



**UNIVERSIDAD TECNICA DE AMBATO**

**FACULTAD DE INGENIERIA EN SISTEMAS**

**CARRERA DE INGENIERIA ELECTRONICA**

**TEMA:**

---

**ADQUISICION DE DATOS DE UN SISTEMA MAESTRO – ESCLAVO  
UTILIZANDO MICROCONTROLADORES MEDIANTE  
COMUNICACIÓN SERIAL PARA “M&B AUTOMATIZACION”**

---

Proyecto de pasantía presentado como requisito previo a la obtención del Título de INGENIERO ELECTRÓNICO

**Autor:**

Freddy Marcelo Ballesteros Jordán

**Tutor:**

Ing. Julio Cuji

**ABRIL - 2007**

**AMBATO - ECUADOR**

## **APROBACION DEL TUTOR**

En calidad de tutor del trabajo de investigación sobre el tema:

ADQUISICION DE DATOS DE UN SISTEMA MAESTRO – ESCLAVO UTILIZANDO MICROCONTROLADORES MEDIANTE COMUNICACIÓN SERIAL PARA “M&B AUTOMATIZACION”, de FREDDY MARCELO BALLESTEROS JORDAN estudiante de la carrera de Ingeniería Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Universidad Técnica de Ambato, considero que dicho informe investigativo reúne los requisitos y méritos suficientes para ser sometidos a la evaluación de conformidad con el Art. 68 del Capítulo IV Pasantías, del reglamento de Graduación de Pregrado de la Universidad Técnica de Ambato

EL TUTOR

Ing. Julio Cuji

## **AUTORIA**

El presente proyecto de pasantía de grado: **ADQUISICION DE DATOS DE UN SISTEMA MAESTRO – ESCLAVO UTILIZANDO MICROCONTROLADORES MEDIANTE COMUNICACIÓN SERIAL PARA “M&B AUTOMATIZACION”**. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprendan del mismo son de exclusiva responsabilidad del autor.

Ambato, Abril 2007-04-26

Freddy Marcelo Ballesteros Jordán

C.C. 1803223559

## **DEDICATORIA**

Este trabajo especial de grado se lo dedico a:

Dios todopoderoso, fuente de sabiduría y esperanza. Me ha proporcionado el camino y me ha iluminado la mente en los momentos más difíciles de la carrera.

A mi Padre Luis Mentor, quien ahora esta junto a Dios, pero a pesar de ello siempre ha permanecido en mi corazón brindándome la fuerza para seguir adelante.

A mi madre Fanny Magdalena, quien con mucho esfuerzo, amor y sabiduría ha sabido levantar esta familia, además con entereza y cariño ha sabido inculcarme buenos principios y por proporcionarme una referencia moral.

A mis hermanos Diego, Daniel, Cristina y Belén por apoyarme en todo momento, nunca han dejado de preocuparse porque llegue hasta mi meta propuesta.

A mi abuelita, mis sobrinos, mi tía, mis primos, mis amigos.

## **AGRADECIMIENTO**

Agradezco a Dios por haberme dado la vida y las fuerzas necesarias para alcanzar mi meta propuesta.

Quiero dar un agradecimiento especial al Ing. Fernando Muñoz, quien ha puesto una valiosa contribución para la realización de este trabajo especial de grado.

Al Ing. Julio Cuji, ingeniero tutor designado por la Facultad de Ingeniería en Sistemas, gracias por el cúmulo de conocimientos técnicos impartidos durante toda la carrera.

A mis amigos y compañeros con quienes he vivido y compartido buenos y malos momentos en esta etapa gratificante de la vida en especial a: Tannia Salvador y Lorena Manzano.

A todos los profesores de esta gran casa de estudios quienes de manera desinteresada han transmitido los conocimientos a todos los que aspiramos ser profesionales dedicados a la ingeniería y al progreso.

Finalmente darle un agradecimiento a mi querida Facultad de Ingeniería en Sistemas, en donde he adquirido una dirección y un enfoque en mi vida.

## RESUMEN EJECUTIVO

El área de automatización desarrolla actividades educativas de investigación y desarrollo y de extensión, en el área de sistemas dinámicos y sus aplicaciones al control automático, teoría de señales, identificación, modelamiento e Instrumentación.

El tema de automatización nos dará una visión muchísimo más amplia de lo que puede ayudar esto a una empresa ya que se va a dar en la misma un proceso de mecanización de las actividades industriales para reducir la mano de obra, simplificar el trabajo para que así se de propiedad a algunas maquinas de realizar las operaciones de manera automática; por lo que indica que se va dar un proceso más rápido y eficiente.

Diariamente se aprecia como en las industrias se va haciendo más fácil el trabajo humano, pues muchos de sus sistemas son reemplazados por sistemas controlados automáticamente, lo que permitirá que el trabajo que lo desarrollaban varias personas sea reemplazado por tan solo una persona, todo esto con la ayuda de la electrónica y de sistemas informáticos.

El presente trabajo especial de grado tiene como objetivo realizar la adquisición de datos en un sistema maestro – esclavo utilizando comunicación serial. Es importante ya que permitirá el avance tecnológico en el área industrial y facilitará a las pequeñas y medianas empresas adquirir o implementar equipos para adquisición de datos e implementarlos en sus fábricas, laboratorios, etc. de manera que este sistema satisfaga sus necesidades y a un precio más accesible.

La metodología empleada permitió seguir de manera ordenada los pasos para el diseño e implementación del sistema para la adquisición de datos, obteniendo un sistema capaz de mostrar la información requerida de una manera segura y eficaz, permitiendo además la visualización en un entorno grafico de los datos obtenidos y de las salidas a manejar desde el computador hacia el lugar de destino donde se podrá tener algún dispositivo electrónico.

Finalmente se realizó una fase de prueba del equipo, en donde se experimentó al máximo sus capacidades.

## INTRODUCCIÓN

El trabajo que a continuación vamos a presentar es acerca de un tema de mucha importancia para nosotros mismos y en especial para toda empresa industrial, el cual lleva el nombre de sistema maestro – esclavo para la adquisición de datos.

El proyecto tiene como objetivo final controlar una cantidad parametrizable de datos, de dos maneras distintas: mediante una interfaz serial (tipo RS232) y mediante una interfaz I2C. La programación de los microcontroladores se realizó en PIC BASIC PRO, de tal manera que los distintos microcontroladores del proyecto puedan ser reprogramados independientemente. Dado que se deseaba obtener componentes reusables, no se utilizó ninguna característica especial de la arquitectura de los microcontroladores.

Para el testeo del sistema, se desarrolló un software para PC que envía los comandos al microcontrolador mediante el puerto serial. Para el testeo del sistema mediante la interfaz del bus I2C el software emula la funcionalidad básica de un dispositivo I2C maestro.

Como dijimos anteriormente al darse una mayor eficiencia en el sector de maquinaria, lograra que la empresa industrial disminuya el tiempo al realizar adquisición de datos, y por lo tanto aumente a una mayor eficiencia en las labores de sus empleados; todo esto ayudara a que la empresa industrial mediante la utilización de inversiones tecnológicas aumente toda su competitividad en un porcentaje considerable con respecto a toda su competencia, y si no se hace, la empresa puede sufrir el riesgo de quedarse rezagado.

Esperamos que con todo esto y más podamos cumplir con todas las expectativas propuestas antes de investigar este tema y logremos alcanzar el objetivo que es aprender acerca de la automatización.

## **INDICE**

### **INDICE GENERAL**

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORIA	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN EJECUTIVO	vi
INTRODUCCION	vii
INDICE GENERAL	viii
INDICE DE CONTENIDO	viii

### **INDICE DE CONTENIDO**

	<b>Página</b>
<b>CAPITULO I</b>	
1 EL PROBLEMA DE INVESTIGACION	
1.1 Tema de Investigación	1
1.2 Planteamiento del problema	1
1.2.1 Contextualización	1
1.2.2 Formulación del problema	2
1.2.3 Delimitación del problema	2
1.3 Justificación	2
1.4 Objetivos	4
1.4.1 Objetivo General	4
1.4.2 Objetivos Específicos	4
<b>CAPITULO II</b>	
2 MARCO TEÓRICO	
2.1 Antecedentes Investigativos	5
2.2 Fundamentación Legal	5
2.3 Categorías Fundamentales	8
Sistema	8
Control	9



MICROCONTROLADORES	10
Controlador y Microcontrolador	10
Aplicaciones de los Microcontroladores	12
El Mercado de los Microcontroladores	13
¿Qué microcontrolador emplear?	14
Recursos comunes a todos los microcontroladores	16
Recursos Especiales	22
Herramientas para el desarrollo de aplicaciones	25
PIC BASIC PRO	27
COMUNICACIÓN SERIAL	27
INTERFACES DE COMUNICACIÓN	30
EL BUS I2C	30
NORMA RS232	35
NORMA RS422/485	37
EL RUIDO	38
Origen del Ruido	38
Factor de Ruido	39
Ruido del Sistema	40
2.4 Hipótesis	40
2.5 Señalamiento de variables	40
- Variable Independiente	
- Variable Dependiente	
<b>CAPITULO III</b>	
<b>3 METODOLOGÍA</b>	
3.1 Modalidad básica de investigación	41
3.2 Tipos de investigación	41
3.3 Población y muestra	42
3.4 Técnicas e instrumentos de investigación	42
3.5 Recolección de información	42
3.6 Procesamiento y Análisis	43

**CAPITULO IV**

4. ANALISIS E INTERPRETACION DE RESULTADOS	56
--	----

**CAPITULO V**

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES	57
-------------------	----

5.2 RECOMENDACIONES	58
---------------------	----

**CAPITULO IV**

6. PROPUESTA	60
--------------	----

BIBLIOGRAFÍA	62
--------------	----

ANEXOS	63
--------	----

# **CAPITULO I**

## **1. EL PROBLEMA DE INVESTIGACION**

### **1.1 TEMA DE INVESTIGACION**

Implementación de un sistema para adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial

### **1.2 PLANTEAMIENTO DEL PROBLEMA**

#### **1.2.1 Contextualización**

En Ecuador se ha estado incentivando a la creación de grandes y pequeñas empresas en diferentes áreas, con el fin de modernizar y actualizar la estructura tecnológica en el área industrial, son varias alternativas que se presentan para mejorar la producción masiva de los productos.

Muchas Industrias en el Ecuador se han visto afectadas con el acelerado crecimiento de la tecnología en maquinaria de tipo industrial por lo que sus propietarios se han visto en la necesidad de equipar sus industrias con nuevos equipos, para poder competir con otras industrias, tanto a nivel nacional como con industrias de fuera del país.

Pero la dificultad de adquirir maquinaria importada se debe a diversos factores tales como los aranceles de importación, el costo de los fletes y del envío de la misma, la asesoría técnica, etc. donde la mayoría de las pequeñas y medianas empresas no poseen la capacidad de enfrentar tales escenarios

En la actualidad ciertos sectores de la industria nacional ya utilizan sistemas de comunicación para el control y funcionamiento de sus operaciones.

Actualmente, la mayoría de los sistemas electrónicos llevan acoplados algún microcontrolador. La industria actual ha requerido de estas unidades para poder aumentar las aplicaciones en sistemas para poder facilitar la vida de los usuarios. Estos “chips” forman parte de lo que será un sistema empotrado, el cual tendrá múltiples funciones y digamos que será el encargado de dirigir el sistema en cuestión.

Desde el punto de vista de la eficiencia, es deseable la reducción de los costos de inversión para que las industrias entren a competir en el mercado nacional, por ello es necesaria la adaptación de tecnologías de acuerdo a las necesidades básicas del cliente.

### **1.2.2 Formulación del Problema**

¿Qué beneficio proporciona a la empresa M&B AUTOMATIZACION la implementación de un sistema para la adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial?

### **1.2.3 Delimitación del Problema**

El presente trabajo investigativo se realizará en la empresa M&B AUTOMATIZACION se abarcará todo lo referente al sistema para la adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial.

## **1.3 JUSTIFICACIÓN**

Los sistemas de control y adquisición de datos en operaciones industriales han venido creciendo con el transcurso del tiempo, es por eso que se han adoptado

sistemas que trabajen en condiciones remotas, ya sea por centralizar los procesos o por tener un control total sin depender de las distancias.

Los sistemas de comunicación para procesos de control, que han trabajado hasta la actualidad, tienen sus ventajas al igual que sus limitantes de acuerdo a la tecnología aplicada. Sin embargo la gran mayoría presenta dificultades principalmente en lo que respecta al factor económico.

Al desarrollar este sistema se ofrecerá a las industrias un proyecto (producto) ecuatoriano a un precio accesible para sus propietarios, además permitirá el avance tecnológico en el área industrial y facilitará a las pequeñas y medianas empresas adquirir equipos de control adaptados a las características propias de sus maquinas.

En este sistema se aplicará los conocimientos adquiridos en la carrera, se aportará a la solución de problemas en las industrias como es la comunicación entre los diferentes periféricos.

Existen dos formas de realizar una comunicación binaria, la paralela y la serial. La comunicación paralela tiene la ventaja de que la transferencia de datos es más rápida, pero el inconveniente es que necesitamos un cable por cada bit de dato, lo que encarece y dificulta el diseño de las placas, otro inconveniente es la capacitancia que genera los conductores por lo que la transmisión se vuelve defectuosa a partir de unos pocos metros.

La comunicación serial en cambio es mucho más lenta debido a que transmite bit por bit pero tiene la ventaja de necesitar menor cantidad de hilos, y además se puede extender la comunicación a mayor distancia, por ejemplo; en la norma RS232 a 15mts., en la norma RS422/485 a 1200mts. Y utilizando un MODEM, púes a cualquier parte del mundo.

En la mayoría de los proyectos es necesario comunicarse con el resto del mundo. La comunicación serial es relativamente fácil de implementar, y es perfecta para aplicaciones que no requieren de un gran ancho de banda.

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

- Implementar un sistema para la adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial para la empresa M&B AUTOMATIZACION, con la finalidad de disminuir el ruido en el envío de la información

### **1.4.2.- OBJETIVOS ESPECÍFICOS**

- Determinar las ventajas al trabajar con microcontroladores.
- Estudiar las condiciones físicas y las variables involucradas para la realización del proyecto
- Controlar una cantidad parametrizable de datos mediante una interfaz I2C.
- Desarrollar un sistema para la adquisición de datos y control de las variables involucradas a través de una interfaz hombre – máquina.
- Desarrollar una aplicación para el manejo de la información con capacidad de programación de parámetros en la industria.
- Obtener información de alta confiabilidad a través de la disminución de ruido mediante la implementación del sistema generado.

## **CAPITULO II**

### **MARCO TEORICO**

#### **2.1 ANTECEDENTES INVESTIGATIVOS**

Revisados los archivos de la Facultad de Ingeniería en Sistemas se determina que no existe ningún trabajo de investigación referente al tema.

#### **2.2 FUNDAMENTACION LEGAL**

Dado que al frente de M&B AUTOMATIZACION se encuentra el Ing. Fernando Muñoz, se ha tomado como referencia:

**La LEY DE EJERCICIO PROFESIONAL DE LA INGENIERIA**  
**(Registro Oficial Núm. 709 – Diciembre 26 de 1974)**

### **CAPITULO I**

#### **NORMAS FUNDAMENTALES**

**Art. 2º.-** Esta Ley garantiza el libre ejercicio de la profesión, dentro de cada rama de la Ingeniería; en consecuencia, condena toda forma de competencia desleal, ya provenga de personas naturales o jurídicas, sean de derecho público o de derecho privado

### **CAPÍTULO II**

#### **DE LOS PROFESIONALES**

**Art. 4º.-** Para los efectos de esta Ley, son profesionales los Ingenieros que hayan obtenido su título en las Universidades, Escuelas Técnicas de Ingenieros, Escuelas

Politécnicas y demás Instituciones de Enseñanza Superior del País reconocidos por la Ley de Educación Superior, o los que hayan revalidado e inscrito en el Ecuador sus respectivos títulos de Ingenieros, obtenidos en el exterior, de conformidad con lo que dispone la indicada Ley.

### **CAPÍTULO III**

#### **DEL EJERCICIO PROFESIONAL**

**Art. 10.-** El ejercicio profesional de los Ingenieros amparados por esta Ley, se realizará exclusivamente en las actividades profesionales inherentes al título obtenido, que será regulado en el respectivo Reglamento.

**Art. 12.-** Para que cualquiera de los documentos técnicos a los que se refiere el Artículo anterior pueda ser presentado y surta efecto en las oficinas públicas y para que su contenido pueda llevarse a ejecución en todo o en parte, por cualquier persona o entidad pública o privada, deberá llevar la firma y el número de licencia profesional del autor intelectual de la respectiva rama, y acompañarse el recibo de pago de la contribución a la que se refiere el artículo 26.

**Art. 16.-** El desempeño de cargos técnicos en las instituciones de derecho público, solo podrán realizarse por los profesionales cuyo título este de acuerdo con las especialidades de las distintas ramas de la Ingeniería atinentes al correspondiente cargo.

#### **Reglamento a La Ley de Ejercicio Profesional de La Ingeniería (Suplemento del Registro Oficial N° 257 del 18 de Enero de 1977)**

### **TÍTULO I**

#### **NORMAS FUNDAMENTALES**

### **CAPITULO II**

#### **De los Profesionales**

**Art. 4°.-** Están amparados por la Ley y sus Reglamentos los profesionales de las siguientes ramas de la Ingeniería: Agrícola, Agronómica, de los Alimentos,



Geográfica, **Eléctrica y Electrónica**, Forestal, Geológica, de Minas y Petróleos, Industrial, Mecánica, Naval, Química, Zootecnistas, Informática, Sistemas y Computación.

## **TITULO II DEL EJERCICIO PROFESIONAL**

### **CAPÍTULO II**

#### **Normas relativas al Ejercicio Profesional**

**Art. 22.-** El ejercicio profesional de la ingeniería se desarrolla a través de los siguientes campos de actividad, en razón del respectivo título académico:

#### **4. INGENIERO ELÉCTRICO Y ELECTRÓNICO:**

- a) En los que por medio de la electricidad se pone en movimiento a un cuerpo en cualquier estado;
- b) Toda obra en que se genere, transforme, transmita, distribuya y utilice la energía eléctrica;
- c) En la que se usen ondas electromagnéticas;
- d) Todo proceso en el que se use electricidad o electrónica para cambiar la estructura de la materia, su estado, su forma o su composición química;
- e) Todos los métodos en los que se usen magnitudes eléctricas para medir o controlar magnitudes físicas;
- f) Todos los campos que correlacionen la luz, temperatura, sonido, fuerza, etc., con la electricidad;
- g) Todo campo en el que se use electricidad para generar, transformar, transmitir, recibir señales e informaciones que sean análogas o digitales;

- h) Todo fenómeno en el que la electricidad se comporte como proceso estocástico o sirva como medio para determinar mejor estos procesos;
- i) Métodos en los que la electricidad sirve directa o indirectamente para terapia o diagnóstico;
- j) Todo fenómeno que pueda ser estudiado con una analogía matemática relacionada con la electricidad;
- k) Todo trabajo teórico o práctico en el que se deban tomar medidas de seguridad por medio de aislaciones, conexiones a tierra, protecciones, etc.;
- l) En lo relacionado con los materiales empleados en la fabricación, construcción, diseño de elementos o componentes eléctricos y/o electrónicos;
- m) Todo proceso telemétrico, de telecontrol y de telecomunicaciones; y,
- n) Cualquier otro trabajo que por su naturaleza o los conocimientos especiales que requiera, esté capacitado para ejercer el Ingeniero Eléctrico y el Ingeniero Electrónico.

## **2.3 CATEGORIAS FUNDAMENTALES**

### **SISTEMA**

Senn, J (1990), define el sistema como: “un conjunto de componentes que interaccionan entre sí para lograr un cierto objetivo, es una colección de elementos o medios que están relacionados y que pueden ser descritos en términos de sus atributos o de sus partes componentes”

Sistema es un todo unitario, organizado, compuesto por dos o más partes y delineado por los límites de un entorno. Cabe destacar que todo proceso, en general, está constituido por varios sistemas individuales mutuamente interactuantes. La adecuada concatenación e interrelación de los diversos hará que cada proceso particular cumpla eficazmente con la misión para la cual se concibió.

## **CONTROL**

El control se ha vuelto parte integral e importante de los procesos industriales y de manufactura moderna, resultando esencial en operaciones industriales debido al aporte de medios para lograr el funcionamiento óptimo de sistemas dinámicos, mejorar la productividad y eliminar las operaciones manuales rutinarias y repetitivas. Gracias al control, las fallas en los procesos se detectan, pudiéndose corregir para eliminar los errores.

Las acciones correctivas de control pueden ser de orden manual o automático. Si todas las tareas son efectuadas sin intervención de un eslabón humano, el control es considerado automático. De acuerdo a Ogata, K (1998), un sistema de regulación automática “es un sistema de control retroalimentado en el que la entrada de referencia o la salida deseada son, o bien constantes o bien varían lentamente en el tiempo, y donde la tarea fundamental consiste en mantener la salida en el valor deseado a pesar de las perturbaciones presentes”

A un sistema de regulación automática en el que la salida es una variable como temperatura, presión, flujo, nivel de líquido o pH, se le denomina sistema de control de procesos. Estos sistemas se caracterizan por su retroalimentación o control a lazo cerrado, donde la señal de error actuante, que es la diferencia entre la señal de entrada y la de retroalimentación (que puede ser la señal de salida o una función de la señal de salida), entra a un controlador para reducir el error y llevar la salida del sistema a un valor deseado

## **MICROCONTROLADORES**

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

Es de hecho una computadora completa situada en un único circuito integrado, que contiene todos los elementos del microprocesador básico además de otras funciones especializadas tales como: unidad central de procesamiento (ALU), memoria RAM, memoria EEPROM, memoria FLASH, periféricos, etc. pero con prestaciones y capacidades limitadas. Las principales ventajas del microcontrolador son: costo de desarrollo e implementación reducido, versatilidad y facilidad de utilización.

Esto los hace atractivos para las aplicaciones de control de procesos y otras áreas conexas, debido a que no requieren una capacidad de memoria de un microprocesador, en estos casos están orientadas más a las operaciones de E/S, adquisición de datos, temporizadores, contadores, interrupciones, PWM, etc.

## **CONTROLADOR Y MICROCONTROLADOR**

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.

- Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

## **APLICACIONES DE LOS MICROCONTROLADORES**

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

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

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

## **EL MERCADO DE LOS MICROCONTROLADORES.**

Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquéllos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

La distribución de las ventas según su aplicación es la siguiente:

- Una tercera parte se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.
- La cuarta parte se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)
- El 16% de las ventas mundiales se destinó al área de las comunicaciones.
- Otro 16% fue empleado en aplicaciones industriales.

- El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.
- También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

### ¿QUÉ MICROCONTROLADOR EMPLEAR?

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.)

**Costes.** Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer demasiado ruido venden 10 veces más microcontroladores que microprocesadores.

Para que nos hagamos una idea, para el fabricante que usa el microcontrolador en su producto una diferencia de precio en el microcontrolador de algunos dólares es importante (el consumidor deberá pagar además el coste del empaquetado, el de los otros componentes, el diseño del hardware y el desarrollo del software). Si el fabricante desea reducir costes debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decanten por microcontroladores pertenecientes a una única familia.

**Aplicación.** Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación:

- **Procesamiento de datos:** puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de



seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión.

- **Entrada Salida:** para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.

- **Consumo:** algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.

- **Memoria:** para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

- **Ancho de palabra:** el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es

de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

- **Diseño de la placa:** la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:

**8048 (Intel).** Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.

**8051 (Intel y otros).** Es sin duda el microcontrolador más popular. Fácil de programar, pero potente. Está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

**80186, 80188 y 80386 EX (Intel).** Versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.

**68HC11 (Motorola y Toshiba).** Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.

**683xx (Motorola).** Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

**PIC (MicroChip).** Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

Es preciso resaltar en este punto que existen innumerables familias de microcontroladores, cada una de las cuales posee un gran número de variantes.

## **RECURSOS COMUNES A TODOS LOS MICROCONTROLADORES.**

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de

periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

### **Arquitectura básica**

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

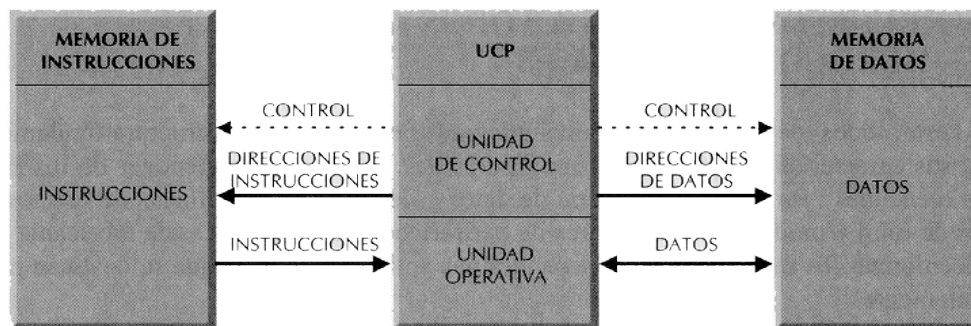


Figura. La arquitectura Harvard dispone de dos memorias independientes para datos y para instrucciones, permitiendo accesos simultáneos.

Los microcontroladores PIC responden a la arquitectura Harvard.

### **El procesador o UCP**

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica

la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

*CISC*: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

*RISC*: Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

*SISC*: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es "específico", o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

### **Memoria**

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

No existen sistemas de almacenamiento masivo como disco duro o disquetes.

Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

#### ***1º. ROM con máscara***

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

#### ***2ª. OTP***

El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

#### ***3ª EPROM***

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only memory) pueden borrarse y grabarse muchas veces. La

grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

#### **4ª EEPROM**

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno.

Este tipo de memoria es relativamente lenta.

#### **5ª FLASH**

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

### **Puertas de Entrada y Salida**

La principal utilidad de las patitas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

### **Reloj principal**

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

## RECURSOS ESPECIALES

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o "Timers".
- Perro guardián o "Watchdog".
- Protección ante fallo de alimentación o "Brownout".
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

### **Temporizadores o "Timers"**

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.



### **Perro guardián o "Watchdog"**

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

### **Protección ante fallo de alimentación o "Brownout"**

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("brownout"). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

### **Estado de reposo ó de bajo consumo**

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando sumido en un profundo "sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

### **Conversor A/D (CAD)**

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

### **Conversor D/A (CDA)**

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

### **Comparador analógico**

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

### **Modulador de anchura de impulsos o PWM**

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

### **Puertos de E/S digitales**

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.

Las líneas digitales de los Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

### **Puertos de comunicación**

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- \* UART, adaptador de comunicación serie asíncrona.
- \* USART, adaptador de comunicación serie síncrona y asíncrona
- \* Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- \* USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- \* Bus *I2C*, que es un interfaz serie de dos hilos desarrollado por Philips.
- \* CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

## **HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES.**

Uno de los factores que más importancia tiene a la hora de seleccionar un microcontrolador entre todos los demás es el soporte tanto software como hardware de que dispone. Un buen conjunto de herramientas de desarrollo puede ser decisivo en la elección, ya que pueden suponer una ayuda inestimable en el desarrollo del proyecto.

Las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

### **Desarrollo del software:**

**Ensamblador.** La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita y en cualquier caso siempre se puede encontrar una versión gratuita para los microcontroladores más populares.

**Compilador.** La programación en un lenguaje de alto nivel (como el C ó el Basic) permite disminuir el tiempo de desarrollo de un producto. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador. Las versiones más potentes suelen ser muy caras, aunque para los microcontroladores más populares pueden encontrarse versiones demo limitadas e incluso compiladores gratuitos.

**Depuración:** debido a que los microcontroladores van a controlar dispositivos físicos, los desarrolladores necesitan herramientas que les permitan comprobar el buen funcionamiento del microcontrolador cuando es conectado al resto de circuitos.

**Simulador.** Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba in-situ.

**Placas de evaluación.** Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. El sistema operativo de la placa recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, puede permitir en cualquier momento realizar ejecución paso a paso, monitorizar el estado del microcontrolador o modificar los valores almacenados los registros o en la memoria.

**Emuladores en circuito.** Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego

irá en el zócalo. Presenta en pantalla toda la información tal y como luego sucederá cuando se coloque la cápsula.

### **PIC BASIC PRO**

El compilador PicBasic Pro (PBP) es nuestro lenguaje de programación de nueva generación que hace más fácil y rápido para nosotros programar micro controladores Pic micro de **Microchip Technology**.

El lenguaje Basic es mucho más fácil de leer y escribir que el lenguaje ensamblador Microchip.

El PBP es similar al “**BASIC STAMP II**” y tiene muchas de las librerías y funciones de los BASIC STAMP I y II. Como es un compilador real los programas se ejecutan mucho más rápido y pueden ser mayores que sus equivalentes STAMP.

PBP no es tan compatible con los BASIC STAMP como nuestro compilador PicBasic es con el BS I.

PBP por defecto crea archivos que corren en un PIC 16F84-04/P con un reloj de 4 Mhz. Solamente muy pocas partes son necesarias capacitores de dos capacitores de 22 pf para el cristal de 4Mhz un resistor de 4.7K en el pin/MCLR y una fuente de 5 volt. Otros micros PIC además del 16F84, así como otros osciladores de frecuencias distintas pueden ser usados por este compilador.

### **COMUNICACIÓN SERIAL**

La información en una cadena serial de bits esta contenida en su forma de onda dependiente del tiempo: los bits se representan por códigos que se transmiten por un periodo de tiempo fijo. El periodo de tiempo usado para transmitir cada código se conoce como periodo *baud*.

El mundo de las comunicaciones internas del computador se realiza en forma paralela alternada, por fuera del computador predominan las comunicaciones seriales; las redes de computadores se basan en dicha comunicación.

El PC utiliza la norma RS232, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios.

La información que maneja un computador puede transmitirse de un lugar a otro en dos formas básicas, en forma serial o en forma paralela. En una transmisión serial se forma un “tren” de bits, uno tras de otro viajan del lugar de emisión al receptor utilizando una sola vía, en este caso será un conductor eléctrico bus Serial, como en caso de los trenes con una sola vía si se desea transmitir en el sentido contrario, se debe esperar que la vía este libre. En la comunicación en paralelo cada bit tiene su vía exclusiva, con la condición de que todos viajen simultáneamente, como en el caso de la comunicación serial para transmitir en el sentido contrario se debe esperar que la vía este libre, a menos que se tenga una exclusiva para el sentido contrario.

Las cadenas seriales de bits generadas por los puertos serie de la PC usan una forma muy simple de codificación. Un bit se transmite durante cada periodo baud, con un bit “1” representado por un voltaje alto TTL y un “0” por un voltaje bajo TTL. Así la velocidad en baudios (baud rate,  $1/[\text{periodo baud}]$ ) de un puerto serie de la PC es igual al número de bits por segundo que se transmiten o reciben.

Para enviar información codificada de esta manera, el transmisor y receptor registran el tiempo, el cual define el periodo baud, deben estar a la misma frecuencia y estar sincronizados. Los bits se transmiten como grupos separados, con una longitud típica de 7 u 8 bits, llamados *caracteres*. El nombre caracter se usa porque cada grupo de bits representan una letra del alfabeto cuando el texto esta codificado en ASCII. Cada caracter se envía en una armazón (frame) consistiendo de un bit “0” llamado un *bit de inicio*, seguido por el caracter mismo, seguido (opcionalmente) por un bit de paridad, y después un bit “1” llamado *bit de paro*. La lógica del bit bajo de inicio le dice al receptor que esta empezando una armazón, y la lógica del bit alto de paro denota el final de la armazón.

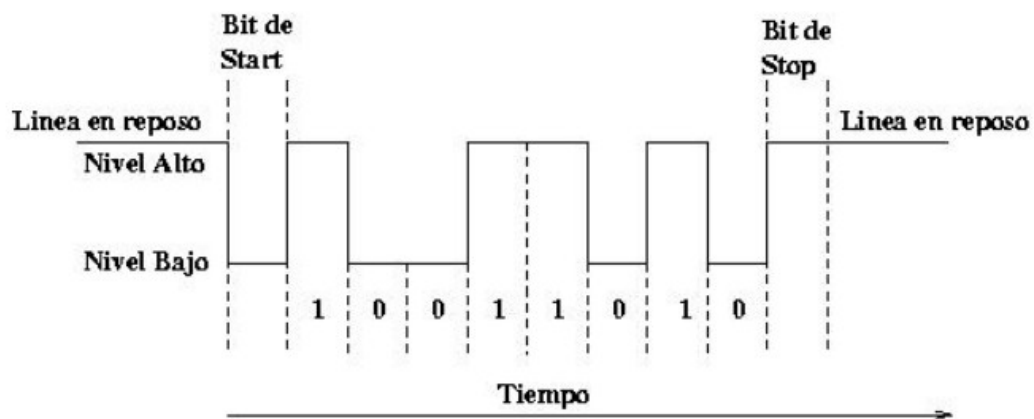
### **Comunicaciones serie asíncronas**

Los datos serie se encuentran encapsulados en tramas de la forma:



Primero se envía un bit de start, a continuación los bits de datos (primero el bit de mayor peso) y finalmente los bits de STOP. El número de bits de datos y de bits de Stop es uno de los parámetros configurables, así como el criterio de paridad par o impar para la detección de errores. Normalmente, las comunicaciones serie tienen los siguientes parámetros: 1 bit de start, 8 bits de Datos, 1 bit de Stop y sin paridad.

En esta figura se puede ver un ejemplo de la transmisión del dato binario 10011010. La línea en reposo está a nivel alto:



Se llama comunicación serial asíncrona porque el receptor se resincroniza el mismo con el transmisor usando el bit de inicio de cada armazón. Los caracteres se pueden transmitir en cualquier tiempo, con un retraso de tiempo arbitrario entre caracteres. Existen también protocolos de comunicación serial síncrona donde los caracteres se envían en bloques sin una armazón de bits circundante. En esta aproximación, el transmisor continuamente transmite señales, con un carácter de sincronización especial que se transmite si no hay datos reales disponibles para transmitir.

Los bits dentro de cada carácter transmitido se envían con el bit menos significativo primero, cada bit durando un periodo baud. Los transmisores y receptores seriales se pueden instruir para enviar o recibir de 5 a 8 bits por carácter (ambos deben de estar de acuerdo en cuantos).

Después de que los bits de cada carácter se envían, puede seguir un bit de paridad opcional. El bit de paridad es útil si la línea de datos esta muy ruidosa como para proporcionar una transmisión fiel. El bit de paridad, P, se puede elegir para dar ya sea paridad par o impar. Para paridad par,  $P = 1$  si el número de 1's en el carácter es impar y  $P = 0$  si el número es par. Es decir, en la paridad par P se elige tal que el número de 1's incluyendo P es par. Para paridad impar, P se elige tal que el número de 1's incluyendo P es impar. El receptor local checa para asegurar que la paridad es aun la misma a pesar de que el cable haya recogido ruido. Si la paridad ha cambiado, algún bit se ha perdido, y el receptor pone una bandera de error de paridad en el registro de estado.

La comunicación serial, como su nombre lo indica, realiza la transferencia de información enviando o recibiendo datos descompuestos en bits, los cuales viajan secuencialmente uno tras otro.

La comunicación serial está compuesta principalmente de dos elementos básicos, el hardware, que hace referencia a la configuración de los conectores y niveles de voltaje, y el software, con el que se controla la información binaria que se quiere transferir. Todo esto está regido por normas o protocolos donde el utilizado por las computadoras convencionales es el protocolo RS-232.

## **INTERFACES DE COMUNICACIÓN**

### **EL BUS I2C**

Diseñado por Philips, este sistema de intercambio de información a través de tan solo dos cables permite a circuitos integrados y módulos OEM interactuar entre sí a velocidades relativamente lentas. Emplea comunicación serie, utilizando un conductor para manejar el timing (pulsos de reloj) y otro para intercambiar datos.

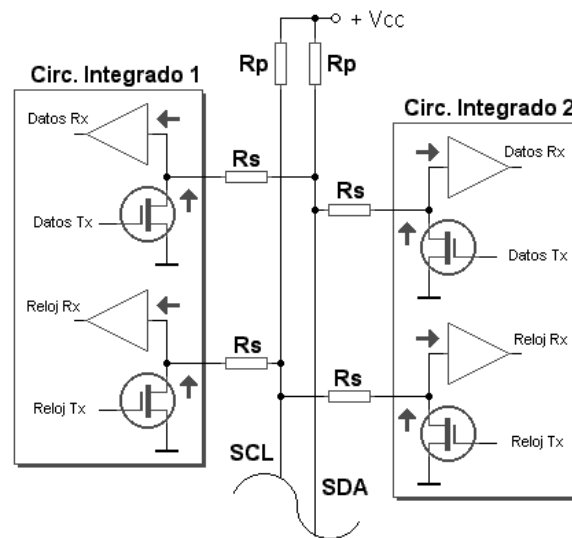
Este bus se basa en tres señales:

- **SDA (System Data)** por la cual viajan los datos entre los dispositivos.



- **SCL (System Clock)** por la cual transitan los pulsos de reloj que sincronizan el sistema.
- **GND (Masa)** Interconectada entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenador abierto, similares a las de colector abierto pero asociadas a un transistor de efecto de campo (ó FET). Se deben poner en estado alto (conectar a la alimentación por medio de resistores Pull-Up) para construir una estructura de bus tal que se permita conectar en paralelo múltiples entradas y salidas.



En el diagrama se observa la configuración eléctrica básica del bus. Las dos líneas de comunicación disponen de niveles lógicos altos cuando están inactivas. Inicialmente el número de dispositivos que se puede conectar al bus es ilimitado, pero obsérvese que las líneas tienen una especificación máxima de 400pF en lo que respecta a capacidad de carga. La máxima velocidad de transmisión de datos que se puede obtener es de aproximadamente 100Kbits por segundo.

Las definiciones o términos utilizados en relación con las funciones del bus I2C son las siguientes:

- **Maestro (Master):** Dispositivo que determina la temporización y la dirección del tráfico de datos en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".

- **Esclavo (Slave):** Cualquier dispositivo conectado al bus incapaz de generar pulsos de reloj. Reciben señales de comando y de reloj proveniente del dispositivo maestro.
- **Bus Desocupado (Bus Free):** Estado en el cual ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Únicamente en este momento es cuando un dispositivo maestro puede comenzar a hacer uso del bus.
- **Comienzo (Start):** Sucede cuando un dispositivo maestro hace ocupación del bus, generando esta condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.
- **Parada (Stop):** Un dispositivo maestro puede generar esta condición dejando libre el bus. La línea de datos toma un estado lógico alto mientras que la de reloj permanece también en ese estado.
- **Dato Válido (Valid Data):** Sucede cuando un dato presente en la línea SDA es estable mientras la línea SCL está a nivel lógico alto.
- **Formato de Datos (Data Format):** La transmisión de datos a través de este bus consta de 8 bits de datos (ó 1 byte). A cada byte le sigue un noveno pulso de reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento, conocido como ACK (del inglés Acknowledge). Esto se logra situando la línea de datos a un nivel lógico bajo mientras transcurre el noveno pulso de reloj.
- **Dirección (Address):** Cada dispositivo diseñado para funcionar en este bus dispone de su propia y única dirección de acceso, que viene pre-establecida por el fabricante. Hay dispositivos que permiten establecer externamente parte de la dirección de acceso. Esto permite que una serie del mismo tipo de dispositivos se puedan conectar en un mismo bus sin problemas de identificación. La dirección 00 es la denominada "de acceso general", por la cual responden todos los dispositivos conectados al bus.
- **Lectura/Escritura (Bit R/W):** Cada dispositivo dispone de una dirección de 7 bits. El octavo bit (el menos significativo ó LSB) enviado durante la operación de direccionamiento corresponde al bit que indica el tipo de operación a realizar. Si este bit es alto el dispositivo maestro lee

información proveniente de un dispositivo esclavo. En cambio, si este bit fuese bajo el dispositivo maestro escribe información en un dispositivo esclavo.

### **Funcionamiento del bus I2C**

Como dijimos, las líneas SDA y SCL transportan información entre los dispositivos conectados al bus.

Cada dispositivo es reconocido por su código (dirección) y puede operar como transmisor o receptor de datos.

Además, cada dispositivo puede ser considerado como Maestro o Esclavo.

El Maestro es el dispositivo que inicia la transferencia en el bus y genera la señal de Clock.

El Esclavo es el dispositivo direccionado.

Las líneas SDA (serial Data) y SCL (serial Clock) son bidireccionales, conectadas al positivo de la alimentación a través de las resistencias de pull-up. Cuando el bus está libre, ambas líneas están en nivel alto.

La transmisión bidireccional serie (8-bits) de datos puede realizarse a 100Kbits/s en el modo standard o 400 Kbits/s en el modo rápido.

La cantidad de dispositivos que se pueden conectar al bus está limitada, solamente, por la máxima capacidad permitida: 400 pF.

### **Condiciones de START y STOP:**

Antes de que se establezca un intercambio de datos entre el circuito Master y los Esclavos, el Master debe informar el comienzo de la comunicación (condición de Start): la línea SDA cae a cero mientras SCL permanece en nivel alto. A partir de este momento comienza la transferencia de datos. Una vez finalizada la comunicación se debe informar de esta situación (condición de Stop). La línea SDA pasa a nivel alto mientras SCL permanece en estado alto.

### **Protocolo del Bus**

Como es lógico, para iniciar una comunicación entre dispositivos conectados al bus I2C se debe respetar un protocolo. Tan pronto como el bus esté libre, un

dispositivo maestro puede ocuparlo generando una condición de inicio. El primer byte transmitido después de la condición de inicio contiene los siete bits que componen la dirección del dispositivo de destino seleccionado y un octavo bit correspondiente a la operación deseada (lectura o escritura). Si el dispositivo cuya dirección se apuntó en los siete bits está presente en el bus éste responde enviando el pulso de reconocimiento ó ACK. Seguidamente puede comenzar el intercambio de información entre los dispositivos.

Cuando la señal R/W está previamente a nivel lógico bajo, el dispositivo maestro envía datos al dispositivo esclavo hasta que deja de recibir los pulsos de reconocimiento, o hasta que se hayan transmitido todos los datos.

En el caso contrario, es decir cuando la señal R/W estaba a nivel lógico alto, el dispositivo maestro genera pulsos de reloj durante los cuales el dispositivo esclavo puede enviar datos. Luego de cada byte recibido el dispositivo maestro (que en este momento está recibiendo datos) genera un pulso de reconocimiento.

El dispositivo maestro puede dejar libre el bus generando una condición de parada (Stop). Si se desea seguir transmitiendo, el dispositivo maestro puede generar otra condición de inicio el lugar de una condición de parada. Esta nueva condición de inicio se denomina "inicio repetitivo" y se puede emplear para direccionar un dispositivo esclavo diferente ó para alterar el estado del bit de lectura/escritura (R/W).

### **Arbitraje**

Una de las características del bus I2C, es que se puede conformar en un sistema multi-maestro, es decir cuando mas de un maestro puede obtener el control del bus, para ello existe un procedimiento que garantiza que, si mas de un maestro accede simultáneamente al bus, únicamente uno de ellos se hace cargo del mismo con el objeto de no alterar la información.

Este procedimiento se le denomina arbitraje y consiste en determinar que maestro se hace cargo del bus en un sistema multi-maestro. Cada maestro debe comparar si el bit de datos que transmite junto con su pulso de reloj en un momento dado, coincide con el nivel lógico presente en la línea de datos SDA. De no ser así, pierde el acceso al bus. Por ejemplo si un maestro saca un nivel lógico "1" al tiempo en que otro saca un "0", en la línea SDA prevalece el segundo, por lo que

el primero debe dejar el bus libre, al menos hasta que se detecte la condición de parada.

### **Direccionamiento**

El proceso de direccionamiento del bus I2C consiste en que el primer byte que envía el maestro tras la condición de inicio es un código que determina y selecciona a un determinado esclavo, el siguiente byte tras la misma indicara la operación a realizar, este byte se denomina el byte de control.

Los 7 bits de mayor peso del primer byte se emplean para direccionar a un determinado esclavo, el de menos peso, determinara si se realizara una operación de escritura o de lectura sobre el esclavo direccionado. Cuando el maestro envía una dirección por el bus, todos los esclavos conectados la comparan con la que poseen internamente.

### **NORMA RS232**

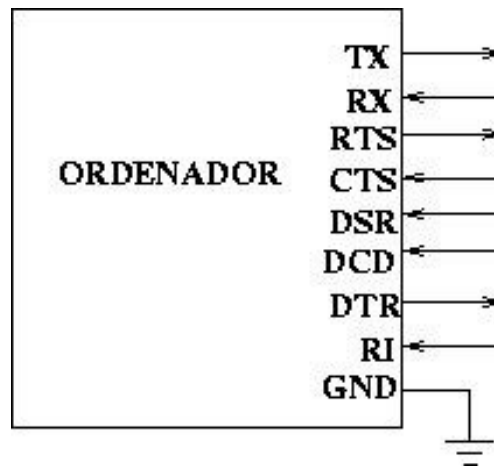
El protocolo RS-232 es una norma o estándar mundial que rige los parámetros de uno de los modos de comunicación serial. Por medio de este protocolo se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, los conectores, etc.

Además de las líneas de transmisión (Tx) y recepción (Rx), las comunicaciones seriales poseen otras líneas de control de flujo (*Handshake*), donde su uso es opcional dependiendo del dispositivo a conectar.

A nivel de software, la configuración principal que se debe dar a una conexión a través de puertos seriales. RS-232 es básicamente la selección de la velocidad en baudios (1200, 2400, 4800, etc.), la verificación de datos o paridad (paridad par o paridad impar o sin paridad), los bits de parada luego de cada dato (1 ó 2), y la cantidad de bits por dato (7 ó 8), que se utiliza para cada símbolo o carácter enviado.

La Norma RS-232 fue definida para conectar un ordenador a un modem. Además de transmitirse los datos de una forma serie asíncrona son necesarias una serie de señales adicionales, que se definen en la norma. Las tensiones empleadas están comprendidas entre +15/-15 voltios.

Puerta serial full dúplex para comunicación punto a punto a una distancia no superior a 30 metros. Desde 3 hilos hasta 19 hilos.



Este protocolo está disponible en los puertos seriales de la mayoría de las computadoras personales (PC).

#### **Conector RS232D (Conector DB9 de 9 pines)**

La siguiente tabla muestra el conector de 9 pines serie que se encuentra en la mayoría de las PC de hoy en día.

SIGNAL	PIN No.
Carrier Detect	1
Receive Data	2
Transmit Data	3
Data Terminal Ready	4
Signal Ground	5
Data Set Ready	6
Request To Send	7
Clear To Send	8
Ring Indicador	9

#### **Señales de la RS-232.**

- **Request To Send (RTS)** Esta señal se envía de la computadora (DTE) al módem (DCE) para indicar que se quieren transmitir datos. Si el módem decide que esta OK, asiente por la línea CTS. Una vez la computadora prende la señal RTS, esperará que el módem asiente la línea CTS. Cuando la señal CTS es afirmado por el módem, la computadora empezará a transmitir datos.

- **Clear To Send (CTS)** Afirmado por el módem después de recibir la señal de RTS indica que la computadora puede transmitir.
- **Data Terminal Ready (DTR)** Esta línea de señal es afirmada por la computadora, e informa al módem que la computadora está lista para recibir datos.
- **Data Set Ready (DSR)** Esta línea de señal es afirmada por el módem en respuesta a una señal de DTR de la computadora. La computadora supervisa el estado de esta línea después de afirmar DTR para descubrir si el módem está encendido.
- **Receive Signal Line Detect (RSLD)** Esta línea de control es afirmada por el módem e informa a la computadora que se ha establecido una conexión física con otro módem. A veces se conoce como detector de portadora (CD). sería un error que una computadora transmita información a un módem si esta línea no está prendida, es decir si la conexión física no funciona.
- **Transmit Data (TD)** es la línea por donde el dato se transmite de un bit a la vez
- **Receive Data (RD)** es la línea por donde el dato se recibe de un bit a la vez.

#### **NORMA RS422/485**

La interfaz serial RS422 y la interfaz serial RS485 se diseñaron para la conexión física entre computadores y terminales directamente. Estos estándares tienen grandes ventajas con respecto a la norma RS232 como por ejemplo, la distancia de comunicación de hasta 1200mts., la velocidad de transmisión de hasta 10 Mbits/seg. y el número de elementos a conectarse, para la interfaz RS422 pueden conectarse un transmisor y hasta diez receptores en un modo de transmisión Full-

duplex, mientras que para la interfaz RS485 se pueden conectar simultáneamente hasta 32 transmisores/receptores en un sistema half-duplex, otra ventaja frente al sistema RS232 es que no requiere fuentes duales sino una fuente de alimentación de 5 voltios.

Ambas interfaces utilizan el método de medida diferencial, en la que utilizan dos líneas para la transmisión y dos para recepción, en cada par de conductores la segunda tiene un nivel de voltaje complementario al del primero, y el receptor responde a la diferencia de voltajes entre los dos conductores. Este tipo de líneas de transmisión se llaman balanceadas, y esto permite la eliminación de ruidos electrostáticos y electromagnéticos común en las dos líneas que se utilizan.

## **EL RUIDO**

Se designa con el nombre de *ruido* a toda señal no deseada que se encuentra superpuesta a una señal útil. De esta definición general resulta que hay un fuerte elemento subjetivo en la cuestión de determinar qué parte de la señal es útil y qué parte es ruido.

En el ámbito de las telecomunicaciones y de los dispositivos electrónicos, en general, se considera ruido a todas las perturbaciones eléctricas que interfieren sobre las señales transmitidas o procesadas.

También, de una forma general el ruido se asocia con la idea de un sonido molesto, bien por su incoherencia, por su volumen o por ambas cosas a la vez.

## **ORIGEN DEL RUIDO**

Los orígenes del ruido son múltiples, pudiendo citarse como más importantes los siguientes:

- La agitación térmica producida en las moléculas del material que forma los conductores y, sobre todo, en las resistencias, por el choque con los electrones en movimiento.
- El movimiento desordenado, en las válvulas termoiónicas y especialmente en los semiconductores, de los electrones y otros portadores de corriente, lo que les lleva a emplear más o menos tiempo en su recorrido de un



electrodo a otro. Este movimiento desordenado de los portadores de carga aumenta considerablemente con la temperatura.

- La naturaleza discreta de los portadores de corriente de los semiconductores (Ruido Shoot).
- La irradiación de los cuerpos negros es otro factor importante en el ruido de las comunicaciones por radio, ya que todos los objetos del universo, dependiendo de su temperatura, emiten energía en forma de ondas electromagnéticas.
- El ruido producido por fuentes tales como contactos defectuosos, artefactos eléctricos, radiación por ignición y alumbrado fluorescente, en general conocidas como señales parásitas.
- El ruido errático producido por fenómenos naturales tales como tormentas eléctricas con relámpagos y rayos, eclipses y otros disturbios en la atmósfera o fuera de ella como las manchas solares.

### **FACTOR DE RUIDO**

La magnitud del ruido generado por un dispositivo electrónico, por ejemplo un amplificador, se puede expresar mediante el denominado **factor de ruido (F)**, que es el resultado de dividir la relación señal/ruido en la entrada **(S/R)<sub>ent</sub>** por la relación señal/ruido en la salida **(S/R)<sub>sal</sub>**, cuando los valores de señal y ruido se expresan en números simples:

$$F = \frac{(S/R)_{ent}}{(S/R)_{sal}}$$

Sin embargo, como los valores de relación señal/ruido suelen expresarse en forma logarítmica, normalmente en decibelios, el factor de ruido en decibelios será, por tanto, la diferencia entre las relaciones S/R en la entrada y en la salida del elemento bajo prueba ya que:

$$10 \cdot \log F = 10 \log (S/R)_{ent} - 10 \log (S/R)_{sal}$$

El factor de ruido cuando se expresa en decibelios se suele llamar **figura de ruido**. El factor de ruido es un parámetro importante en los sistemas de

transmisión, ya que mientras el ruido externo nunca se podrá eliminar totalmente, la reducción del ruido generado por los equipos depende del cuidado de su diseño.

## **RUIDO DEL SISTEMA**

Idealmente, una señal que es digitalizada y entregada por un instrumento es la misma señal que es ingresada al instrumento. Frecuentemente se pueden atribuir las diferencias entre las señales de entrada y salida al ruido del sistema, el cual proviene de un diverso número de fuentes, incluyendo el medio ambiente y el instrumento mismo.

El diseño de un instrumento requiere que se comprenda como el ruido puede afectar la adquisición de datos, el diseño del hardware y el medio ambiente.

Si no se entienden las preocupaciones relativas tales como la interferencia electromagnética (EMI), manejo de la fuente de potencia, puesta a tierra, la configuración electrónica, etc., entonces no se puede diseñar un instrumento que sea exacto dentro de un medio ambiente eléctricamente ruidoso.

## **2.4 HIPÓTESIS**

La implementación de un sistema para la adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial permitirá la reducción de ruido al obtener información.

## **2.5 SEÑALAMIENTO DE VARIABLES**

### **2.5.1 VARIABLE INDEPENDIENTE**

La implementación de un sistema para la adquisición de datos

### **2.5.2 Variable dependiente**

La reducción de ruido al obtener información

## **CAPITULO III**

### **METODOLOGÍA**

#### **3.1 MODALIDAD BÁSICA DE INVESTIGACIÓN**

En el desarrollo del sistema para la adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial se utilizará el método científico, debido a que nos permite trabajar por etapas siguiendo un proceso lógico, ordenado y controlado; además es adecuado utilizar el método Inductivo que resuena en el estudio de lo General a lo Particular, ya que en el campo de la Ingeniería Electrónica abarca una extensa área de técnicas y aplicaciones, de ello hemos tomado una parte para aplicarla a este proyecto.

También la presente investigación se contextualizará en la modalidad de investigación de campo, debido a que los hechos serán estudiados en M&B AUTOMATIZACION, y el presente proyecto se desarrollará en las instalaciones de la empresa ya mencionada, lo cual será de gran ayuda para llegar a la implementación de nuestro sistema.

#### **3.2 TIPOS DE INVESTIGACIÓN**

La investigación abarcará el nivel exploratorio pues reconoce las variables que nos competen, el nivel descriptivo permitirá caracterizar la realidad investigada, el nivel correlacional deducirá el grado de relación entre las variables en estudio y finalmente el nivel explicativo detectará las causas de determinados comportamientos y canalizará la estructuración de propuestas de solución la problemática analizada.

Para el enfoque será la investigación cualitativa pues obtendrá información directa de los investigados, en virtud de los cuales será factible desarrollar un análisis crítico de los resultados y proponer alternativas de solución.

### **3.3 POBLACIÓN Y MUESTRA**

El presente proyecto se desarrollará en la Empresa M&B AUTOMATIZACION con la supervisión de diferentes profesionales en el área y abarcará todo lo referente al sistema para la adquisición de datos desde un microcontrolador esclavo hacia un microcontrolador maestro mediante comunicación serial.

### **3.4 TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN**

Las técnicas empleadas en la presente investigación serán: la recolección actual de los datos, estudio de los documentos relacionados a cada microcontrolador a utilizar.

La recolección de los datos se emplea para conocer el estado de funcionamiento de los elementos del sistema para la adquisición de datos.

El estudio de los documentos relacionados a cada microcontrolador se realizará para identificar cada una de las partes que conforman el sistema para la adquisición de datos y para poder tener una visión clara de los elementos que vamos a utilizar, en nuestro caso los microcontroladores.

### **3.5 RECOLECCIÓN DE INFORMACIÓN**

Para la recolección eficaz de la información de campo, se recurrirá a las siguientes estrategias:

- Elaboración de los instrumentos para una segura y eficaz implementación de un sistema para la adquisición de datos.

### **3.6 PROCESAMIENTO Y ANALISIS**

Una vez aplicados los instrumentos y analizada la validez, se procederá al análisis integral, enriquecido gracias a los elementos de juicio desprendidos del marco teórico, objetivos y variables de la investigación.

En esta sección se describen de manera completa y detallada los procedimientos, lineamientos y técnicas utilizados en la implementación del sistema de control para la adquisición de datos de un sistema maestro – esclavo utilizando microcontroladores mediante comunicación serial, buscando satisfacer los requerimientos establecidos en el capítulo II del presente trabajo especial de grado, en base a las exigencias de la Empresa M&B AUTOMATIZACION, adaptado a las necesidades de las diferentes industrias, teniendo como norte la innovación tecnológica.

#### **DESCRIPCIÓN GENERAL DEL SISTEMA:**

Se dispone de dos microcontroladores PIC16F877, un maestro y un esclavo. El maestro y el esclavo se comunican a través de una interfaz I2C. En el esclavo están conectadas ocho entradas digitales (un dip switch conectado para las pruebas necesarias pero se podría reemplazar por sensores digitales) y ocho salidas digitales (se visualizan mediante LEDs que de igual manera se los reemplazará por actuadores). Adicionalmente, en el esclavo se encuentra conectado un teclado matricial.

El maestro tiene una salida para manejar un relé y una entrada de un pulsante para desactivar al relé. El maestro se conecta por el puerto serial de un PC (se utiliza un MAX232 para convertir los niveles de voltaje) para visualizar las entradas y para manejar las salidas que se encuentran en el esclavo.

El maestro permanentemente se comunica (mediante I2C) con el esclavo a fin de obtener información de las entradas y también para manejar las salidas. Mediante el teclado se puede ingresar una clave de 4 dígitos. La clave correcta se encuentra

almacenada en la memoria EEPROM del maestro. Si la clave ingresada es correcta, el maestro activa el relé, el cual puede ser apagado mediante el pulsante.

Se implementó una aplicación para PC, la cual permite visualizar el estado de las entradas y manejar las salidas del esclavo. Adicionalmente se puede cambiar la clave y se lleva una cuenta de las veces que el relé ha sido activado (mediante el ingreso correcto de la clave).

Para una mejor comprensión vamos a ir detallando por pasos las diferentes etapas en la implementación del sistema para la adquisición de datos.

## **1. ELEMENTOS A UTILIZAR**

Brevemente describiremos los elementos que consideramos más importantes para el desarrollo de nuestro proyecto, pues necesitamos un estudio previo para seleccionarlos.

### **a) Microcontrolador PIC16F877**

Los microcontroladores PIC16F87X, poseen procesadores RISC, con arquitectura HARVARD, esta se caracteriza por la independencia entre la memoria de códigos y la memoria de datos.

El microcontrolador PIC16F877, posee una memoria de programa de 8192 words, memoria de datos EEPROM de 256 bytes, memoria RAM de 368 bytes y 33 pines de entrada y salida, los cuales se dividen en:

Puerto A trabaja a 6 bits

Puerto B trabaja a 8 bits

Puerto C trabaja a 8 bits

Puerto D trabaja a 8 bits

Puerto E trabaja a 3 bits

Además de ello posee ocho conversores Análogo – Digital A/D, una de las principales diferencias frente a los otros pic es la capacidad que posee, y por esta razón es el más utilizado en proyectos avanzados que requieren mayor número de entradas y/o salidas, como automatización de procesos industriales, alarmas

residenciales, etc. Una desventaja es que debe ser conectado con un cristal externo, ya que no tiene incorporado el mismo y necesita también en el Master Clear una resistencia Pull – Up, ya que no hay forma de deshabilitar esta opción.

Una de las razones porque elegimos utilizar este microcontrolador es por poseer el Módulo de comunicación serie síncrona (MSSP) y también el Transmisor/Receptor - Síncrono/Asíncrono serie (USART), entre otras características.

A continuación se describen brevemente estos módulos que conforman un microcontrolador de la familia PIC16F877 de MICROCHIP:

### ***Módulo de Comunicación Serie Síncrona: MSSP***

En los PIC16F87X, *Microchip* ha implantado en el silicio de sus chips el módulo MSSP (*Master Synchronous Serial Port*), que proporciona un excelente interfaz de comunicación de los microcontroladores con otros microcontroladores y diversos periféricos, entre los que destacan las memorias EEPROM serie, los conversores A/D, los controladores de displays, etc. De esta forma, el usuario dispone de algunas patitas de la puerta C, que correctamente programadas sirven para manejar directamente la comunicación serie. Además, el módulo MSSP admite dos de las alternativas más usadas en la comunicación serie síncrona.

1ª. SPI (Serial Peripheral Interface)

2ª. I2C (Inter Integrated Circuit)

*El módulo MSSP consta básicamente de dos registros: el SSPSR, que es un registro de desplazamiento que transforma la información serie en paralelo y viceversa, y el registro SSPBUF, que actúa como buffer de la información que se recibe o transmite.*

### ***El USART: Transmisor/Receptor – Síncrono/Asíncrono serie***

Los PIC16F87X contienen un módulo MSSP con dos puertas para la comunicación serie síncrona, o sea, con señal de reloj. Además, también disponen de un módulo USART capaz de soportar la comunicación serie síncrona y asíncrona.

El USART, llamado SCI (Serial Communications Interface), puede funcionar como un sistema de comunicación *full duplex* o bidireccional asíncrono, adaptándose a multitud de periféricos y dispositivos que transfieren información de esta forma, tales como el monitor CRT o el ordenador PC. También puede trabajar en modo síncrono unidireccional o *half duplex* para soportar periféricos como memorias, conversores, etc. En resumen, el USART puede trabajar de tres maneras:

- 1ª. ASINCRONA (Full duplex, bidireccional)
- 2ª. SINCRONA-MAESTRO (Half duplex)
- 3ª. SINCRONA-ESCLAVO (Half duplex, unidireccional)

#### **b) CI MAX 232**

El CI. MAX232 es la solución para transmitir a mayor distancia, ya que incrementa los niveles de voltaje de 5 V a  $\pm 10V$ , gracias a un juego de capacitores que le ayuda a doblar los voltajes, por lo que para su alimentación solo se requiere una fuente de 5V que puede ser la misma que utiliza el PIC. El MAX232 dispone de dos juegos de transmisores y receptores, de los cuales solo ocuparemos un par de ellos.

El MAX232 en este caso nos ayudará a convertir los voltajes TTL del PIC en voltajes de la norma RS232, quiere decir que si enviamos un estado lógico alto (5V), a la salida del Tout del CI MAX232 tendremos -10V, y si enviamos un 0 lógico desde el PIC (0V), el MAX232 enviará +10V.

## **2. ¿POR QUÉ UTILIZAMOS COMUNICACIÓN SERIAL?**

Existen dos formas de realizar una comunicación binaria, la paralela y la serial. La comunicación paralela, como por ejemplo la comunicación de un PIC con un circuito integrado 7447 (decodificador BCD), que utilizamos para manejar un display de siete segmentos, en donde los datos viajan simultáneamente a través de cuatro hilos, tiene la ventaja de que la transferencia de datos es más rápida, pero el inconveniente es que necesitamos un cable por cada bit de dato, lo que encarece y dificulta el diseño de las placas, otro inconveniente es la capacitancia que generan los conductores por lo que la transmisión se vuelve defectuosa a partir de unos pocos metros.



La comunicación serial en cambio es mucho más lenta debido a que trasmite bit por bit pero tiene la ventaja de necesitar menor cantidad de hilos, y además se puede extender la comunicación a mayor distancia, por ejemplo; en la norma RS232 a 15 mts., en la norma RS422/485 a 1200 mts., y utilizando un MODEM, pues a cualquier parte del mundo.

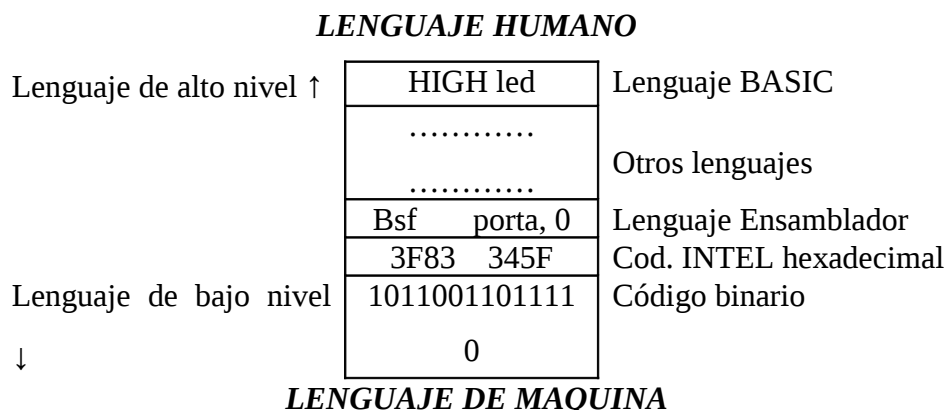
Existe dos formas de realizar la comunicación serial: la síncrona y la asíncrona, la diferencia entre estas dos formas de comunicación es que la comunicación sincrónica además de la línea para la transmisión de datos, necesita otra línea que contenga los pulsos de reloj, estos a su vez indican cuando un dato es válido. Por otra parte la comunicación serial asíncrona no necesita pulsos de reloj, en su lugar utiliza mecanismo como referencia tierra (RS232) o voltajes diferenciales (RS422/485), en donde la duración de cada bit es determinada por la velocidad de transmisión de datos que se debe definir previamente entre ambos equipos.

### 3. SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN

Este punto fue muy importante, pues para realizar la programación necesitamos un lenguaje de fácil comprensión y fácil de aprender.

Vamos a ver la diferencia entre el lenguaje ensamblador y el lenguaje Basic.

Para poder entender la diferencia entre los dos lenguajes de programación, debemos entender que es un lenguaje de alto nivel y que es un lenguaje de bajo nivel, a través del siguiente cuadro podemos ver los niveles de programación.



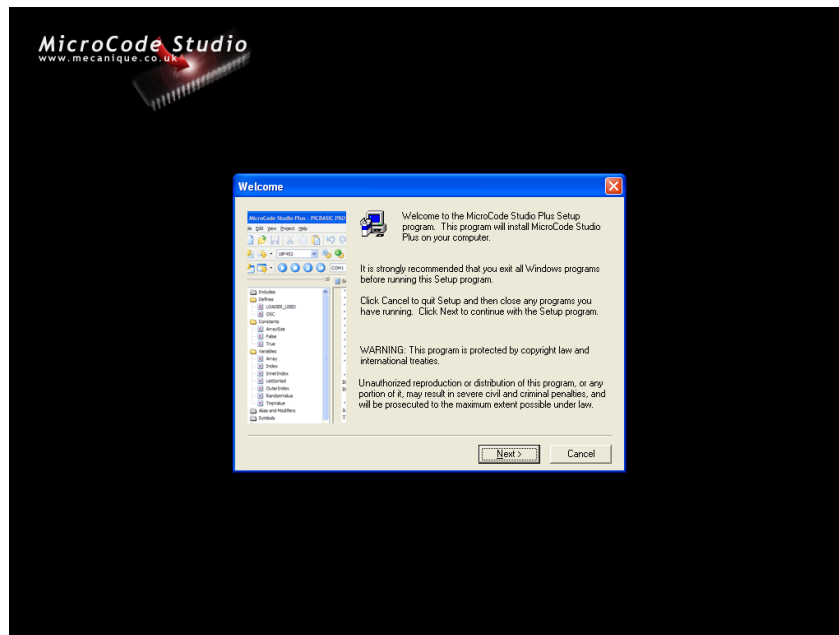
En el cuadro podemos ver los niveles de programación, el lenguaje que más se acerca a los humanos es el de más alto nivel. El lenguaje más próximo al tipo de datos que entiende el microcontrolador es el lenguaje de bajo nivel.

Por consiguiente vamos a trabajar con un programador de alto nivel, el que más entendemos los humanos.

## 4. INSTALACIÓN DEL SOFTWARE PARA PROGRAMACIÓN DE LOS PIC'S

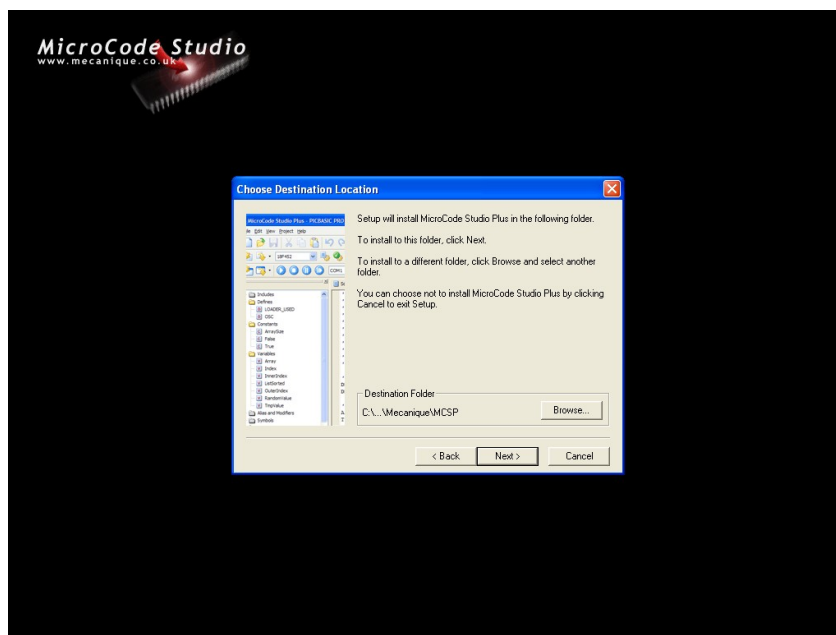
### 4.1 Instalación del Software MicroCode Studio

Para instalar nuestro editor debemos tener el CD de Microtechnica, una vez que lo tengamos entramos en la carpeta de MicroCodeStudioPlus y aquí buscamos el archivo SETUP.EXE y lo ejecutamos, veremos una nueva pantalla en la que presionamos next, luego yes.





A continuación nos indicará la carpeta en donde se instalará C:\Archivos de Programa\Mecanique, si deseamos lo dejamos ahí, pero es recomendable mejor instalar en C:\ directamente, presionamos Browse, cambiamos a C:\Mecanique\MCS y presionamos OK



Inmediatamente preguntará si quiere crear esta nueva ubicación presionamos Si, luego tendremos la pantalla destino de la instalación que debe decir C:\Mecanique\MCS, presionamos Next.

Al cabo de unos segundos la instalación habrá terminado, entonces presionamos Finish.

## **4.2 Instalación de PBP246 (PicBasic Pro versión 2.46)**

Este es nuestro compilador que nos facilitará las cosas ya que disponemos un programador de PIC, el IC-Prog 105C, también tenemos un editor de texto el MicrodeStudioPlus y solo nos falta un programa compilador que se encargará de generar los archivos .HEX, necesarios para poder grabar en un microcontrolador PIC.

Este programa debemos buscarlo de igual manera en el CD de Microtechnica, una vez localizado entramos en la carpeta PBP246 y ejecutamos el archivo SETUP.EXE,

## **5. DESARROLLO DE LOS PROGRAMAS**

### **5.1 Programas de los Microcontroladores**

Con los detalles técnicos y especificaciones que se relatan en la DESCRIPCION GENERAL DEL SISTEMA se procedió a realizar el programa para el microcontrolador esclavo con sus respectivas entradas y salidas, de igual manera se realizó el programa adecuado para el microcontrolador maestro.

#### **Descripción del Programa del Esclavo:**

El esclavo está ejecutando un lazo infinito que comprende las siguientes tareas:

Esperar a que el maestro inicie comunicación I2C para lectura o escritura del esclavo;

Barrer el teclado, para detectar si se presionó alguna tecla

Actualizar los valores de las entradas y los dígitos ingresados (por teclado),

En registros que se envían vía I2C al maestro.

Escribir en las salidas con los valores enviados por el maestro.

#### **Descripción del programa del maestro:**

El maestro está ejecutando un lazo infinito que comprende las siguientes tareas:

Comunicarse serialmente con el PC para establecer valores deseados de las salidas y enviar los valores de las entradas. El PC también puede enviar una nueva clave para almacenar en el maestro.

Comunicarse vía I2C con el esclavo para enviar valores de salidas deseados y para leer el estado de las entradas y la clave ingresada por teclado.

Activar el relé en caso de que la clave digitada (y enviada desde el esclavo) sea correcta.

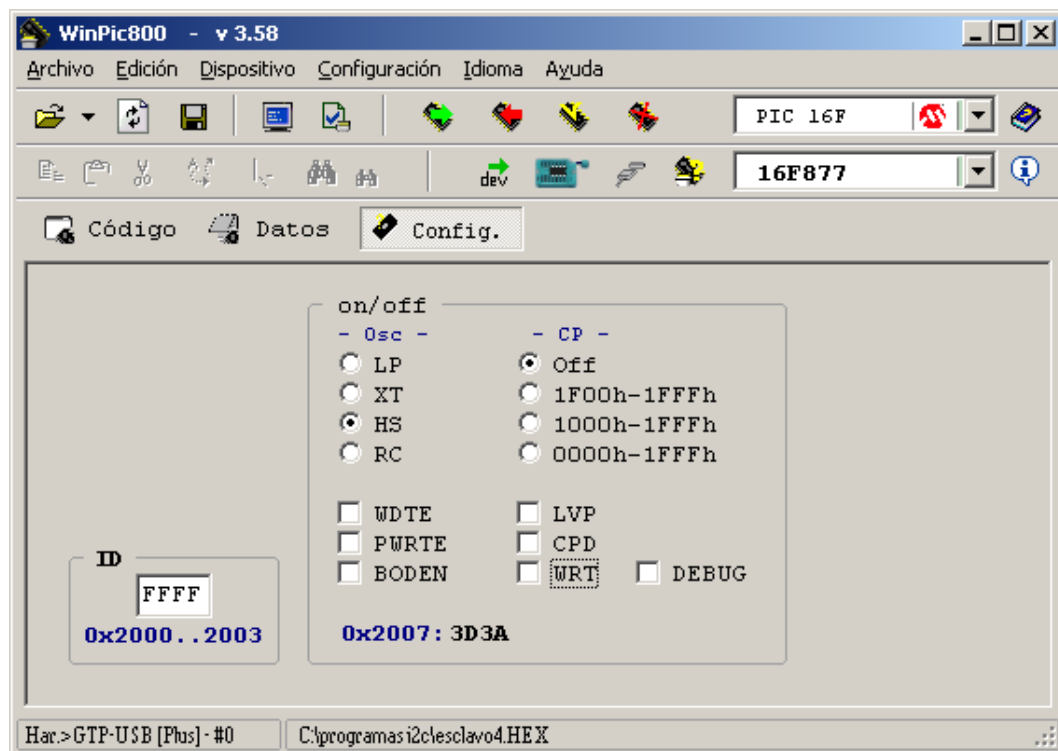
En caso de que se presione el pulsante, apagar el relé.

### Configuración de programación:

Una vez que los programas están listos, los compilamos para generar el archivo .HEX y procedemos a grabarlo, en el microcontrolador PIC.

Para grabar utilizamos el software programador WinPic800 versión 3.58

Abrimos el archivo a grabar y antes de hacerlo se debe configurar de la siguiente manera, luego de esto procedemos a grabar los microcontroladores.



## 5.2 Programa para el PC

Esta aplicación se la desarrolló en Visual Basic 6.0

La aplicación para PC inicia y mantiene permanente comunicación con el maestro, a fin de enviarle valores de salida deseados y leer las entradas del esclavo y el estado del relé. Adicionalmente, la aplicación permite enviar una nueva clave al maestro.



## 6. DISEÑO DEL CIRCUITO ELECTRONICO

El circuito electrónico lo vamos a detallar en tres secciones para mejor comprensión.

### 6.1 Esclavo

El microcontrolador PIC16F877 esclavo esta conectado de la siguiente manera:

- En la puerta A, bits del 0 al 5, puerta E bits 0 y 1, tenemos conectadas las ocho entradas
- En la puerta B conectamos el teclado hexadecimal. Los cuatro bits más bajos los usamos para conectar las filas y en los bits más altos conectamos las columnas.
- En la puerta D vamos a visualizar las ocho salidas
- En la puerta C, bit 0, conectamos un led que nos indicará cuando en el teclado se presionó una tecla, y se apagará una vez que se ha ingresado el cuarto dígito (independientemente si la clave es correcta o no)
- Los pines 13 y 14 son utilizados para conectar el oscilador.
- Los pines 18 y 23, son las líneas SDA (datos) y SCL (reloj) respectivamente, que utilizaremos para la comunicación I2C con el microcontrolador maestro

## 6.2 Maestro

El microcontrolador PIC16F877 maestro esta conectado de la siguiente manera:

- En el pin 40, que corresponde a la puerta B.7 conectamos el relé que será activado previo correcto ingreso de la clave
- En el pin 33, correspondiente a la puerta B.0 tenemos el pulsante que nos ayudará a desactivar el relé.
- Los pines 13 y 14 son utilizados para conectar el oscilador.
- Los pines 18 y 23, son las líneas SDA (datos) y SCL (reloj) respectivamente, que utilizaremos para la comunicación I2C con el microcontrolador esclavo
- En la puerta C: RC6/TX/CK y RC7/RX/DT (pines 26 y 27 respectivamente) es por donde vamos a realizar la comunicación con el computador (Transmisión y Recepción).

## 6.3 El Modulo de Comunicación

Es una interfaz entre el puerto serial del ordenador y el USART del microcontrolador maestro, el cual transforma los niveles lógicos del estándar RS – 232 a niveles lógicos TTL aceptados por la familia de microcontroladores PIC 16F877.

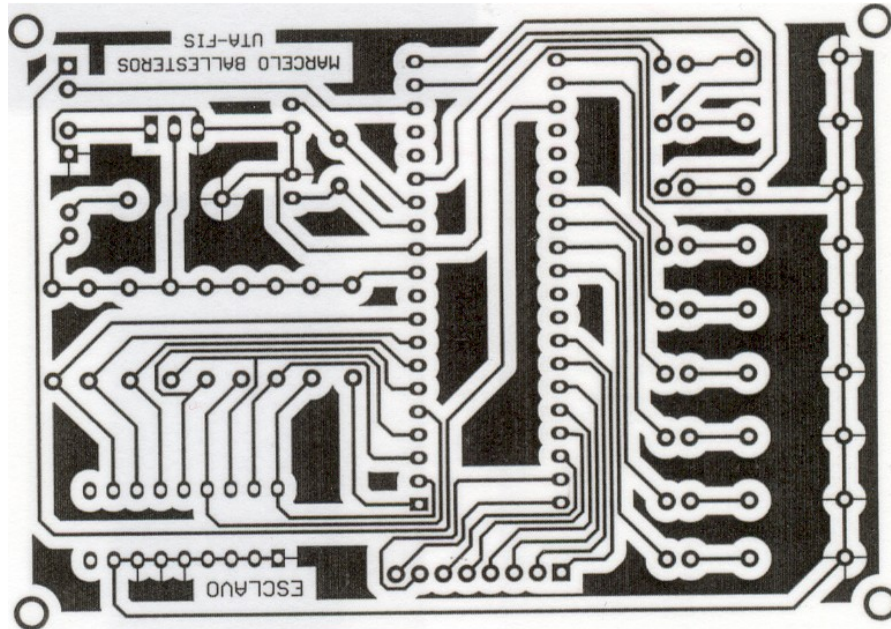
**Nota:** Ver los diagramas esquemáticos (Maestro, esclavo y módulo de comunicación) en la sección de anexos.

## 7. DISEÑO Y CONSTRUCCION DE LAS TARJETAS DE CIRCUITO IMPRESO

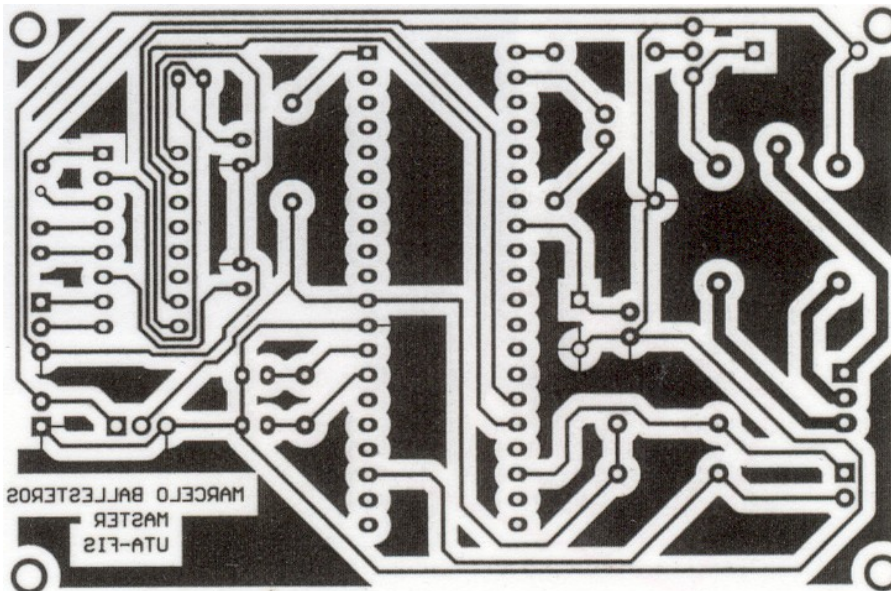
El diseño de las tarjetas de circuito impreso, se realiza en base a la selección de los componentes nombrados en la sección anterior, donde se hace referencia a cada uno de ellos.

Con la utilización del programa PROTEL se realizó el diseño de dos placas: una para el microcontrolador esclavo con sus respectivos elementos, y otra placa para el microcontrolador maestro de igual manera con sus respectivos elementos.

**Trazado para el esclavo**



**Trazado para el maestro**





## **8. PRUEBAS E IMPLEMENTACION**

Una vez concluida la fase del diseño del sistema para adquisición de datos se procedió a la etapa de pruebas y puesta en marcha, el cual comprendió la revisión del funcionamiento de los componentes electrónicos, de la programación de los microcontroladores y la comunicación con el ordenador, además de realizar los ajustes necesarios para la puesta a punto. Posteriormente se procedió a la implementación con la manufactura de las tarjetas de circuito impreso y la selección de los componentes más adecuados para el desarrollo del diseño.

### **PRUEBAS**

Las pruebas que se realizaron fueron necesarias para evaluar el comportamiento de los programas elaborados tanto para el ordenador como para los microcontroladores. Uno de los puntos más importantes consistió en la sincronización de la comunicación entre los tres elementos (PC, U1, U2), sobre todo por la velocidad de transferencia de los datos.

En este sentido, se realizó el montaje en protoboard del circuito electrónico, para la construcción y depuración de los programas diseñados, usando solo los componentes necesarios.

### **IMPLEMENTACION**

Una vez que se realizaron las respectivas pruebas y correcciones, con el ensamblaje en protoboard, procedimos a implementar los circuitos en las respectivas tarjetas diseñadas previamente.

## **CAPITULO IV**

### **ANALISIS E INTERPRETACION DE RESULTADOS**

Con las respectivas pruebas que se realizaron hemos podido determinar por ejemplo, la longitud del cable entre microcontrolador maestro y computador, dado que si no usamos el CI MAX232 la longitud de cable entre estos es mucho menor. Esto permitió además determinar el tipo de cable mas adecuado para tal fin.

Se puede decir que el sistema para adquisición de datos es un sistema confiable puesto que los datos obtenidos durante todas las pruebas realizadas han sido correctos, por eso podríamos decir con toda seguridad que el bus I2C lleva datos de una manera eficaz y segura hasta su destino.

La comunicación I2C se verificó, utilizando una rutina en el microcontrolador que nos indicará mediante el encendido de un led en caso de existir fallas en la comunicación.

De acuerdo a las pruebas realizadas se pudieron determinar los elementos adecuados para el correcto funcionamiento de nuestro sistema.

Los resultados obtenidos han sido exitosos ya que se cumplió con los requerimientos, esto es llevar datos, y esto aplicado a las necesidades de una industria nos facilitará el control de elementos remotos.

En los PIC16F877, el bus I2C está implementado en silicio tanto en el modo maestro como en el modo esclavo. Debemos recordar que en modo maestro es el dispositivo, el que inicia y finaliza la comunicación y genera los impulsos de reloj, también selecciona el esclavo al que se destina la información.

Como ya se ha comentado al hablar del bus I2C, en general, cada transferencia comienza con la condición de inicio (start) y termina con la condición de parada (stop).

Al empezar a manejar el bus I2C puede ser de mucho dolor de cabeza para el estudiante pero una vez que se logra controlarlo se puede verificar como los datos viajan de una manera eficaz y segura.

## **CAPITULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. CONCLUSIONES**

Con la elaboración del presente trabajo especial de grado se desarrolló un sistema para la adquisición de datos, adecuado a las necesidades de las diferentes industrias.

Tanto Philips como otros fabricantes de dispositivos compatibles con I2C disponen de una amplia gama de circuitos integrados, incluyendo memorias RAM y E<sup>2</sup>PROM, microcontroladores, puertos de E/S, codificadores DTMF, trancceptores IR, conversores A/D y D/A, relojes de tiempo real, calendarios, etc.

Dado que no siempre se requiere alta velocidad de transferencia de datos este bus es ideal para sistemas donde es necesario manejar información entre muchos dispositivos y, al mismo tiempo, se requiere poco espacio y líneas de circuito impreso. Por ello es común ver dispositivos I2C en video grabadoras, sistemas de seguridad, electrónica automotriz, televisores, equipos de sonido y muchas otras aplicaciones más.

Incluso, y gracias a que el protocolo es lo suficientemente simple, usualmente se ven dispositivos I2C insertados en sistemas microcontrolados que no fueron diseñados con puertos I2C, siendo el protocolo es generado por el firmware.

También hay dispositivos de adaptación que permiten conectar buses originalmente paralelos a sistemas I2C. Tal es el caso del chip PCD 8584 de Philips el cual incorpora bajo su encapsulado todo lo necesario para efectuar dicha tarea.

Hay, además, circuitos integrados cuya única misión es adaptar los niveles presentes en el bus I2C y convertirlos desde y hacia TTL, permitiendo resolver fácil y rápidamente la interconexión de dispositivos de dicha familia con el I2C.

## 5.2 RECOMENDACIONES

Primero vamos a dar unas recomendaciones básicas pero muy útiles a la hora de montar un proyecto con microcontroladores PIC

- Recuerde que el PIC tiene tecnología CMOS, esto quiere decir que consume muy poca corriente pero que a la vez es susceptible a daños por estática, por eso se recomienda utilizar pinzas para manipular y así poder transportar desde el grabador al protoboard o viceversa, o a su vez utilizar una manilla antiestática
- Procure utilizar un regulador de voltaje como el 7805 que nos entrega exactamente 5V y no un adaptador de pared, ya que el voltaje de salida no siempre es el mismo del que indica su fabricante, por último puede utilizar un circuito con un diodo zener de 5.1V
- No sobrepase los niveles de corriente, tanto de entrada como de salida, recuerde que el PIC puede entregar por cada uno de sus pines una corriente máxima de entrada de 25 mA. Así mismo soporta una corriente máxima de entrada de 25 mA., esto quiere decir que puede encender un led con una resistencia de 330  $\Omega$
- En algunos proyectos es necesario conectar un capacitor de 0,1  $\mu$ F en paralelo al PIC, este evita mal funcionamientos que podrían ocurrirle, en especial cuando utilizamos teclados matriciales y tenemos adicionalmente un buzzer activo (parlante activo o chicharra)

Ahora veamos algunas recomendaciones en caso de fallas con el bus I2C

- Ante un falla en el funcionamiento de alguno de los integrados conectados al bus y antes de probar cambiando CIs, tenemos que hacer algunas verificaciones. Como primera medida comprobamos el estado de las resistencias de pull up
- Luego con una punta lógica verificamos que cuando el bus está inactivo, las dos líneas se encuentren en estado alto.
- A continuación, chequeamos que el Master dirija alguno de los integrados, esta verificación también podemos hacerla con la punta lógica, pero tengamos en cuenta dos cosas: con la punta lógica solamente estamos detectando actividad en el bus, pero no podemos

saber si se lleva a cabo satisfactoriamente. En segundo lugar, si hay más de un integrado conectado al bus, no podremos determinar cuál de ellos está siendo solicitado.

- Para analizar más a fondo las comunicaciones y avanzar en el diagnóstico, tendremos que contar con un osciloscopio digital o construir un sencillo probador con unos pocos componentes.

Un caso especial son las memorias 24Cxx que trabajan con el protocolo I2C y son utilizadas en televisores, monitores, reproductores de DVD, etc.

Los microcontroladores como son re programables, permiten realizar modificaciones al sistema de control en un futuro.

## **CAPITULO VI**

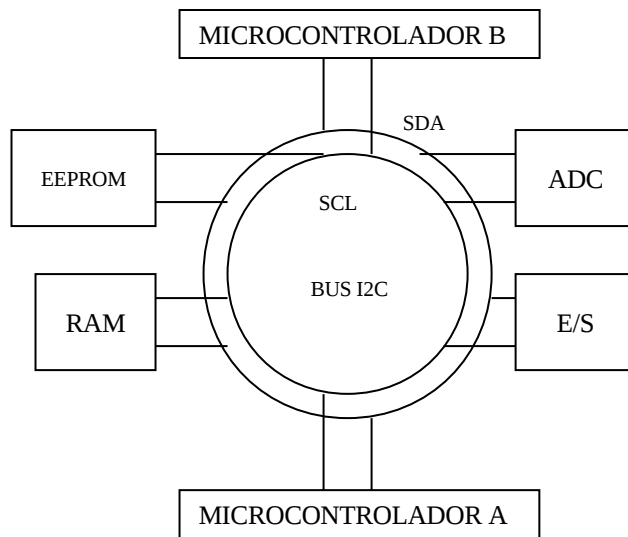
### **PROPUESTA**

Cuando empezamos a estudiar el desarrollo de este trabajo especial de grado principalmente nos propusimos estudiar los diferentes usos que se le pueden dar a los microcontroladores y de ahí poder aplicarlos para dar solución a las diferentes necesidades de las industrias. Pero una vez empezado los estudios previos pudimos ver que los microcontroladores los podemos tener desde aplicaciones muy sencillas en el hogar, estando presentes en un sencillo sistema de alarma o también controlando el encendido y apagado automático de luces, por ejemplo; como también los podemos tener en aplicaciones más complicadas como en la construcción de un PLC, proyectos con teclados que permitan mediante el ingreso de una clave activar una cerradura por ejemplo, y claro también sistema de comunicación entre un PC y PIC, entre PICs, etc.

El diseño de proyectos con microcontroladores también lo podemos extender en la industria agropecuaria para lo que se podría desarrollar un equipo capaz de efectuar mediciones de temperatura, humedad relativa hasta una distancia de x metros del mismo, como por ejemplo en un sistema de control de humedad, temperatura para invernaderos industriales y con empleo de microcontroladores podemos facilitar las tareas de los obreros, pues mediante software se puede reducir notablemente el hardware lo que nos ayudará inclusive a la implementación de los circuitos electrónicos.

Para futuras investigaciones se propone el estudio del BUS I2C en sus otros modos de operación.

Sería recomendable revisar al bus I2C trabajando en un sistema *multi-maestro*; es decir un sistema en el que puede haber más de un maestro conectado y controlando el bus. Normalmente se trata de microcontroladores o microcomputadoras.



Supongamos que un momento dado el microcontrolador A actúa como maestro, esta configuración puede no ser permanente. En otro momento, puede ser el B quien actué de maestro. Todo depende de la dirección del dato a transferir en un momento dado.

Dada la posibilidad de que existieran varios maestros, podría ocurrir que más de uno inicie la transferencia en un mismo instante de tiempo. Para evitar el caos que pueda producir esta situación, se ha desarrollado un sistema de arbitraje de Bus, el mismo que se recomienda estudiarlo más a fondo para posteriores investigaciones.

De igual manera todos los requerimientos tanto para software y hardware van a ser los que hemos utilizado para el presente proyecto, si bien es cierto que para la parte de software se deberá realizar las configuraciones necesarias para trabajar en el modo multi-maestro, en la parte de hardware en cambio simplemente aumentaran el número de elementos maestros que se van a utilizar como también los dispositivos esclavos a controlar (pueden ser memorias, ADC, dispositivos de entrada/salida, etc.)

## **BIBLIOGRAFÍA**

ANGULO, José María (2006). Microcontroladores PIC Diseño práctico de aplicaciones 2ª Parte. Segunda Ed.

CORRALES, Santiago (2006). Electrónica Practica con Microcontroladores PIC. Imprenta Gráfica. Ecuador

REYES, Carlos (2004). Aprenda rápidamente a programar microcontroladores PIC. Gráficas Ayerve. Ecuador

MICROENGINEERING Labs, Inc.(2005). PicBasic Pro™ Compiler

## **Internet**

[www.i-micro.com/pdf/articulos/spi.pdf](http://www.i-micro.com/pdf/articulos/spi.pdf)

<http://www.fi.uba.ar/laboratorios/lie/Publicaciones/100.pdf#search=%22datos%20desde%20un%20microcontrolador%20esclavo%20hacia%20un%20maestro%22>

<http://www.fi.uba.ar/laboratorios/lisi/JISIC-2004-metodologia-diseno-sistemas-tutoriales-inteligentes.pdf#search=%22datos%20desde%20un%20microcontrolador%20esclavo%20hacia%20un%20maestro%22>

<http://atc.inf-cr.uclm.es/Asignaturas/Grado/ECo/lab/Practicas/i2c.pdf#search=%22datos%20desde%20un%20microcontrolador%20esclavo%20hacia%20un%20maestro%22>

<http://www.olimex.cl/tutorial5.pdf?osCsid=dd9075550003fd93552c75517c3fb95e#search=%22comunicacion%20serial%22>

<http://www.prodigyweb.net.mx/saucedo8/sci.pdf#search=%22comunicacion%20serial%22>

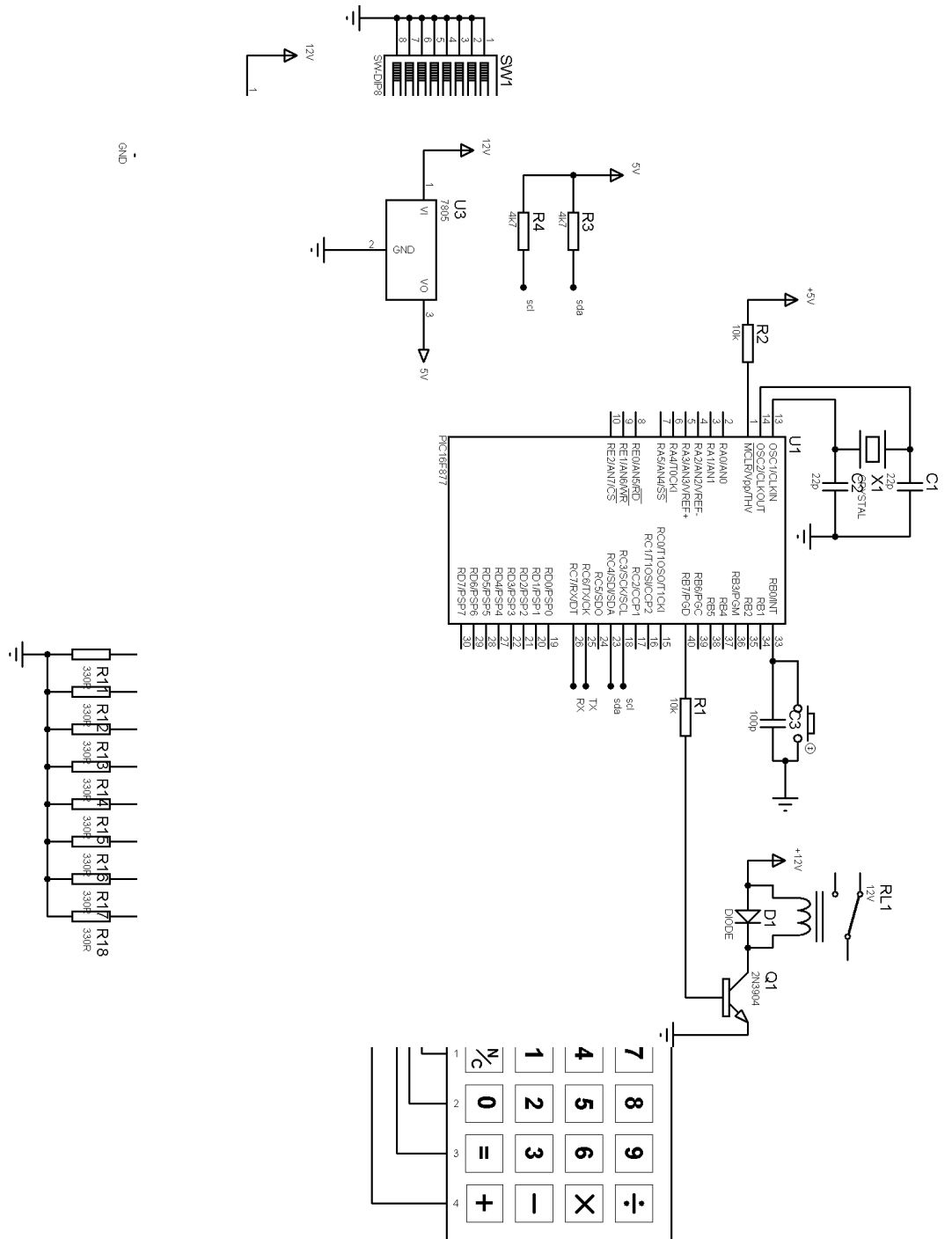
[http://es.wikipedia.org/wiki/Ruido\\_\(f%C3%ADsica\)](http://es.wikipedia.org/wiki/Ruido_(f%C3%ADsica))



**PRIMERA SECCION**

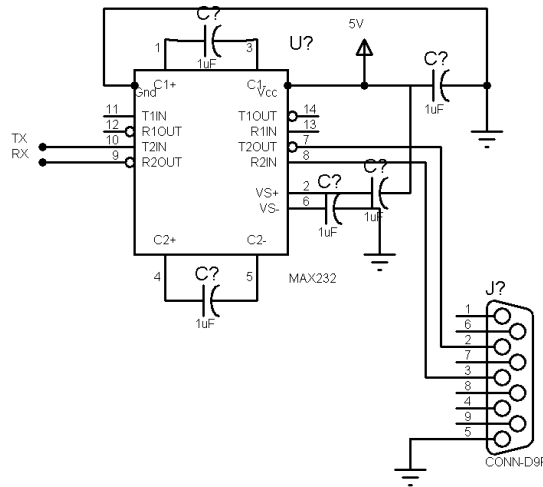
**DIAGRAMAS ESQUEMÁTICOS PARA EL SISTEMA DE ADQUISICIÓN DE DATOS**

**Esclavo**



**Maestro**

## Módulo de Comunicación



## SEGUNDA SECCION

### EL BUS I2C Y LOS PIC16F87X

En los PIC16F87X, el bus I2C está implementado en silicio tanto en el modo maestro (master) como en el esclavo (slave).

Recuérdese que en modo maestro es el dispositivo, en este caso el PIC16F87X, el que inicia y finaliza la transferencia y genera los impulsos de reloj. También selecciona el esclavo al que se destina la información. (Figura A1.)

Como ya se ha comentado al hablar del bus I2C, en general, cada transferencia comienza con la condición de inicio (start) y termina con la condición de parada (stop). Ambas condiciones las genera el maestro, y la primera consiste en un flanco descendente en SDA mientras SCL tiene nivel alto. La condición de parada es un flanco ascendente mientras SCL tiene nivel alto.

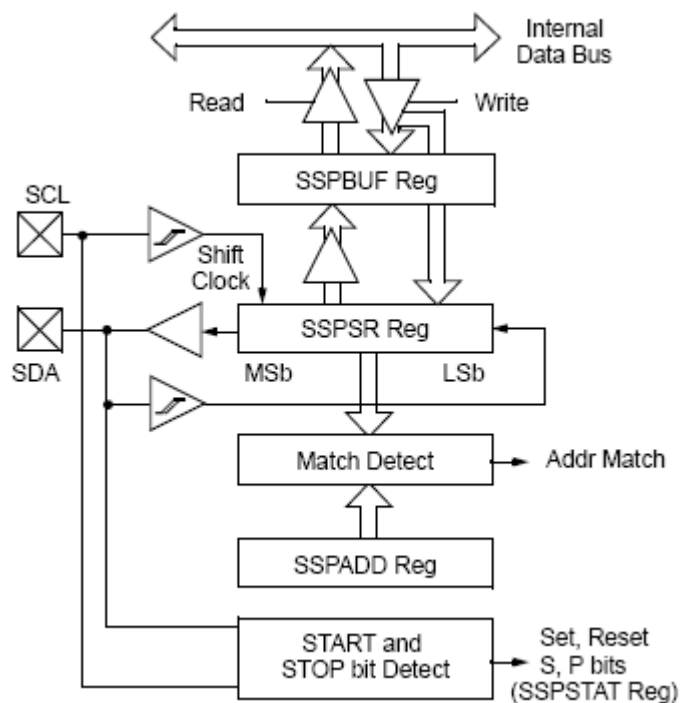
Los datos que se colocan sobre la línea SDA son bytes de 8 bits, que comienzan con el MSB y terminan con el bit de menos peso, al cual sigue en el noveno bit la condición de Reconocimiento (ACK). Para este caso, el transmisor pone SDA = 1, mientras que el receptor pone SDA = 0 en dicho impulso de reloj, prevaleciendo sobre la línea el nivel bajo.

El primer byte que envía el maestro tras la condición de inicio contiene la dirección del esclavo con el que se desea realizar la comunicación. El código 0 se usa para realizar una «llamada general» sobre todos los esclavos. La dirección en realidad consta de 7 bits, estando destinado el octavo a indicar la operación a

realizar (R/W#: Lectura/Escritura#). Tras este byte inicial de direccionamiento se manda otro byte que especifica las características de la operación a realizar.

Para activar el bus I2C hay que poner el bit SSPEN = 1 que es el bit 5 del registro SSPCON (SSPCON<5>». A partir de ese momento las patitas SDA y SCL quedan configuradas para soportar el protocolo I2C. Previamente dichas patitas de la Puerta C (RC3/SCL) y RC4/SDA) han de estar configuradas como entradas mediante la escritura de los bits correspondientes del registro TRISC.

Como se aprecia en la Figura A1, hay seis registros para controlar el bus I2C.



**Figura A1.** Estructura general de los registros que controlan la transferencia por las patitas SDA y SCL en el modo maestro del bus I2C

1. SSPCON: registro de control
2. SSPCON: registro de control 2
3. SSPSTAT: registro de estado
4. SSPBUF: buffer para los datos
5. SSPSR: registro de desplazamiento no accesible directamente
6. SSPADD: registro de dirección

Mediante el bit CKE se ponen a nivel alto las patitas SDA y SCL en el modo maestro y esclavo. Cuando CKE = 0 los niveles de dichas patitas se configuran según las especificaciones del bus I2C.

Los 4 bits de menos peso del registro SSPCON sirven para seleccionar el modo de trabajo entre estos posibles:

1. Modo I2C maestro.  $\text{Relej} = (\text{SPADD} + j) * \text{OSC} / 4$
2. Modo I2C esclavo con 7 bits para la dirección
3. Modo I2C esclavo con 10 bits para la dirección

El registro de Estado SSPSTAT contiene la información que representa el estado de la transferencia de datos. Detecta las condiciones de inicio y parada, así como la recepción del byte de la dirección.

SSPBUF es el registro que actúa como buffer y soporta el dato que se va a transmitir o que se ha recibido. El registro de desplazamiento SSPSR realiza la conversión serie/paralelo y viceversa junto al SSPBUF. Cuando se ha completado la llegada del byte, que se recibe en el registro SSPSR, se traspa a SSPBUF y se activa el señalizador SSPIF. Si se recibe otro byte antes de haber leído al SSPBUF, se activa el señalizador de desbordamiento SSPOV (SSPCON<6>).

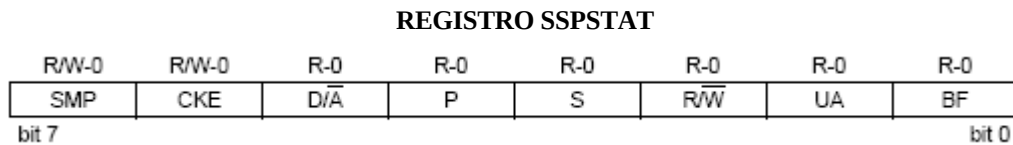
El registro SSPADD guarda la dirección del esclavo. Cuando se emplean direcciones de 10 bits, el usuario debe escribir el byte alto de la dirección con el siguiente código donde A9:A8 son los dos bits de más peso: 1-1-1-1-0-A9-A8-0.

En la Figura A2 se muestra la distribución de los bits del registro SSPSTAT, que ocupa la dirección 94h de los registros específicos de la memoria RAM.

El bit SMP es SSPSTAT en el modo maestro es un bit de muestreo que vale 1 cuando los bits de datos se muestrean al final del período, ó vale 0 cuando se hace en la mitad del período. El bit CKE selecciona los niveles de las patitas SCL y SDA en el modo maestro o multi-maestro. El bit D/A# (Dato/Dirección#) indica si el dato recibido es de información (1) o es una dirección (0). La llegada del bit o condición de Stop la detecta el bit P, que se pone a 1. El bit S detecta la condición de Inicio o Start. R/W# indica si se trata de una Lectura/Escritura#. El bit UA cuando vale 1 indica que la dirección es de 10 bits y que hay que cargar el byte alto, según la codificación antes expresada, en el registro SSPADD. Si es 0 la

dirección es de 7 bits. Finalmente, el bit BF actúa como señalizador del buffer de datos. Si BF = 1 indica que tiene un dato y la transmisión está en progreso sin terminar.

En la Figura A3 se muestra la estructura del registro de control SSPCON, que ocupa la dirección 14h del mapa de memoria.



**Figura A2.** Distribución de los bits del registro de Estado SSPSTAT. Todos los bits son leíbles, siendo sólo escribibles los dos de más peso. Tras un reset todos se ponen a 0



**Figura A3.** Registro de control SSPCON. Todos sus bits son leíbles y escribibles y se ponen a 0 tras un reset

El bit WCOL es un detector de colisiones en escritura y cuando vale 1 significa que se ha intentado escribir en SSPBUF en condiciones no válidas para el protocolo. El bit SSPON si vale 1 indica desbordamiento, o sea, que llega un byte a SSPBUF sin haberse leído el anterior. Si SSPEN = 1 la Puerta Serie queda configurada con las patitas SCL y SDA. Si SSPEN = 0 las patitas RC3/SCL y RC4/SDA funcionan como líneas de E/S digitales. El bit CKP sirve para activar el reloj en el modo esclavo, no usándose en el modo maestro. Mediante los 4 bits de menos peso de SSPCON se selecciona la frecuencia de reloj. Se indican los códigos más interesantes para el modo maestro:

<b>SSPCON3-0</b>	<b>FRECUENCIA DE RELOJ</b>
0000	Reloj = Fosc/4

0001	Reloj = Fosc/16
0010	Reloj = Fosc/61
0011	Reloj = Salida del TMR2/2
1000	Reloj = (SSPADD+1)*Fosc/4

En la Figura A4 se muestra la distribución de los bits del registro de control 2 SSPCON2. El bit GCEN sólo se usa en el modo esclavo. Cuando se pone el bit ACKSTAT = 1 significa que se ha recibido el bit de reconocimiento ACK del esclavo. ACKDT es el bit de reconocimiento en el modo maestro en recepción. Si ACKDT = 0 el maestro ha transmitido el bit de Reconocimiento, pero si vale 1 no lo ha hecho. Cuando se pone ACKEN = 1 se inicia la secuencia de generación de la condición de reconocimiento. Este bit se borra automáticamente por hardware. Para habilitar el modo de recepción del maestro hay que poner el bit RCEN = 1. Para generar la condición de parada en las líneas SCL y SDA hay que poner PEN= 1. El bit RSEN cuando se pone a 1 inicia la repetición de la condición de inicio. Finalmente, para iniciar la condición de inicio hay que poner SEN = 1.

**REGISTRO SSPCON2**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

**Figura A4.** Registro de control 2 SSPCON2. Todos sus bits son leíbles y escribibles y quedan a 0 tras un Reset.

**CHULETA – RESUMEN DE LOS REGISTROS USADOS PARA EL BUS I2C**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	—	(2)	—	EEIF	BCLIF	—	—	CCP2IF	-x-0 0--0	-x-0 0--0
8Dh	PIE2	—	(2)	—	EEIE	BCLIE	—	—	CCP2IE	-x-0 0--0	-x-0 0--0
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
93h	SSPADD	I <sup>2</sup> C Slave Address/Master Baud Rate Register								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in I<sup>2</sup>C mode.

Note 1: These bits are reserved on PIC16F873/876 devices; always maintain these bits clear.

2: These bits are reserved on these devices; always maintain these bits clear.

**Figura A5.** *Tabla que recoge los principales registros que intervienen en el control del módulo MSSP con el protocolo bus I2C*