realizado previamente bajando la velocidad de giro y luego cambiar el sentido de rotación.

2.2.4.4 Diagrama circuital del Motor Paso a Paso

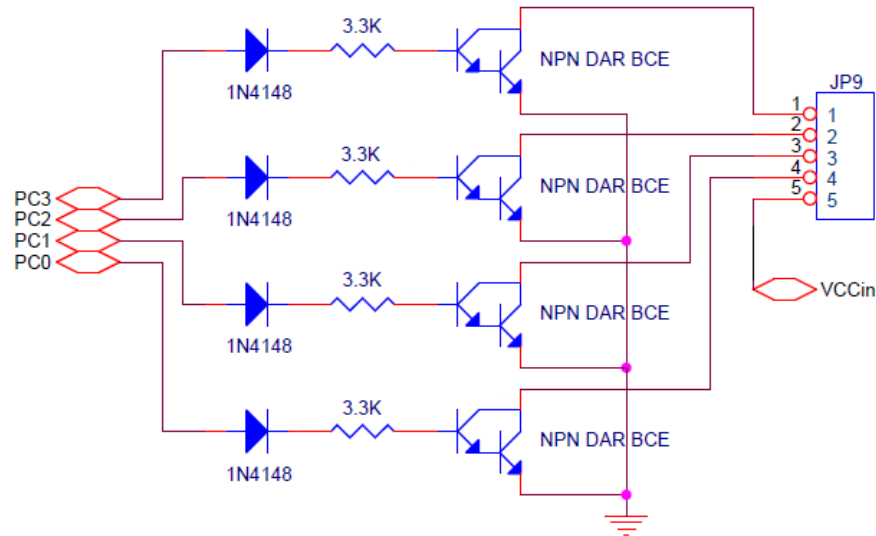


Figura 2.20 Diagrama circuital del motor paso a paso

El motor paso a paso utiliza las líneas PC0, PC1, PC2 y PC3 del pòrtico C del microcontrolador ATmega164P, en las cuales se ubica la secuencia de control correspondiente para el manejo del motor haciendo que gire en sentido horario, antihorario o reduzca los giros dependiendo lo que desee realizar el programador.

Debido a que el microcontrolador ATmega164P responde a especificaciones de la familia CMOS (Complementary Metal Oxide Semiconductor) es capaz de suministrar 0.4mA a 3.3V en cada una de sus líneas, el nivel de voltaje y corriente son relativamente pequeños para el control de motores paso a paso, es por tal motivo que se ha implementado un circuito de fuerza para superar esta limitación, en donde se destacan cuatro transistores de potencia tipo Darlington (TIP 122) que nos permite manejar cargas de hasta 500mA (5A) como podemos observar en la tabla 2.8 de los datos del dispositivo. Para proteger al AVR de picos de corriente inversa que se generan al momento de arrancar el motor se han diseñado cuatro diodos (1N4148) en serie con la resistencia de base (3.3K Ω) de cada transistor de potencia, figura 2.21.

TIP122	CARACTERÍSTICAS BÁSICAS				Distribución de terminales
	Símbolo	Mínimo	Típico	Máximo	
Transistor dárlington NPN de mediana potencia (65W) diseñado para funcionar como interruptor	I_c			5A	
	V_{ceo}			100v	
	V_{be}		2.5v		
	β	20		200	
	$V_{ce(sat)}$	2v		4v	

Tabla 2.8 Características del transistor NPN Darlington (TIP122)

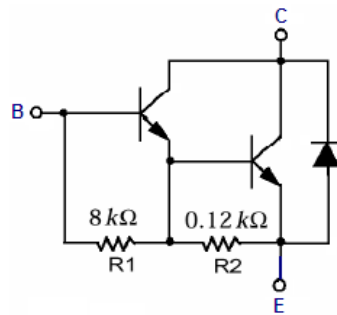


Figura 2.21 Circuito equivalente al transistor NPN Darlington (TIP122)

Como se puede observar, el transistor Darlington tiene una ganancia mucho mayor que un transistor normal, pues se multiplican las ganancias de los dos transistores. Los transistores Darlington se emplean en circuitos donde se necesitan controlar grandes cargas con corrientes muy pequeñas. Por otro lado cabe decir que la caída de tensión entre la base **B** y el emisor V_{BE} en este tipo de transistores es el doble que en los ordinarios (entre unos 1.4v) y a la vez el colector de cada transistor TIP122 se conecta a una bornera, para su conexión con cada cable de salida del motor PAP que se usa.

Como se mencionó en la fuente de alimentación, el motor paso a paso posee una fuente de voltaje DC externa que también será suministrada por la fuente que alimenta al módulo receptor. Para lo cual se ha dotado con un circuito de protección encendido/apagado (ON/OFF), permitiendo detectar la presencia de voltaje DC. Y además tiene un conector (bornera) adicional que nos permite trabajar con una fuente externa para controlar un motor paso a paso de mayor consumo.

2.2.5 SENSORES INFRARROJOS

Los sensores infrarrojos que se van a utilizar, constan de un diodo emisor (transforma la energía eléctrica en energía luminosa) y un fotodiodo receptor (transforma la energía luminosa en energía eléctrica), que tienen un diámetro de 5mm cada uno, figura 2.22.



Figura 2.22 Diodo emisor y fotodiodo receptor

Estos diodos deben estar ubicados uno frente al otro, para que el haz de luz que es emitido por el diodo emisor sea recibido por el fotodiodo, si estos no están correctamente alineados no van a funcionar adecuadamente.

Tiene la finalidad de determinar la presencia de un objeto en un intervalo de distancia especificado (en nuestro caso la distancia es alrededor de 20cm) y suelen basarse en el cambio provocado en alguna característica del sensor debido a la proximidad del objeto.

2.2.5.1 Módulo del Diodo Emisor

2.2.5.1.1 Circuito Integrado LM555²¹

El **circuito integrado 555**, es un circuito integrado de bajo costo y de grandes prestaciones. Entre sus aplicaciones principales cabe destacar las de multivibrador astable (dos estados metaestables) y monoestable (un estado estable y otro metaestable), detector de impulsos, etc.

²¹ http://es.wikipedia.org/wiki/Circuito_integrado_555

Este integrado consta internamente de 23 transistores, 2 diodos y 12 resistencias. Opera con tensiones de alimentación desde 4.5v hasta 18v y puede manejar corrientes de salida de hasta 200mA, con una capacidad suficiente para impulsar directamente entradas TTL, LED, zumbadores, bobinas de relé, parlantes piezoeléctricos y otros componentes.

En la figura 2.23 observamos el integrado y su distribución de pines.

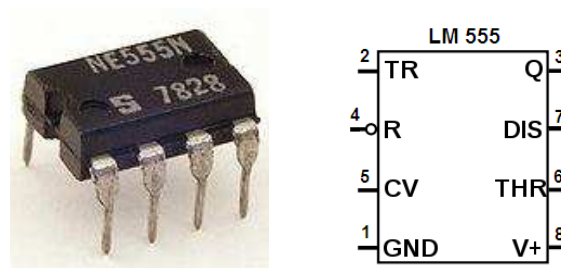


Figura 2.23 Integrado y distribución de pines del LM555

2.2.5.1.2 Funcionamiento del diodo emisor

El módulo emisor es el encargado de emitir una luz infrarroja la cual es manejada principalmente por un oscilador LM555, el cual nos permite generar una onda cuadrática y constante, la que es controlada mediante un potenciómetro para variar la frecuencia de salida de la onda, esto nos permitirá sincronizar la señal emitida por el infrarrojo hacia el circuito receptor.

La salida del oscilador lo conectamos a una resistencia y luego a un transistor el cual lo vamos a utilizar como un amplificador de corriente, que nos permite aumentar el alcance de la señal emitida por el LED infrarrojo.

También utilizamos un LED indicador, el cual nos permite observar si el circuito emisor se encuentra en funcionamiento o no.

En la siguiente figura 2.24, se puede observar el diagrama circuital del diodo emisor de luz.

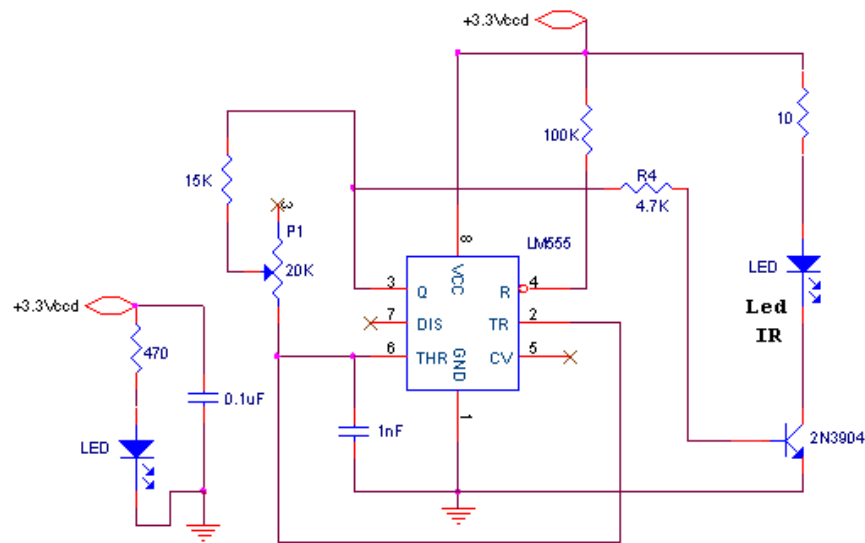


Figura 2.24 Diagrama circuital del emisor IR

2.2.5.2 Módulo del Fotodiodo Receptor

2.2.5.2.1 Amplificador Operacional LM358²²

El amplificador LM358, consta de dos ganancias independientes altas, internamente una frecuencia de compensación por aumento de unidad, amplificadores operacionales que fueron diseñados específicamente para operar en una amplia gama de voltajes de alimentación desde 3v a -32v.

En la siguiente figura 2.25 se puede ver el integrado y su distribución de pines.

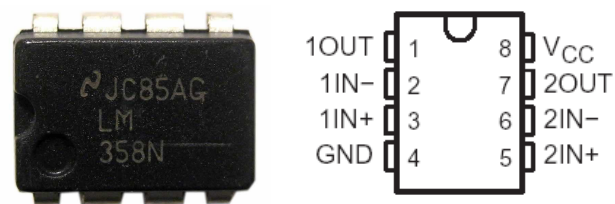


Figura 2.25 Integrado y distribución de pines del LM358

²² <http://www.national.com/mpf/LM/LM358.html>

2.2.5.2.2 Funcionamiento del fotodiodo receptor

Este módulo consta esencialmente de un detector de infrarrojo o foto transistor, el cual trabaja como un interruptor, es decir, con luz conduce y sin luz no conduce, y este emite una señal que va dirigida hacia un amplificador operacional. El cual está conectado como seguidor de tensión (su característica es que $V_i=V_o$) y también se utiliza como un acoplador de impedancias, que a su salida se divide en dos ramas.

La primera nos permite visibilizar el funcionamiento del circuito por medio de un LED indicador el cual es activado mediante un transistor. Y la segunda se dirige a otro amplificador operacional igualmente conectado como seguidor de tensión, con el que obtenemos una señal de salida con un voltaje constante y su salida se la va conectar al microcontrolador para procesar la señal.

En la siguiente figura 2.26, se puede observar el diagrama circuital del fotodiodo receptor.

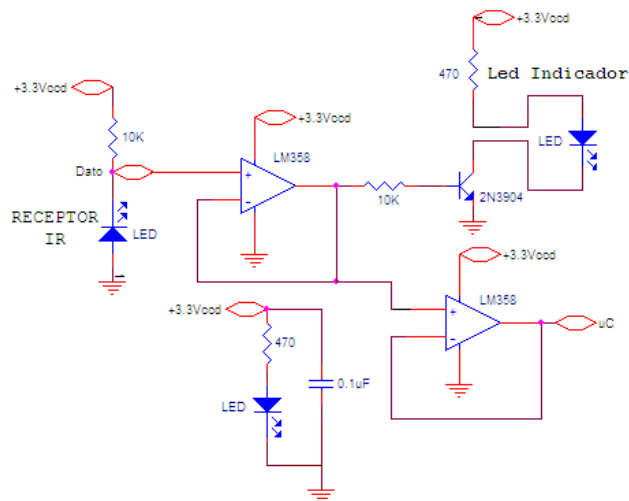


Figura 2.26 Diagrama circuital del receptor IR

2.2.6 RELÉ

Un relé es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un

electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Su composición física está dada por:

- Un electroimán.
- Juegos de contactos abiertos y cerrados.
- Y elementos mecánicos.

2.2.6.1 Características generales

Las características generales de cualquier relé son:

- El aislamiento entre los terminales de entrada y de salida.
- Adaptación sencilla a la fuente de control.
- Posibilidad de soportar sobrecargas, tanto en el circuito de entrada como en el de salida.

Las dos posiciones de trabajo en los bornes de salida de un relé se caracterizan por:

- En estado abierto, alta impedancia.
- En estado cerrado, baja impedancia.

Existen relés de dos tipos: los relés de DC y los relés de AC, la diferencia entre los dos tipos es el tipo de voltaje que se necesita para polarizar las bobinas que activan los contactos del relé, por lo cual para el relé de DC se necesita un voltaje de hasta $12V_{DC}$ y para el de AC hasta $120V_{AC}$.

El relé empleado para el desarrollo de nuestro proyecto es un relé de DC, figura 2.27.



Figura 2.27 Relé DC

2.3 MÓDULO DEL CONTROL REMOTO.

2.3.1 MANEJO DEL TECLADO PARA EL INGRESO DE CLAVES

El usuario va tener una clave personal de ingreso, esta clave está compuesta por cuatro dígitos, la cual es ingresada por el usuario mediante un teclado hexadecimal que está conectado en los pines B y D (PB0, PB1, PD3, PD4, PD5, PD6, PD7, PD2) del microcontrolador ATmega8 para que realice la lectura y almacenamiento, una vez realizado el ingreso de los cuatro dígitos el microcontrolador envía la clave de forma serial al módulo de radiofrecuencia XBee.

2.3.1.1 Introducción Teclado Hexadecimal

Dispositivo de entrada de datos que consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (Single In Line) macho de 8 pines que corresponden a las cuatro filas y a las cuatro columnas que tiene. En la siguiente figura 2.28 se observar el teclado hexadecimal y su matriz.

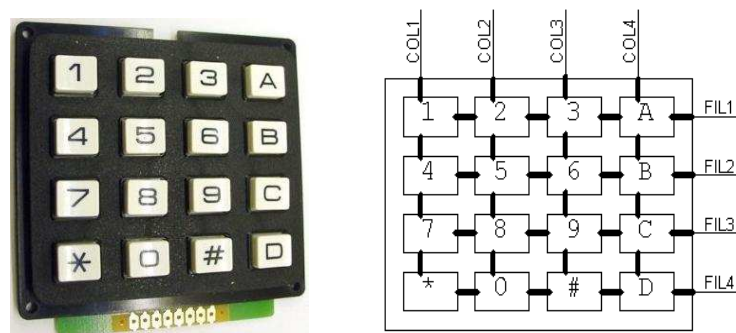


Figura 2.28 Estructura del teclado hexadecimal

2.3.1.2 Diagrama circuital del teclado hexadecimal

En la figura 2.29 se muestra el diagrama esquemático de conexión del teclado hexadecimal con el microcontrolador ATmega8.

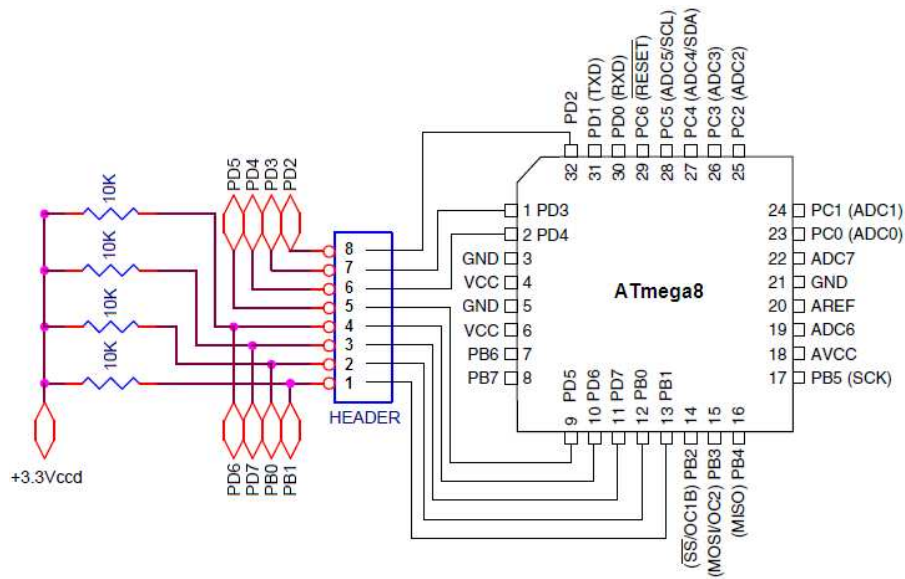


Figura 2.29 Conexión del teclado hexadecimal con el ATmega8

2.3.2 TRANSMISIÓN DE LA CLAVE INGRESADA

La clave que ingresa el usuario es almacenada en la memoria del microcontrolador ATmega8 hasta que se verifica que se ha ingresado los cuatro dígitos, para luego ser enviada en forma serial al módulo de radiofrecuencia XBee.

La señal que ingresa al módulo XBee, es modulada en QPSK (Modulación con desplazamiento de fase en cuadratura) con una frecuencia de 2.4GHz, esta señal es transmitida hacia el módulo receptor para su posterior procesamiento del cual hablaremos más adelante.

2.4 MÓDULO RECEPTOR

2.4.1 RECEPCIÓN DE LA CLAVE INGRESADA

Está formada por un módulo XBee, el cual recibe la señal que transmite el módulo de control remoto, a su vez la cuantifica, codifica y entrega la señal con niveles de voltaje CMOS.

Este módulo receptor opera en la frecuencia ISM de 2.4GHz, tiene una sensibilidad del receptor de -92dBm y opera con un voltaje de alimentación de 3.3v de corriente continua de 0.4mA.

La salida del módulo XBee ingresa al microcontrolador ATmega164P a través del p rtico PD0, el cual recibe una se al serial para ser almacenada en la memoria del microcontrolador y para luego ser procesada. Figura 2.30.

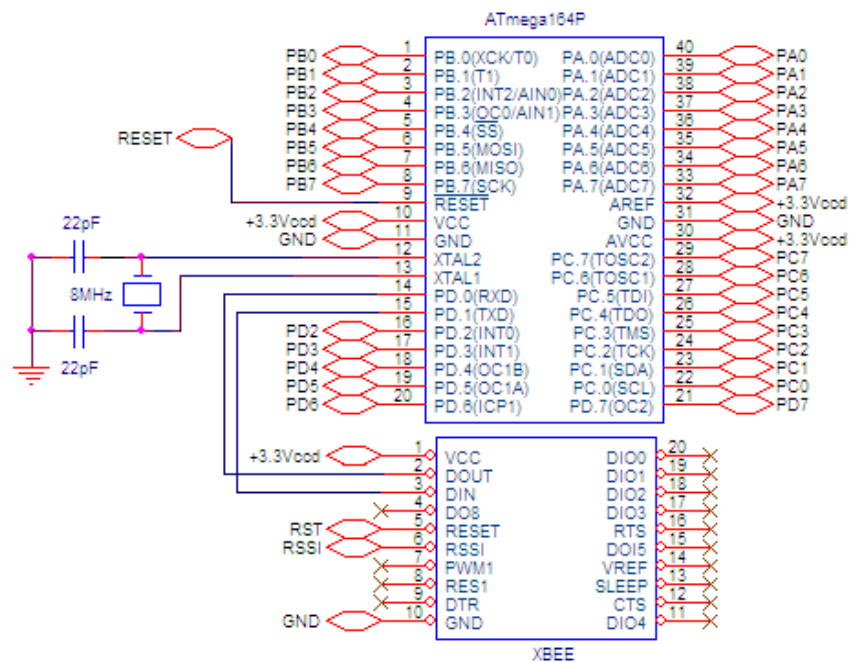


Figura 2.30 Diagrama de conexi n del microcontrolador con el XBee

El dispositivo receptor XBee presenta en el pin RSSI la se al proveniente del m dulo de control remoto, la misma que es enviada al pin PD0 para ser receptada por el microcontrolador que recibe el paquete de los cuatro d gitos para luego ser enviada en forma serial al computador, es decir los datos que recibe el m dulo de receptor son enviados directamente al computador.

Cuando se realiza este envi  el microcontrolador se queda en modo de espera de una respuesta del computador, ya que si la clave es correcta el computador enviara un car cter de aceptaci n (para nuestro caso "I"), el cual significa "clave correcta" y por ende el microcontrolador manda una secuencia de movimiento al

motor paso a paso permitiendo abrir o cerrar la puerta de garaje y si es incorrecta enviara un carácter de negación “N” (clave incorrecta).

2.4.2 COMUNICACIÓN SERIAL

La comunicación serial consiste en el envío de un bit de información de manera secuencial, esto es, un bit a la vez y a un ritmo acordado entre el emisor y el receptor. Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII.

Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

En la actualidad existen diferentes ejemplos de puertos que comunican información de manera serial. Uno de ellos es el conocido como “puerto serial” el cual ha sido gradualmente reemplazado por el puerto USB (Universal Serial Bus) que permite mayor versatilidad en la conexión de múltiples dispositivos.

La comunicación serial en computadores ha seguido los estándares definidos en 1969 por el RS-232 (Recommended Standard 232) que establece niveles de voltaje, velocidad de transmisión de los datos, etc.

En el presente proyecto se utiliza la interfaz de comunicación serial asincrónica por ser la más universal y compatible con el AVR utilizado, que nos permite comunicarnos serialmente entre el dispositivo USB y el computador.

2.4.2.1 Interfaz de comunicación serial usando USB

Para la interfaz de comunicación entre el módulo receptor y el computador, se va utilizar una comunicación serial, la cual nos permite conectarnos con el computador mediante la interfaz serial USB, por lo que incluye el software de red básica.

En la figura 2.31 se muestra la interfaz de comunicación serial USB de transmisión que enlaza el computador con el módulo receptor.



Figura 2.31 Interfaz de comunicación serial USB

El microcontrolador ATmega8 se comunica con el computador, utilizando el pÓrtico USART, tanto para la recepci3n de datos desde el computador como para la transmisi3n de datos hacia el m3dulo receptor.

Esta comunicaci3n serial USB requiere la operaci3n del microcontrolador ATmega8 a 16MHz con un cristal externo de 12MHz el cual est3 conectado en paralelo a los pines 9 (XTAL1) y 10 (XTAL2) de nuestro microcontrolador como se muestra en la figura 2.32.

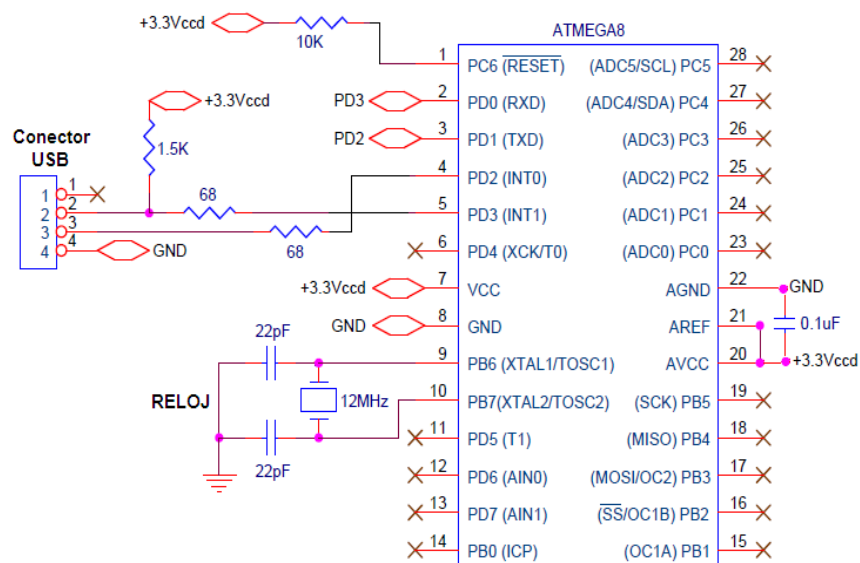


Figura 2.32 Diagrama circuital del interfaz USB

2.4.2.2 Conexión del microcontrolador con el interfaz USB

Desde el computador al adaptador de niveles se usa el cable de comunicación USB socket figura 2.33 del cual se utilizan los siguientes pines como se observa en la tabla 2.9.



Figura 2.33 USB Socket

PIN μ C	CONEXIÓN	USB SOCKET-PIN
	Vcc (opcional)	1
5	PD3	2
4	PD2	3
8-22	Vss (tierra)	4

Tabla 2.9 Conexión del microcontrolador al USB Socket

Como se puede apreciar en la conexión del microcontrolador ATmega8 con el conector de bajada USB (terminal tipo B), no se toma en cuenta el voltaje V_{DC} de polarización del microcontrolador como fuente de voltaje para el USB, que como se indica es opcional, ya que el utilizar o no esta fuente no repercute en el funcionamiento de las comunicaciones USB entre nuestro módulo y el computador.

2.5 DESARROLLO DEL SOFTWARE

El sistema de control de acceso desarrollado en el presente proyecto requiere de diferentes tipos de software como son:

- Para los módulos XBee.
- Para los microcontroladores ATmega8 y Amega164P.
- Y para el computador.

2.5.1 PROGRAMACIÓN DEL MÓDULO XBee

Los módulos XBee pueden ser programados a través de una hyperterminal de Windows y una interface serial con un MAX232 y una serie de comandos llamados AT, pero este método es más complicado.

Existen dos tipos de interfaces, serial y USB que pueden ser utilizadas para programar los módulos XBee con un software propietario llamado X-CTU, el cual permite realizar las operaciones de manera más natural, fácil y rápida. Con este software podemos definir de una forma rápida todos los parámetros que queramos modificar en nuestros módulos XBee.

Una de las ventajas es ²³que podemos tener hasta 65000 combinaciones distintas de red y se pueden hacer redes de punto a punto y punto a multi-punto. El protocolo que utiliza los módulos XBee es 802.15.4 que pertenece a las redes PAN (Personal Área Network).

En la figura 2.34 se muestra la ventana del software X-CTU, con el cual se programan los módulos XBee. En nuestro caso los XBee van a trabajar en modo TX/RX, para lo cual se va a modificar los siguientes parámetros bastante sencillos para poder conectar nuestros módulos que a continuación se observan.

- MY** dirección origen.
- DL** dirección destino.
- BD** baud rate (velocidad de transmisión).

²³ <http://www.decelectronics.com/html/XBEE/XBEE.htm>

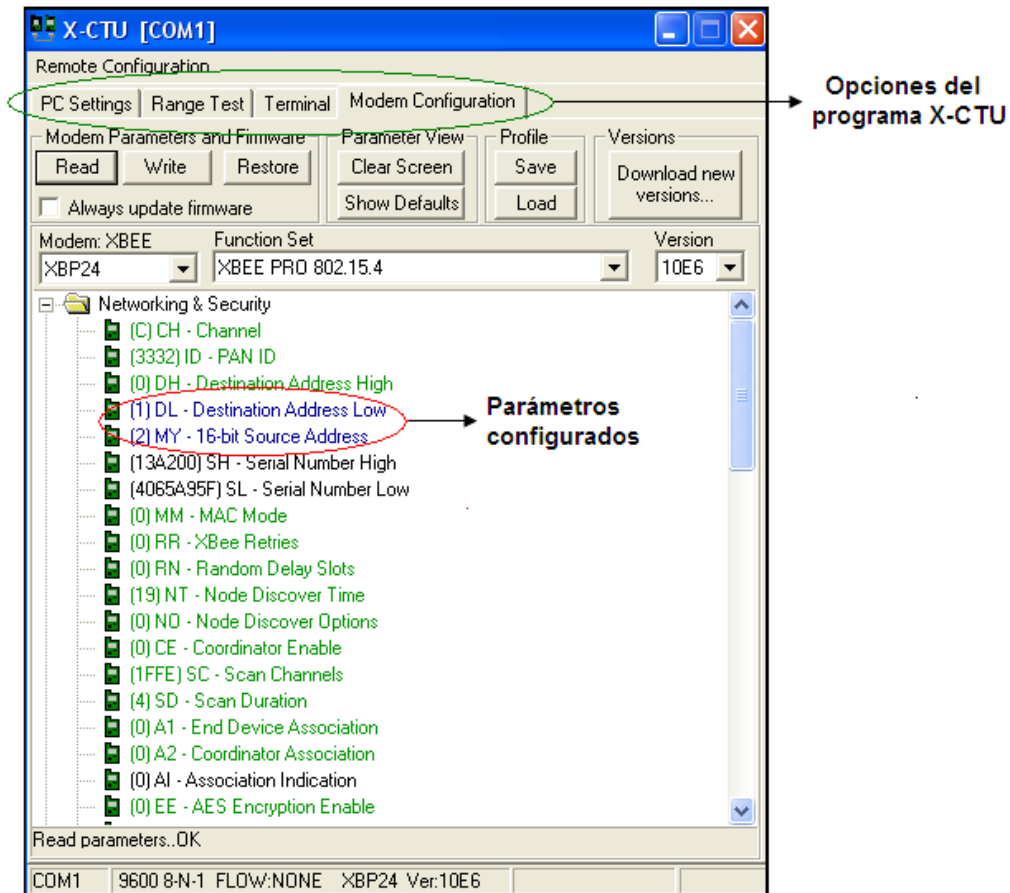


Figura 2.34 Opciones y parámetros del programa X-CTU

2.5.1.1 Proceso de transmisión del módulo XBee

En la siguiente figura 2.35 se muestra la forma de comunicación que tiene el módulo XBee, para la transmisión de datos en forma bidireccional.

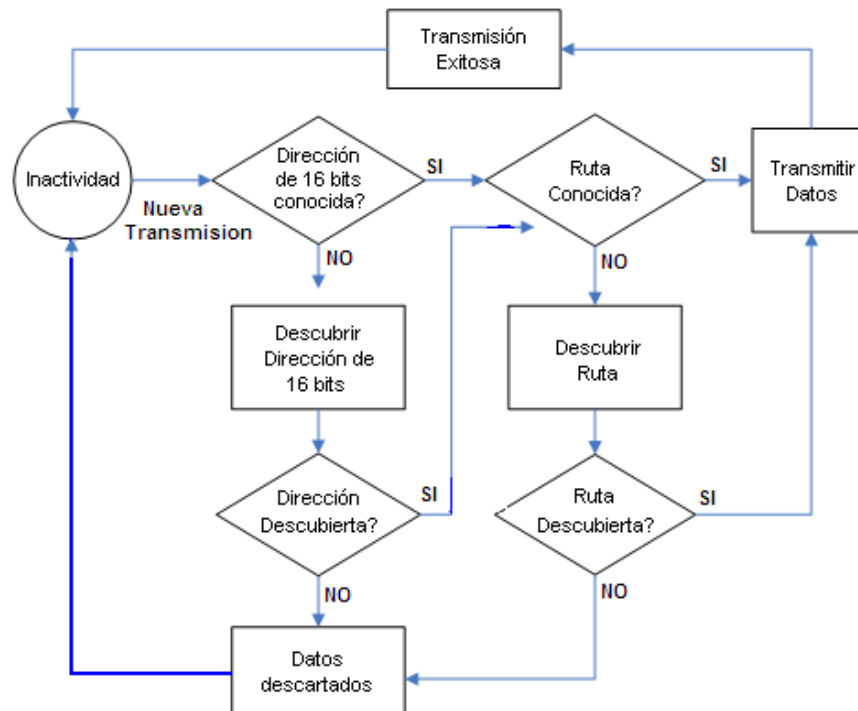


Figura 2.35 Proceso de transmisión del módulo XBee

2.5.2 PROGRAMACIÓN DE LOS MICROCONTROLADORES

Para la programación de los microcontroladores de cada uno de nuestros módulos tanto del control remoto como del receptor, se puede utilizar distintos lenguajes de programación, unos de bajo nivel como el “Assembler” y otros de alto nivel como por el ejemplo AVR Studio basado en lenguaje C, en nuestro caso vamos a utilizar el compilador conocido como Bascom AVR que trabaja con Visual Basic.

En la figura 2.36 se observar la interfaz gráfica del compilador Bascom AVR.



Figura 2.36 Interfaz del Bascom AVR

Este compilador y grabador AVR será el encargado de realizar todas las funciones necesarias para la configuración lógica de los microcontroladores de nuestro prototipo.

2.5.2.1 Interfaz de trabajo del Bascom AVR - Control Remoto

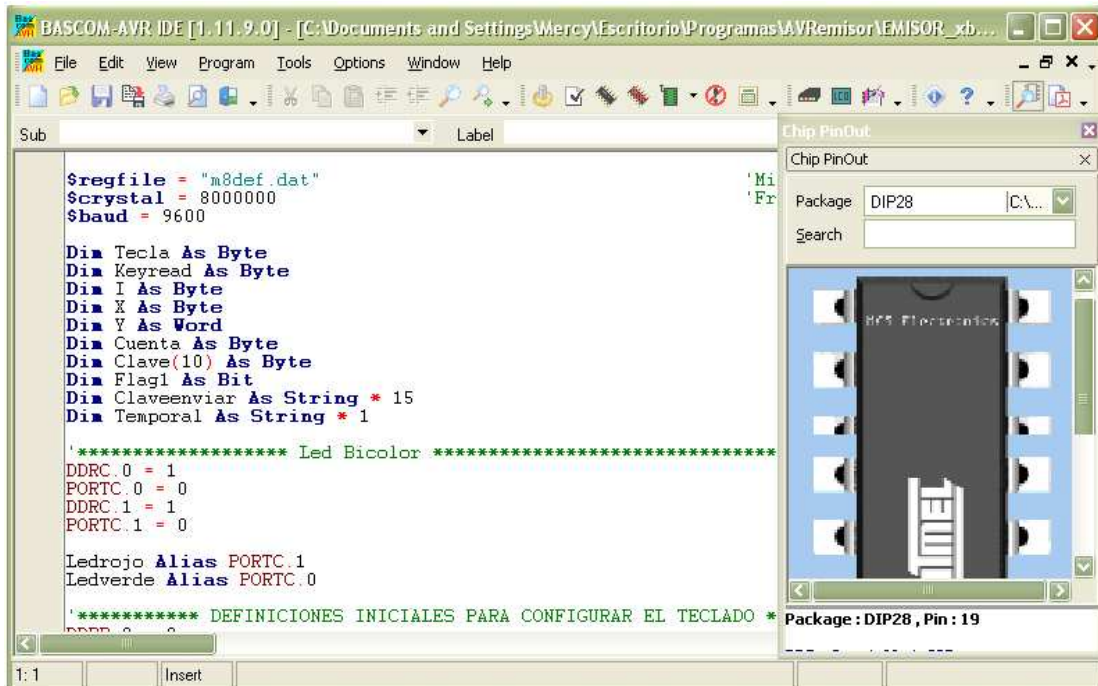


Figura 2.37 Interfaz de trabajo del Bascom AVR - control remoto

Este programa realiza las siguientes funciones:

2.5.2.1.1 Lectura del teclado hexadecimal

El teclado hexadecimal es el componente por donde ingresara cada usuario su clave personal de acceso, este teclado consta de 16 teclas las mismas que se han dividido en 4 filas cuyos nombres son Y1, Y2, Y3 y Y4 y en 4 columnas X1, X2, X3 y X4, conectadas como se puede observar en la figura 2.29. El programa del microcontrolador debe realizar un barrido constante de cada una de las filas y las columnas para realizar el censo al momento de presionar los dígitos del 0 al 9 y las letras A, B, C, y D respectivamente.

2.5.2.1.2 Envió de datos por el pórtico DOUT pin 2 del dispositivo XBee

El programa una vez que realiza la lectura de la clave ingresada por el usuario, ésta clave es almacena en las localidades de la memoria de trabajo y al momento de ser ingresados los cuatro dígitos los envía en un solo paquete al dispositivo de transmisión XBee. Mientras el usuario no presione los cuatro dígitos de su clave, ésta no será transmitida, es decir el microcontrolador no enviará los datos al dispositivo transmisor XBee.

2.5.2.2 Interfaz de trabajo del Bascom AVR - Módulo Receptor

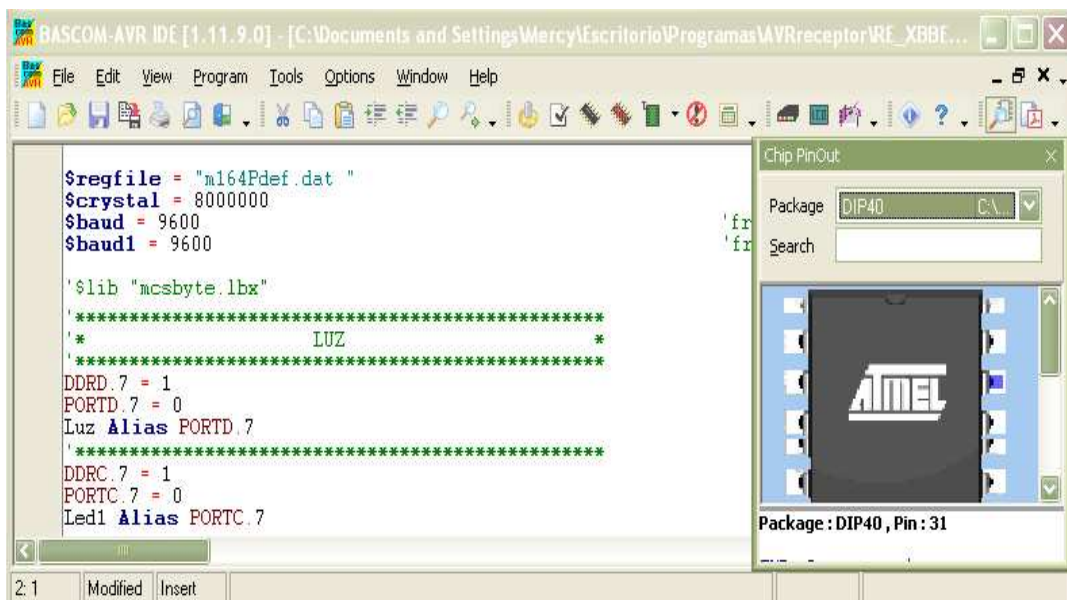


Figura 2.38 Interfaz de trabajo del Bascom AVR - módulo receptor

A continuación se presenta un diagrama de flujo con las secuencias que debe seguir el software del microcontrolador Atmega164P del módulo receptor.

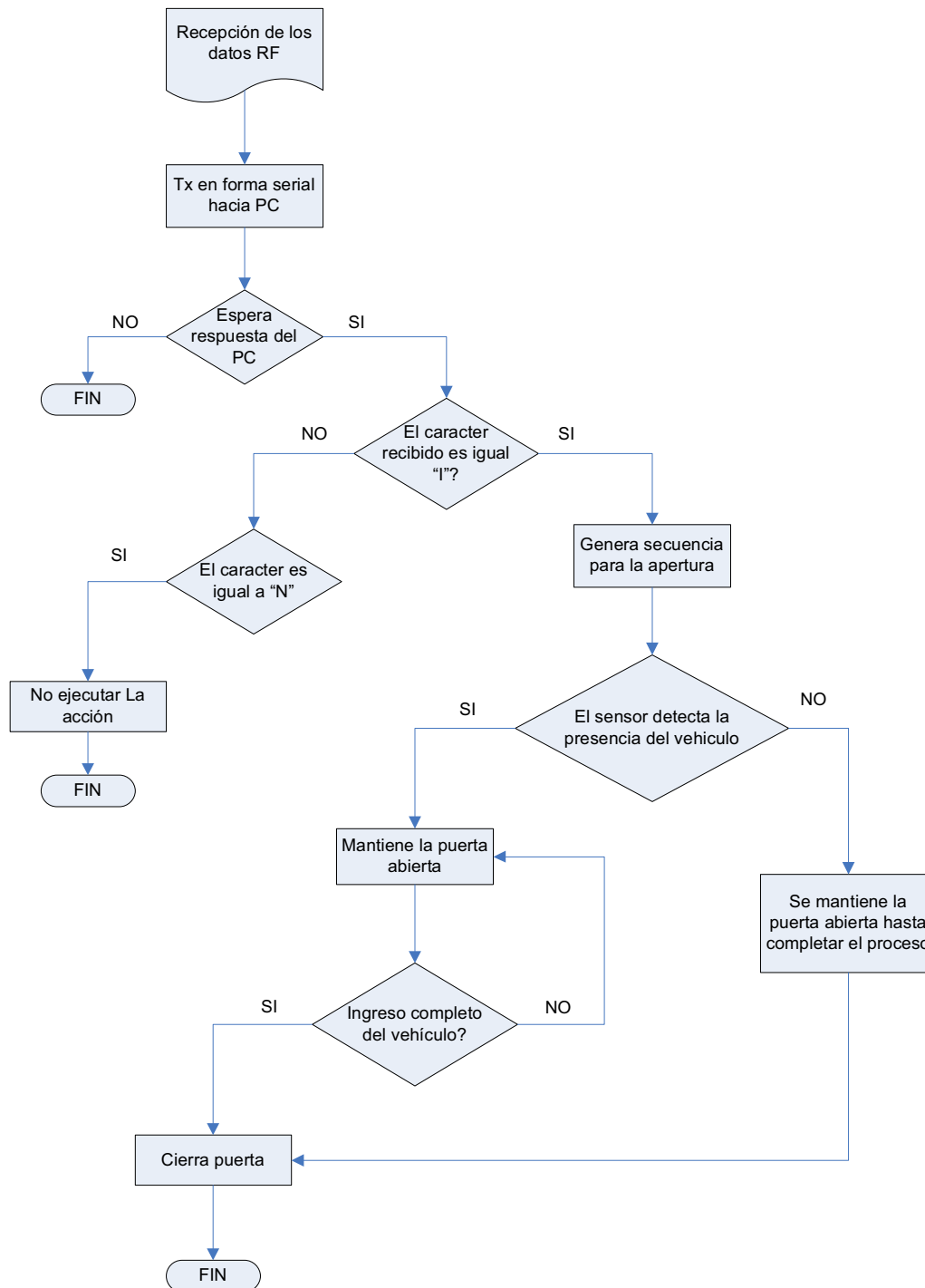


Figura 2.39 Diagrama de flujo del módulo control receptor

El programa del microcontrolador del módulo receptor va realizar las siguientes funciones:

2.5.2.2.1 Recepción de los datos provenientes del control remoto

La clave que es digitada por el usuario desde cualquier posición, es transmitida en forma serial en grupos de 4 dígitos hacia el módulo receptor y son recibidos uno tras otro por el módulo de radiofrecuencia XBee.

Cuando este evento ocurre el microcontrolador recibe el dato y lo almacena temporalmente para luego enviarlo al computador, esperando una respuesta positiva o negativa de la clave ingresada.

2.5.2.2.2 Comunicación serial USB entre el módulo receptor y el computador

La recepción de los datos se realiza a través de la interrupción de comunicación serial UART del microcontrolador, la cual realiza la recepción y control de las respuestas que va enviar el computador.

Por lo tanto, esta interrupción va seguir diferentes secuencias para la recepción de los datos. La primera secuencia consiste en leer el dato y almacenarlo en un registro de memoria interna para luego compararlo con el dato guardado en su memoria EEPROM.

La segunda secuencia es la de realizar una acción para el control del motor paso a paso, esta acción consiste en girar el eje del motor gradualmente el mismo que se encuentra ensamblado a la puerta permitiendo el ingreso o salida del vehículo y devolverlo a su posición original (de cerrado).

2.5.2.2.3 Control del movimiento del eje del motor paso a paso

El dispositivo que genera el movimiento de la puerta de garaje es un motor paso a paso unipolar, modelo 6C794-9212K de 7.5v y 1.8° que cumple con las características y requerimientos necesarios.

Una vez que el microcontrolador recibe una respuesta positiva del computador, es decir el caracter "I", este va a la parte del programa en donde se encuentra la

secuencia para activar el movimiento del eje del motor paso a paso desde una posición especificada.

El eje del motor paso a paso va dar más de 2 vueltas para abrir o cerrar la puerta de garaje, ayudándose también de los fines de carrera lo cual permite un mejor control de apertura, permitiendo el acceso del vehículo, además cabe recalcar casos especiales de funcionamiento los cuales explicaremos a continuación:

- Si el vehículo al ingresar o salir no atraviesa completamente la puerta, es decir por diferentes razones se queda en medio de la puerta, el microcontrolador no dará la orden de cerrarla hasta que se haya censado que no exista ningún vehículo.
- Si se desea el ingreso de otras personas ajenas al domicilio que no tienen el control remoto, la puerta se abrirá y cerrará mediante el uso de un pulsador que hará que el microcontrolador trabaje en modo apertura manual.
- Además se van utilizar otras teclas del control para su apertura como son: la tecla “A” para que la puerta se pare en el momento que se desee y “B” para acceder de nuevo al control remoto o al pulsador para abrir y cerrar la puerta dependiendo de las circunstancias.

2.5.2.2.4 Inicio y control de los sensores infrarrojos

Los elementos que permiten censar si el vehículo ingresa completamente o no, son los sensores infrarrojos que están colocados a una distancia de 20cm.

Estos sensores constan de un haz de luz, la cual si es interrumpida por el vehículo mandan un pulso al microcontrolador indicando que no se debe cerrar la puerta, caso contrario los sensores indican que el vehículo a ingresado ó salido completamente y se cierra la puerta.

2.5.3 SOFTWARE PARA EL COMPUTADOR

El programa en el computador personal constituye una interfaz entre el operador autorizado y el sistema de control de acceso, a través del cual se definen las claves de los usuarios autorizados y registra la actividad de entrada y salida en una base de datos que permitirá llevar un informe detallado de los registros del mismo. Todo esto se da mediante el programa Microsoft Visual Basic 6.0.

2.5.3.1 Microsoft Visual Basic 6.0

Es un lenguaje de fácil aprendizaje pensado tanto para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios que facilita el rápido desarrollo de aplicaciones gráficas. Su sintaxis, derivada del antiguo BASIC, que ha ido evolucionando con modernos lenguajes estructurados.

Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL, etc) a través de ADO.

2.5.3.1.1 Características de Microsoft Visual Basic 6.0

- Es un diseñador de entorno de datos: donde es posible generar, de manera automática, conectividad entre controles y datos mediante la acción de arrastra y colocar sobre formularios o informes.
- Asistente para formularios: sirve para generar de manera automática formularios que administren registros de tablas o consultas pertenecientes a una base de datos, hoja de cálculo u objeto.
- En aplicaciones HTML: se combinan instrucciones de Visual Basic con código HTML para controlar los eventos que se realizan con frecuencia en una página Web.

2.5.3.2 Interfaz de trabajo del Microsoft Visual Basic 6.0

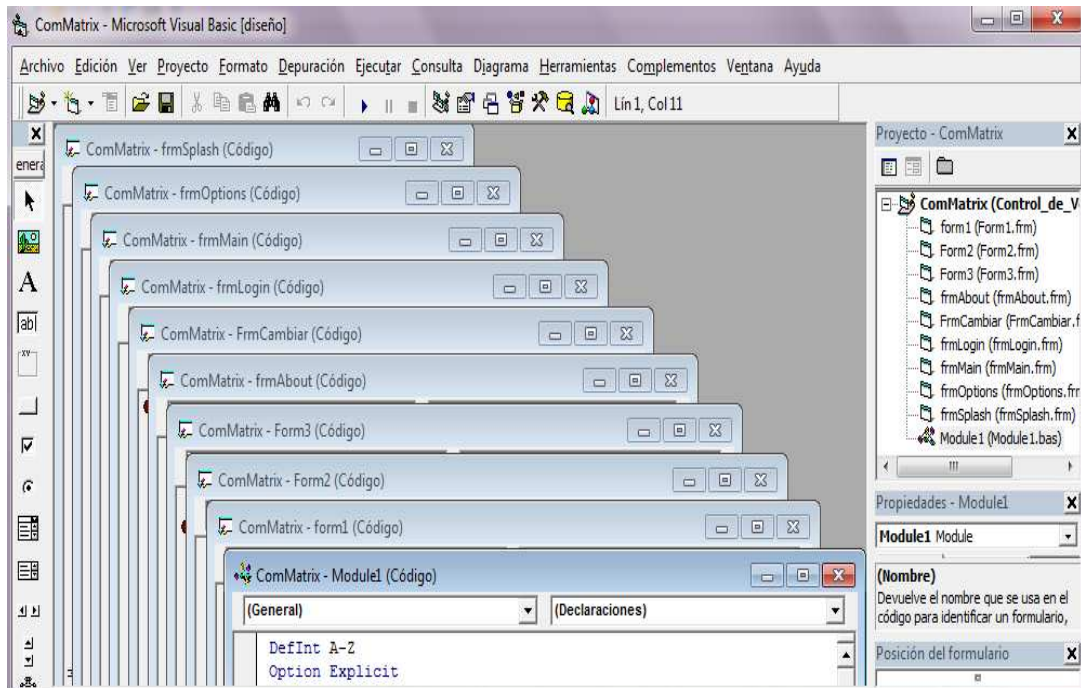


Figura 2.40 Interfaz de trabajo del Microsoft Visual Basic 6.0

2.5.3.3 Diagrama de bloques del Software en el computador

El programa Microsoft Visual Basic 6.0 comprende la organización coherente de los datos de entrada y salida, es decir, no va existir una actividad directa entre el operador y la base de datos, pero si se mantiene una relación directa entre máquina y operador.

En la siguiente figura 2.41 se observa las actividades que desarrolla el Microsoft Visual Basic 6.0 las cuales son: una interfaz para el operador, una comunicación serial con el microcontrolador del módulo receptor y una comunicación con su base de datos.



Figura 2.41 Diagrama de bloques del software del computador

Este interfaz de Microsoft Visual Basic 6.0 va comprender los siguientes aspectos:

- Una interfaz amigable hacia el operador del equipo. En el que constan las diferentes pantallas para el ingreso, registro, cambios y visualización de datos de forma accesible y comprensible para el operador, permitiendo una comunicación serial con el microcontrolador.

Al momento de iniciar el programa el computador pide al usuario que describa el puerto de comunicación libre para el intercambio de datos, debido a que no todos los computadores tienen el mismo puerto libre. Se crea una ventana de comunicación a fin de chequear constantemente el puerto, para almacenar en cualquier momento los datos enviados por el microcontrolador.

- La programación del protocolo de comunicación entre el microcontrolador y el computador.
- Una comunicación del Microsoft Visual Basic 6.0 con el operador, para el ingreso y consultas desde y hacia la base de datos.

En el siguiente gráfico se muestra el diagrama de flujo que refleja todo lo explicado anteriormente correspondiente a la recepción serial de datos.

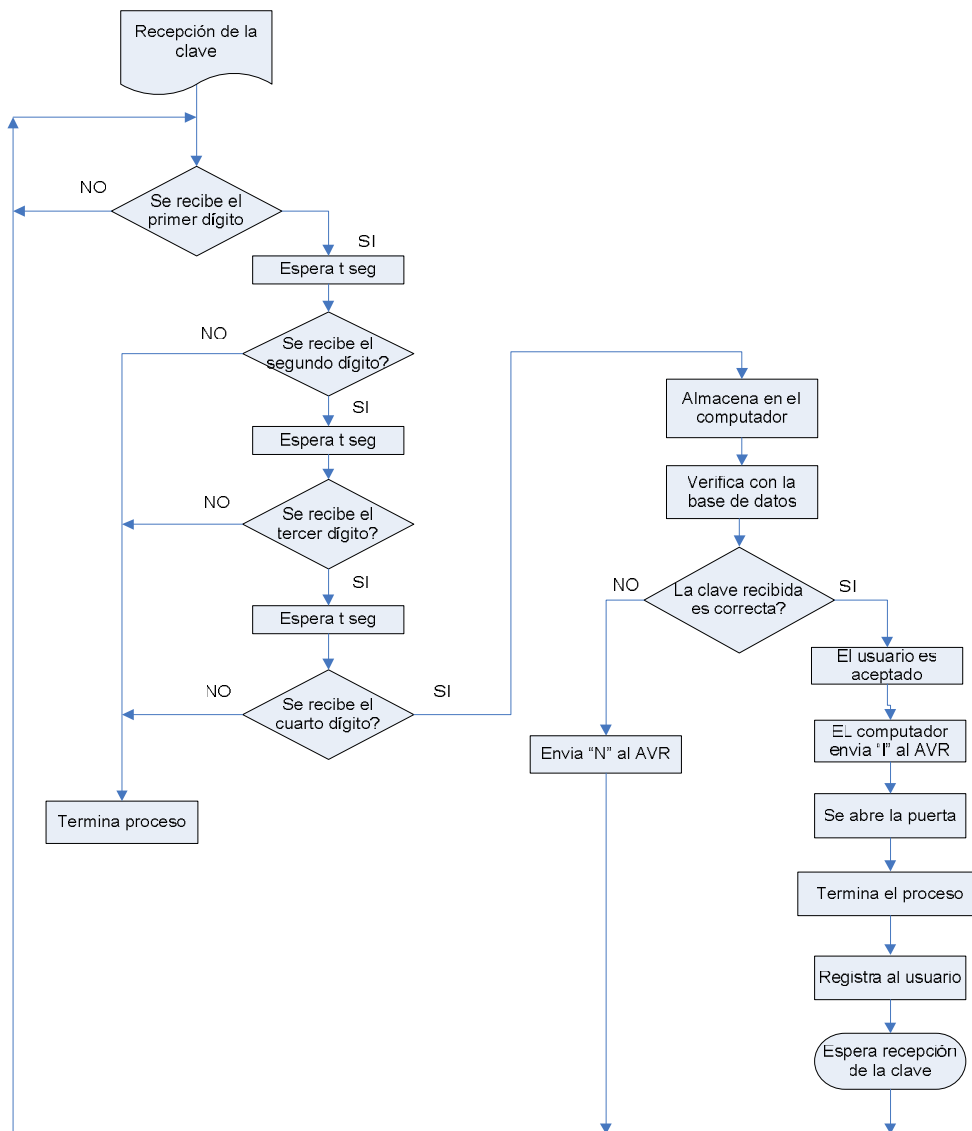


Figura 2.42 Diagrama de flujo del software en el computador

Cuando el usuario es registrado, se guardara en la base de datos: su entrada, su salida, la fecha y hora en la que se realizó el evento.

2.6 CONSTRUCCIÓN DEL PROTOTIPO

El siguiente paso para la elaboración del proyecto fue la adquisición de los elementos que intervienen en los diferentes módulos, tomando en cuenta los diagramas descritos anteriormente.

2.6.1 SOFTWARE UTILIZADO

Para la implementación de nuestros módulos el primer paso fue el diseño y ruteo de los circuitos electrónicos utilizando el programa Orcad Capture for Windows. Figura 2.43.

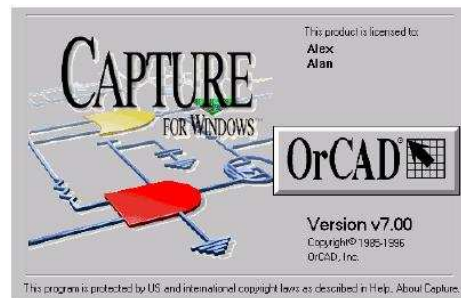


Figura 2.43 Pantalla de inicio ORCAD CAPTURE for Windows

El cual nos genera páginas y diseños esquemáticos, por lo cual todos los diagramas y partes del esquemático son almacenados dentro de una librería y pueden fácilmente ser copiados, movidos y editados por el usuario, además nos presenta una gran cantidad de librerías con una variedad de dispositivos, lo que nos permite crear librerías, dispositivos, nombrarlos, modificarlos, copiarlos, etc.

Orcad Capture for Windows nos permite crear las pistas que serán transferidas a la placa de baquelita para su posterior tratamiento. Además el programa layout for Windows nos permite diseñar y realizar el ruteo de los circuitos impresos de manera automática o manual, partiendo del diagrama circuital creado en el Orcad Capture for Windows.



Figura 2.44 ORCAD LAYOUT for Windows

2.6.2 DIAGRAMA CIRCITAL DEL CONTROL REMOTO ELABORADO EN

ORCAD

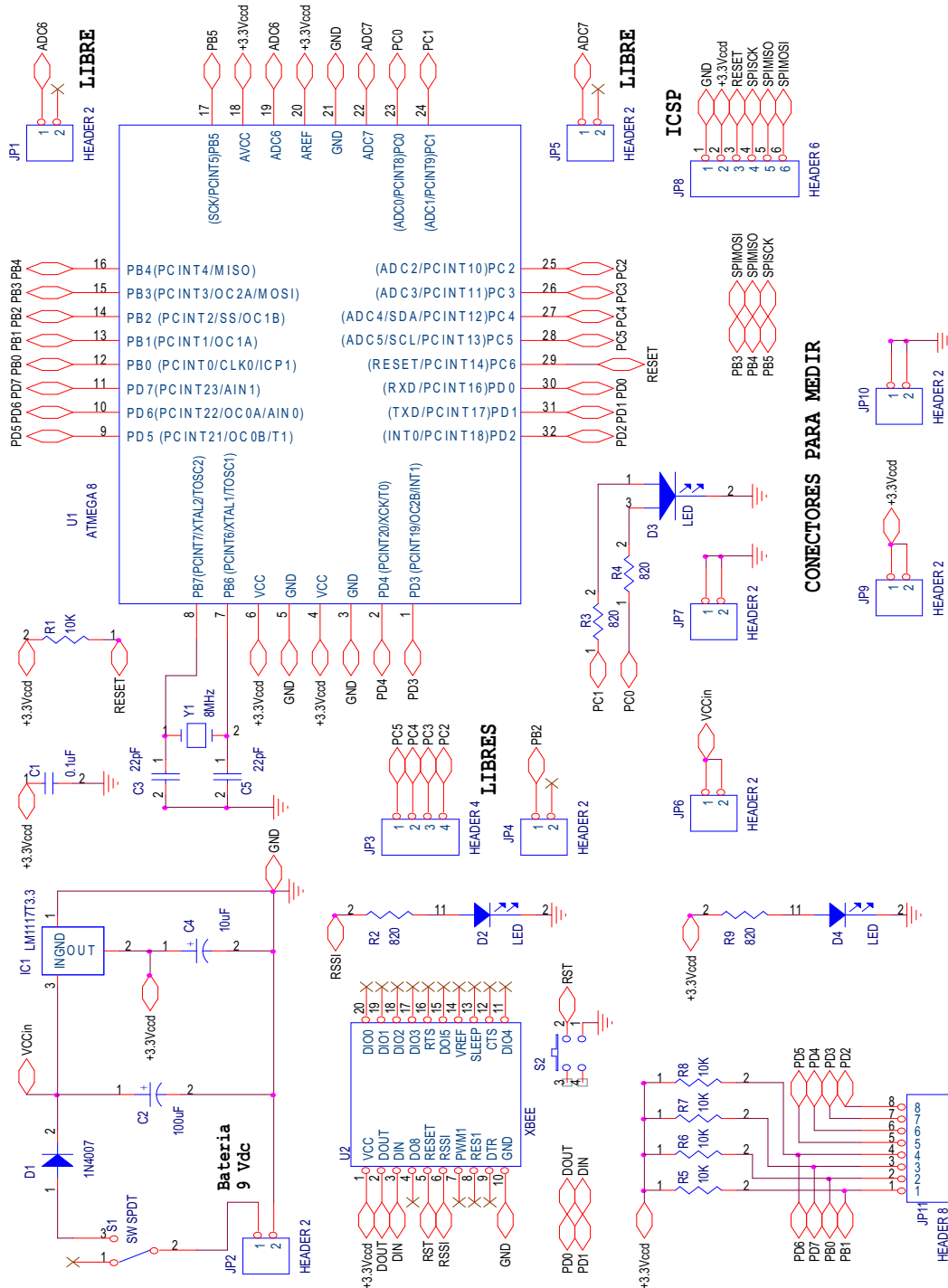


Figura 2.45 Diagrama circuital del control remoto

2.6.2.1 Circuito impreso y Screen de la tarjeta

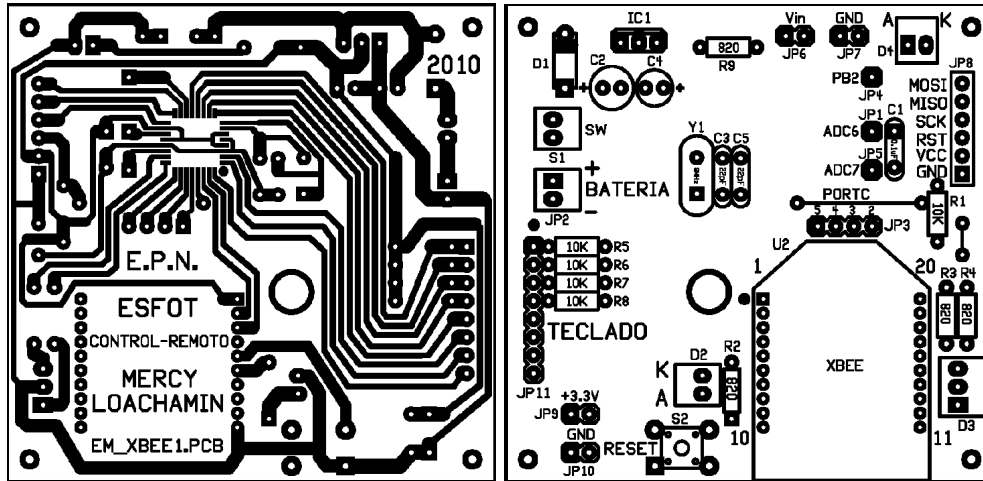


Figura 2.46 Circuito impreso y screen final del módulo control remoto

2.6.2.2 Elaboración de la baquelita y tarjeta electrónica ensamblada

Una vez que se ha concluido el proceso de diseño de todas las pistas correspondientes al control remoto, procedemos a imprimirlas en papel termotransferible, para luego pasarlo a la baquelita.

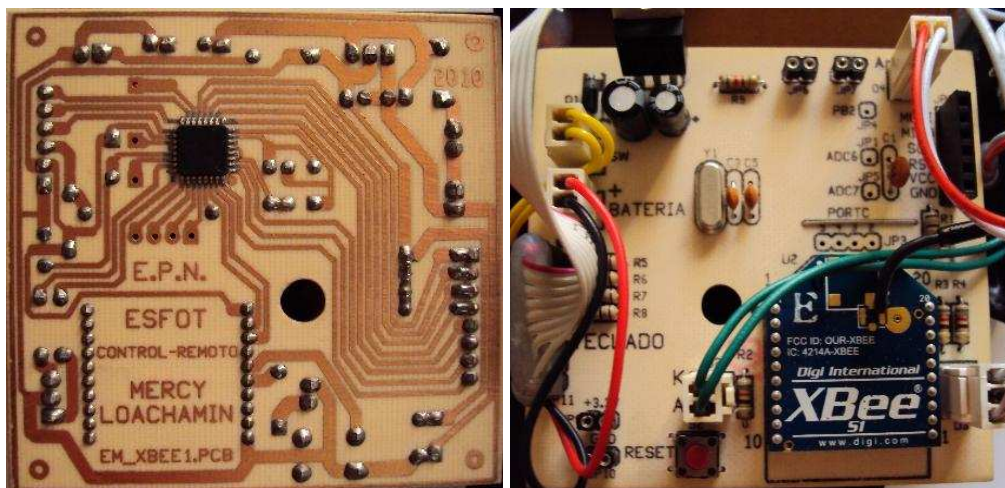


Figura 2.47 Baquelita impresa y tarjeta electrónica final

2.6.3 DIAGRAMA CIRCITAL DEL MÓDULO RECEPTOR ELABORADO EN

ORCAD

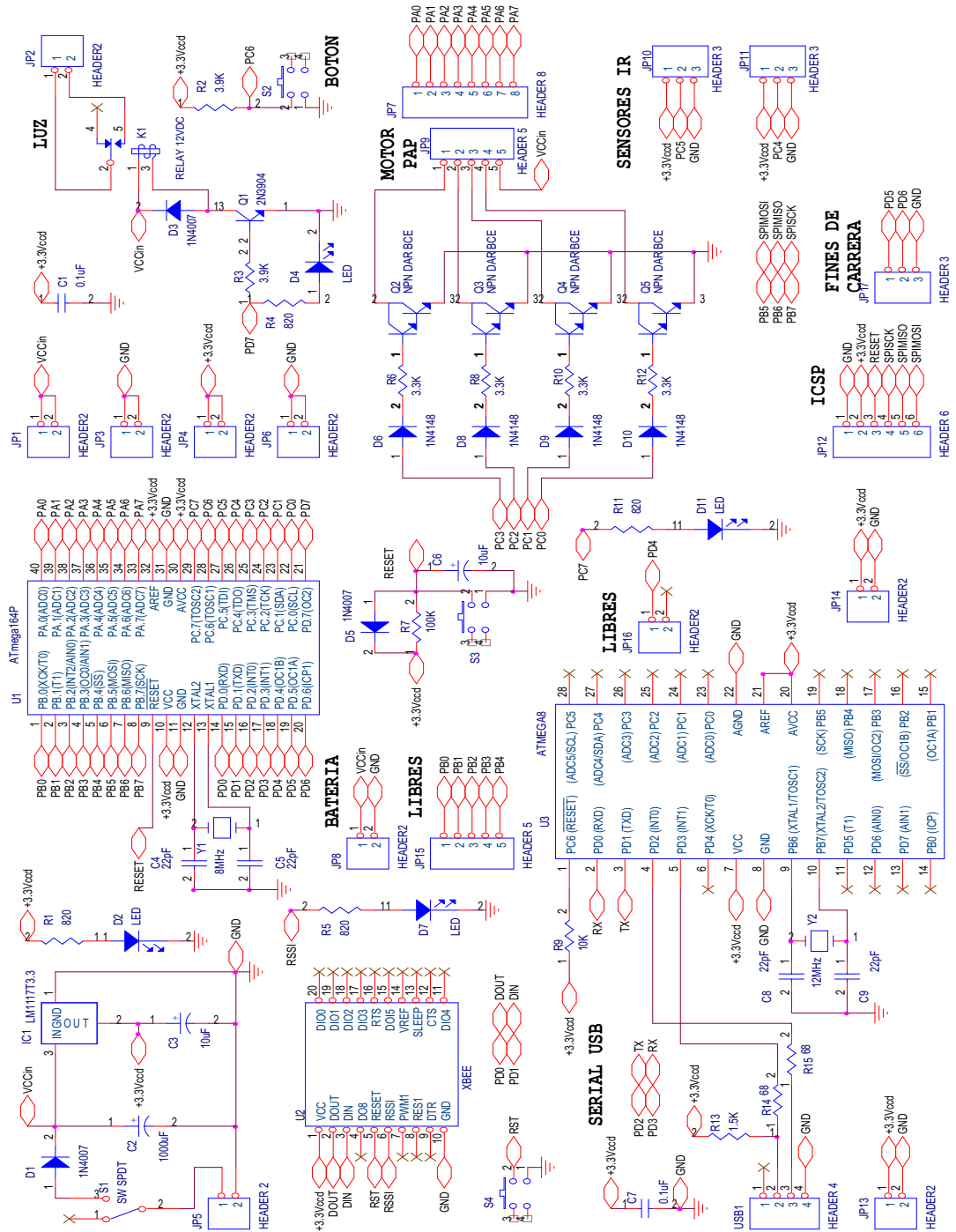


Figura 2.48 Diagrama Circital del módulo receptor

2.6.3.1 Circuito impreso

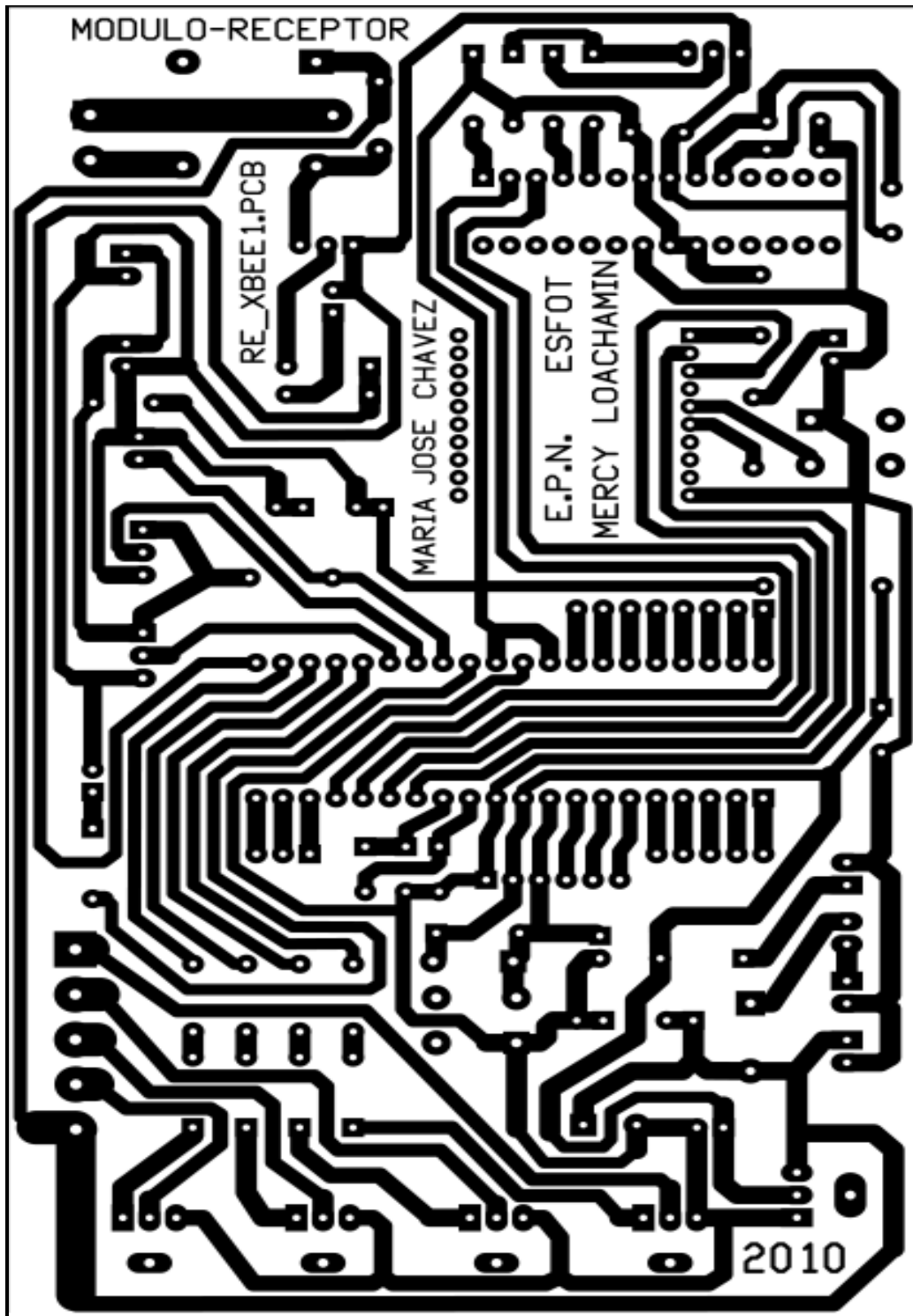


Figura 2.49 Circuito impreso final módulo receptor

2.6.3.2 Screen de tarjeta

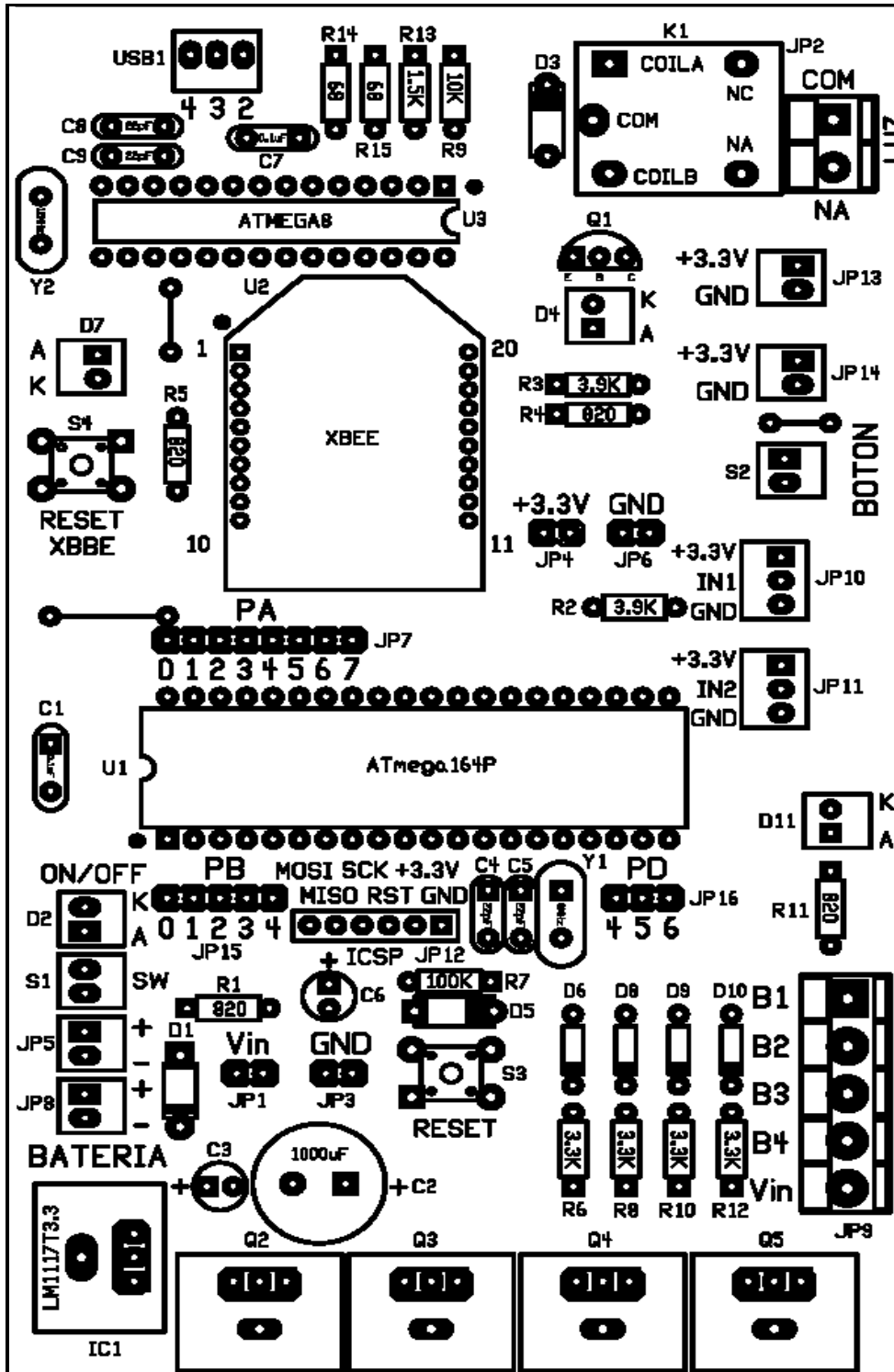


Figura 2.50 Screen de tarjeta ensamblada módulo receptor

2.6.3.3 Elaboración de la baquelita

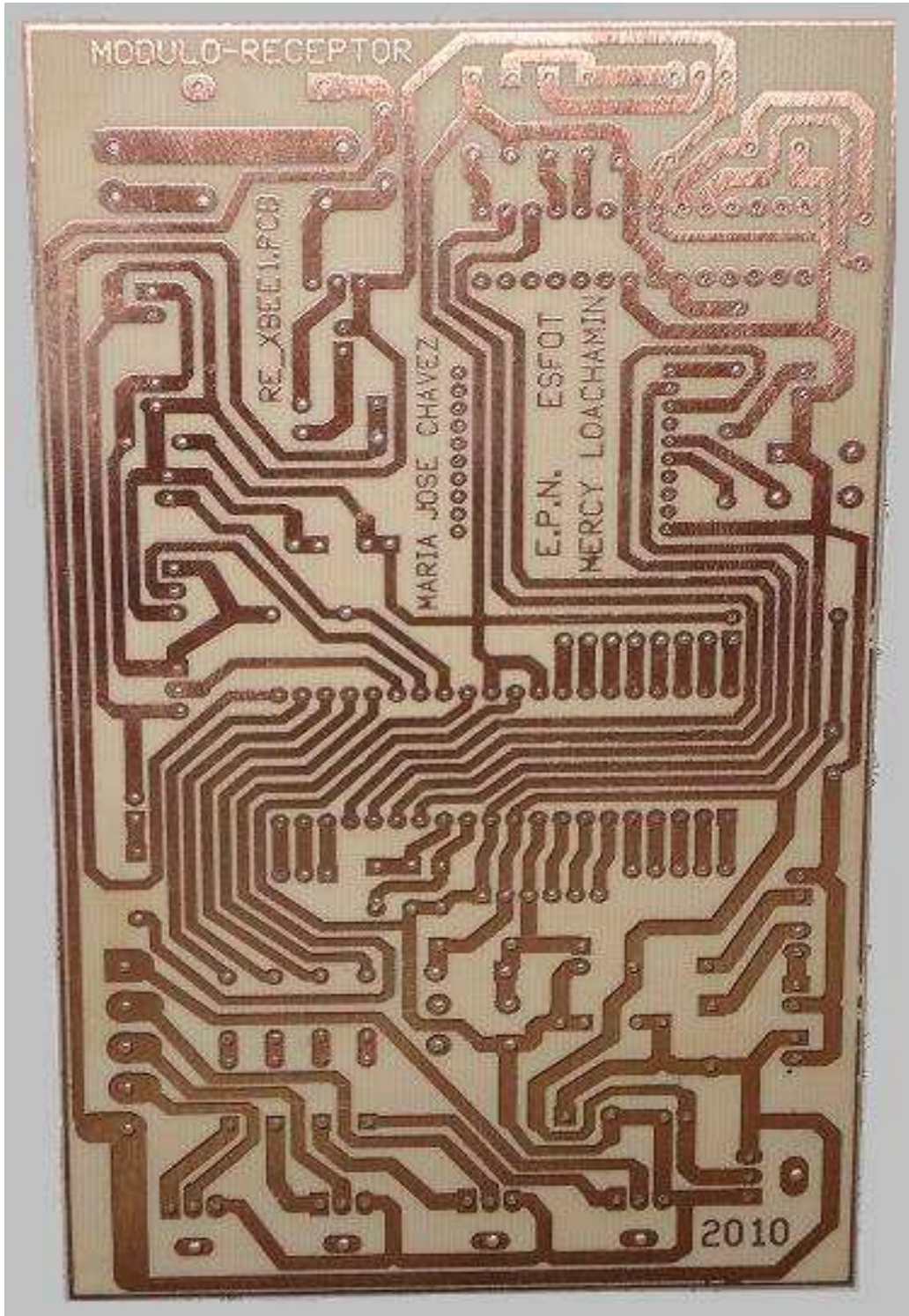


Figura 2.51 Baquelita impresa

2.6.3.4 Tarjeta electrónica ensamblada



Figura 2.52 Materiales para el módulo receptor

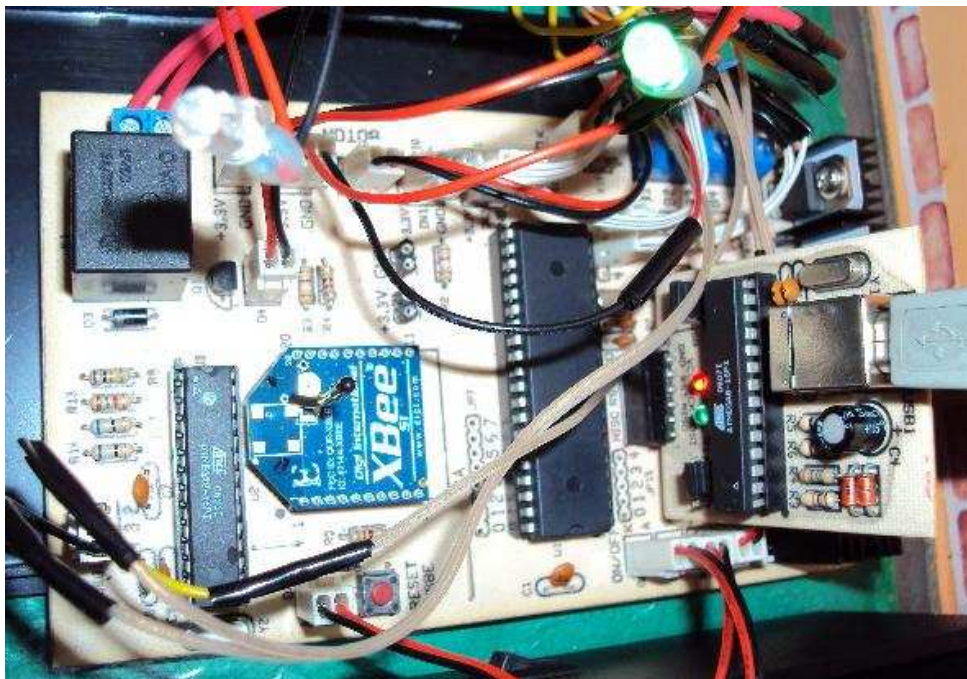


Figura 2.53 Tarjeta electrónica al final del proceso de soldadura

2.6.4 DIAGRAMA CIRCUITAL DE LOS SENSORES INFRARROJOS ELABORADOS EN ORCAD

2.6.4.1 Módulo EMISOR

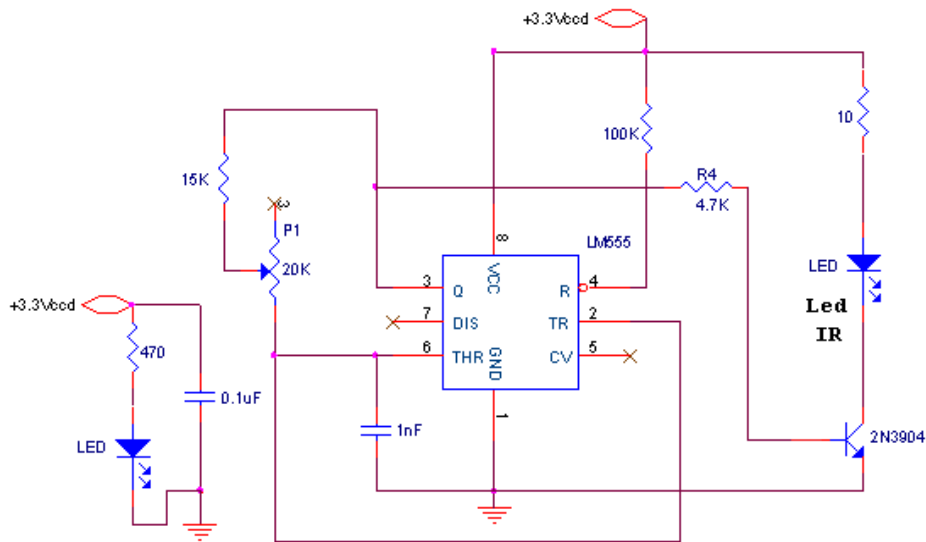


Figura 2.54 Diagrama circuital del módulo EMISOR

2.6.4.1.1 Circuito impreso y Screen de la tarjeta

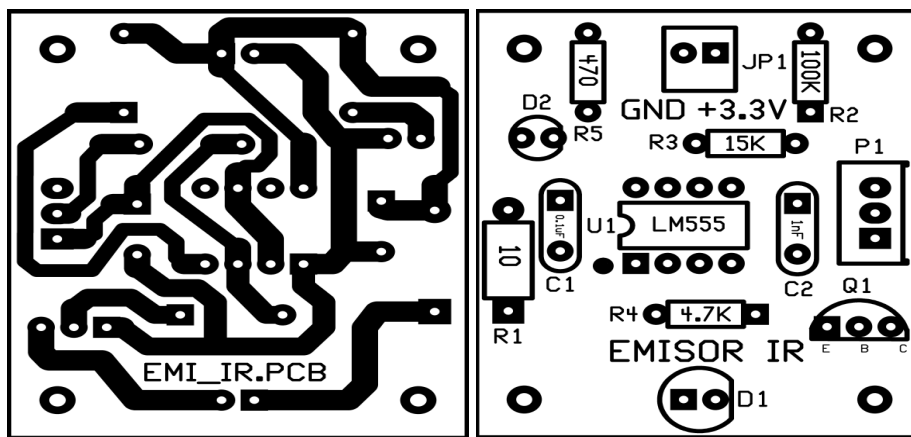


Figura 2.55 Circuito impreso y screen final del EMISOR

2.6.4.2 Módulo RECEPTOR

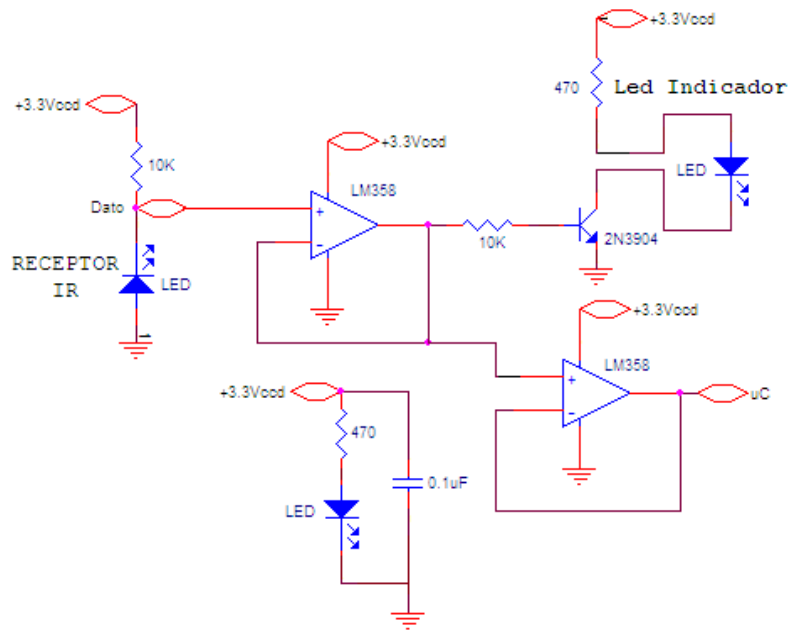


Figura 2.56 Diagrama circuital del módulo RECEPTOR

2.6.4.2.1 Circuito impreso y Screen de la tarjeta

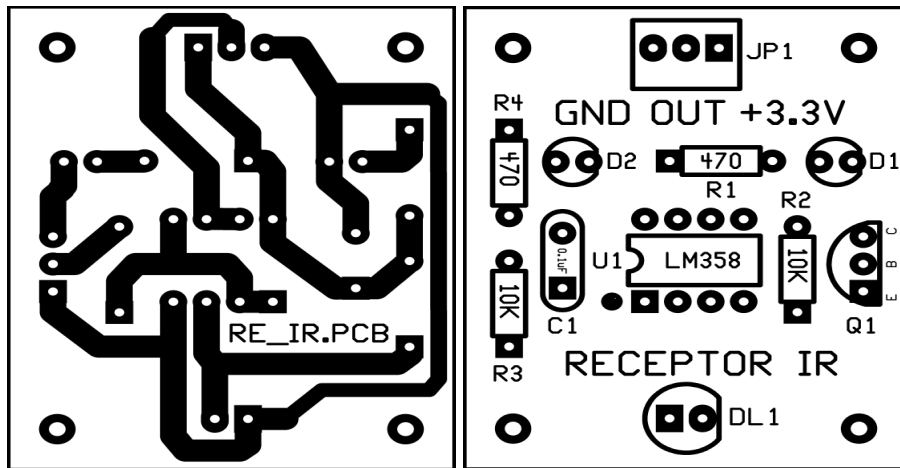


Figura 2.57 Circuito impreso y screen final del RECEPTOR

2.6.4.3 Elaboración de las baquelitas

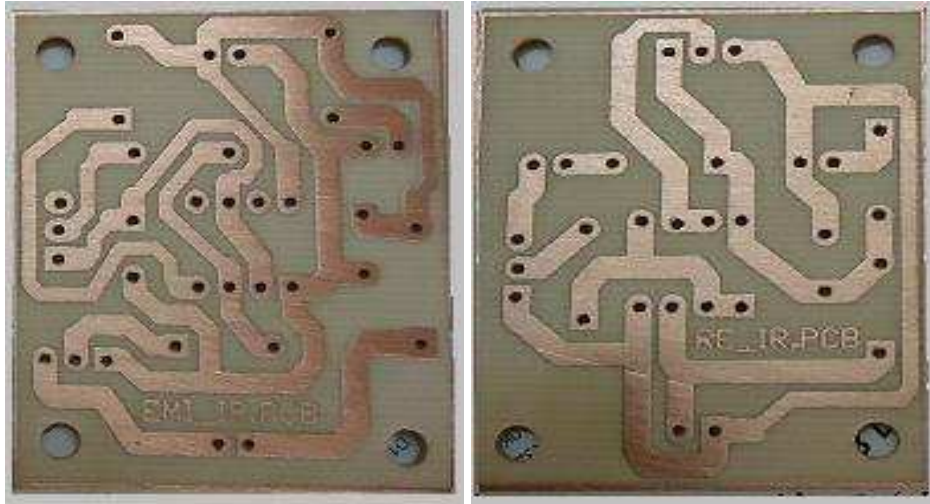


Figura 2.58 Baquelita impresa

2.6.4.4 Tarjetas electrónicas ensambladas

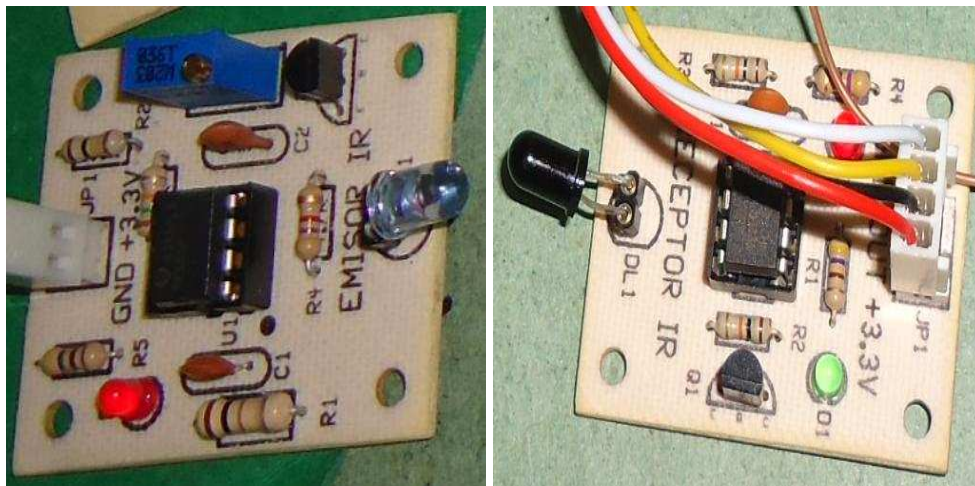


Figura 2.59 Tarjeta electrónica al final del proceso de soldadura

2.7 COSTO DE LOS EQUIPOS Y MATERIALES UTILIZADOS

A continuación se detalla el listado de los componentes utilizados durante la construcción del prototipo, el cual se lo ha dividido en ítems según los diferentes módulos que se realizaron con sus respectivos costos.

2.7.1 MATERIALES DEL MÓDULO CONTROL REMOTO

ELEMENTOS O DISPOSITIVOS	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Capacitor 0.1uF	1	0.15	0.15
Capacitor 100uF	1	0.3	0.3
Capacitor 22pF	2	0.15	0.3
Capacitor 10uF	1	0.15	0.15
Diodo 1N4007	1	0.15	0.15
LEDs	3	0.3	0.9
Regulador LM1117T3.3	1	2	2
Jumpers 2 pines	8	0.3	2.4
Jumpers 4 pines	1	0.5	0.5
Jumpers 6 pines	1	0.7	0.7
Jumpers 8 pines	1	0.9	0.9
Resistencia 10K Ω	5	0.05	0.25
Resistencia 820 Ω	4	0.05	0.2
SW micro interruptor	1	0.4	0.4
Pulsador	1	0.4	0.4
ATmega8	1	8	8
Zócalo para el módulo XBee	1	4	4
Cristal de 8MHz	1	1	1
Conectores 2 pines	3	0.3	0.9
Conectores XBee	1	2	2
Caja	1	5	5
Teclado	1	10	10
Módulo XBee	1	40	40
VALOR 1			80.6

2.7.2 MATERIALES DEL MÓDULO RECEPTOR

ELEMENTOS O DISPOSITIVOS	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Capacitor 0.1uF	2	0.15	0.3
Capacitor 1000uF	1	0.4	0.4
Capacitor 10uF	2	0.15	0.3
Capacitor 22pF	4	0.15	0.6
Diodos 1N4007	3	0.15	0.45
LEDs	4	0.4	1.6
Diodos 1N4148	4	0.15	0.6
Regulador LM1117T3.3	1	2	2
Jumpers 2 pines	9	0.3	2.7
Jumpers 8 pines	1	0.9	0.9
Jumpers 5 pines	2	0.6	1.2
Jumpers 3 pines	3	0.4	1.2
Jumpers 6 pines	1	0.7	0.7
Relé de 12VDC	1	1	1
Transistor 2N3904	1	0.1	0.1
Transistor NPN TIP 122	4	1	4
Resistencia 820Ω	4	0.05	0.2
Resistencia 3.9K	2	0.05	0.1
Resistencia 3.9K	4	0.05	0.2
Resistencia 100K	1	0.05	0.05
Resistencia 10K	1	0.05	0.05
Resistencia 1.5K	1	0.05	0.05
Resistencia 68Ω	2	0.05	0.1
SW micro interruptor	1	0.4	0.4
Pulsador	3	0.4	1.2
Puerto USB	1	1	1
ATmega164P	1	6	6
Zócalo para el módulo XBee	1	4	4
ATmega8	1	4	4
Cristal de 8MHz	1	1	1
Cristal de 12MHz	1	1	1
Conectores 2 pines	6	0.3	1.8
Conectores XBee	2	2	4
Caja	1	10	10
Módulo XBee	1	40	40
VALOR 2			93.2

2.7.3 MATERIALES PARA LOS SENSORES INFRARROJOS

ELEMENTOS O DISPOSITIVOS	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Capacitor 0.1uF	2	0.15	0.3
Capacitor 0.1nF	1	0.15	0.15
LED's	4	0.4	1.6
LM555	1	0.25	0.25
LM358	1	0.25	0.25
Jumpers 2 pines	1	0.3	0.3
Jumpers 3 pines	1	0.4	0.4
Resistencia 5KΩ	1	0.05	0.05
Transistor 2N3904	2	1	2
Resistencia 10Ω	1	0.05	0.05
Resistencia 10KΩ	2	0.05	0.1
Resistencia 100KΩ	1	0.05	0.05
Resistencia 15KΩ	1	0.05	0.05
Resistencia 4.7KΩ	1	0.05	0.05
Resistencia 470Ω	2	0.05	0.1
VALOR 3			5.7

2.7.4 OTROS MATERIALES

ELEMENTOS	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Motor paso a paso	1	15	15
Fuente DC 12V – 1.2A	1	13	13
Batería 9 V _{DC}	1	3	3
Maqueta	1	50	50
Cable serial	1	5	5
VALOR 4			86

COSTO TOTAL= VALOR1 + VALOR2 + VALOR3 + VALOR4

COSTO TOTAL= 80.6 + 93.2 + 5.7 + 86

COSTO TOTAL= 265.5

CAPÍTULO 3

PRUEBAS Y RESULTADOS

Una vez diseñado y construido el prototipo nos disponemos a realizar las pruebas respectivas de cada uno los módulos de nuestro prototipo.

3.1 PRUEBAS DEL MÓDULO DE CONTROL REMOTO

Para poder verificar la correcta transmisión de cada uno de los 4 dígitos ingresados por el teclado hexadecimal y el alcance de transmisión del módulo de control remoto, se va usar el programa de comunicación MicroCode Studio, el cual nos permite visualizar cada uno de los dígitos que son enviados desde el control remoto al módulo receptor.

En la siguiente figura 3.1 se muestra las diferentes funciones del teclado del control remoto.

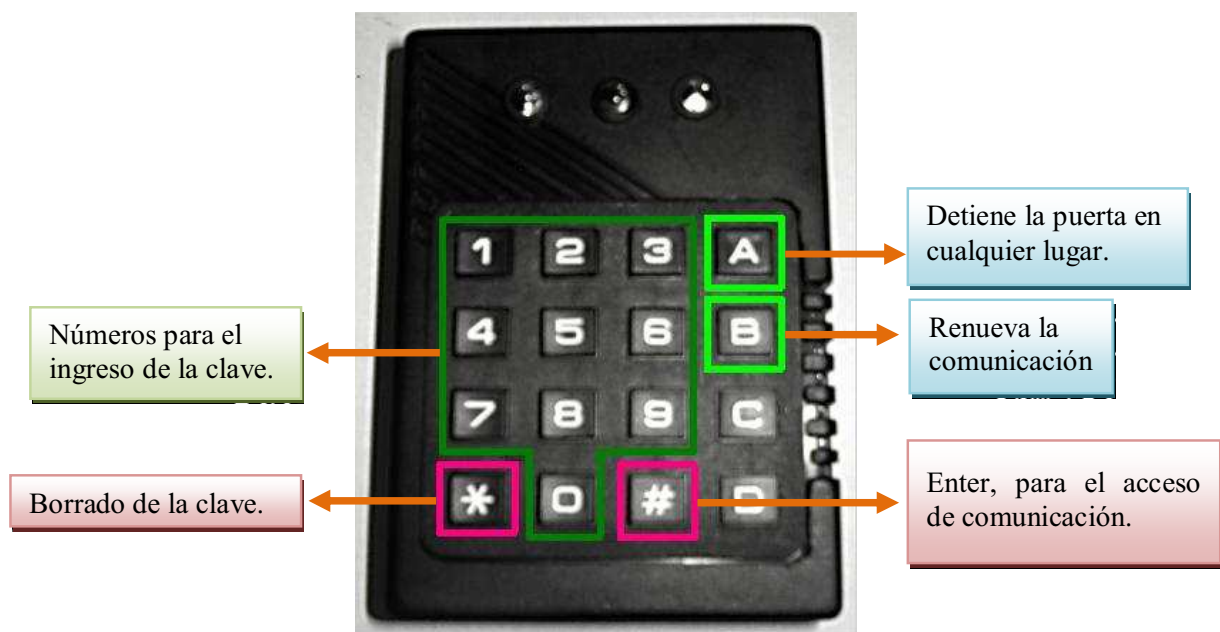


Figura 3.1 Foto del control remoto

El correcto funcionamiento del control remoto depende de la energía de la batería que se utiliza, que en óptimas condiciones alcanzan una distancia de 6,00 metros y entre paredes hasta 1.5 metros aproximadamente esto es un promedio de las pruebas realizadas.

3.1.1 ANÁLISIS DE LA LECTURA DE COMUNICACIÓN SERIAL

Para verificar que nuestro módulo receptor está transmitiendo se ha optado por realizar una prueba de comunicación serial RS232 hacia el computador utilizando el software MicroCode, para lo cual se ha enviado una clave de acceso como se observa en la figura 3.2 y la comunicación serial se la configuró a 9600 buadios, con 1 bit de inicio, 8 bits de datos, 1 bit de parada y ninguno de paridad.

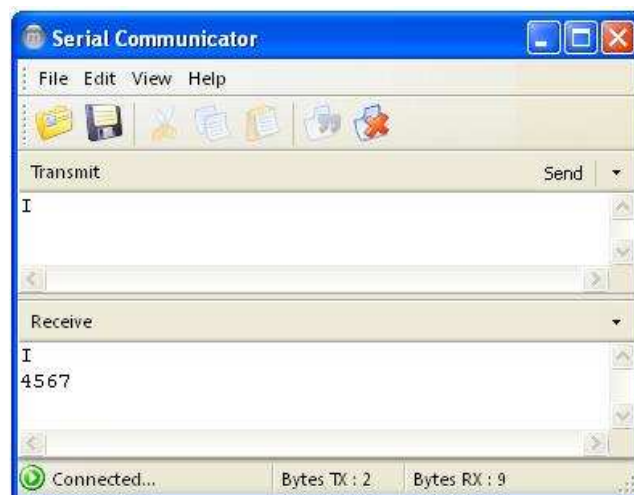


Figura 3.2 Ventana de comunicación MicroCode Studio

3.2 PRUEBAS DEL MÓDULO RECEPTOR Y EL MOTOR PASO A PASO

Para poder revisar la comunicación serial del microcontrolador ATmega164P con el computador se utiliza la ventana de comunicación del Microsoft Visual Basic 6.0 figura 3.3 permitiéndonos visualizar la información que se transmite y se recibe por medio de un puerto com3 de comunicación.

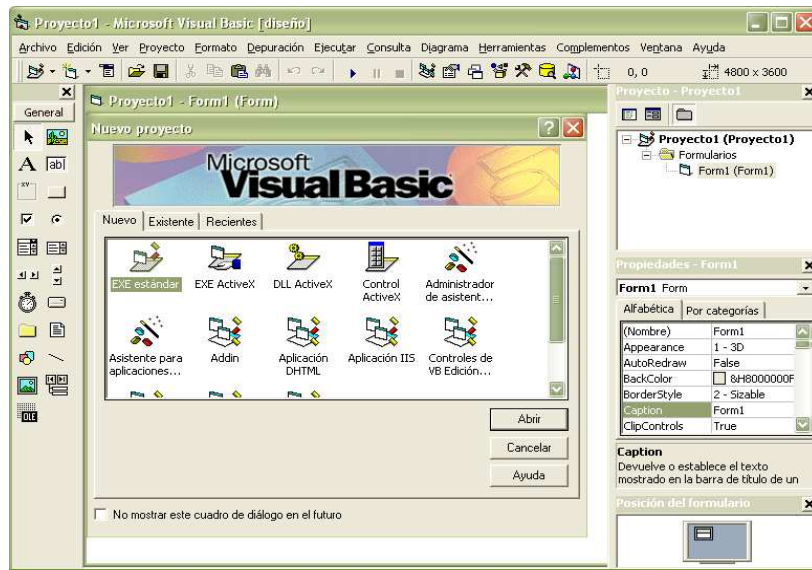


Figura 3.3 Ventana Microsoft Visual Basic 6.0

Al enviar la clave mediante el control remoto el sistema de la computadora comprueba en la base de datos si es o no correcta, y envía la letra "I" al módulo receptor el cual ejecuta el programa para la apertura de la puerta de garaje.



Figura 3.4 Puerta de garaje en reposo



Figura 3.5 El automóvil cruza la puerta



Figura 3.6 Automóvil entre los sensores infrarrojos

3.3 PRUEBAS DEL CONTROL DE ACCESO

El control de acceso está constituido por los siguientes elementos: módulo de control remoto, módulo control receptor, motor paso a paso y el computador para mediante el software adecuado monitorear el sistema.

Las pruebas realizadas son las siguientes:

- Comunicación entre el software del computador y el control de acceso.
- Almacenamiento correcto de la información en el sistema.

Las pruebas se realizaron cuando el equipo del control de acceso y el computador están comunicados utilizando una interfaz USB serial.

3.3.1 COMUNICACIÓN ENTRE EL SOFTWARE DEL PC Y EL CONTROL DE ACCESO

La comunicación entre el computador y el módulo receptor se la realiza mediante un puerto USB del computador que está conectado mediante un cable serial al módulo receptor, para poder habilitar este puerto de comunicación se debe realizar las siguientes configuraciones COM:

- Iniciamos el proceso de configuración del COM, abriendo la ventana de comunicación como se observa en la ventana de la figura 3.7A.



A

- Seguidamente ingresamos al menú para seleccionar el p rtico de nuestra comunicaci n, en nuestro caso se va escoger el puerto Com7 de los varios que existen, figura 3.7B.



B

- Para luego poder inicializar correctamente nuestra comunicaci n, figura 3.7C.



C

Figura 3.7 Ventanas para seleccionar el puerto de comunicaci n

Si el p rtico de comunicaci n no es seleccionado correctamente, se observa el siguiente mensaje, figura 3.8.

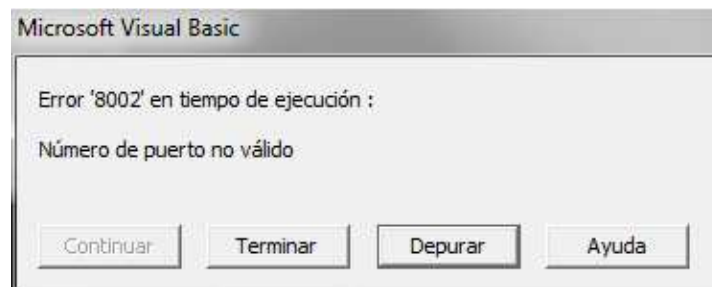


Figura 3.8 Pórtico de comunicación incorrecto.

Cuando el usuario del garaje está haciendo uso del mismo, se muestra la siguiente ventana donde nos indica la comunicación que hay ente el control remoto y la base de datos del sistema, figura 3.9.



Figura 3.9 Ventana de comunicación del control remoto y el sistema

En el momento que se establece la comunicación como se muestra en la figura anterior el motor paso a paso se encuentra realizando el proceso de apertura de la puerta.

3.3.2 ALMACENAMIENTO CORRECTO DE INFORMACIÓN EN EL SISTEMA

La pantalla de interfaz principal para la comunicación y almacenamiento de los datos de nuestro proyecto se muestra en la figura 3.10.



Figura 3.10 Pantalla principal de comunicación

El programa que tenemos en el Microsoft Visual Basic 6.0 nos permite realizar modificaciones como el nombre del usuario y su clave, el proceso es el siguiente:

Ingresamos a la ventana de comunicación, seleccionamos claves/usuarios figura 3.11A, nos aparece la pantalla denominada Lectura/cambio de Usuarios figura 3.11B, leemos los usuarios y si deseamos modificar o añadir ingresamos los datos y guardamos los cambios efectuados como se observa en la figura 3.11C.



Figura 3.11A Ingreso al sistema usuarios

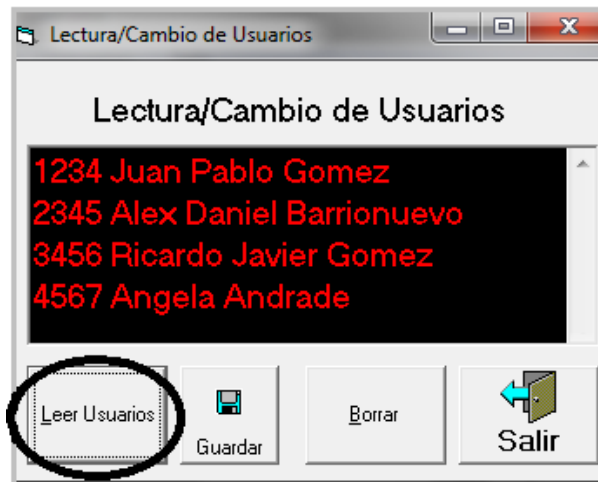


Figura 3.11B Lectura y cambio de usuarios

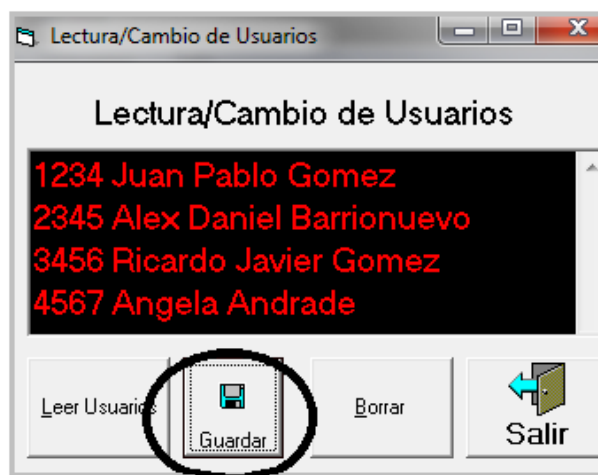


Figura 3.11C Grabación de Usuarios

El Microsoft Visual Basic 6.0 presenta una base de datos donde se guarda los diferentes eventos que se produjeron en el sistema, en el cual podemos identificar si el evento es Entrada o Salida del vehículo ya que cada uno de los sensores está diseñado para identificar cada evento, figura 3.12.

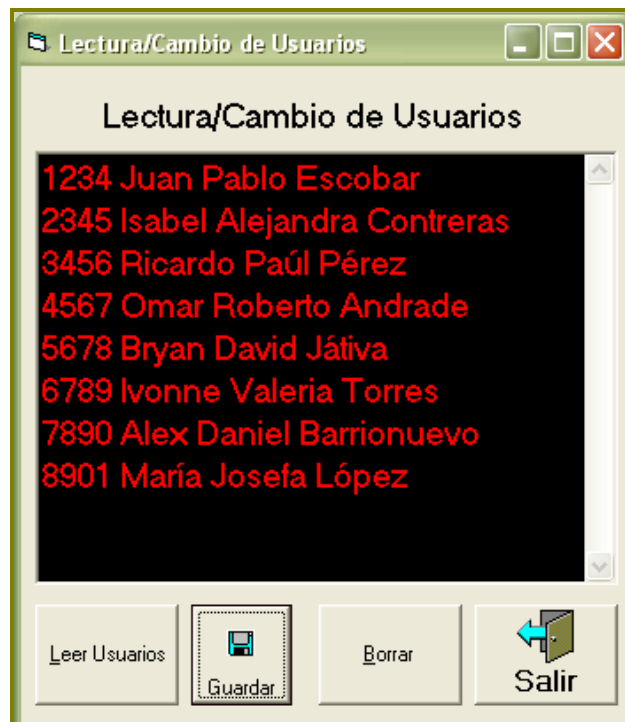


Figura 3.12 Base de datos de los usuarios

Además de presentar una base de datos en Bloc de notas como se muestra en la figura 3.13.

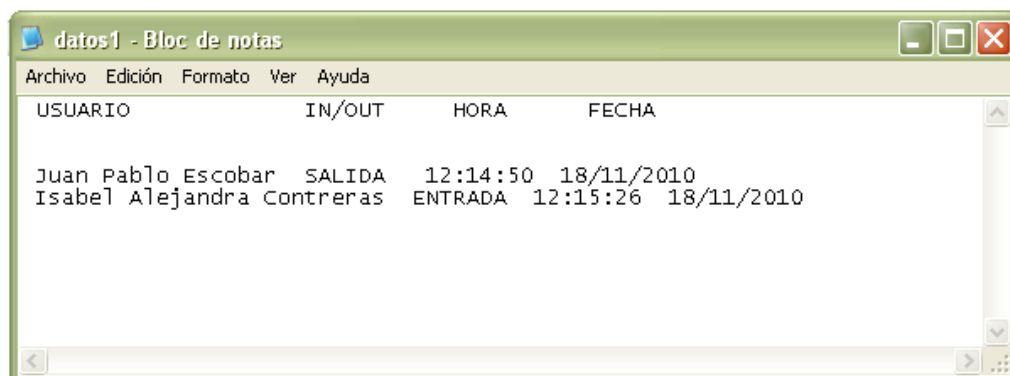


Figura 3.13 Base de datos de los usuarios en Bloc de Notas

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Es indispensable que la línea donde se propaga una señal, este en nivel alto es decir 1L (5v ó 3,3v), ya que los XBee funcionan a mayor capacidad a este voltaje.
- El motor paso a paso unipolar utilizado tiene un grado de precisión de 1.8°, el cual es muy importante ya que así nos proporciona una buena precisión al momento de abrir o cerrar la puerta de garaje.
- El programa elaborado para el motor paso a paso unipolar, va controlar el sentido de giro del motor con simples cambios de voltaje en sus líneas de entrada, así como la inversión de las mismas para cambiar su sentido de giro.
- Para el desarrollo del software de los microcontroladores ATmega8 y ATmega164P, se utiliza un lenguaje de alto nivel como es el Bascom AVR y para la comunicación del módulo receptor con el computador utilizamos el Microsoft Visual Basic 6.0 que nos facilita el almacenamiento de acceso de los vehículos.
- Es necesario modular la información para que viaje por el espacio, para transmitir información vía radiofrecuencia.
- Es indispensable la transmisión de los datos hacerlo por tramas más no bit a bit.
- Se utilizó un cable USB, para comunicarse con el usuario por su fácil adquisición, manejo y por su flexibilidad para usarlo.

RECOMENDACIONES

- Los detectores infrarrojos fueron diseñados para instalarse en una casa que no posea características especiales ni dispositivos adicionales, además mediante ellos podemos identificar si el vehículo está ingresando o saliendo de la casa.
- Los transmisores y receptores son dispositivos que se los puede cambiar, es decir que si uno de ellos presenta algún daño es fácil sustituirlo.
- Es indispensable verificar el estado de la batería utilizada, ya que esto provoca la variación de transmisión de los datos enviadas por el control remoto.
- Tenemos una base de datos, la cual podemos: leerla, modificarla y revisarla siempre y cuando la persona autorizada sepa la clave de ingreso. Y además se puede decir que los datos de los usuarios están encriptados para mayor seguridad.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS EN TEXTOS

[1] CARLOS A. REYES. Aprenda rápidamente a programar Microcontroladores PIC.

PAGINAS DE INTERNET

[2] SISTEMAS DE CONTROL DE ACCESO.

http://es.wikipedia.org/wiki/Token_de_seguridad

<http://es.wikipedia.org/wiki/RFID>

http://es.wikipedia.org/wiki/ISO_7816

<http://es.wikipedia.org/wiki/Biometr%C3%ADa>

[3] MEDIOS DE TRANSMISIÓN.

http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_inal%C3%A1brica

<http://www.monografias.com/trabajos17/medios-de-transmision/medios-de-transmision.shtml>

<http://transmision.galeon.com/mediosguiados.html>

[4] RADIO FRECUENCIA.

<http://es.wikipedia.org/wiki/Radiofrecuencia>

<http://www.monografias.com/trabajos/redesconcep/redesconcep.shtml>

<http://www.redeya.com/electronica/tutoriales/radio/radio.htm>

http://es.wikipedia.org/wiki/Banda_ISM

http://es.wikipedia.org/wiki/IEEE_802.15.4

[5] MICROCONTROLADORES.

<http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

<http://www.mitecnologico.com/Main/ArquitecturaDelMicrocontrolador>

<http://trecedb.wordpress.com/2009/02/13/partes-del-microcontrolador/>

[6] MOTORES.

http://es.wikipedia.org/wiki/Motor_el%C3%A9ctrico

<http://www.x-robotics.com/motorizacion.htm>

<http://html.rincondelvago.com/motores-de-corriente-continua.html>

[7] SENSORES

<http://es.wikipedia.org/wiki/Sensor>

http://www.profesormolina.com.ar/tecnologia/sens_transduct/tipos.htm

[8] COMUNICACIÓN SERIAL.

<http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

<http://es.scribd.com/doc/17048589/Comunicacion-Usb>

<http://es.kioskea.net/contents/pc/usb.php3>

<http://www.apuntesdeelectronica.com/microcontroladores/comunicacion-usb-pic-18f4550.htm>

http://es.wikipedia.org/wiki/Universal_Serial_Bus

[9] MÓDULOS XBEE.

<http://www.decelectronics.com/html/XBEE/XBEE.htm>

[XBee-Datasheet.pdf](#)

[10] MICROCONTROLADORES AVR

[ATmega8-Datasheet.pdf](#)

[ATmega164P-Datasheet.pdf](#)

[11] MOTORES PASO A PASO

http://www.sebyc.com/crr/descargas/motores_pap.pdf

<http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>

[12] INTEGRADOS

http://es.wikipedia.org/wiki/Circuito_integrado_555

<http://www.national.com/mpf/LM/LM358.html>

[12] RELÉ

<http://es.wikipedia.org/wiki/Rel%C3%A9>

ANEXOS

ANEXO A

DATASHEET - XBEE MANUAL

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements
 DSSS (Direct Sequence Spread Spectrum)
 Each direct sequence channels has over 65,000 unique network addresses available
 Source/Destination Addressing
 Unicast & Broadcast Communications
 Point-to-point, point-to-multipoint
 and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 μ A

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 μ A

ADC and I/O line support

Analog-to-digital conversion, Digital I/O
 I/O Line Passing

Easy-to-Use

No configuration necessary for out-of box RF communications
 Free X-CTU Software (Testing and configuration software)
 AT and API Command Modes for configuring module parameters
 Extensive command set
 Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements.
 Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.



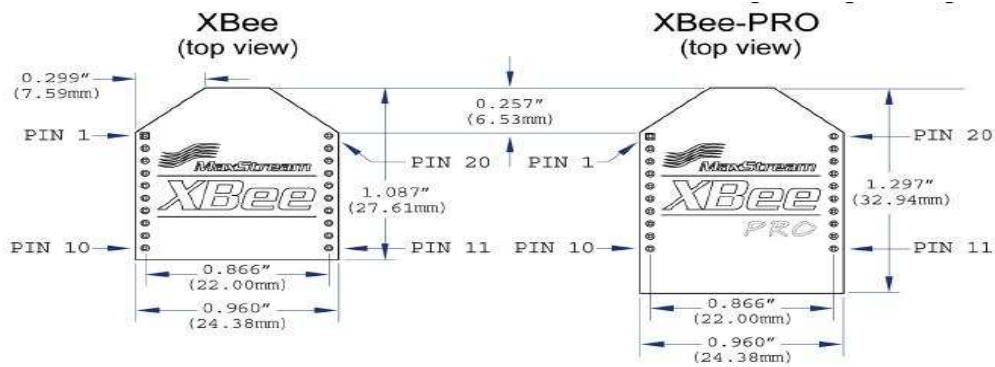
Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

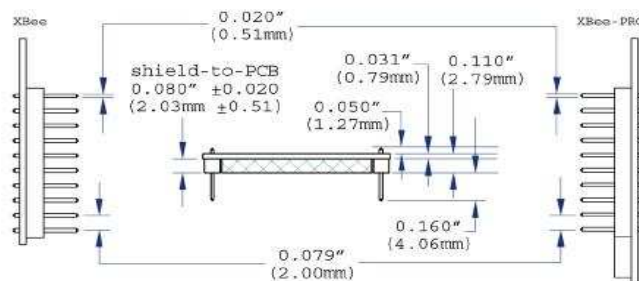
Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

* See Appendix A for region-specific certification requirements.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antennas" Knowledgebase Article located on Digi's Support Web site



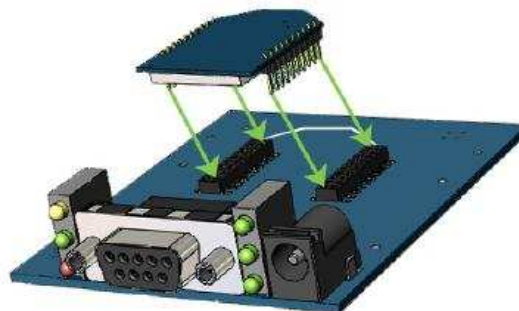
XBee & XBee-PRO
(side views)



Mounting Considerations

The XBee®/XBee-PRO® RF Module was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-02. XBee Module Mounting to an RS-232 Interface Board.

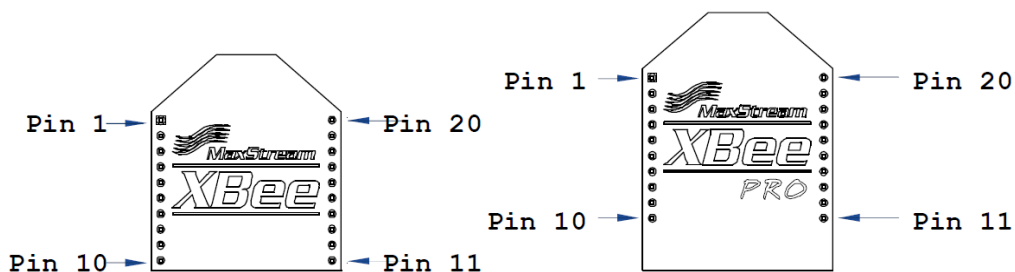


The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

Pin Signals

XBee®/XBee PRO RF Module Pin Numbers.



Pin Assignments for the XBee and XBee PRO Modules (Low asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	<u>RESET</u>	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	<u>CTS</u> / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	<u>RTS</u> / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k Ω pull-up resistor attached to RESET
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

Electrical Characteristics

DC Characteristics (VCC = 2.8 3.4 VDC).

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{IL}	Input Low Voltage	All Digital Inputs	-	-	0.35 * VCC	V
V _{IH}	Input High Voltage	All Digital Inputs	0.7 * VCC	-	-	V
V _{OL}	Output Low Voltage	I _{OL} = 2 mA, VCC >= 2.7 V	-	-	0.5	V
V _{OH}	Output High Voltage	I _{OH} = -2 mA, VCC >= 2.7 V	VCC - 0.5	-	-	V
I _{IN}	Input Leakage Current	V _{IN} = VCC or GND, all inputs, per pin	-	0.025	1	μA
I _{OZ}	High Impedance Leakage Current	V _{IN} = VCC or GND, all I/O High-Z, per pin	-	0.025	1	μA
TX	Transmit Current	VCC = 3.3 V	-	45 (XBee) 215, 140 (PRO, Int)	-	mA
RX	Receive Current	VCC = 3.3 V	-	50 (XBee) 55 (PRO)	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	μA

ADC CHARACTERISTICS (OPERATING).

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{REFH}	VREF - Analog-to-Digital converter reference range		2.08	-	V _{DDAD} *	V
I _{REF}	VREF - Reference Supply Current	Enabled	-	200	-	μA
		Disabled or Sleep Mode	-	< 0.01	0.02	μA
V _{INDC}	Analog Input Voltage ¹		V _{SSAD} - 0.3	-	V _{DDAD} + 0.3	V

ADC Timing/Performance Characteristics.

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
R _{AS}	Source Impedance at Input ²		-	-	10	kΩ
V _{AIN}	Analog Input Voltage ³		V _{REFL}		V _{REFH}	V
RES	Ideal Resolution (1 LSB) ⁴	2.08V ≤ V _{DDAD} ≤ 3.6V	2.031	-	3.516	mV
DNL	Differential Non-linearity ⁵		-	±0.5	±1.0	LSB
INL	Integral Non-linearity ⁶		-	±0.5	±1.0	LSB
E _{ZS}	Zero-scale Error ⁷		-	±0.4	±1.0	LSB
F _{FS}	Full-scale Error ⁸		-	±0.4	±1.0	LSB
E _{IL}	Input Leakage Error ⁹		-	±0.05	±5.0	LSB
E _{TU}	Total Unadjusted Error ¹⁰		-	±1.1	±2.5	LSB

ANEXO B

DATASHEET

ESPECIFICACIONES TÉCNICAS DEL MICROCONTROLADOR AVR ATMEGA8

Features

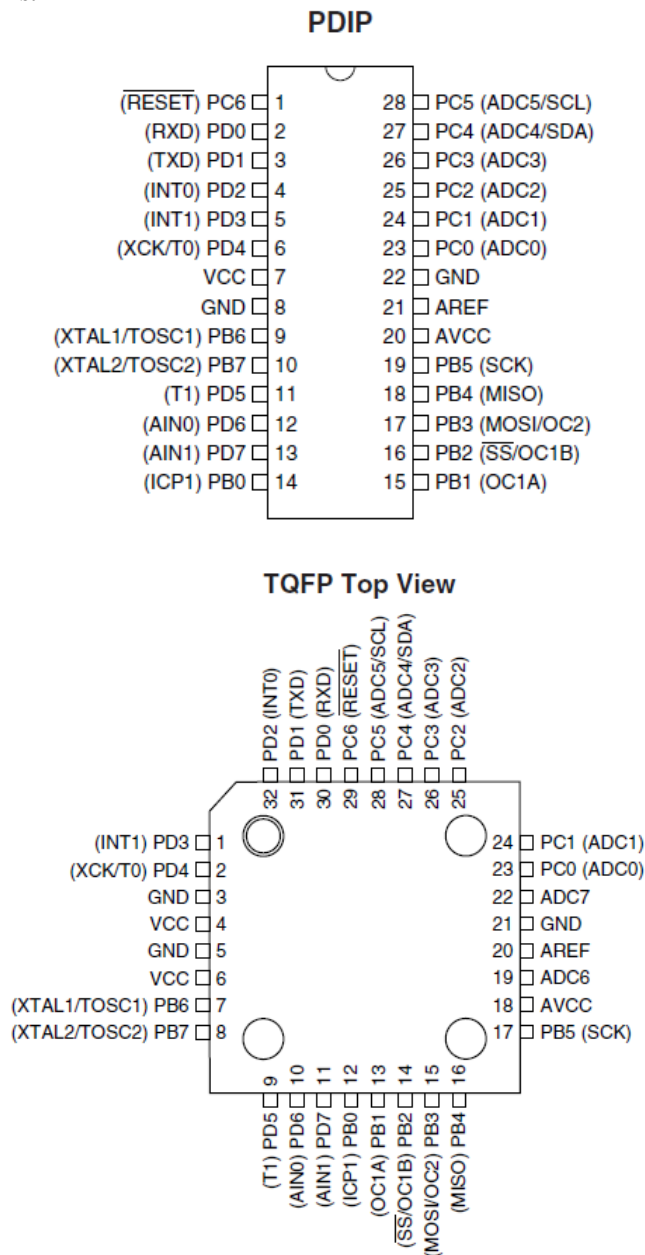
- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 8K Bytes of In-System Self-programmable Flash program memory
 - 512 Bytes EEPROM
 - 1K Byte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Three PWM Channels
 - 8-channel ADC in TQFP and QFN/MLF package
 - Eight Channels 10-bit Accuracy
 - 6-channel ADC in PDIP package
 - Six Channels 10-bit Accuracy
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V (ATmega8L)
 - 4.5 - 5.5V (ATmega8)
- Speed Grades
 - 0 - 8 MHz (ATmega8L)
 - 0 - 16 MHz (ATmega8)
- Power Consumption at 4 Mhz, 3V, 25°C
 - Active: 3.6 mA
 - Idle Mode: 1.0 mA
 - Power-down Mode: 0.5 µA



8-bit **AVR[®]**
with 8K Bytes
In-System
Programmable
Flash

ATmega8
ATmega8L

Pin Configurations.

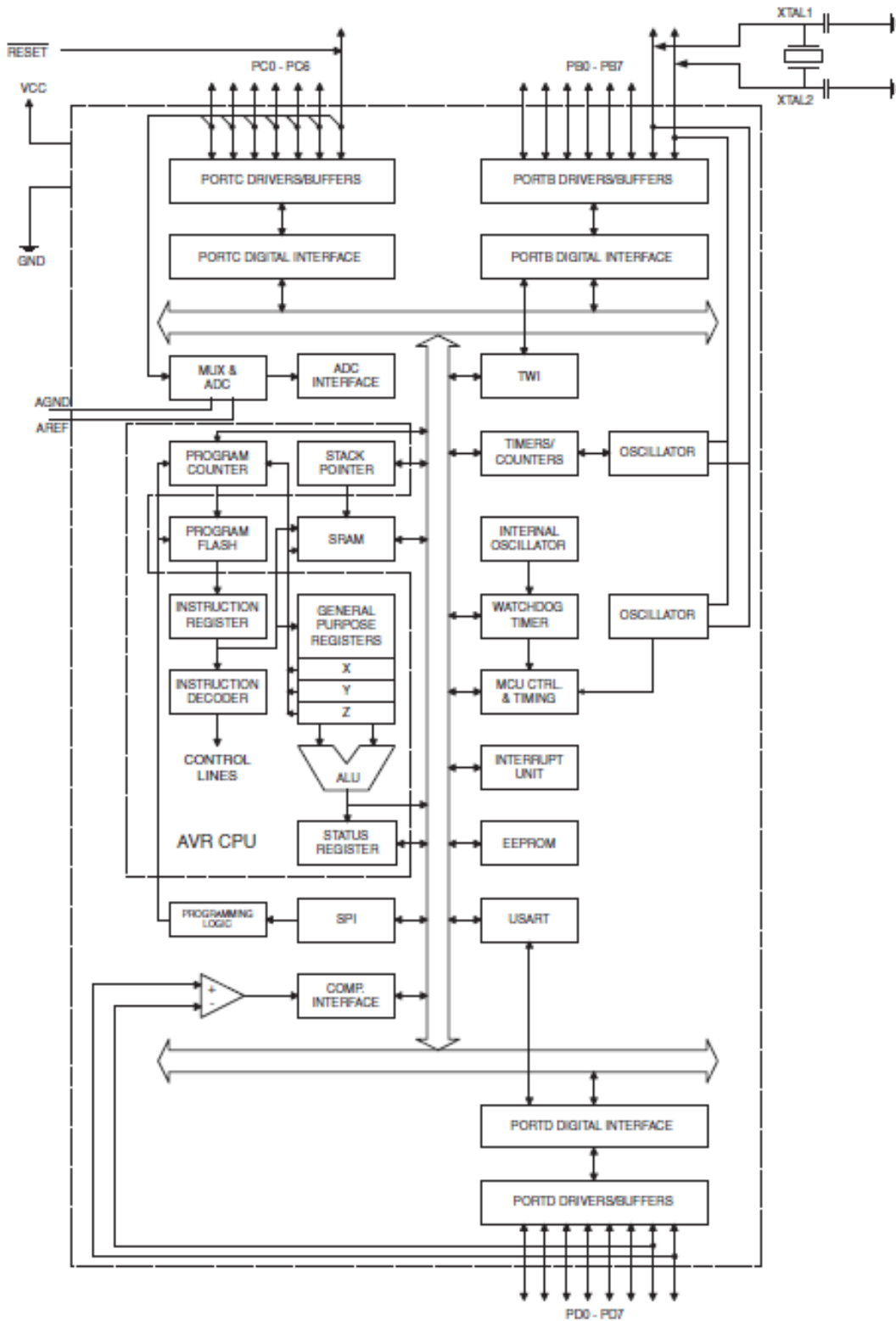


Overview

The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture.

By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

Block Diagram.





The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega8 AVR is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

Disclaimer.

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Pin Descriptions.

VCC	Digital supply voltage.
GND	Ground.
Port B (PB7..PB0) XTAL1/XTAL2/TOSC1/ TOSC2	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p> <p>Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.</p> <p>If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.</p> <p>The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 58 and "System Clock and Clock Options" on page 25.</p>
Port C (PC5..PC0)	<p>Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
PC6/RESET	<p>If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.</p> <p>If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a Reset.</p> <p>The various special features of Port C are elaborated on page 61.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8 as listed on page 63.</p>
RESET	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a reset.</p>
AV_{CC}	<p>AV_{CC} is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that Port C (5..4) use digital supply voltage, V_{CC}.</p>
AREF	<p>AREF is the analog reference pin for the A/D Converter.</p>
ADC7..6 (TQFP and QFN/MLF Package Only)	<p>In the TQFP and QFN/MLF package, ADC7..6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.</p>

ANEXO C

DATASHEET

ESPECIFICACIONES TÉCNICAS DEL MICROCONTROLADOR AVR ATMEGA164P



**8-bit AVR[®]
Microcontroller
with 16/32/64K
Bytes In-System
Programmable
Flash**

**ATmega164P/V
ATmega324P/V
ATmega644P/V**

Características:

- ❖ **Microcontrolador AVR de 8 bits de alto rendimiento y bajo consumo.**
- ❖ **Arquitectura Avanzada RISC**
 - 131 instrucciones. La mayoría de un solo ciclo de reloj de ejecución.
 - 32 registros de trabajo de 8 bits para propósito general.
 - Funcionamiento estático total.
 - Capacidad de procesamiento de unos 20 MIPS a 20 MHz.
 - Multiplicador por hardware de 2 ciclos
- ❖ **Memorias de programa y de datos no volátiles de alta duración**
 - 16/32/44 K bytes de FLASH auto programable en sistema
 - 512B/1K/2K bytes de EEPROM
 - 1/2/4K bytes de SRAM Interna
 - Ciclos de escritura/borrado: 10.000 en Flash / 100.000 en EEPROM
 - Retención de Datos: 20 años a 85°C / 100 años a 25°C
 - Sección opcional de código Boot con bits de bloqueo independientes. Programación en sistema del programa Boot que se encuentra dentro del mismo chip. Operación de lectura durante la escritura.
 - Bloqueo programable para la seguridad del software.
- ❖ **Interfase JTAG**
 - Capacidades de Boundary Scan de acuerdo con el estándar JTAG
 - Soporte Extendido Debug dentro del chip
 - Programación de FLASH, EEPROM, fusibles y bits de bloqueo a través de la interfase JTAG.

- ❖ **Características de los periféricos**
 - Dos Timer/Contadores de 8 bits con prescalamiento separado y modo comparación.
 - Un Timer/Contador de 16 bits con prescalamiento separado, modo comparación y modo de captura.
 - Contador en Tiempo Real con Oscilador separado
 - 6 Canales para PWM
 - ADC de 10 bits y 8 canales
 - Modo Diferencial con ganancia seleccionable a $\times 1$, $\times 10$ o $\times 200$.
 - Interfase serie de dos hilos con byte orientado.
 - Dos puertos Seriales USART Programables
 - Interfaz Serial SPI maestro-esclavo
 - Watchdog Timer programable con oscilador independiente, dentro del mismo chip.
 - Comparador Analógico dentro del mismo Chip
 - **Interrupt and Wake-up on Pin Change**

- ❖ **Características especiales del microcontrolador**
 - Power-on Reset (en el encendido) y detección de Brown-out (pérdida de polarización) programable.
 - Oscilador RC interno calibrado.
 - Fuentes de interrupción externas e internas.
 - 6 modos de descanso: Idle, Reducción de Ruido ADC, Power-save, Power-down, Standby y Standby extendido.

- ❖ **Encapsulados para Entradas/Salidas (E/S)**
 - 32 líneas de E/S programables.
 - PDIP de 40 pines, TQFP y QFN/MLF de 44 pines.

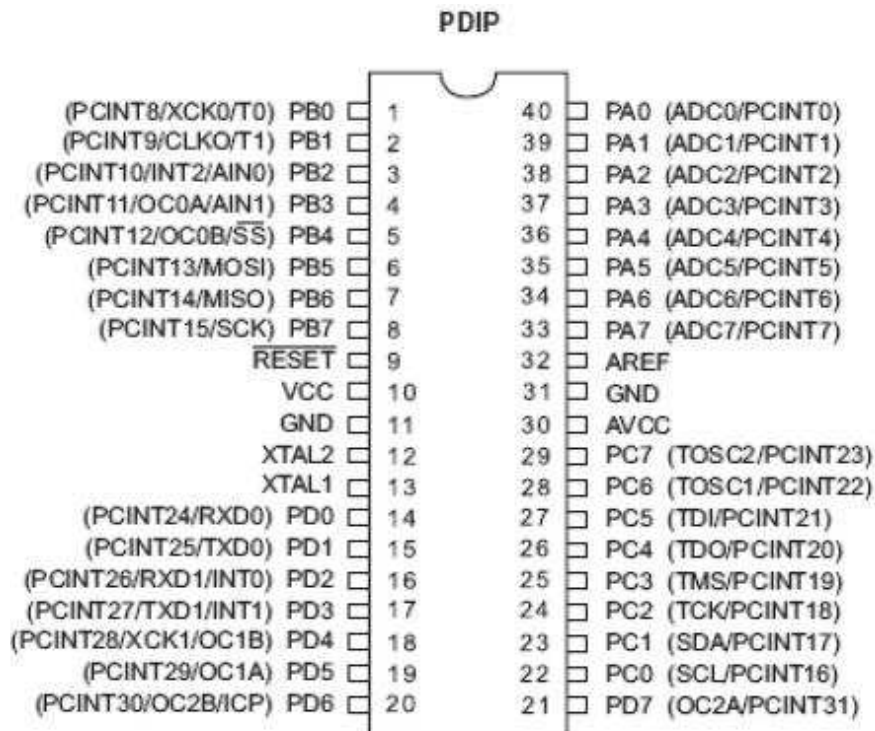
- ❖ **Voltajes de Operación**
 - 1.8 - 5.5V para el ATMEGA 164P/324P/644PV
 - 2.7 - 5.5V para el ATMEGA 164P/324P/644P

- ❖ **Velocidad de Funcionamiento**
 - ATMEGA 164P/324P/644PV: 0 - 4MHz @ 1.8 - 5.5V - 10MHz @ 2.7 - 5.5V
 - ATMEGA 164P/324P/644P: 0 - 10MHz @ 2.7 - 5.5V - 20MHz @ 4.5 - 5.5V

- ❖ **Consumo de energía a 1MHz, 1.8V, 25°C para el ATMEGA 164P/324P/644P**
 - Activo: 0.4mA
 - Modo Power-down: 0.1uA
 - Modo Power-Save: 0.6uA (Incluye RTC de 32 Khz)

1. CONFIGURACIÓN DE PINES

Figura 1-1. Pines de Salida del ATmega164P/324P/644P

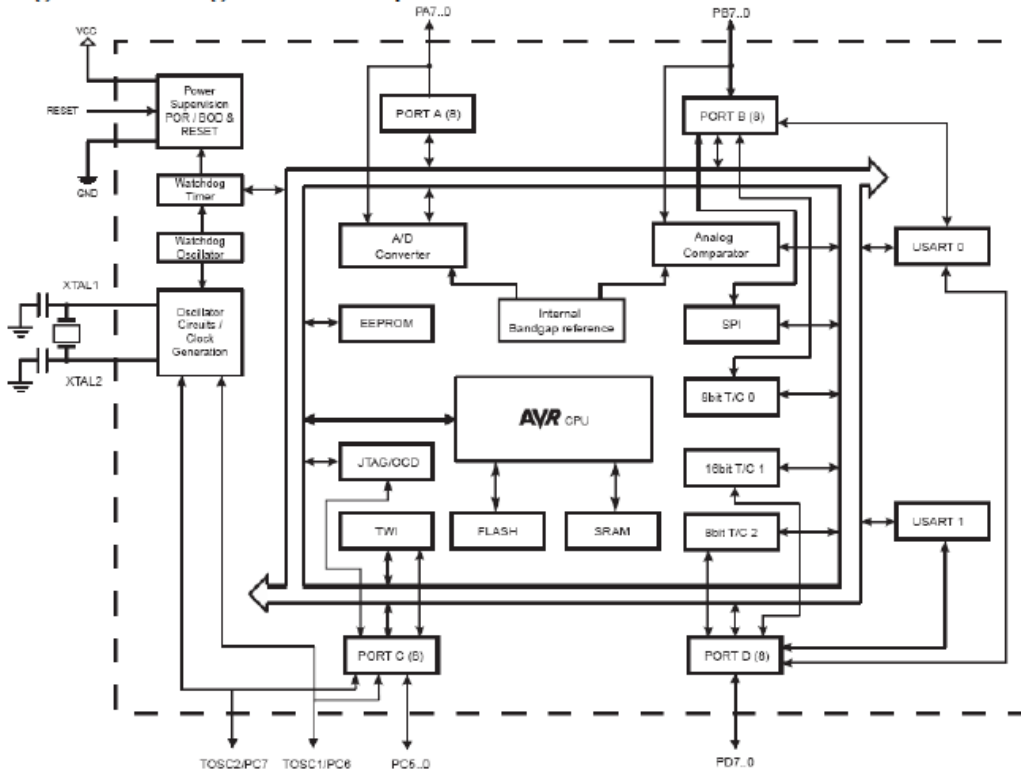


2. Revisión Global

El ATmega164P/324P/644P es un microcontrolador CMOS de 8 bits de bajo consumo basado en la arquitectura RISC mejorada. Sus instrucciones se ejecutan en un ciclo de máquina, el ATmega164P / 324P / 644P consigue transferencia de información alrededor de 1 MIPS por MHz admitido por el sistema, permitiendo al diseñador del sistema optimizar el consumo de energía versus la velocidad de procesamiento.

2.1 Diagrama de Bloque

Figura 2 -1. Diagrama de Bloques



El core (núcleo) AVR combina un conjunto de instrucciones RISC con 32 registros para uso de propósito general. Todos los 32 registros están directamente relacionados con la Unidad Aritmética Lógica (ALU), admitiendo dos registros independientes al ejecutarse una instrucción en un ciclo de máquina. El resultado de esta arquitectura es más eficiente, se consigue un caudal de flujo y transferencia hasta diez veces más rápido que microcontroladores CISC convencionales.

El ATmega164P / 324P / 644P provee las siguientes características: 16/32 / 64K bytes en el sistema de Flash Programable con capacidad de lectura y escritura de 512B/1K/2K bytes en la EEPROM, 1/2/4K bytes en la SRAM, 32 pines de E/S para propósito general, 32 registros de propósito general, Contador en Tiempo real (RTC), tres Timer/Contadores flexibles con modo de Comparación y PWM, 2 USARTs, un byte orientado a la Interfaz Serial de 2 hilos, 8 canales ADC de 10 bits con opción de entrada Diferencial con ganancia programable, Watchdog Timer programable con oscilador interno, un Puerto serial SPI, Interfase de prueba JTAG, también usado para acceder al sistema On-chip Debug y seis modos de programación seleccionable para ahorro de energía. El modo Idle detiene al CPU mientras permite a la SRAM, Timer/Contador, Puerto SPI y al sistema de interrupciones continuar funcionando.

El ATmega164P / 324P / 644P AVR es soportado con un juego completo de programas y herramientas de desarrollo del sistema incluyendo: compiladores de C, ensambladores de macro, depurador / simuladores de programa, emuladores de circuitos y equipos de evaluación.

2.2 Comparación entre el ATmega164P, ATmega324P and ATmega644P

Table 2-1. Differences between ATmega164P and ATmega644P

Device	Flash	EEPROM	RAM
ATmega164P	16 Kbyte	512 Bytes	1 Kbyte
ATmega324P	32 Kbyte	1 Kbyte	2 Kbyte
ATmega644P	64 Kbyte	2 Kbyte	4 Kbyte

2.3 Descripción de Pines

2.3.1 VCC

Alimentación de Voltaje Digital

2.3.2 GND

Tierra

2.3.3 Puerto A (PA7:PA0)

El puerto A sirve como entradas analógicas para el conversor Análogo Digital. El puerto A también sirve como un puerto bidireccional de 8 bits con resistencias internas de pull up (seleccionables para cada bit). Los buffers de salida del puerto A tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puerto A están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no este corriendo. El puerto A también sirve para varias funciones especiales del ATmega164P/324P/644P como la Conversión Análogo Digital.

2.3.4 Port B (PB7:PB0)

El puerto B es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up.

Las salidas de los buffers del puerto B tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puesto B están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo. El puerto B también sirve para varias funciones especiales del ATmega164P/324P/644P como se menciona en las páginas iniciales.

2.3.5 Port C (PC7:PC0)

El puerto C es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up (seleccionadas por cada bit). Las salidas de los buffers del puerto C tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puerto C están en tri-estado cuando las condiciones de reset están activadas siempre y cuando el reloj no este corriendo. El puerto C también sirve para las funciones de Interfaz del JTAG, con funciones especiales del ATmega164P/324P/644P como se menciona en las páginas iniciales.

2.3.6 Port D (PD7:PD0)

El Puerto D es un puerto bidireccional de entradas y salidas con resistencias internas de pull up (seleccionadas por cada bit). Las salidas de los buffers del puerto D tienen características simétricas controladas con sumideros de fuentes de alta capacidad.

Los pines del Puerto D están en tri-estado cuando llega una condición de reset activa, siempre y cuando el reloj no esté corriendo.

El puerto D también sirve para varias funciones especiales del ATmega164P/324P/644P como se menciona en las páginas iniciales.

2.3.7 $\overline{\text{RESET}}$

Entrada del Reset. Un pulso de nivel bajo en este pin por períodos de pulso mínimo genera un reset, siempre y cuando el reloj no esté corriendo.

La longitud del pulso mínimo está especificada en las Características y Sistemas de Reset (Páginas 331 del Data Sheet). Pulsos cortos no son garantizados para generar un reset.

2.3.8 XTAL1

Entrada para el amplificador del oscilador invertido y entrada para el circuito de operación del reloj interno.

2.3.9 XTAL2

Salida del Oscilador amplificador de salida.

2.3.10 AVCC

AVCC es la alimentación de voltaje para el pin del Puerto F y el Conversor Análogo a Digital. Este debe ser conectado externamente a VCC, siempre y cuando el ADC no sea usado. Si el ADC es usado, este deberá ser conectado a VCC a través de un filtro paso bajo.

2.3.11 AREF

Está es la referencia para el pin de la conversión Análoga a Digital.

3. Recursos

Un set comprensible de herramientas, notas de aplicación y datasheet se puede descargar desde <http://www.atmel.com/avr>.

4. Acerca de los Ejemplos de Código

Este documento contiene unos simples ejemplos de código que brevemente muestran como usar varias partes del dispositivo. Sea consiente que no todos los vendedores de compiladores en C incluyen la definición de bits en el archivo de cabecera y el manejo de interrupciones en el compilador C es dependiente. Por favor confirme con la documentación del compilador C para más detalles.

Los ejemplos de código asumen que la parte de archivo de cabecera esta incluido antes de la compilación. Para las localidades de los registros de E/S en el mapa extendido de E/S, las instrucciones "IN", "OUT", "SBIS", "SBIC", "CBI", y "SBI" serían reemplazadas con instrucciones de acceso extendido E/S. Típicamente "LDS" y "STS" combinadas con "SBR", "SBRC", "SBR", y "CBR".

5. Retención de Datos

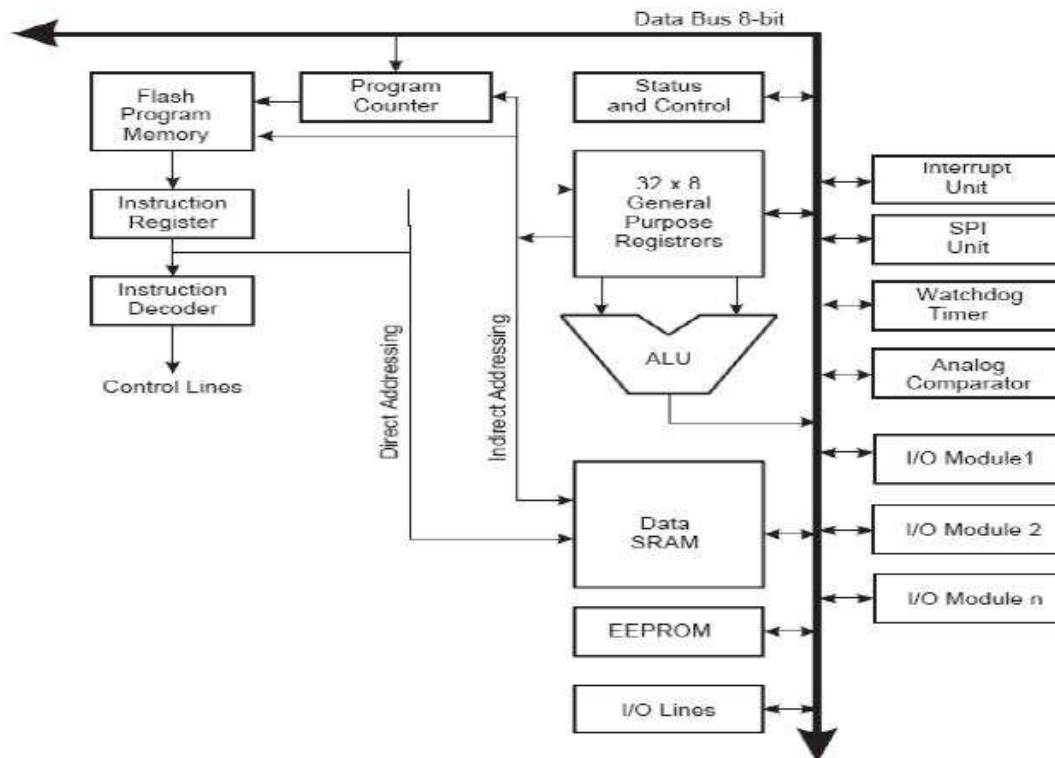
La fiabilidad de la calificación de resultados muestra que la velocidad de falla de un proyecto es mucho menor que 1 PPM en 20 años a 85°C ó 100 años a 25°C.

6. AVR CPU Core

6.1 VISIÓN GENERAL

Esta sección discute la arquitectura general del AVR. La principal función del AVR es asegurar la correcta ejecución del programa.

La CPU debe ser capaz de acceder a la memoria, llevar a cabo cálculos, control de periféricos y atención de interrupciones.



Para maximizar el rendimiento y el paralelismo, el AVR usa una arquitectura de Hardware con separador de memorias y buses para programa y datos. Las instrucciones en la memoria de programa son ejecutadas con un simple nivel de colas. Mientras una instrucción es ejecutada, la siguiente instrucción es ejecutada desde la memoria de programa. Este concepto permite que las instrucciones sean ejecutadas en cada ciclo de máquina. La memoria de programa está en la memoria Flash re-programable.

El Archivo del Registro (Register File) de rápido acceso contiene 32 registros de propósito general de 8 bits trabajando en un simple ciclo de reloj.

Esto permite una operación de ciclo simple en la Unidad Aritmética lógica. En una operación típica de la ALU, dos operandos están fuera del Archivo de Registro, la operación es ejecutada, y el resultado es guardado en el Archivo de Registro en un ciclo de máquina.

Seis de los 32 registros pueden ser usados como tres registros punteros de 16 bits de dirección, para direccionar los Datos y permitir los cálculos de direcciones diferentes.

Uno de estos tres punteros puede ser usado como un puntero de direcciones para tablas en la memoria de programa de la Flash. Estos registros de función adicionales son el X, Y y Z de 16 bits, descritos después en esta sección. La ALU soporta operaciones lógicas y aritméticas entre registros o entre constantes y registros.

Simple operaciones de registros pueden ser ejecutadas en la ALU. Después de una operación aritmética, el registro de estado es actualizado para reflejar información acerca de los resultados de la operación. El flujo del programa es provisto por un salto condicional e incondicional y llamado de interrupciones, capaz de direccionar espacios de direcciones completamente.

La mayoría de instrucciones del AVR tienen un formato simple de una palabra de 16 bits. Cada dirección de memoria contiene instrucciones de 16 o 32 bits. El espacio de memoria en la flash está dividido en dos secciones, la sección Baja del programa y la sección de aplicación de programa.

Ambas secciones están dedicadas para bloqueo de escritura y protección de lectura/escritura.

La instrucción SMP que se escribe en la Sección de la memoria Flash debe residir en la sección Baja del programa.

Durante los llamados de interrupción y subrutinas, la dirección de regreso del Contador de Programa (CP) es almacenado en la pila (stack).

La pila (stack) está localizada efectivamente en la SRAM (RAM estática) de datos y consecuentemente el tamaño de la pila está limitado solo por el tamaño total de la SRAM (RAM estática) y su uso.

Todos los programas a usarse deben inicializar en el SP (Stack Pointer) en la rutina del Reset (antes de que sea ejecutada una interrupción o una subrutina).

El puntero de pila (Stack pointer SP) es la lectura/escritura accesible en el espacio de E/S. La RAM estática de datos puede ser fácilmente penetrada a través de los cinco diferentes modos de direccionamiento soportados en la arquitectura de AVR.

El espacio de memoria en la arquitectura de los mapas de memoria de los AVR son todos lineales y regulares. Un módulo de interrupción flexible tiene sus registros de control en los espacios de E/S con una Habilidad de Interrupción Global en el Registro de Estado.

Todas las interrupciones tienen separado un vector de interrupciones en la tabla del vector de interrupciones. Las interrupciones tienen prioridad de conformidad con su vector de interrupciones. La dirección más baja del vector de interrupciones tiene alta prioridad.

El espacio de memoria de E/S contiene 64 direcciones para las funciones periféricas de la CPU, el Registro de Control SPI y otras funciones de Entrada y Salida. La memoria de Entrada y Salida puede ser accedida directamente o como localidades de espacio de datos siguiendo estos Archivos de Registro: 0x20 - 0x5F. En suma, el ATmega164P/324P/644P tiene espacios extendidos de entrada y salida desde la dirección 0x60 - 0xFF en la SRAM donde solo las instrucciones ST/STS/STD y LD/LDS/LDD pueden ser usadas.

ANEXO D
PROGRAMAS REALIZADOS

PROGRAMA DEL MÓDULO DE CONTROL REMOTO

```

$regfile = "m8def.dat"           'Micro a Utilizar ATMEGA8
$crystal = 8000000              'Frecuencia del cristal 8Mhz
$baud = 9600

Dim Tecla As Byte
Dim Keyread As Byte
Dim I As Byte
Dim X As Byte
Dim Y As Word
Dim Cuenta As Byte
Dim Clave(10) As Byte
Dim Flag1 As Bit
Dim Claveenviar As String * 15
Dim Temporal As String * 1

Const Rebote = 100
***** INTERRUPCION de serial *****
Dim Texto2 As String * 20

On Urxc Rec_isr1                'Define subrutina de Interrupcion
Enable Urxc
Enable Interrupts

***** Led Bicolor *****
Ddrc.0 = 1
Portc.0 = 0
Ddrc.1 = 1
Portc.1 = 0

Ledrojo Alias Portc.1
Ledverde Alias Portc.0

***** DEFINICIONES INICIALES PARA CONFIGURAR EL TECLADO *****
Ddrb.0 = 0
Portb.0 = 1
Ddrb.1 = 0
Portb.1 = 1
Ddrd.7 = 0
Portd.7 = 1
Ddrd.6 = 0
Portd.6 = 1

Ddrd.5 = 1
Portd.5 = 1
Ddrd.4 = 1
Portd.4 = 1
Ddrd.3 = 1
Portd.3 = 1
Ddrd.2 = 1

```

Portd.2 = 1

F1 Alias Portd.5 'nombres para los pines de las filas
 F2 Alias Portd.4
 F3 Alias Portd.3
 F4 Alias Portd.2

C1 Alias Pinb.1 'nombres para los pines de las columnas
 C2 Alias Pinb.0
 C3 Alias Pind.7
 C4 Alias Pind.6

***** Animacion Inicial *****

For X = 0 To 5

 Ledverde = 1

 Ledrojo = 1

 Waitms 200

 Ledverde = 0

 Ledrojo = 0

 Waitms 200

Next X

Do

 Flag1 = 1

 Cuenta = 1

 While Flag1 = 1

 Tecla = 255

 Gosub Barrido

 Gosub Rebote1

 Select Case Tecla

 If Cuenta < 5 And Tecla < 10 Then

 Clave(cuenta) = Tecla

 Cuenta = Cuenta + 1

 Gosub Toggle_ledrojo

 End If

 Case 10

 Gosub Toggle_tomate

 Print "A000"

 Gosub Toggle_tomate

 Case 11

 Gosub Toggle_tomate

 Print "B000"

 Gosub Toggle_tomate

 Case 15

 If Cuenta = 5 Then

 Gosub Toggle_tomate

 Claveenviar = ""

 Claveenviar = Str(clave(1)) + Str(clave(2)) + Str(clave(3)) + Str(clave(4))

 Print Claveenviar

 Gosub Toggle_tomate

 Flag1 = 0

 Else

```

        Gosub Toggle_ledrojo
        Flag1 = 0
        Gosub Toggle_ledverde
    End If
Case 14
    Gosub Toggle_ledverde
    Cuenta = 1          'Borra dato mal ingresado
End Select
Wend
Loop

***** Subrutina de Barrido de Teclas *****
Barrido:
    F1 = 0
    If C1 = 0 Then : Tecla = 1 : Return : End If
    If C2 = 0 Then : Tecla = 2 : Return : End If
    If C3 = 0 Then : Tecla = 3 : Return : End If
    If C4 = 0 Then : Tecla = 10 : Return : End If
    F1 = 1 : F2 = 0
    If C1 = 0 Then : Tecla = 4 : Return : End If
    If C2 = 0 Then : Tecla = 5 : Return : End If
    If C3 = 0 Then : Tecla = 6 : Return : End If
    If C4 = 0 Then : Tecla = 11 : Return : End If
    F2 = 1 : F3 = 0
    If C1 = 0 Then : Tecla = 7 : Return : End If
    If C2 = 0 Then : Tecla = 8 : Return : End If
    If C3 = 0 Then : Tecla = 9 : Return : End If
    If C4 = 0 Then : Tecla = 12 : Return : End If
    F3 = 1 : F4 = 0
    If C1 = 0 Then : Tecla = 14 : Return : End If
    If C2 = 0 Then : Tecla = 0 : Return : End If
    If C3 = 0 Then : Tecla = 15 : Return : End If
    If C4 = 0 Then : Tecla = 13 : Return : End If
    F4 = 1
    Waitms 1
Return
***** Subrutina de Rebote de Teclas *****
Rebote1:
    While C1 = 0 : Wend
    While C2 = 0 : Wend
    While C3 = 0 : Wend
    While C4 = 0 : Wend
    Waitms 1
Return
*****
Toggle_ledverde:
    For I = 0 To 5
        Toggle Ledverde
        Waitms 25
    Next
    Reset Ledverde
    Reset Ledrojo

```

```
Return
*****
Toggle_ledrojo:
  For I = 0 To 5
    Toggle Ledrojo
    Waitms 25
  Next
  Reset Ledrojo
  Reset Ledverde
Return
*****
Toggle_tomate:
  For I = 0 To 5
    Toggle Ledverde
    Toggle Ledrojo
    Waitms 25
  Next
  Reset Ledverde
  Reset Ledrojo
Return
*****
Rec_isr1:
  Disable Urxc
  Input Texto2 Noecho
  Temporal = Mid(texto2 , 1 , 1)
  Select Case Temporal
    Case "R"
      Ledrojo = 1
      Ledverde = 0
    Case "V"
      Ledrojo = 0
      Ledverde = 1
    Case "T"
      Ledrojo = 1
      Ledverde = 1
    Case "X"
      Ledrojo = 0
      Ledverde = 0
  End Select
  Enable Urxc
Return
*****
End
```

PROGRAMA DEL MÓDULO RECEPTOR

```

$regfile = "m164Pdef.dat "
$crystal = 8000000
$baud = 9600                'frecuencia del cristal 8 Mhz
$baud1 = 9600              'frecuencia del cristal 8 Mhz

$lib "mcsbyte.lbx"

Config Com1 = Dummy , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Config Com2 = Dummy , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0

'Abre el segundo UART por hardware
Open "com2:" For Binary As #1

***** INTERRUPCION de serial *****
On Urxc Rec_isr            'Define subrutina de Interrupcion XBEE
Enable Urxc
On Urxc1 Rec_isr1         'Define subrutina de Interrupcion PC
Enable Urxc1

*****
'*****
'          LUZ              *
'*****

Ddrd.7 = 1
Portd.7 = 0
Luz Alias Portd.7

*****
'*****
'          LED1             *
'*****

Ddrc.7 = 1
Portc.7 = 0
Led1 Alias Portc.7

*****
'*****
'          IR1              *
'*****

Ddrc.5 = 0
Portc.5 = 1
Ir1 Alias Pinc.5

*****
'*****
'          IR2              *
'*****

Ddrc.4 = 0
Portc.4 = 1
Ir2 Alias Pinc.4

*****
'*****
'          BOTON            *
'*****

```

Ddrc.6 = 0
 Portc.6 = 1
 Boton Alias Pinc.6

* FIN DE CARRERA 1 *

Ddrc.6 = 0
 Portd.6 = 1
 Fc1 Alias Pind.6

* FIN DE CARRERA 2 *

Ddrc.5 = 0
 Portd.5 = 1
 Fc2 Alias Pind.5

* MOTOR PAP *

Ddrc.0 = 1
 Portc.0 = 0
 B1 Alias Portc.0
 Ddrc.1 = 1
 Portc.1 = 0
 B2 Alias Portc.1
 Ddrc.2 = 1
 Portc.2 = 0
 B3 Alias Portc.2
 Ddrc.3 = 1
 Portc.3 = 0
 B4 Alias Portc.3

Dim Flag1 As Bit 'variable tipo bit

Dim Flag2 As Bit 'variable tipo bit

Dim Flag3 As Bit 'variable tipo bit

Dim Flag4 As Bit 'variable tipo bit

Dim Flag5 As Bit 'variable tipo bit

Dim Flag6 As Bit 'variable tipo bit

Dim Clave1(5) As Byte

Dim B As Byte

Dim I As Byte

Dim J As Byte

Dim X As Word

Dim Y As Word

Dim Z As Word

Dim Temporal As String * 1

Dim Temporal1 As String * 10

Dim Serial1 As String * 20

Dim Serial2 As String * 20

```

Const Delay1 = 80
Const Retardo = 20

Waitms 500
Cursor Off
Cls
Flag1 = 0
Flag2 = 0
Flag3 = 0
Flag4 = 0
Flag5 = 0
Flag6 = 0
Enable Interrupts
'Cierra la puerta si esta abierta
Gosub Cerrarpuertadondesea

```

```

'-----
Do
  If Flag2 = 1 Then
    Flag2 = 0
    Gosub Abrirpuerta
  If Flag4 = 0 Then
    While Ir1 = 0 And Ir2 = 0
      Gosub Toggle_led1
    Wend

    If Ir1 = 1 Then
      While Ir1 = 0
        Gosub Toggle_led1
      Wend
      While Ir1 = 1
        Gosub Toggle_led1
      Wend
      While Ir2 = 0
        Gosub Toggle_led1
      Wend
      While Ir2 = 1
        Gosub Toggle_led1
      Wend
      For I = 1 To 50
        Gosub Toggle_led1
        Gosub Toggle_led1
      Next I
      Gosub Cerrarpuerta
      Print #1 , "E000";
      Luz = 0
    End If
    If Ir2 = 1 Then
      While Ir2 = 0
        Gosub Toggle_led1
      Wend

```

```

While Ir2 = 1
  Gosub Toggle_led1
Wend
While Ir1 = 0
  Gosub Toggle_led1
Wend
While Ir1 = 1
  Gosub Toggle_led1
Wend
For I = 1 To 50
  Gosub Toggle_led1
  Gosub Toggle_led1
Next I
Gosub Cerrarpuerta
Print #1 , "S000";
Luz = 0
End If
Else
  Gosub Cerrarpuertadondesea
End If
End If

If Boton = 0 Then
  If Flag6 = 1 Then
    Luz = 1
  End If
  Gosub Toggle_led1
  Gosub Toggle_led1
  Gosub Toggle_led1
  While Boton = 0
    Gosub Toggle_led1
  Wend
  Gosub Toggle_led1
  Gosub Toggle_led1
  Gosub Toggle_led1
  Toggle Flag3
  If Flag3 = 1 Then
    Gosub Abrirpuerta
  Else
    Gosub Cerrarpuerta
    Luz = 0
  End If
End If
Gosub Toggle_led1
Loop
*****

Toggle_led1:
  Toggle Led1
  Waitms 20
Return
*****

Cerrarpuertadondesea:

```



```

While Fc1 = 1
  While Fc2 = 1
    B1 = 1
    Gosub Toggle_led1
    B1 = 0 : B2 = 1
    Gosub Toggle_led1
    B2 = 0 : B3 = 1
    Gosub Toggle_led1
    B3 = 0 : B4 = 1
    Gosub Toggle_led1
    B4 = 0
  Wend
  Gosub Toggle_led1
  Gosub Toggle_led1
  Gosub Toggle_led1
  While Fc1 = 1
    B4 = 1
    Gosub Toggle_led1
    B4 = 0 : B3 = 1
    Gosub Toggle_led1
    B3 = 0 : B2 = 1
    Gosub Toggle_led1
    B2 = 0 : B1 = 1
    Gosub Toggle_led1
    B1 = 0
  Wend
Wend
Return
*****

Abrirpuerta:
  While Fc2 = 1 And Flag4 = 0
    B1 = 1
    Gosub Toggle_led1
    B1 = 0 : B2 = 1
    Gosub Toggle_led1
    B2 = 0 : B3 = 1
    Gosub Toggle_led1
    B3 = 0 : B4 = 1
    Gosub Toggle_led1
    B4 = 0
  Wend
  Gosub Toggle_led1
Return
*****

Cerrarpuerta:
  While Fc1 = 1 And Flag4 = 0
    B4 = 1
    Gosub Toggle_led1
    B4 = 0 : B3 = 1
    Gosub Toggle_led1
    B3 = 0 : B2 = 1
    Gosub Toggle_led1

```

```

    B2 = 0 : B1 = 1
    Gosub Toggle_led1
    B1 = 0
Wend
    Gosub Toggle_led1
Return
*****
Rec_isr:                                'INTERRUPCION DEL XBEE
    Disable Interrupts
    Disable Urx
    Input Serial1 Noecho
    Temporal1 = Mid(serial1 , 1 , 4)
    Select Case Temporal1
        Case "A000"
            Flag4 = 1
            Print #1 , "A111";
        Case "B000"
            Flag4 = 0
            Print #1 , "B111";
        Case Else
            Print #1 , Temporal1;
    End Select
    Do
        B = Inkey()
    Loop Until B = 0
    Enable Interrupts
    Enable Urx
Return
*****
Rec_isr1:                                'INTERRUPCION DEL PC
    Disable Interrupts
    Disable Urx1
    Input #1 , Serial2 Noecho
    Temporal = Mid(serial2 , 1 , 1)
    Select Case Temporal
        Case "I"
            Flag2 = 1
            Print "V"
        Case "J"
            Luz = 1
            Flag6 = 1
        Case "K"
            Luz = 0
            Flag6 = 0
        Case "L"
            Print "R"
    End Select
    Do
        B = Inkey(#1)
    Loop Until B = 0
    Enable Interrupts
    Enable Urx1

```

Return

End

PROGRAMA EN EL VISUAL BASIC 6.0



```

Dim dato1 As String
Dim Cl As String
Dim DatoAs Integer
Dim NumeroUsuariosAs Integer
Dim NumeroAs Integer
Dim X As Integer
Dim i As Integer
Dim UltimoUsuarioAs Integer
Dim tiempo As Date
Dim fecha As Date
Dim tiempo1 As String
Dim fecha1 As String
Dim clave(100) As String
Dim Usuario(100) As String
Dim flag1 As Byte
Private Sub Command1_Click()
Dim OpenFlag
OpenFlag = form1.MSComm1.PortOpen
If OpenFlag Then
Else
form1.MSComm1.PortOpen = Not form1.MSComm1.PortOpen
End If
Command1.Enabled = False
'Open "datos1.txt" For Append As #6
End Sub
Private Sub CmdSalir_Click()
If form1.MSComm1.PortOpen Then form1.MSComm1.PortOpen = False
Close #6
End
End Sub
Private Sub Form_Load()
Dim c As Integer
form1.lblportico.Caption = "Com" & form1.MSComm1.CommPort
form1.lblvelocidad.Caption = Left$(form1.MSComm1.Settings, Len(form1.MSComm1.Settings) - 6)
Open App.Path & "\data1.dat" For Input As #2
Line Input #2, h
Text1.Text = h
Close
Dim TextoriginalAs String
Textoriginal = EncryptString("1", Text1, DECRYPT)
Open App.Path & "\data2.dat" For Output As #3
Print #3, Textoriginal
Close #3
Numero = 1
Open App.Path & "\data2.dat" For Input As #4
Do WhileNotEOF(4) ' Repite el bucle hasta el final del archivo.
Line Input #4, LíneaTexto ' Lee el carácter en la variable.
clave(Numero) = Mid(LíneaTexto, 1, 4)
Usuario(Numero) = Mid(LíneaTexto, 5)
Numero = Numero + 1

```

```

Loop
' Text2.Text = ""
' For i = 1 ToNumero - 1
'   Text2.Text = Text2.Text + clave(i) + " " + Usuario(i) + vbCrLf
' Next i
Close #4 ' Cierra el archivo.
Kill "data2.dat"
NumeroUsuarios = Numero - 1
Open "datos1.txt" For Append As #6
End Sub
Private Sub MnuAcerca_Click()
frmAbout.ShowvbModal
End Sub
Private Sub MnuClavesUsuarios_Click()
Form3.Show vbModal
Open App.Path & "\data1.dat" For Input As #2
Line Input #2, h
Text1.Text = h
Close
Dim TextoriginalAs String
Textoriginal = EncryptString("1", Text1, DECRYPT)
Open App.Path & "\data2.dat" For Output As #3
Print #3, Textoriginal
Close #3
Numero = 1
Open App.Path & "\data2.dat" For Input As #4
Do WhileNotEOF(4) ' Repite el bucle hasta el final del archivo.
Line Input #4, LíneaTexto ' Lee el carácter en la variable.
clave(Numero) = Mid(LíneaTexto, 1, 4)
Usuario(Numero) = Mid(LíneaTexto, 5)
Numero = Numero + 1
Loop
' Text2.Text = ""
' For i = 1 ToNumero - 1
'   Text2.Text = Text2.Text + clave(i) + " " + Usuario(i) + vbCrLf
' Next i
Close #4 ' Cierra el archivo.
Kill "data2.dat"
NumeroUsuarios = Numero - 1
Open "datos1.txt" For Append As #6
End Sub
Private Sub MnuPropiedades_Click()
Form2.Show vbModal
End Sub
Private Sub MnuSalir_Click()
CmdSalir_Click
End Sub
Private Sub MSComm1_OnComm()
Dim Cl As String
Select Case MSComm1.CommEvent
' Controlar cada evento o error escribiendo
' código en cada instrucción Case
' Errores
Case comBreak ' Se ha recibido una interrupción.
Case comEventFrame ' Error de trama
Case comEventOverrun ' Datos perdidos.
Case comEventRxOver ' Desbordamiento del búfer
' de recepción.
Case comEventRxParity ' Error de paridad.
Case comEventTxFull ' Búfer de transmisión lleno.
Case comEventDCB ' Error inesperado al recuperar DCB.

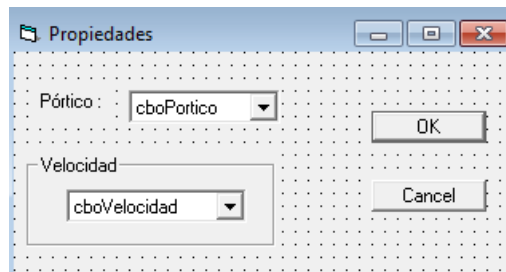
' Eventos
Case comEvCD ' Cambio en la línea CD.
Case comEvCTS ' Cambio en la línea CTS.
Case comEvDSR ' Cambio en la línea DSR.
Case comEvRing ' Cambio en el indicador de
' llamadas.
Case comEvReceive ' Recibido nº RThreshold de
' caracteres.
Cl = form1.MSComm1.Input
Label3.Caption = Cl
ci1 = Mid(Cl, 1, 4)
flag1 = 0

```

```

        Select Case ci1
        Case "E000"
        tiempo1 = Time
            fecha1 = Date
            Text2.Text = Text2.Text + Usuario(i) + " ENTRADA" + " " + tiempo1 + " " + fecha1 + vbCrLf
            cCadena = Usuario(UltimoUsuario) + " ENTRADA" + " " + tiempo1 + " " + fecha1
            Print #6, cCadena
        Case "S000"
        tiempo = Time
            fecha = Date
            Text2.Text = Text2.Text + Usuario(i) + " SALIDA " + " " + tiempo1 + " " + fecha1 + vbCrLf
            cCadena = Usuario(UltimoUsuario) + " SALIDA " + " " + tiempo1 + " " + fecha1
            Print #6, cCadena
        Case "A111"
        Case "B111"
        Case Else
            For i = 1 To NumeroUsuarios
            If clave(i) = ci1 Then
                form1.MSComm1.Output = "I" + Chr(13)
                Do
                    Ret = DoEvents()
                Loop Until form1.MSComm1.OutBufferCount = 0
            tiempo = Time
            If tiempo > "18:00:00" Or tiempo < "6:00:00" Then
                form1.MSComm1.Output = "J" + Chr(13)
                Do
                    Ret = DoEvents()
                Loop Until form1.MSComm1.OutBufferCount = 0
            Else
                form1.MSComm1.Output = "K" + Chr(13)
                Do
                    Ret = DoEvents()
                Loop Until form1.MSComm1.OutBufferCount = 0
            EndIf
            UltimoUsuario = i
            flag1 = 1
        ExitFor
            End If
        Next i
        If flag1 = 0 Then
            form1.MSComm1.Output = "L" + Chr(13)
            Do
                Ret = DoEvents()
            Loop Until form1.MSComm1.OutBufferCount = 0
        End If
    End Select
    Case comEvSend ' Hay un SThreshold
    ' caracteres en el búfer
    ' de transmisión.
    Case comEvEOF ' Se ha encontrado un carácter
    ' EOF en la entrada.
    EndSelect
EndSub

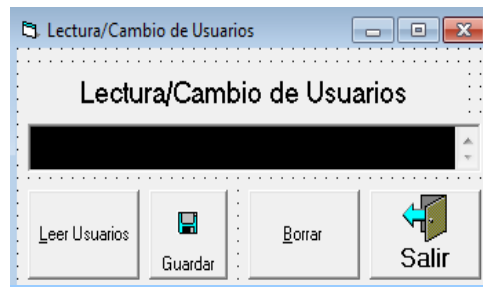
```



```

Private Sub cmdCancel_Click()
Unload Me
End Sub

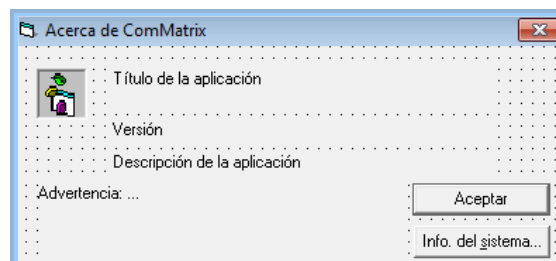
```



```

Dim y As Byte
Private Sub Command1_Click()
    Unload Me
End Sub
Private Sub Command2_Click()
    Text1.Text = ""
    Open App.Path & "\data1.dat" For Input As #2
    Line Input #2, h
    Text1.Text = h
    Close
    Dim Textoriginal As String
    Textoriginal = EncryptString("1", Text1, DECRYPT)
    Text1.Text = Textoriginal
End Sub
Private Sub Command3_Click()
    Text1.Text = ""
End Sub
Private Sub Command4_Click()
    Dim textocodificado As String
    textocodificado = EncryptString("1", Text1, ENCRYPT)
    Open App.Path & "\data1.dat" For Output As #2
    Print #2, textocodificado
    Close
End Sub

```



```

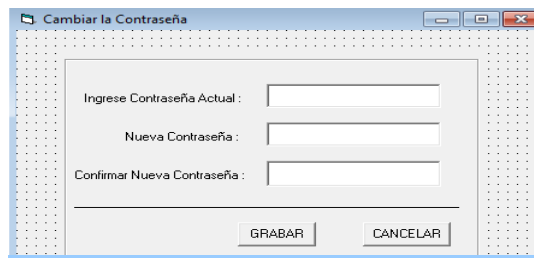
' Opciones de seguridad de clave del Registro...
Const KEY_ALL_ACCESS = &H2003F
' Tipos ROOT de claves del Registro...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1          ' cadena terminada en valor nulo Unicode
Const REG_DWORD = 4      ' número de 32 bits
ConstgREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
ConstgREGVALSYSINFOLOC = "MSINFO"
ConstgREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
ConstgREGVALSYSINFO = "PATH"
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByValhKey As Long, ByValpSubKey As String, ByValulOptions As Long, ByValsamDesired As Long, ByRefphkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByValhKey As Long, ByValpValueName As String, ByValpReserved As Long, ByReflpType As Long, ByValpData As String, ByReflpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByValhKey As Long) As Long
Private Sub Form_Load()
    lblVersion.Caption = "Versión "&App.Major& "." &App.Minor& "." &App.Revision
    lblTitle.Caption = App.Title
End Sub

```

```

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub
Private Sub cmdOK_Click()
    Unload Me
End Sub
Public Sub StartSysInfo()
    On Error GoToSysInfoErr
    Dim rc As Long
    DimSysInfoPath As String
    ' Intentar obtener el nombre y la ruta del programa en el Registro...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath) Then
    ' Intentar obtener sólo la ruta del programa en el Registro...
    ElseIfGetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
    ' Validar la existencia de versión conocida de 32 bits de archivo
    If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
        SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
    ' Error: no se encuentra el archivo...
    Else
        GoToSysInfoErr
    EndIf
    ' Error: no se encuentra la entrada del Registro...
    Else
        GoToSysInfoErr
    End If
    Call Shell(SysInfoPath, vbNormalFocus)
Exit Sub
SysInfoErr:
MsgBox "La información del sistema no está disponible en este momento", vbOKOnly
End Sub
Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRefKeyVal As String) As Boolean
    Dim i As Long
    Dimrc As Long
    DimhKey As Long
    DimhDepth As Long
    DimKeyValType As Long
    DimtmpVal As String
    DimKeyValSize As Long
    ' Contador de bucle
    ' Código de retorno
    ' Controlador a una clave de Registro abierta
    '
    ' Tipo de datos de una clave del Registro
    ' Almacenamiento temporal para un valor de clave del Registro
    ' Tamaño de variable de clave del Registro
    ' AbrirRegKeybajoKeyRoot {HKEY_LOCAL_MACHINE...}
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Abrir la clave del Registro
    If (rc <> ERROR_SUCCESS) Then GoToGetKeyError ' Controlar error...
    tmpVal = String$(1024, 0) ' Asignar espacio de variable
    KeyValSize = 1024 ' Marcar tamaño de variable
    ' Obtener valor de clave del Registro...
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value
    If (rc <> ERROR_SUCCESS) Then GoToGetKeyError ' Controlar errores
    tmpVal = VBA.Left(tmpVal, InStr(tmpVal, VBA.Chr(0)) - 1)
    ' Determinar el tipo de valor de clave para conversión...
    Select Case KeyValType
        Case REG_SZ ' Buscar tipos de datos...
            ' Tipo de datos String de clave del Registro
            KeyVal = tmpVal ' Copiar valor String
        Case REG_DWORD
            ' Tipo de datos Double Word de clave del Registro
            For i = Len(tmpVal) To 1 Step -1
                ' Convertircada bit
                KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Generar valor carácter a carácter
            Next
            KeyVal = Format$("&h" + KeyVal) ' Convertir Double Word a String
        EndSelect
    GetKeyValue = True ' Operación realizada correctamente
    rc = RegCloseKey(hKey) ' Cerrar clave del Registro
    ExitFunction ' Salir
    GetKeyError: ' Limpiar después de que se produzca un error...
    KeyVal = "" ' Establecer el valor de retonor a la cadena vacía
    GetKeyValue = False ' La operación no se ha realizado correctamente
    rc = RegCloseKey(hKey) ' Cerrar clave del Registro
EndFunction

```



```

Function Comprobar() As Boolean
    If TxtActual.Text = "" Then
        NCA = MsgBox("Debe ingresar la contraseña actual.", vbExclamation, "Cambiar Contraseña de Usuario.")
        TxtActual.SetFocus
        Comprobar = False
        Exit Function
    End If
    If TxtNueva.Text = "" Then
        NCN = MsgBox("Debe ingresar la nueva contraseña.", vbExclamation, "Cambiar Contraseña de Usuario.")
        TxtNueva.SetFocus
        Comprobar = False
        Exit Function
    End If
    If TxtConfirmar.Text = "" Then
        NCon = MsgBox("Debe confirmar la contraseña.", vbExclamation, "Cambiar Contraseña de Usuario.")
        TxtConfirmar.SetFocus
        Comprobar = False
        Exit Function
    End If
    If TxtActual.Text <> frmLogin.txtpasswd.Text Then
        NCA = MsgBox("Es incorrecta la contraseña actual.", vbExclamation, "Cambiar Contraseña de Usuario.")
        TxtActual.SetStart = 0
        TxtActual.SetLength = Len(TxtContraseñaA)
        TxtActual.SetFocus
        Comprobar = False
        Exit Function
    End If
    Comprobar = True
End Function
Private Sub CmdCancelar_Click()
    FrmCambiar.Hide
    Unload Me
End Sub
Private Sub CmdGrabar_Click()
    If Comprobar = True Then
        Preguntar = MsgBox("Confirma cambiar contraseña de usuario?", vbQuestion + vbYesNo, "Cambiar Contraseña de Usuario.")
        If Preguntar = 6 Then
            If TxtConfirmar.Text <> TxtNueva.Text Then
                OK = MsgBox("La contraseña Nueva y Confirmar son diferentes.", vbInformation, "Cambiar Contraseña de Usuario.")
            Else
                If Len(TxtNueva.Text) < 6 Or Len(TxtNueva.Text) > 15 Then
                    MsgBox "La contraseña debe tener un minimo de 6 caracteres y un máximo de 15", vbExclamation, "Atencion!"
                    TxtNueva.SetFocus
                    SendKeys "{home}+{end}"
                Else
                    Dim textocodificado As String
                    frmLogin.txtpasswd.Text = TxtNueva.Text
                    textocodificado = EncryptString("1", TxtNueva, ENCRYPT)
                    Open App.Path & "\data.dat" For Output As #1
                    Print #1, textocodificado
                    Close
                    OK = MsgBox("La contraseña se ha cambiado satisfactoriamente.", vbInformation, "Cambiar Contraseña de Usuario.")
                End If
            End If
        End If
    End Sub
End Sub

```




```

Private Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" (ByVal pBuffer As String, nSize As Long)
As Long
Public OK As Boolean
Private Sub CmdCambiarp_Click()
FrmCambiar.ShowvbModal
End Sub
Private Sub Form_Load()
Dim h As String
On Error GoTo a
Dim sBuffer As String
Dim lSize As Long
sBuffer = Space$(255)
lSize = Len(sBuffer)
Call GetUserName(sBuffer, lSize)
If lSize > 0 Then
txtUserName.Text = Left$(sBuffer, lSize)
Else
txtUserName.Text = vbNullString
End If
Open App.Path & "\data.dat" For Input As #1
Line Input #1, h
txtdpassw.Text = h
Close
Dim Textoriginal As String
Textoriginal = EncryptString("1", txtdpassw, DECRYPT)
txtdpassw.Text = Textoriginal
a:
If Err.Number = 76 Then
Mensaje = " No existe el archivo DATA.DAT " ' Define el mensaje.
Estilo = vbOKOnly + vbExclamation ' Define los botones.
Titulo = " Inicio de sesión " ' Define el título.
Respuesta = MsgBox(Mensaje, Estilo, Titulo)
End
End If
End Sub
Private Sub CmdSalir_Click()
OK = False
Me.Hide
End Sub
Private Sub cmdOK_Click()
'Pendiente: crear comprobación de contraseña
'comprobar si la contraseña es correcta
Static numintentos As Byte
If TxtPassword.Text = frmLogin.txtdpassw.Text Then
OK = True
Me.Hide
Else
MsgBox "La contraseña no es válida; vuelva a intentarlo", vbCritical, "Inicio de sesión"
TxtPassword.SetFocus
TxtPassword.SelStart = 0
TxtPassword.SelLength = Len(TxtPassword.Text)
numintentos = numintentos + 1
End If

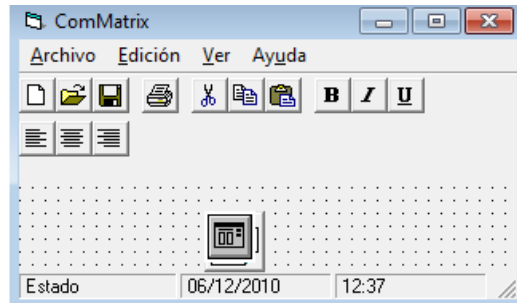
If numintentos > 2 Then
OK = False
Me.Hide
End If
End Sub
Private Sub TxtPassword_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then

```

```

    Call cmdOK_Click
EndIf
End Sub

```



```

Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByValhWnd&, ByValHelpFile$, ByValwCommand%,
dwData As Any)

```

```

Private Sub Form_Load()
Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Dim i As Integer

```

```

'close all sub forms
For i = Forms.Count - 1 To 1 Step -1
Unload Forms(i)
Next
If Me.WindowState<>vbMinimized Then
SaveSettingApp.Title, "Settings", "MainLeft", Me.Left
SaveSettingApp.Title, "Settings", "MainTop", Me.Top
SaveSettingApp.Title, "Settings", "MainWidth", Me.Width
SaveSettingApp.Title, "Settings", "MainHeight", Me.Height
End If
End Sub

```

```

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)
On Error Resume Next
Select Case Button.Key
Case "Nuevo"
'TareasPendientes: Agregar código de botón 'Nuevo'.
MsgBox "Agregar código de botón 'Nuevo'."
Case "Abrir"
mnuFileOpen_Click
Case "Guardar"
mnuFileSave_Click
Case "Imprimir"
mnuFilePrint_Click
Case "Cortar"
mnuEditCut_Click
Case "Copiar"
mnuEditCopy_Click
Case "Pegar"
mnuEditPaste_Click
Case "Negrita"
'TareasPendientes: Agregar código de botón 'Negrita'.
MsgBox "Agregar código de botón 'Negrita'."
Case "Cursiva"
'TareasPendientes: Agregar código de botón 'Cursiva'.
MsgBox "Agregar código de botón 'Cursiva'."
Case "Subrayado"
'TareasPendientes: Agregar código de botón 'Subrayado'.

```

```

MsgBox "Agregar código de botón 'Subrayado'."
    Case "Alinear a la izquierda"
        'TareasPendientes: Agregar código de botón 'Alinear a la izquierda'.
MsgBox "Agregar código de botón 'Alinear a la izquierda'."
    Case "Centrar"
        'TareasPendientes: Agregar código de botón 'Centrar'.
MsgBox "Agregar código de botón 'Centrar'."
    Case "Alinear a la derecha"
        'TareasPendientes: Agregar código de botón 'Alinear a la derecha'.
MsgBox "Agregar código de botón 'Alinear a la derecha'."
End Select
End Sub
Private Sub mnuHelpAbout_Click()
    frmAbout.ShowvbModal, Me
End Sub
Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer
    'si no hay archivo de ayuda para este proyecto, mostrar un mensaje al usuario
    'puede establecer el archivo de Ayuda para su aplicación en el cuadro
    'de diálogo Propiedades del proyecto
    If Len(App.HelpFile) = 0 Then
        MsgBox "No se puede mostrar el contenido de la Ayuda. No hay Ayuda asociada a este proyecto.", vbInformation,
            Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
        If Err Then
            MsgBoxErr.Description
        End If
    End If
End Sub
Private Sub mnuHelpContents_Click()
    Dim nRet As Integer
    'si no hay archivo de ayuda para este proyecto, mostrar un mensaje al usuario
    'puede establecer el archivo de Ayuda para la aplicación en el cuadro
    'de diálogo Propiedades del proyecto
    If Len(App.HelpFile) = 0 Then
        MsgBox "No se puede mostrar el contenido de la Ayuda. No hay Ayuda asociada a este proyecto.", vbInformation,
            Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBoxErr.Description
        End If
    End If
End Sub
Private Sub mnuViewWebBrowser_Click()
    'TareasPendientes: Agregar código 'mnuViewWebBrowser_Click'.
    MsgBox "Agregar código 'mnuViewWebBrowser_Click'."
End Sub
Private Sub mnuViewOptions_Click()
    frmOptions.ShowvbModal, Me
End Sub
Private Sub mnuViewRefresh_Click()
    'TareasPendientes: Agregar código 'mnuViewRefresh_Click'.
    MsgBox "Agregar código 'mnuViewRefresh_Click'."
End Sub
Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub
Private Sub mnuViewToolBar_Click()
    mnuViewToolBar.Checked = Not mnuViewToolBar.Checked
    tbToolBar.Visible = mnuViewToolBar.Checked
End Sub
Private Sub mnuEditPasteSpecial_Click()
    'TareasPendientes: Agregar código 'mnuEditPasteSpecial_Click'.
    MsgBox "Agregar código 'mnuEditPasteSpecial_Click'."
End Sub
Private Sub mnuEditPaste_Click()
    'TareasPendientes: Agregar código 'mnuEditPaste_Click'.
    MsgBox "Agregar código 'mnuEditPaste_Click'."
End Sub

```

```

Private Sub mnuEditCopy_Click()
'TareasPendientes: Agregar código 'mnuEditCopy_Click'.
MsgBox "Agregar código 'mnuEditCopy_Click'."
End Sub
Private Sub mnuEditCut_Click()
'TareasPendientes: Agregar código 'mnuEditCut_Click'.
MsgBox "Agregar código 'mnuEditCut_Click'."
End Sub
Private Sub mnuEditUndo_Click()
'TareasPendientes: Agregar código 'mnuEditUndo_Click'.
MsgBox "Agregar código 'mnuEditUndo_Click'."
End Sub
Private Sub mnuFileExit_Click()
'descargar el formulario
Unload Me
End Sub
Private Sub mnuFileSend_Click()
'TareasPendientes: Agregar código 'mnuFileSend_Click'.
MsgBox "Agregar código 'mnuFileSend_Click'."
End Sub
Private Sub mnuFilePrint_Click()
'TareasPendientes: Agregar código 'mnuFilePrint_Click'.
MsgBox "Agregar código 'mnuFilePrint_Click'."
End Sub
Private Sub mnuFilePrintPreview_Click()
'TareasPendientes: Agregar código 'mnuFilePrintPreview_Click'.
MsgBox "Agregar código 'mnuFilePrintPreview_Click'."
End Sub
Private Sub mnuFilePageSetup_Click()
On Error Resume Next
With dlgCommonDialog
.DialogTitle = "Configurar página"
.CancelError = True
.ShowPrinter
End With
End Sub
Private Sub mnuFileProperties_Click()
'TareasPendientes: Agregar código 'mnuFileProperties_Click'.
MsgBox "Agregar código 'mnuFileProperties_Click'."
End Sub
Private Sub mnuFileSaveAll_Click()
'TareasPendientes: Agregar código 'mnuFileSaveAll_Click'.
MsgBox "Agregar código 'mnuFileSaveAll_Click'."
End Sub
Private Sub mnuFileSaveAs_Click()
'TareasPendientes: Agregar código 'mnuFileSaveAs_Click'.
MsgBox "Agregar código 'mnuFileSaveAs_Click'."
End Sub
Private Sub mnuFileSave_Click()
'TareasPendientes: Agregar código 'mnuFileSave_Click'.
MsgBox "Agregar código 'mnuFileSave_Click'."
End Sub
Private Sub mnuFileClose_Click()
'TareasPendientes: Agregar código 'mnuFileClose_Click'.
MsgBox "Agregar código 'mnuFileClose_Click'."
End Sub
Private Sub mnuFileOpen_Click()
Dim sFile As String
With dlgCommonDialog
.DialogTitle = "Abrir"
.CancelError = False
'Pendiente: establecer los indicadores y atributos del control comondialog
.Filter = "Todos los archivos (*.*)|*.*"
.ShowOpen
If Len(.FileName) = 0 Then
Exit Sub
End If
sFile = .FileName
End With
'Pendiente: agregar código para procesar el archivo abierto
End Sub
Private Sub mnuFileNew_Click()
'TareasPendientes: Agregar código 'mnuFileNew_Click'.
MsgBox "Agregar código 'mnuFileNew_Click'."

```

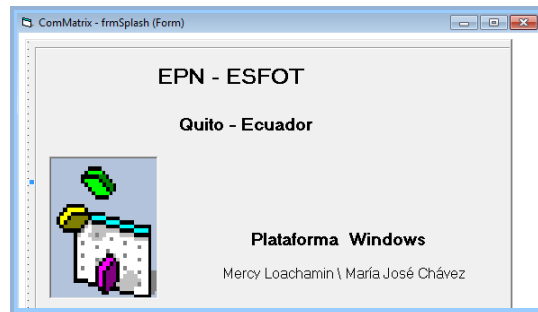
End Sub



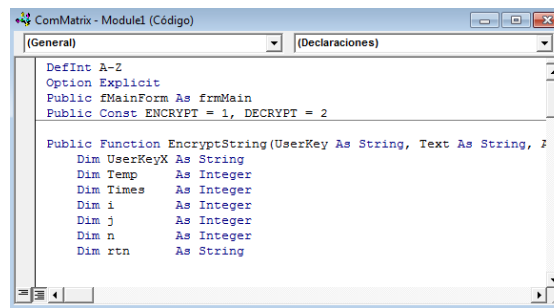
```

Private Sub cmdApply_Click()
    'ToDo: Add 'cmdApply_Click' code.
    MsgBox "Aplicar código va aquí para establecer opciones sin cerrar el cuadro de diálogo"
End Sub
Private Sub cmdCancel_Click()
    Unload Me
End Sub
Private Sub cmdOK_Click()
    'Pendiente: Agregar código para 'cmdOK_Click'.
    MsgBox "Aquí se coloca código para establecer opciones y cerrar el cuadro de diálogo"
    Unload Me
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    Dim i As Integer
    i = tbsOptions.SelectedItem.Index
    'controlar ctrl+tab para mover a la siguiente ficha
    If (Shift And 3) = 2 And KeyCode = vbKeyTab Then
        If i = tbsOptions.Tabs.Count Then
            'última ficha, por lo que hay que volver a la primera ficha
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(1)
        Else
            'incrementar la ficha
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(i + 1)
        End If
    ElseIf (Shift And 3) = 3 And KeyCode = vbKeyTab Then
        If i = 1 Then
            'última ficha, por lo que hay que volver a la primera ficha
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(tbsOptions.Tabs.Count)
        Else
            'incrementa la ficha
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(i - 1)
        End If
    End If
End Sub
Private Sub tbsOptions_Click()
    Dim i As Integer
    'mostrar y habilitar los controles seleccionados de la ficha
    'y ocultar y deshabilitar los demás
    For i = 0 To tbsOptions.Tabs.Count - 1
        If i = tbsOptions.SelectedItem.Index - 1 Then
            picOptions(i).Left = 210
            picOptions(i).Enabled = True
        Else
            picOptions(i).Left = -20000
            picOptions(i).Enabled = False
        End If
    Next
End Sub

```



```
Private Sub Form_Load()
'   lblVersion.Caption = "Version "&App.Major& "." &App.Minor& "." &App.Revision
lblProductName.Caption = App.Title
End Sub
Private Sub frmMainFrame_DragDrop(Source As Control, X As Single, y As Single)
End Sub
```



```
DefInt A-Z
Option Explicit
Public fMainFormAsfrmMain
PublicConst ENCRYPT = 1, DECRYPT = 2
Public Function EncryptString(UserKey As String, Text As String, Action As Single) As String
    Dim UserKeyXAs String
    Dim Temp As Integer
    Dim Times As Integer
    Dim i As Integer
    Dim j As Integer
    Dim n As Integer
    Dimrtn As String
    n = Len(UserKey)
    ReDimUserKeyASCIIS(1 To n)
    For i = 1 To n
    UserKeyASCIIS(i) = Asc(Mid$(UserKey, i, 1))
    Next
    ReDimTextASCIIS(Len(Text)) As Integer
    For i = 1 To Len(Text)
    TextASCIIS(i) = Asc(Mid$(Text, i, 1))
    Next
    IfAction = ENCRYPT Then
        For i = 1 To Len(Text)
            j = If(j + 1 >= n, 1, j + 1)
            Temp = TextASCIIS(i) + UserKeyASCIIS(j)
            If Temp > 255 Then
                Temp = Temp - 255
            End If
            rtn = rtn + Chr$(Temp)
        Next
    ElseIfAction = DECRYPT Then
        For i = 1 To Len(Text)
            j = If(j + 1 >= n, 1, j + 1)
            Temp = TextASCIIS(i) - UserKeyASCIIS(j)
            IfTemp < 0 Then
                Temp = Temp + 255
            End If
            rtn = rtn + Chr$(Temp)
        Next
    End If
    Return rtn
End Function
```

```
        End If
    rtn = rtn + Chr$(Temp)
Next
    End If
EncryptString = rtn
End Function
Sub Main()
    Dim e As Currency
    Dim f As Currency
    Dim fLogin As New frmLogin
    fLogin.ShowvbModal
    If Not fLogin.OKThen
        'Fallo al iniciar la sesión, se sale de la aplicación
    End
    End If
    Unload fLogin
    frmSplash.Show
    frmSplash.Refresh
    e = Timer
    Do
        f = Timer - e
    Loop While f < 2
    Set fMainForm = New frmMain
    Load fMainForm
    Unload frmSplash
    form1.Show
End Sub
```