

Universidad Tecnológica de El Salvador

FACULTAD: INFORMATICA Y CIENCIAS APLICADAS

CARRERA: TÉCNICO EN INGENIERÍA DE HARDWARE



TEMA:

**DISEÑO Y CONSTRUCCIÓN DE UNA PANTALLA LUMINOSA DIGITAL
PARA EL DESPLIEGUE DE MENSAJES ALFANUMÉRICOS, COMO APOYO
VISUAL E INFORMATIVO EN OFICINAS, PASILLOS Y/O AULAS DE LA
UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR**

TRABAJO DE GRADUACION PRESENTADO POR:

DIANA CAROLINA HERNANDEZ REYES

MARIA CONCEPCION RIVAS RODRIGUEZ

PARA OPTAR AL GRADO DE:

TECNICO EN: INGENIERIA DE HARDWARE

FECHA DE GRADUACION:

SEPTIEMBRE DE 2010

SAN SALVADOR, EL SALVADOR, CENTROAMERICA

PAGINA DE AUTORIDADES

**LIC. JOSE MAURICIO LOUCEL
RECTOR**

**ING. NELSON ZÁRATE SÁNCHEZ
VICERRECTOR ACADEMICO**

**LIC. LISETT CRISTALINA CANALES DE RAMIREZ
DECANA**

JURADO EXAMINADOR

**ING. HERBERT ISRAEL CARDONA
PRESIDENTE (A)**

**ING. JOSE MAURICIO RIVERA MEJIA
PRIMER VOCAL**

**ING. OMAR OTONIEL FLORES CORTEZ
SEGUNDO VOCAL**

**SEPTIEMBRE DE 2010
FECHA DE GRADUACION**

SAN SALVADOR, EL SALVADOR, CENTOAMERICA

AGRADECIMIENTO

En primer lugar, le doy **GRACIAS A DIOS TODOPODEROSO** por haberme iluminado y darme la fortaleza y sabiduría con las cuales pude superar cada una de las malas circunstancias en el desarrollo de este trabajo de graduación. Gracias te doy Señor por permitirme gozar de tu respaldo incondicional como el de ningún otro me le ha brindado.

AGRADEZCO A MIS PADRES; que estuvieron presentes en cada una de las etapas que se fueron presentando a lo largo de este trabajo, por darme el apoyo que necesite para culminar este sueño, y por esa confianza que depositaron en mi, para no desmayar cuando los tiempos eran difíciles; Gracias...

ASI COMO A TODA MI FAMILIA; que también forma parte de este logro que con cada uno de sus consejos y motivaciones lo hicieron posible.

AGRADEZCO A NUESTRO ASESOR EL ING. OTONIEL FLORES; por vivir juntamente con nosotros cada uno de los éxitos y fracasos que transcurrieron en este trabajo, por brindarnos un poco de sus grandes conocimientos y sobre todo su apoyo incondicional y disponibilidad para ayudarnos. Gracias...

A MI COMPAÑERA DE GRUPO; en la que pude descubrir una gran persona y sobre todo una gran amiga, con grandes virtudes y deseos de seguir adelante. Por la tolerancia y conocimientos que en su momento intercambiamos para culminar exitosamente este trabajo de graduación.

Y... a todas aquellas personas que de una u otra forma nos ayudaron e hicieron posible que mi compañera y yo estuviésemos realizando este trabajo de graduación con mucho orgullo y éxito.

DIANA CAROLINA HERNANDEZ REYES

AGRADECIMIENTO

Agradezco en primer lugar a mi Señor Jesucristo, por haberme prestado la vida, la salud y abrirme una puerta para obtener la ayuda económica. Le agradezco a i Señor por guardarme día a día en mi camino y por darme la sensatez necesaria para no equivocarme en las decisiones tomadas. Le doy gracias por permitirme estudiar y dar por terminada mi carrera, ahora en este tiempo le agradezco por que sin El nada puedo alcanzar.

Agradezco también a mi familia a mis padres y hermanos que de una y de otra forma me han ayudado y sobre todo me han apoyado, les doy las gracias por la paciencia que me han tenido y lo que les puedo decir es que mi señor les bendiga y les guíe en su andar.

Digo gracias a mi compañera Diana Reyes por acompañarme en este corto, pero difícil camino, le doy las gracias porque, aunque hubo momentos difíciles juntas salimos adelante, y al terminar esta etapa le digo que siempre puede contar conmigo, gracias compañera y amiga.

Doy las gracias a los docentes que me han acompañado durante este trayecto, por la paciencia y la dedicación en su labor, así también agradezco al asesor Ing. Otoniel Flores por su ayuda en el desarrollo del proyecto final de graduación y a los jurados que n os evaluaron en el momento de la defensa del proyecto, y en general a la Universidad por abriarnos las puertas y conducirnos a camino laboral; a todos ellos muchas gracias.

MARIA CONCEPCION RIVAS RODRIGUEZ

INDICE

No. De Página

INTRODUCCION.....	i
-------------------	---

CAPITULO I

SITUACION ACTUAL

1.1 Situación problemática.....	1
1.2 Enunciado del problema.....	2
1.3 Justificación.....	3
1.4 Objetivos.....	5
1.5 Alcances.....	6
1.6.- Estudio de factibilidad.....	7
1.6.1 Estudio económico.....	8
1.6.2 Estudio técnico.....	11
1.7 Carta de aceptación.....	16

CAPITULO II

MARCO TEÓRICO DE REFERENCIA

2.1. Marco teórico del problema.....	17
2.1.2 Que es un sistema electrónico embebido.....	17
2.1.3 Características principales de un sistema electrónico embebido.....	19
2.1.4 Estructura básica de un sistema electrónico embebido.....	23
2.1.5 Áreas de aplicación de los sistemas electrónicos embebidos.....	27
2.1.6 Componentes principales del proyecto a desarrollar.....	30
2.2 Marco teórico de solución.....	33

2.2.1 Esquema general en bloque de una pantalla luminosa digital.....	34
2.3 Marco teórico conceptual.....	36

CAPITULO III

DESARROLLO DE LA SOLUCIÓN

3.- Propuesta de la Solución.....	41
Conclusiones.....	50
Recomendaciones.....	51
ANEXOS.....	52
Referencias.....	90

INTRODUCCION

En el primer capítulo de este documento se realiza la formulación del proyecto a desarrollar, se plantean los ítems que describen la necesidad de diseñar y construir un prototipo electrónico de una **pantalla luminosa digital** como proyecto de graduación.

Para iniciar el documento se plantea la situación actual de la Escuela de Informática de la Facultad de Informática y Ciencias Aplicadas de la Universidad Tecnológica de El Salvador, con respecto a la forma en la que la universidad publica sus fechas y horarios de las actividades más importantes a realizar con relación al alumnado, para permitirles su participación como aporte a su desarrollo académico.

Al adentrarse en este tema tiende a generarse una situación problemática, ya que el método que se emplea no es el más apropiado. Pero además de describir el problema, también se da una posible solución, la cual se justifica como una forma muy viable de resolver dicho problema. Y para resolverlo es necesario plantearse objetivos y alcances que garanticen que, si se cumplen, el problema será resuelto. Además, el documento describe que tan factible es la solución que se plantea desde el punto de vista económico y técnico, que tan apropiado es el proyecto que se propone desarrollar y que beneficios se obtendrán al aprobarlo y desarrollarlo.

En el segundo capítulo se describen el área en general en la que está enmarcado el proyecto, se describe cada una de las partes lógicas y físicas que lo componen y el funcionamiento de las mismas. Además, se muestran conceptos básicos y fáciles de comprender para el usuario.

En el marco teórico del problema se describe el área en la que se clasifica un sistema embebido y las partes principales que lo componen, así como también los tipos de sistemas embebidos y sus características. En el marco teórico de solución se muestra el esquema electrónico en bloques del proyecto ha desarrollar, cada una de sus partes y su función lo que en conjunto hace posible que la pantalla luminosa digital funcione. Y en el marco teórico conceptual se describen los conceptos básicos de todo lo descrito en el marco teórico.

Y para finalizar este proyecto se concluye con la descripción general de todo el proyecto, que es, como se ha realizado, etc. Preguntas que se van respondiendo a medida se le da lectura al capítulo tres

Para dar por terminado el diseño y contracción de un prototipo de una pantalla luminosa digital.

CAPITULO I: SITUACIÓN ACTUAL

1.1.- Situación problemática

Dentro del quehacer del estudiante universitario dentro de su alma mater le es necesario estar informado sobre avisos de las actividades que la universidad realiza, especialmente a los estudiantes de Técnico en Ingeniería de Hardware de la Escuela de Informática en la Facultad de Informática y Ciencias Aplicadas de la Universidad Tecnológica de El Salvador, con respecto a las actividades educativas o cambios que se efectúan dentro de su área de estudios.

Al realizar una indagación sobre este tema dentro de las instalaciones de la universidad es posible observar que en la actualidad la Universidad Tecnológica de El Salvador no cuenta con una forma efectiva de anunciarse, capaz de atraer la atención del estudiantado, un medio que les brinde la información necesaria y adecuada de forma breve, de tal forma que les mantenga informados de las actividades y/o cambios que se puedan realizar dentro de su área de estudios.

Este problema es muy evidente en la escuela de informática de la universidad, ya que en muchas ocasiones los estudiantes no tienen acceso a las fechas y horas en las que se realizarán las actividades, tales como: fechas de pagos, cambios de aulas, actividades que realiza la universidad como seminarios, cursos libres o diplomados, las prácticas en los laboratorios, etc.

Esta falta de un método que permita anunciar de forma segura y accesible se vuelve una gran necesidad dentro de la universidad. Y este problema se da

debido a que actualmente el medio que la escuela de informática está implementando no es el mejor, porque no logran captar la atención del alumno, esto debido a que muchas veces resulta tedioso leer un cartel o un volante pegado en una rota folio o en un pizarrón, y en ocasiones la información no está al alcance de todos, o quizá muchos no tengan acceso a Internet a cualquier hora del día para informarse.

Por todas estas situaciones es necesario que de alguna forma se dé una pronta solución a esta problemática, para que el estudiante se mantenga bien informado de lo que acontece en su universidad, para que pueda ser puntual en sus responsabilidades y horarios correspondientes. Por ello es necesario buscar una alternativa que logre resolver este problema y pueda mantener de forma visible la información más importante y de forma actualizada.

1.2.- Enunciado del problema

¿Cómo logrará la escuela de informática en la facultad de informática y ciencias aplicadas de la Universidad Tecnológica de El Salvador obtener una herramienta moderna que les permita proporcionar mensajes informativos en oficinas y pasillos de forma práctica y actualizada, que logre la atención visual de los estudiantes?

1.3.- Justificación

Es necesario e indispensable que la escuela de informática de la universidad tenga acceso a posibles soluciones para sus problemas.

Por este motivo en esta oportunidad el proyecto propuesto ofrece una muy viable solución a sus problemas, una opción muy oportuna y adecuada que les permita mostrar la información más importante a los estudiantes de forma práctica y actualizada,

Debido a esto el proyecto propuesto trata de un prototipo electrónico de una **pantalla luminosa digital** para el despliegue de mensajes alfanuméricos, es decir que en la pantalla se podrán publicar mensajes cortos de forma clara y específica proporcionando así apoyo visual informativo en oficinas y/o pasillos de la escuela de informática.

Éste innovador proyecto le facilitará el trabajo a la hora de publicar información.

Algunos de los beneficios que la universidad obtendrá al aceptar esta propuesta son:

1. Se dispondrá de un prototipo electrónico moderno de calidad, diseñado y fabricado en la universidad, el cual mostrará un ambiente limpio y ordenado dentro de las instalaciones.
2. Los componentes electrónicos a utilizar son de calidad y a un bajo costo por lo que será un equipo durable y además integrará una fuente de

alimentación eléctrica que se conectará a 110 voltios, esto genera un bajo consumo de corriente, por lo que el costo económico será menor al costo en papelerías.

3. El uso del dispositivo será fácil y práctico, sólo será necesario modificar las líneas en el firmware en las que se indicará el o los mensajes a desplegar.

Todas estas bondades se obtienen con la pantalla luminosa digital que se propone en este proyecto. Además, su uso es fácil y práctico, ya que es reprogramable vía software. Esto es posible usando un Software gratuito como lo es el IDE BASCOM AVR, este software posee un editor y compilador que convierte el texto escrito en lenguaje Basic a lenguaje de maquina para que pueda ser entendido e interpretado por el microcontrolador y así el circuito electrónico cumpla con las funciones requeridas.

Para hacer los cambios, con respecto a los textos que se publicarán basta con cambiar una línea del código del programa que ya incluirá el prototipo, en esta línea se definirá el número de caracteres que compondrán los anuncios, y cuantos se publicarán continuamente (esto dependerá de la memoria del microcontrolador que se use), garantizando una muy bonita apariencia que cautive la atención de los estudiantes.

1.4.- Objetivos

Objetivo General:

Desarrollar un prototipo electrónico que consiste en una pantalla luminosa digital que permita desplegar mensajes alfanuméricos, como apoyo visual informativo en oficinas y/o pasillos de la Universidad Tecnológica de El Salvador.

Objetivos Específicos:

- Diseñar y construir un prototipo electrónico moderno de calidad y que sea de bonita apariencia.
- Utilizar componentes electrónicos de calidad y durabilidad, que sean accesibles económicamente y que su consumo eléctrico sea mínimo, generando así un bajo costo económico.
- Elaborar un código de programación en lenguaje BASIC que al descargarlo a la memoria del dispositivo éste despliegue los mensajes indicados.

1.5.- Alcances

Para la realización de éste proyecto se han propuesto determinados alcances, los cuales se convierten en promesas que se proponen alcanzar como proyecto, y además se describen los productos, que son los resultados finales que se alcanzarán con este prototipo electrónico, los cuales se describen a continuación.

PROMESAS	PRODUCTO
1.- Proporcionar un prototipo electrónico moderno de calidad y que sea de bonita apariencia.	1.- Un prototipo de una pantalla luminosa digital.
2.- Los componentes electrónicos que se utilizarán son de muy buena calidad lo que garantiza durabilidad a muy buen precio y su consumo eléctrico es bajo, por lo que reducirá costos en comparación a los gastos en papelería.	2.- Reducción de costos y durabilidad del producto.
3.- Diseñar un código de programación en lenguaje BASIC, que indique a la memoria del prototipo que mensaje desplegar, y que pueda ser compilado y descargado por el IDE Bascom-AVR.	3.- Código de programación en lenguaje BASIC.

1.6.- Estudio de factibilidad

Para analizar la factibilidad técnica y económica de éste proyecto primero se deben definir que clase de elementos se utilizarán, se debe conocer cada uno de los componentes electrónicos que forman el circuito, para poder iniciar la investigación del funcionamiento y del costo de cada componente electrónico. Luego se define la ubicación de los elementos en el circuito para que esté funcione correctamente.

En éste caso el proyecto propuesto está compuesto por dos etapas bien definidas, tal como lo muestra la siguiente figura.

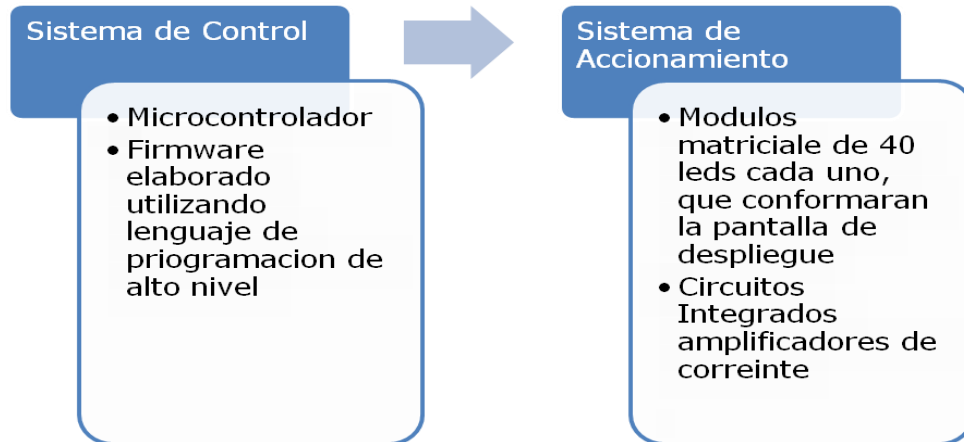


Figura 1.1: Esquema de bloques del proyecto Pantalla Luminosa Digital

La figura 1.1 muestra el esquema de bloques general del proyecto, la forma lógica de cómo funcionará el prototipo. Dicho esquema consta de dos etapas principales por las que está formado el prototipo, que son: Sistema de Control y Sistema de Accionamiento. Cada esquema muestra las principales etapas que

componen el prototipo, haciendo posible que el circuito electrónico logre controlar la pantalla y despliegue los textos indicados en el firmware.

A continuación, se describen las características y precios de cada uno de los principales componentes a utilizar para la realización de éste prototipo electrónico. Se describen tres productos distintos para cada elemento principal del proyecto y así poder tener una comparación entre ventajas económicas y técnicas de cada producto a considerar, y poder elegir el más conveniente en ambos aspectos.

1.6.1.-Estudio económico

En éste apartado se muestra una comparación de precios de las opciones de productos a utilizar en los principales componentes del proyecto, con el objetivo de observar la factibilidad económica que ofrece cada producto, así como observar la factibilidad logística de compra de cada uno. En algunos casos se ha elegido el de mayor o menor precio, esto dependerá de las características de funcionamiento de cada elemento, que se verán en el apartado de factibilidad económica.

Es necesario mencionar que todas las herramientas y componentes a describir han sido cotizadas en tiendas localizadas en Estados Unidos, y los costos mostrados ya incluyen gastos de envío. Esto debido a que dentro del país no se encuentran ninguna de estas herramientas; pero es fácil comprarlas por

medio del Internet, debido a ello en cada cuadro se muestra el nombre de la página web en donde se pueden cotizar y comprar.

IDE para el desarrollo del firmware de aplicación			
	CODEVISION	WINAVR	BASCOMAVR
Precio de cotización	Demo	GNU	Demo
Lugar de cotización	www.hpinfotech.ro	www.sourceforge.net	www.meselec.com

Elección IDE: Para el desarrollo del firmware del proyecto se optará por el IDE Bascom-AVR. Desde el punto de vista económico ofrece la ventaja de ser gratis la versión Demo, la cual es más que suficiente para los propósitos del proyecto, esto desde el punto de vista de las opciones y capacidades que ofrece esta versión.

Microcontrolador			
	PIC16F876	PICAXE28	ATMEGA-328
Precio de cotización (USD)	\$15.00	\$17.50	\$10.00
Lugar de cotización	www.mouser.com	www.rev-ed.co.uk	www.mouser.com

Elección Microcontrolador: Para el desarrollo de este prototipo electrónico el microcontrolador a utilizar es uno de los principales componentes dentro del circuito, ya que es el 'cerebro' del circuito que controla la función de los demás elementos del sistema. En éste proyecto se opta por el microcontrolador ATMEGA-328 de la familia AVR fabricado por la compañía ATMEL se ha

elegido este modelo, ya que es el de menor costo económico y cumple con las exigencias del proyecto.

Circuito integrado buffer de corriente			
	TPIC6B596	TPIC6B595	74HC595
Precio de cotización (USD)	\$4.5	\$4.5	\$3.-
Lugar de Cotización	www.mouser.com	www.mouser.com	www.mouser.com

Elección de Circuito Integrado Buffer de Corriente: El buffer de corriente es el encargado de regular la corriente que llegará a las matrices de leds, se encarga de adecuar la corriente transistada por las matrices asegurándose que sea la adecuada para evitar que éstas se dañen, en éste caso se optó por el buffer de corriente TPIC6B596, tomando en cuenta que su precio es similar al 595, y aunque es un poco más caro que el 74HC595, se han tomado en cuenta sus características de funcionamiento.

Modulo matriz de 40 leds.			
	NFM-34581	NFM-14581	NFM-23581
Precio de cotización (USD)	\$5.-	\$5.-	\$5.-
Lugar de Cotización	www.mouser.com	www.mouser.com	www.mouser.com

Elección de Matriz de Leds: las matrices de leds son los dispositivos encargados de mostrar el producto final, en éste caso los mensajes alfanuméricos que contendrán la información a publicar. En esta oportunidad

se optó por la Matriz NFM-23581 y se ha elegido por sus características de funcionamiento y por su tamaño, ya que el precio de los tres dispositivos es el mismo y además es muy accesible. Y el consumo de potencia del proyecto es:

- Voltios: 5 volt.
- Amperios: 2.5 amper
- Potencia: 0.5 wats.

Basándose en esto puede decirse que el consumo eléctrico del proyecto, es mínimo.

1.6.2.- Estudio técnico

En este apartado se describe el porque se han elegido los componentes, desde el punto de vista técnico, tomando en cuenta las características electrónicas, capacidad y/o velocidad física y lógica del dispositivo para que sea posible lograr que el circuito funcione correctamente y puedan cumplirse los objetivos planteados en éste proyecto.

IDE para el desarrollo del firmware de aplicación							
<i>Característica</i>	<i>Peso</i>	CODEVISION		WINAVR		BASCOMAVR	
Editor	10%	2	20%	2	20%	2	20%
Compilador	30%	1	30%	1	30%	2	60%
Modulo Programador	25%	0	0%	0	0%	2	50%
Modulo Simulador	25%	0	0%	1	25%	2	50%
Configurable	10%	1	10%	1	10%	2	20%
Total		30%		42.5%		100%	

Escala de ponderación: 0 = No cumple -- 1= Bueno -- 2 = Excelente

Elección IDE: El ambiente de desarrollo integrado BASCOM AVR es el elegido para desarrollar este proyecto, desde el punto de vista técnico, ya que integra un compilador que convierte el lenguaje Basic a lenguaje Maquina haciendo posible que el microcontrolador lo pueda interpretar y ejecutarlo. Además, es un programa muy completo y proporciona una interfaz que permite descargar el firmware a la memoria del microcontrolador a través del puerto paralelo de la PC, Además el programa incluye herramientas de ayuda para su utilización y se opto por el porque éste software es el que se ha utilizado durante el transcurso de la carrera, por lo tanto, se nos facilita su uso.

Microcontrolador							
<i>Característica</i>	<i>Peso</i>	PIC16F876		ATMEGA328		PICAXE28	
Velocidad	20%	2	40%	2	40%	2	40%
Memoria	30%	2	60%	2	60%	2	60%
Lenguaje de Programación	20%	2	40%	2	40%	2	40%
Vía de Programación	15%	1	15%	2	30%	2	30%
Precio	15%	1	15%	2	30%	1	15%
TOTAL		85%		100%		92.5%	

Escala de ponderación: 0 = No cumple -- 1= Bueno -- 2 = Excelente

Elección: Microcontrolador ATMEGA328 de la familia AVR del fabricante ATMEL, esto debido a que cumple con todos los requisitos técnicos necesarios para la implementación de este proyecto, una de las características técnicas más importantes es el Lenguaje y Compilador para su programación: BASIC, usando el BASCOM-AVR el cual es un IDE y compilador de lenguaje Basic

para microcontrolador de la familia AVR, además el microcontrolador elegido tiene la ventaja de poder realizar la descarga del programa a la memoria interna usando el puerto paralelo de la PC, haciendo posible programarlo vía software desde la PC, además la velocidad y la memoria de éste uC es la necesaria para el correcto funcionamiento del prototipo electrónico.

Circuito integrado buffer de corriente							
<i>Característica</i>	<i>Peso</i>	TPIC6B596		TPIC6B595		74HC595	
Numero de pines	10%	2	20%	2	20%	2	20%
Encapsulado	20%	2	40%	2	40%	2	40%
Máxima corriente	30%	2	60%	2	60%	0	0%
Voltaje de operación	20%	2	40%	2	40%	2	40%
Entrada / Salida	20%	2	40%	2	40%	2	40%
Total		100%		100%		70%	

Escala de ponderación: 0 = No cumple -- 1= Bueno -- 2 = Excelente

Elección: Como circuito amplificador de corriente para el correcto funcionamiento de los leds de los módulos matriciales, se ha elegido el chip TPIC6B596 ya que cumple los requerimientos técnicos para éste proyecto, tanto la amplificación de corriente como con el beneficio de tener una entrada serial de datos, interconexión con el microcontrolador, y salida paralela de datos para conexión con la matriz de leds.

Modulo Matriz de 40 leds.							
Característica	Peso	NEM-34581		NEM-14581		NEM-23581	
No pines	10%	2	20%	2	20%	2	20%
Color	20%	2	40%	1	20%	2	40%
Tamaño	20%	2	40%	2	40%	2	40%
No. de leds	30%	2	60%	0	0%	2	60%
Pin común (AC/CC)	20%	2	40%	2	40%	2	40%
Total		100%		60%		100%	

Escala de ponderación: 0 = No cumple -- 1= Bueno -- 2 = Excelente

Elección: Matriz de 40 Leds NEM-23581, se opto por este modelo debido a que su tamaño es más grande que los demás modelos, se encuentran en color rojo o verde y algo muy importante es que cada matriz consta de 40 leds, y en el proyecto se utilizaran 8 matrices, haciendo un total de 320 leds, es decir 320 píxeles, los cuales permitirán una muy bonita vistosidad y claridad en las frases alfanuméricas que se desplegarán por ellas.

Al finalizar el estudio de factibilidad económico y técnico se opto por los siguientes componentes:

- **BASCOM-AVR:** Se eligió porque es una versión DEMO gratis, y técnicamente es un ambiente que integra las características necesarias para la implementación del firmware en el proyecto.
- **Microcontrolador ATMEGA-328:** Entre los componentes mencionados este modelo es el más accesible económicamente, y sus características técnicas son las necesarias para el correcto funcionamiento del prototipo.

- **Circuito integrado Buffer de corriente TPIC6B596:** Económicamente su costo es similar al de los otros modelos, pero técnicamente su funcionamiento es el adecuado para el circuito electrónico.
- **Módulos matrices de 40 leds NFM23581:** El precio económico es igual entre las matrices mencionadas, pero desde el punto de vista técnico éste modelo posee mejores características y su tamaño es más grande, lo que garantiza una mejor visibilidad al momento de mostrar el producto final.

Al concluir puede decirse que es posible realizar éste prototipo electrónico, porque sus costos económicos no son muy elevados y técnicamente cada elemento cumple con los requisitos necesarios para que la pantalla luminosa digital logre cumplir los objetivos planteados como proyecto de graduación y así poder demostrar nuestros conocimientos técnicos adquiridos a través de la carrera.

1.7.- Carta de aceptación

San Salvador, marzo de 2010

Estimado Señor

LIC. MARVÍN HERNÁNDEZ

Presente

En esta grata oportunidad le saludan las alumnas del Técnico en Ingeniería de Hardware deseándole éxito en sus labores cotidianas.

Aprovechando la oportunidad para presentarle nuestro proyecto de graduación, el cual consiste en una **pantalla luminosa digital**, que permitirá desplegar mensajes en forma visual informativa en oficinas y/o pasillos de la Universidad, siendo así una herramienta muy adecuada para mostrar la información más importante y de interés para los alumnos.

Este proyecto beneficiara a la Escuela de Informática de la Facultad de Informática y Ciencias Aplicadas de la Universidad, por lo que solicitamos su valiosa aprobación para poder implementar nuestro proyecto.

Atentamente,

F_____

Maria Concepción Rivas, alumna

F_____

Diana Carolina Hernández, alumna

F_____

LIC. MARVIN HERNÁNDEZ, (Firma de Aceptación)

CAPITULO II: MARCO TEÓRICO DE REFERENCIA

2.1- Marco teórico del problema

A continuación, en el marco teórico del problema se describe el área del conocimiento en la que está ubicado el proyecto, es decir el entorno teórico de la descripción de cada una de las etapas por las que esta compuesto el proyecto. Se definen conceptos básicos y claros para que el lector pueda comprender cual es la finalidad de éste proyecto.

El proyecto “pantalla luminosa digital”, es un sistema de hardware informático que se enmarca dentro del área de los sistemas digitales programables, específicamente los sistemas electrónicos embebidos (Embebed System).

2.1.2.- ¿Qué es un sistema electrónico embebido?

Un sistema embebido (SE) es un conjunto de elementos de electrónica, informática y hasta mecánica, diseñado para la realización de una función, aplicación o tarea específica. En algunos casos éstos sistemas están embebidos o empotrados dentro de un sistema de mayor escala. Los sistemas embebidos se utilizan para usos muy específicos, y en la mayoría de sistemas los componentes se encuentran incluidos en la placa base del dispositivo.

Algo que es muy notable en estos sistemas es el precio y el consumo eléctrico. Esto debido a que los sistemas embebidos se pueden fabricar por unidad, decenas, centenas, o por millones de unidades, logrando así reducir los costos. Los sistemas embebidos suelen usar un procesador relativamente pequeño y una memoria pequeña para reducir los costos y reducir el consumo eléctrico. Usualmente estos sistemas están basados en microprocesadores o microcontroladores y la tarea o función que realizan está especificada por el firmware, o programa de aplicación específico, alojado en la memoria del sistema. Y en cuanto a la velocidad, dependerá de los elementos que se utilicen, ya que lo que se pretende con estos sistemas es simplificar toda la arquitectura haciendo posible que un sólo dispositivo realice una función específica de forma práctica y rápida.

Por ello un sistema embebido debe estar diseñado para representar un tiempo real las entradas, activando salidas, es decir su funcionamiento es en tiempo real y según lo que se guarde en sus entradas así activará las salidas:



Figura 2.1: Esquema general de un sistema electrónico

La figura 2.1, muestra los tres bloques generales por los que está compuesto el sistema embebido, como lo son las entradas por las que se introducen los datos de un usuario externo, otro sistema o señales provenientes del entorno del sistema, y el bloque central es el sistema de control, el bloque de control de todo el dispositivo en el que se guarda el firmware, y como salida el producto final, en esté caso los mensajes que se desplegarán por la pantalla.

Como se menciona anteriormente parte fundamental del sistema embebido es el software o firmware. El software es en general todo lo que trae, lo que compone internamente al sistema embebido, el entorno lógico, y también incluye un firmware que es para indicarle las tareas específicas a realizar.

El firmware funciona para SE de la igual diseño y iguales componentes, es decir un sistema embebido diseñado para un modelo específico, no es compatible con otro modelo.

Por ejemplo: Dentro de la PC, están la tarjeta de video, el CD-ROM, estos son sistemas embebidos; es como un BIOS, específico para determinada marca o modelo y no compatible con otro.

2.1.3.- Características principales de un sistema electrónico embebido

Un sistema embebido en general debe poseer ciertas características indispensables con las funciones adecuadas, haciendo posible que el sistema

embebido cumpla con la función que se ha diseñado. Estas características se listan y describen a continuación.

Diseñado para una tarea específica: Se dice que un sistema embebido debe ser diseñado para realizar una única tarea, por lo que son sistemas muy eficientes y dedicados, pudiendo así ser sistemas que se integren dentro de sistemas de mayor escala. Es por eso que los sistemas embebidos son dispositivos electrónicos usados para controlar y operar equipos, dispositivos, maquinas, aparatos domésticos, equipos móviles, PDA's, automóviles, instrumentos electrónicos, etc.

Integran hardware y software: además de ser dispositivos formados por elementos electrónicos, estos sistemas embebidos integran un software o firmware, el cual es un código de programa que se le descarga a la memoria del dispositivo, dicho programa debe estar bien diseñado específicamente para responder al hardware electrónico diseñado, ya que de el depende el correcto funcionamiento de la parte lógica del dispositivo. Este firmware debe ser escrito en cualquier lenguaje de programación compatible con el CPU o procesador central del sistema embebido.

Trabajan en tiempo real: existen dispositivos diseñados para realizar funciones específicas como los controladores que se utilizan en centrales telefónicas, ascensores, sistemas de adquisición de datos. Sistemas de diagnóstico y control: todas estas funciones son realizadas en tiempo real. Es decir, el sistema debe ser capaz de responder instantáneamente a cualquier cambio de sus entradas (sensores, pulsadores, interruptores), realizando algún proceso o modificando su entorno.

Manejan entradas: Todo sistema necesitara entradas que procesar, para decidir como actuar en dependencia del estado de estas, en el caso de los sistemas embebidos es común el uso de sensores, los sensores son dispositivos electrónicos capaces de captar las señales físicas y enviarlas al bloque de control del dispositivo, para que esté reaccione de acuerdo a su funcionamiento. Estas entradas pueden provenir del entorno del sistema, por ejemplo, un sensor de temperatura, un sensor de presencia, etc., o estas entradas pueden ser internas al sistema, por ejemplo, la hora actual, el valor de una variable, etc.

Manejas salidas Como se menciona, los sistemas reaccionaran a sus entradas modificando salidas, las cuales pueden ser internas o externas, una salida interna puede ser: la escritura en una posición de memoria, el envío de información a un chip interno, etc., por el contrario una salida externa se define

como aquella que puede modificar el entorno externo del sistema, para esto se dota de actuadores, los actuadores son los encargados de interactuar y/o modificar el entorno físico del sistema embebido, algunos ejemplos de estos son los diodos emisores de luz, parlantes, motores, etc.

Bajo costo económico: Debido a su especialización y a que los sistemas embebidos combinan tanto software como hardware electrónico, su costo de elaboración es relativamente bajo ya que son sistemas diseñados para realizar tareas específicas, por lo que el costo de los materiales no es muy elevado, lo que garantiza un costo económico accesible.

Eficiencia (bajo consumo de potencia): Los sistemas embebidos deben ser diseñados teniendo en mente el consumo de potencia y la eficiencia de esta, debido a que son sistemas que estarán dedicados a tareas específicas y serán sistemas que no dependerán de una constante intervención de un usuario. Debido a su poca complejidad en su diseño son dispositivos eficientes y su consumo potencial es mínimo, ya que los materiales que se utilizan son de calidad y por su tamaño que suele ser pequeño su consumo eléctrico es mínimo.

2.1.4.- Estructura básica de un sistema electrónico embebido

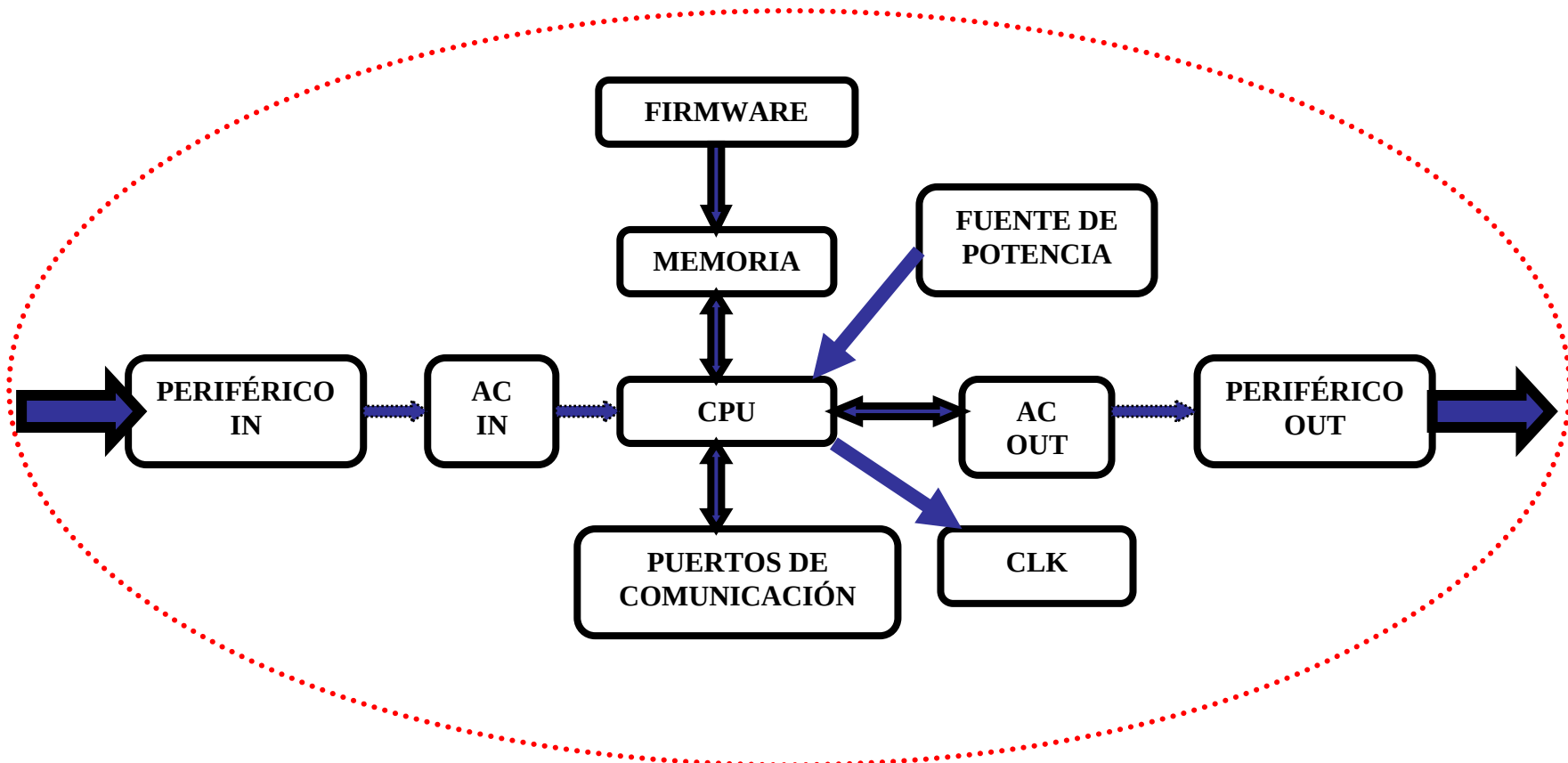


Figura 2.2: Detalle de bloques de la estructura interna de un sistema embebido.

En la figura 2.2, se muestra un esquema de bloques internos de un sistema embebido en general, en esta figura se puede observar que un sistema embebido internamente debe estar compuesto por bloques específicos que cumplen funciones que son parte de un todo. Cada bloque será constituido por diversos componentes de electrónica, a continuación, se hace una breve descripción de cada bloque referente a la figura 2.2.

Periféricos de Entrada: Un periférico es un componente electrónico que hace posible la conexión con su entorno físico, es decir con otro dispositivo, en el caso de los sistemas embebidos, los periféricos de entrada le permiten al sistema embebido recibir información desde el equipo externo, por medio de estos dispositivos se hace posible transferir la información necesaria desde un ambiente físico externo hacia el interior del sistema embebido. Por ejemplo, están los sensores de luz, teclados, micrófonos, etc.

Acondicionamiento de Señal de entrada: este bloque es el encargado de tomar las señales provenientes de los periféricos de entrada y enviárselas al bloque correspondiente dentro del sistema embebido, por ejemplo, pueden ser los sensores ya que estos se encargan de captar la señal física

Acondicionamiento de Señal de entrada: este bloque es el encargado de tomar las señales provenientes de los periféricos de entrada y enviárselas al

bloque correspondiente dentro del sistema embebido, por ejemplo, pueden ser los sensores ya que estos se encargan de captar la señal física.

Algunos casos la convierten en energía eléctrica para enviársela al bloque de control.

Firmware: Todo sistema embebido debe estar controlado por órdenes que específicas dentro de un programa o software, es un sistema operativo de bajo nivel diseñado específicamente para una marca o modelo. Este programa de aplicación puede estar escrito en cualquier lenguaje de programación compatible con el procesador a utilizar. Este firmware puede ser actualizado, pero debe tomarse muy en cuenta el tipo de sistema embebido que se está manejando, ya que según la implementación hardware que se tenga así debe ser la estructura lógica del firmware que se le debe descargar para que el dispositivo cumpla con la función que se pretende.

Memoria: ¿Que es la memoria? en este caso, la memoria que se utilizará ya viene incorporada en el microcontrolador, y en cuanto al tamaño de esta, debe considerarse cual es la finalidad del dispositivo, ya que así será el tamaño del firmware que se le debe descargar. Este bloque puede estar formando por memoria de tipo ROM y/o RAM

Unidad Central de Proceso CPU: este es el bloque encargado de ejecutar las acciones, se encarga de dar las ordenes a los demás bloques, con respecto a que acción se realizará y se encuentra en la parte central del esquema en donde se ubica el microprocesador o microcontrolador, es decir la CPU o unidad que aporta capacidad al sistema, teniendo la opción de incluir memoria interna o externa.

Puertos de Comunicación: estos dispositivos son parte fundamental en el circuito, ya que proveen al SE comunicación bi-direccional con otros sistemas. Por ejemplo, **RS-232**, [I²C](#), **ISP**, éstos son protocolos que permiten la programación del sistema embebido.

Fuente de Potencia: para implementar este bloque puede utilizarse una fuente de corriente o baterías alcalinas, esto es opcional, pero debe tomarse en cuenta que el bloque de potencia es el encargado de proveer la energía necesaria a todo el circuito electrónico, en este caso deben tomarse en cuenta las exigencias de los dispositivos electrónicos y la finalidad del sistema embebido para determinar que magnitud de corriente se le debe adaptar, esto depende del consumo del sistema embebido.

Reloj del sistema CLK: el clock, es un reloj que marca el tiempo real, desde Mhz hasta Ghz, y es posible implementarlo con un cristal externo, esto si es necesario.

Acondicionamiento de Señal de Salida: como salidas están los actuadores, que pueden ser motores. Estos son los encargados de interactuar con el entorno exterior, los que se encargan de hacer posible el desplazamiento del dispositivo, si fuere el caso, este bloque es el que proporciona la energía necesaria a los periféricos de salida, para que sea posible mostrar el funcionamiento del sistema embebido.

Periféricos de Salida: son los bloques o puertos de conexión encargados de permitir las salidas del sistema, a dispositivos externos o actuadores, los periféricos o componentes que permiten obtener el resultado final del funcionamiento completo del sistema embebido, en estos se pueden encontrar, leds, pantallas, display, etc.

2.1.5.- Áreas de aplicación de los sistemas electrónicos embebidos.

En el mundo actual Los sistemas embebidos están inmersos dentro de nuestra vida diaria. Sus aplicaciones abarcan muchos sistemas y aparatos, los cuales

son utilizados por los seres humanos en el diario vivir, algunos de estos se mencionan a continuación:

HOGAR: en los hogares es muy común encontrar sistemas embebidos, aunque muchas veces no están muy visibles, pero se encuentran dentro de las lavadoras de ropa, dentro de los televisores digitales, sistemas de audio, reproductores de MP3, controles remotos, alarmas electrónicas, en los sistemas de climatización y temperatura, calefacción, sistemas de video como grabadores y reproductores, teléfonos celulares, en módems, routers, entre otros.

CONSUMO: los equipos que se clasifican en esta área son de uso muy común, en especial en los hogares, ya que son sistemas diseñados para realizar tareas específicas en tiempo real, entre ellos esta: televisores, pantallas, microondas, refrigeradores lavadores, mp3playes, celulares, entre otros.

OFICINA: en las oficinas también es posible encontrar estos sistemas en equipos que se han vuelto en la actualidad indispensables, como lo son los routers para conectividad a Internet, cortafuegos, switchs para redes locales, impresoras, scanner, digitalizadores de imágenes, pizarras electrónicas, proyectores de video, cámaras fotográficas digitales, alarmas digitales, sistemas de control de acceso, aire acondicionado, y muchos mas.

TRANSPORTE: uno de los ejemplos más comunes e indispensables que se usan en la actualidad son, los automóviles modernos, estas maquinas tiene más de 200 sistemas embebidos en su infraestructura. Ya que para controlar muchas partes del vehiculo, se utilizan sistemas muy precisos, como lo son los frenos ABS, el sistema de inyección, combustible, aceleración, carburación, sistema de luces, y señalización, luces de cabina, climatización y aire acondicionado, limpia vidrios, entre otros.

COMUNICACIONES: en esta área es fácil identificar muchos aparatos que están basados en sistemas embebidos, como los ya mencionados routers, Access Point para comunicación WiFi, switches de comunicación administrables remotamente, sistemas de seguridad informática como los cortafuegos, sistemas de filtrado de paquetes TCP/IP, módems, sistemas utilizando fibra óptica, estos entre una gran variedad de sistemas embebidos clasificados en esta área.

MEDICINA: en la actualidad en el área de la salud también se encuentran equipos basados en sistemas embebidos, los cuales van desde simples termómetros digitales, hasta sistemas de monitoreo para pacientes con problemas cardiacos, también ecógrafos digitales, monitores de apnea del sueño, y en otros aparatos electrónicos que ya existen y sin duda se seguirán creando sistemas embebido en el área de la medicina.

ROBÓTICA: la mayoría de robots diseñados en el área de entretenimiento y muchos industriales, se basan en sistemas embebidos, capaces de manejar los sistemas articulares, de video, transporte por el control de velocidad y dirección de motores, en el área de comunicaciones, en oficinas y muchas otras áreas en las que se utilizan sistemas electrónicos embebidos.

DOMÓTICA: en esta área se encuentran los sistemas encargados de automatizar las viviendas, se basan completamente en sistemas embebidos, los cuales están desde la climatización y calefacción, sistemas de seguridad para el hogar, sistema de iluminación, control de consumo de energía, medios de comunicaciones del hogar, servicios de TV digital, sistemas de riegos, estos entre otros mas, ubicados en el área de la domótica.

2.1.6.- Componentes principales del proyecto a desarrollar

¿Qué es un prototipo electrónico?

Es un primer modelo o molde original que se fabrica y de este modelo se parte para la elaboración de lo que se pretende, un objeto, herramienta, dispositivo electrónico, etc.

¿Qué tipos de prototipos electrónicos existen?

Existe una gran variedad de prototipos electrónicos que luego se convierten en ejemplares para el diseño y desarrollo de muchos equipos de los mismos, entre

ellos se encuentren: prototipos mecánicos, eléctricos, análogos, digitales, industriales, médicos, hogareños, prototipos diseñados para el área de comunicación, etc.

¿Qué es una pantalla?

Es una lámina que se sujeta delante o alrededor de la luz artificial, en la cual pueden proyectarse imágenes o textos, lo que ofrece una vista bien definida del objeto o imagen que se proyecta en ella. En este caso la pantalla estará formada con matrices de leds.

¿Qué es una matriz de leds?

Es un cartel formado por varias filas y columnas de leds, puede servir para pasar mensajes publicitarios, decorar cierta área, decorar nuestro ordenador, o lo más usual para elaborar proyectos electrónicos, como una pantalla en la que se puedan exhibir textos, imágenes, fijas o animadas.

¿Cómo desplegar los mensajes?

Una pantalla compuesta por matrices de leds, es un dispositivo electrónico y para que realice la función de desplegar mensajes debe estar debidamente programada, ya que parte importante del funcionamiento de una pantalla es el software que se le descargue a la memoria de ese circuito. En este prototipo se usará un microcontrolador que tenga la memoria necesaria y adecuada para que pueda guardarse en ella la información o imágenes que se pretenden mostrar en la pantalla. Para lograr esto el programa que se le descargue debe

estar muy bien diseñado para que la pantalla despliegue lo que el programa le indique.

¿Qué es un microcontrolador y como funciona?

Un microcontrolador (uC) es un chip o circuito integrado en cuyo interior incorpora los componentes o bloque básicos para conformar un sistema embebido.

Una consecuencia de su pequeño tamaño es que los recursos (memoria, velocidad) están limitados, en comparación a una PC o a un sistema embebido basado en microprocesadores.

En términos funcionales un uC es un chip reprogramable que controla un sistema embebido, y físicamente el uC es un chip con muchos pines (desde 8 hasta más de 40). Estos pines son usados para alimentación del reloj, puertos digitales, puertos análogos, comunicación y más.

¿Cómo funciona el buffer de corriente?

Funciona como circuito amplificador de corriente para el correcto funcionamiento de los dispositivos que se utilizan en el circuito electrónico, encargándose de que la corriente que llegue a ellos sea la adecuada.

¿Qué método de programación se debe utilizar?

Existe una amplia gama de software para desarrollar sistemas embebidos, hay programas diseñados específicamente para la descarga de códigos hacia los sistemas embebidos, así como también existen muchas herramientas,

diferentes lenguajes de programación de bajo o de alto nivel, siendo herramientas disponibles para que el usuario elija la que más le parezca y se Adecue a la finalidad del sistema embebido.

2.2.- Marco teórico de solución

El proyecto propuesto en este documento se encuentra ubicado en el área de los sistemas electrónicos embebidos. Un sistema embebido es un conjunto de elementos de electrónica, informática y hasta mecánica, diseñado para la realización de una función / aplicación / tarea específica. En algunos casos estos sistemas están embebidos o empotrados dentro de un sistema de mayor escala. Los sistemas embebidos se utilizan para usos muy específicos, y en la mayoría de estos sistemas los componentes se encuentran incluidos en la placa base del dispositivo.

Dentro de los sistemas embebidos se encuentran diversas aplicaciones, entre ellas están los dispositivos de consumo. Los sistemas embebidos de consumo son sistemas diseñados para realizar tareas específicas en tiempo real, por ejemplo: televisión, lavadoras, microondas, refrigeradores, entre otros. En esta clasificación se encuentra el proyecto descrito en este documento, el prototipo de una ***pantalla luminosa digital***.

Una pantalla luminosa digital es un sistema electrónico embebido capaz de realizar una función, la cual consiste en desplegar mensajes alfanuméricos en tiempo real y además alternará la hora, la que podrá ser modificada por switches.

2.2.1.- Esquema general en bloque de una pantalla luminosa digital

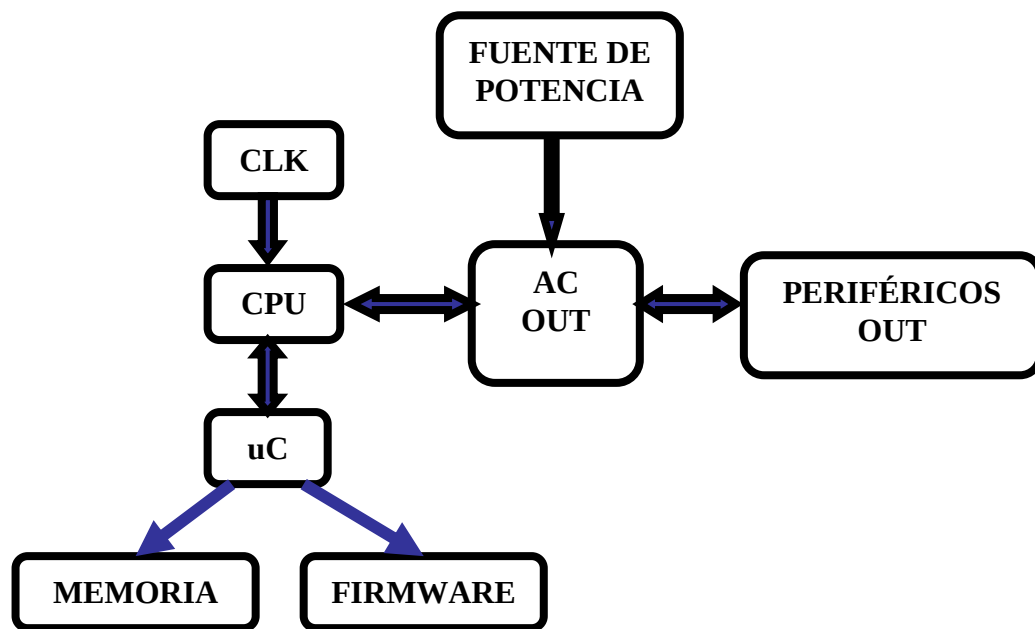


Figura 2.3: esquema general de una pantalla luminosa digital

La figura 2.3, muestra el esquema electrónico, cada unos de los componentes por los que estará compuesto el prototipo.

CLK: este bloque viene integrado dentro del microcontrolador y puede adaptarse a un cristal externo para hacer que funcione como reloj que marque el tiempo real.

CPU: para implementar el bloque del CPU se utilizará un microcontrolador Atmega 168, de la familia Atmel, este uC tiene las características necesarias para la elaboración de este proyecto, tomando en cuenta que este es el cerebro del prototipo, el bloque principal por el que está formado el CPU.

Uc (microcontrolador): el microcontrolador a usar en este prototipo es el Atmega 168, este uC tiene 16kb de memoria, la cual es suficiente para guardar el firmware del prototipo.

Memoria: como se menciona anteriormente la memoria viene incluida en el microcontrolador y es de 16kb, el necesario para guardar el código que le indique al dispositivo lo que debe hacer.

Firmware: es un código de programa diseñado en lenguaje BASIC en el que se le indica al microcontrolador que ordenes debe enviarles a los demás elementos para que éstos cumplan con la función correspondiente y así lograr que la pantalla despliegue los mensajes que se indican en el firmware.

Fuente de potencia: la fuente de potencia es la encargada de brindar la corriente eléctrica al dispositivo, y se usará una fuente de 5 voltios, la cual podrá conectarse a 110 voltios.

Ac Out: como salida AC estará el buffer de corriente TPIC6B596, éste funciona como un regulador de voltaje para las matrices de leds, evitando que llegue un sobrecarga de energía y éstas se dañen.

Estos bloques son los que conforman el proyecto a desarrollar, cada uno tiene su propia función tal como se ha descrito, haciendo posible que la pantalla luminosa digital despliegue mensajes y alterne la hora en tiempo real.

2.3.- Marco teórico conceptual

Sistema: conjunto de partes o cosas coordinadas por reglamentos diseñados para un fin, o una aplicación específica.

Electrónica: área de la ingeniería y de la física aplicada al diseño de dispositivos, por lo general circuitos electrónicos, cuyo funcionamiento depende del flujo de electrones haciendo posible la transmisión, recepción y almacenamiento de información. Esta información puede consistir en voz (señales de voz) o en música en un receptor de radio, en una imagen en una pantalla de televisión, o en números u otros datos en un ordenador o computadora.

Embebido: es cuando un conjunto de componentes de electrónica o informática están embebidos o empotrados dentro de un sistema de mayor escala, es decir el pequeño sistema esta embebido o integrado en otro de mayor nivel.

Prototipo: es un primer modelo o molde original que se fabrica y de este modelo se parte para la elaboración de lo que se pretende, un objeto, herramienta, dispositivo electrónico, etc.

Hardware: es la parte tangible, lo que se puede ver y tocar, de un determinado sistema o equipo electrónico o informático.

Software: es la parte interna que en algunos casos no se ve y no se puede tocar, es el bloque lógico encargado de indicar que es lo que se debe hacer, esto según lo que se le haya programado al dispositivo.

Periférico: son los dispositivos de entrada y salida que permiten la conexión con el entorno físico, como entradas de información y salidas, es decir donde se muestra la función final del equipo o dispositivo.

AC In: es un bloque electrónico en el que se ubica un determinado dispositivo, el cual se encarga de proporcionar la corriente alterna que debe llegar hasta el interior del equipo electrónico.

Firmware: es un sistema operativo de bajo nivel similar al BIOS en el que todas las aplicaciones más importantes utilizan su funcionalidad. Aunque a diferencia del BIOS en el firmware no hay una versión común disponible, ya que cada firmware es diseñado específicamente para un modelo, y no hay compatibilidad con otras marcas.

Memoria: es un dispositivo diseñado y fabricado específicamente para guardar en ella información. La capacidad de la memoria dependerá del tipo de equipo que se este implementando, y puede ser que la memoria sea externa o que ya venga incorporada en uno de los dispositivos que se están usando, estas pueden ser ROM, PROM, EPROM, EEPROM, RAM, entre otras.

CPU: unidad central de procesamiento, este es un bloque principal dentro de un dispositivo, es el cerebro, se encarga de dar las ordenes a los demás dispositivos, en algunos casos pueden usarse microcontroladores o microprocesadores como parte fundamental del bloque de control.

Puertos Electrónicos: son dispositivos físicos electrónicos que sirven como puertos de conexión que permiten la entrada y salida de información al dispositivo, es el medio donde se conecta la entrada y la salida.

Fuente de potencia: es el bloque encargado de proporcionar la potencia eléctrica necesaria para que el equipo funcione correctamente, la fuente está

alimentando el equipo durante el tiempo que está encendido, y además de una fuente también pueden utilizarse baterías alcalinas, sólo debe tomarse en cuenta cual es la magnitud que se necesita.

CLK: el reloj puede ser interno o externo, éste se encarga de mostrar el tiempo en el que trabaja el dispositivo puede ser en tiempo real o no, esto depende de la finalidad del equipo.

Cristal: es un especio de vidrio transparente, con el cual es posible hacer funcionar el reloj del dispositivo, además el cristal puede implementarse de forma externa y en conjunto con el reloj, se mide el tiempo del funcionamiento.

AC out: es el bloque de salida de corriente, en el que puede implementarse el dispositivo adecuado para que regule la corriente que sale hacia el dispositivo que muestra el producto final de equipo.

Robótica: conjunto de conocimientos prácticos que permiten obtener, realizar y automatizar sistemas basándose en estructuras metálicas empleando un cierto grado de inteligencia, para que sea posible lograr un equipo destinado a desempeñarse en un determinado ambiente.

Domótica: este termino proviene de la unión de las palabras *domus* (que significa casa en [latín](#)) y *tica* (de *automática*, palabra en griego, 'que funciona por sí sola'). Se entiende por domótica al conjunto de sistemas capaces de

automatizar una vivienda, proporcionando una posible gestión en cuanto a la seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y en el hogar podría definirse como la integración de una tecnología en el diseño inteligente de un determinado lugar.

Pantalla: es una lámina que se sujeta delante o alrededor de la luz artificial, en la cual pueden proyectarse imágenes o textos, lo que ofrece una vista bien definida del objeto o imagen que se proyecta en ella. En este caso la pantalla estará formada con matrices de leds.

Matriz de leds: es un cartel formado por varias filas y columnas de leds, puede servir para pasar mensajes publicitarios, decorar cierta área, decorar nuestro ordenador, o lo más usual para elaborar proyectos electrónicos, como una pantalla en la que se puedan exhibir textos, imágenes, fijas o animadas.

Microcontrolador: un microcontrolador (uC) es un chip o circuito integrado en cuyo interior incorpora los componentes o bloque básicos para conformar un sistema embebido.

Buffer de corriente: este es un chip que en su interior integra los componentes necesarios para la regulación de la corriente transitante en el circuito y que va dirigida a un determinado componente, no permitiendo que llegue sobrecarga de energía a ese dispositivo, evitando que este se dañe o arruine.

IDE Bascom-AVR: El ambiente de desarrollo integrado BASCOM AVR, es un software bastante completo ya que integra un compilador que convierte el lenguaje Basic a lenguaje Maquina haciendo posible que el microcontrolador lo pueda interpretar y ejecutarlo. Además, es un programa muy completo y proporciona una interfaz que permite descargar el firmware a la memoria del microcontrolador a través del puerto paralelo de la PC, y también el programa incluye herramientas de ayuda para su utilización, lo que facilita la programación del sistema embebido.

CAPITULO III: DESARROLLO DE LA SOLUCIÓN

3.1.- Propuesta de la solución

Durante todo el desarrollo de este documento se ha descrito el porque es necesario diseñar e implementar un prototipo que sea capaz de suplir en parte la necesidad de mostrar la información más importante y oportuna de forma práctica y actualizada.

De acuerdo a lo descrito en este documento es necesario desarrollar un equipo que sea capaz de facilitar la forma de publicar eventos o cambios a realizar. Por este motivo en este trabajo de graduación se describe una propuesta, la

cual es un medio capaz de mostrar información breve y oportuna que mantenga la información visible y al alcance de quien se pretende informar.

Debido a esto se ha diseñado un equipo moderno capaz de cumplir estas exigencias. Esté equipo es la solución más práctica y económica apta para ser un medio en el que se pueda publicar la información y mantenerla actualizada.

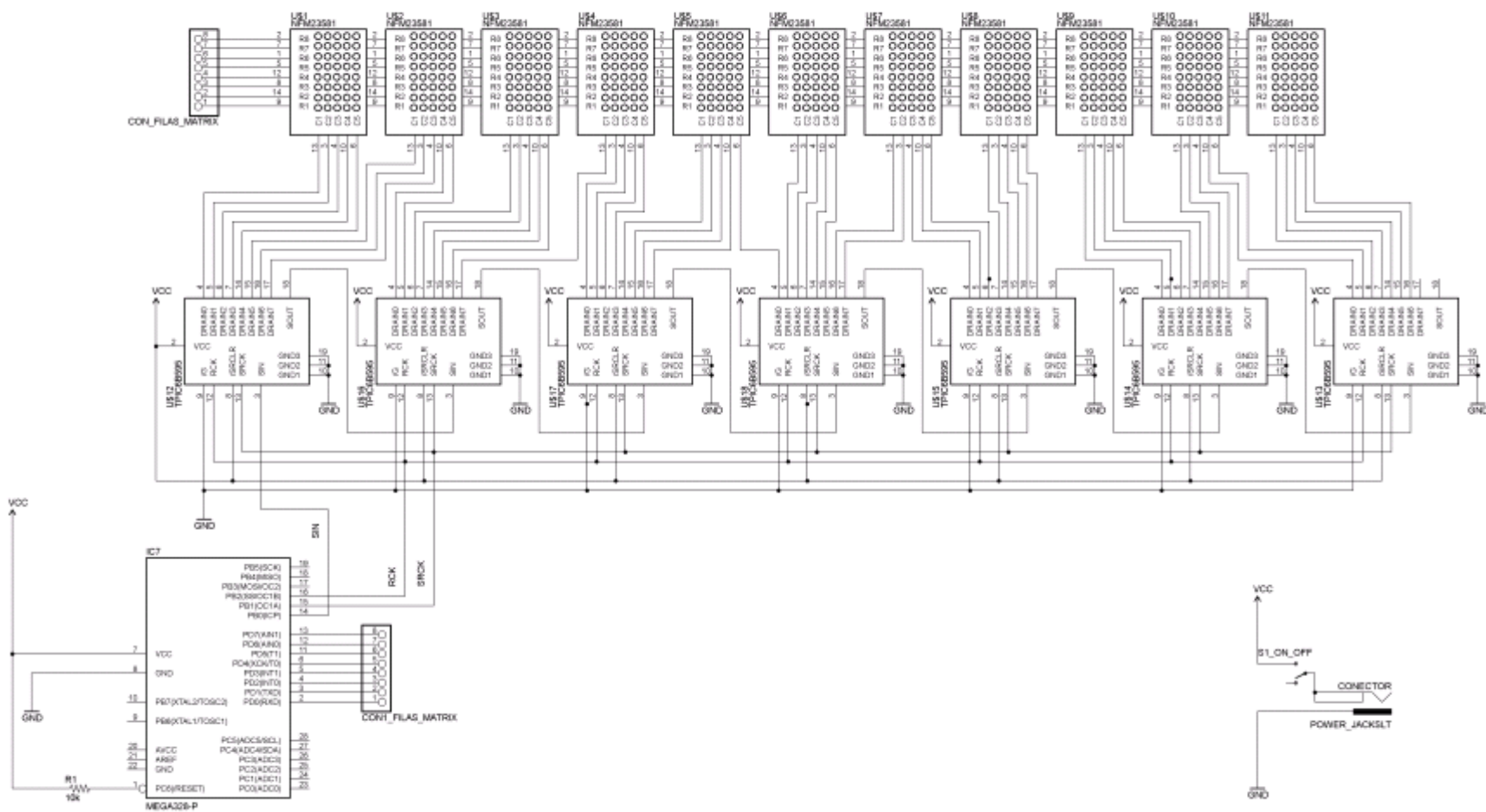
Esté prototipo es un equipo moderno y actualizado, un ejemplar de una *pantalla luminosa digital*. Una pantalla es una lamina o un rotulo que al acoplarlo con luz artificial permite proyectar imágenes o textos de forma bien definida, lo que hace que el receptor capte el mensaje que ahí se esta mostrando. En este caso la pantalla es un dispositivo de alta tecnología y durabilidad, es un *sistema electrónico embebido* capaz de desempeñar una función especifica en tiempo real, es un equipo electrónico reprogramable vía software en el cual es posible publicar mensajes alfanuméricos breves.

Su desarrollo inicia en el diseño de un circuito eléctrico, esté circuito es un diagrama en el cual se definen los componentes que se utilizarán y como deben ir interconectados electrónicamente entre si para lograr que el equipo cumpla con lo que se pretende. Este diagrama debe ir muy bien diseñado de acuerdo a las entradas eléctricas y salidas de cada componente electrónico.

Es necesario que se tenga una idea bien definida a cerca de lo que se pretende diseñar, deben considerarse los componentes que se utilizarán, algo muy necesario y de mucha ayuda son las hojas técnicas de cada componente, estas hojas técnicas pueden buscarse en Internet y de acuerdo a ellas se define como deben conectarse los elementos.

Esto debido a que las polaridades de los componentes no son las mismas, varían, así como también las entradas y salidas. Todo esto debe identificarse previamente para poder diseñar el circuito.

A continuación, se muestra el circuito electrónico de éste proyecto.



El equipo está diseñado para realizar una función específica que consiste en mostrar información breve de forma práctica y actualizada. Este equipo tiene como bloque central o CPU un microcontrolador Atmega-328, este microcontrolador incorpora memoria interna, la cual es la necesaria para almacenar en ella el firmware o código principal en el cual van las instrucciones que les indican a los componentes como deben trabajar entre si.

Además, como alimentación eléctrica el equipo tendrá una fuente de potencia de 5 voltios de corriente directa que se conectarán a 110 voltios de corriente alterna, lo que garantizará un bajo consumo a un muy buen costo económico.

Además, en este bloque se encuentra un buffer de corriente que conecta la salida del microcontrolador con una matriz de leds, y como salidas están las matrices de leds, las cuales muestran el producto final que son los mensajes textuales que se publicarán.

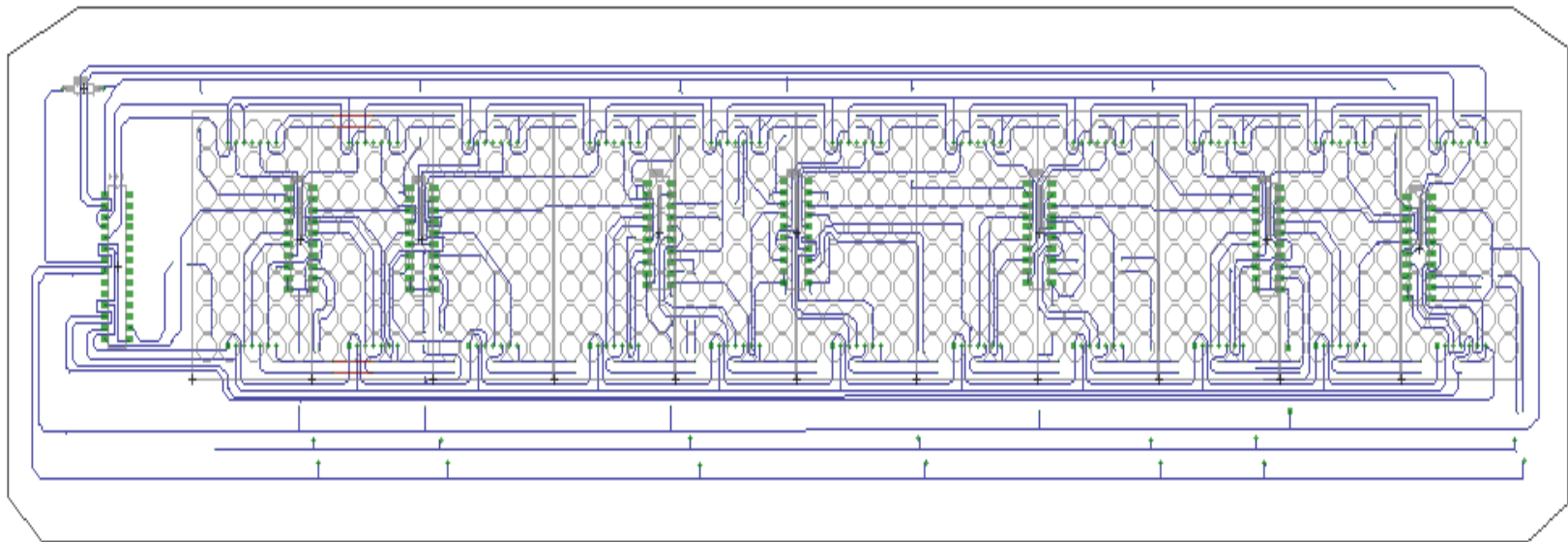
Logrando así que funcione como un medio de publicidad, en el que se podrán publicar mensajes cortos que estarán siempre visibles y podrán cambiarse o modificarse por medio de un software que es un ambiente de desarrollo integrado para programar en lenguaje Basic. Por medio de este software se le descargará el código o firmware al equipo electrónico. Este código siempre será el mismo solo será necesario cambiar una línea en la que se indique el mensaje que se desea mostrar y reprogramarlo para poder observar en la pantalla la ultima modificación que se le hizo, y así mantener la información actualizada.

Luego de tener el circuito eléctrico diseñado correctamente se realiza la compra de los componentes, asegurándose que sean los que se ubicaron en el circuito o que sean similares con las mismas características técnicas y de funcionamiento. Ya teniendo todos los componentes es recomendable ensamblarlos en un breadbord para asegurarse que funcionará como se tiene planeado.

Después de haberlo ensamblando y estando seguros que funcionará correctamente o si es necesario hacer algún cambio en cuanto a los componentes o a las conexiones eléctricas, es prudente hacerlo, porque, así como funcione en el breadbord así funcionara también en la placa base del dispositivo.

Ya estando seguros que funcionan correctamente se diseña el circuito impreso. Este circuito es la forma final de cómo Irán conectados los componentes, es el circuito impreso, es decir el circuito que se quemara en la placa cobreada, para ensamblar y soldar en ella los componentes, siendo esta la placa base del prototipo.

CIRCUITO IMPRESO



Este proyecto ha sido diseñado y elaborado durante el proceso de graduación con el fin de ofrecer una opción viable que ayude a la universidad a mantener la información siempre de forma visible.

Y esta dirigido a la escuela de informática de la Universidad Tecnológica de El Salvador, ofreciendo un dispositivo electrónico actualizado moderno que además de ser de bajo costo económico es fácil de usar, ya que solo será necesario modificar una línea del código que ira guardado en la memoria del cpu del equipo. Este dispositivo le facilitará el trabajo de impresión y pega a la universidad, ya que bastará con modificar una línea del programa.

La iniciativa de diseñar este prototipo inicio debido a que en ocasiones cuando se es estudiante no se tiene un completo alcance a la información, porque en algunos casos la información es publicada por Internet y no todos tienen acceso a Internet a cualquier hora del día, o en otros casos la información está impresa en volantes dentro de las instalaciones de la universidad. Por este motivo como estudiantes optamos por esta opción planteándonos un objetivo: dejar a la universidad un ejemplar de un equipo moderno y económico capaz de facilitar la publicación de la información para los estudiantes.

Este dispositivo ha sido diseñado y desarrollado con componentes capaces de realizar lo que se pretende, pero debe tomarse en cuenta que hay una gran variedad de componentes electrónicos de mayor capacidad y durabilidad que

los que se están utilizando en este prototipo. Es claro que en esta ocasión se opto por los materiales de menor costo económico, dado que solo es un ejemplar, un prototipo y no se implementará. Por esto el dispositivo no tendrá el rendimiento y durabilidad que uno compuesto con materiales de mayor calidad. Pero si la universidad opta por implementar este proyecto es recomendable integrarle al equipo un microcontrolador de mayor capacidad, ya que este es el que trae incorporada la memoria y según la capacidad y/o velocidad de este así será la cantidad de caracteres o mensajes que podrán mostrarse. También algo que se puede mejorar son las matrices de leds, puede optarse por usar matrices mas grandes y que intercambien colores en su iluminación, lo que producirá un aspecto de mejor visibilidad y agradable al ambiente estudiantil.

Conclusiones

- La investigación realizada durante este proceso de graduación con respecto a los sistemas electrónicos embebidos ha reforzado los conocimientos de electrónica obtenidos a través de la carrera, lo que nos ayuda a poder desempeñarnos en el amplio mundo de la tecnología electrónica.
- El diseño y desarrollo de este prototipo nos ha permitido conocer más el hardware de los equipos electrónicos, así como también la programación en lenguaje Basic a través de un software diseñado para programar sistemas electrónicos.
- Al finalizar el prototipo podemos decir que le ofrecemos a la universidad un método muy viable y oportuno que les reducirá el trabajo de papelería y el gasto económico, y que además logrará captar la atención de los estudiantes.
- Logrando así finalizar nuestro proyecto de graduación y concretando la construcción de un sistema electrónico embebido como lo es una pantalla luminosa digital diseñada y desarrollada para desplegar mensajes alfanuméricos permitiendo ser actualizada cuanto sea necesario.

Recomendaciones

- Para una mejor implementación de este proyecto es recomendable invertir más dinero en la compra de componentes de mayor capacidad y/o velocidad, para lograr un óptimo funcionamiento.
- Es posible cambiar el diseño del circuito electrónico, si se desea conectar los componentes de forma diferente o si se quiere cambiar algún componente para mejorar el resultado.
- En cuanto a la capacidad de almacenamiento de la memoria donde se guarda el firmware puede variarse, según sea la finalidad del equipo así debe elegirse el microcontrolador o para obtener un mejor resultado puede utilizarse un microprocesador.
- Además, el proyecto queda abierto para que otros estudiantes puedan continuar su desarrollo, por ejemplo, pueden adaptarle un teclado para facilitar la programación del dispositivo.

ANEXOS

ANEXO 1: MATRIZ DE CONGRUENCIA

A. MATRIZ DE CONGRUENCIA

TEMA: "Diseño y construcción de una pantalla luminosa digital para el despliegue de mensajes alfanuméricos, como apoyo visual informativo en oficinas y/o pasillos de la Universidad Tecnológica de El Salvador"		
ENUNCIADO DEL PROBLEMA: ¿Cómo logrará la Escuela de Informática en la Facultad de Informática y Ciencias Aplicadas de la Universidad Tecnológica de El Salvador obtener una herramienta moderna que les permita proporcionar mensajes informativos en oficinas y pasillos de forma practica y actualizada, que logre la atención visual de los estudiantes?		
Objetivo General: Desarrollar un prototipo electrónico que consiste en una pantalla luminosa digital que permita desplegar mensajes alfanuméricos, como apoyo visual informativo en oficinas y/o pasillos de la Universidad Tecnológica de El Salvador.		
Objetivo específico 1: Diseñar y construir un prototipo electrónico moderno de calidad y que sea de bonita apariencia.	Objetivo específico 2: Utilizar componentes electrónicos de calidad y durabilidad, que sean accesibles económicamente y que su consumo eléctrico sea mínimo, generando así un bajo costo económico.	Objetivo específico 3: Elaborar un código de programación en lenguaje BASIC que al descargarlo a la memoria del dispositivo esté despliegue los mensajes indicados.
Alcance 1: Proporcionar un prototipo electrónico moderno de calidad y que sea de bonita apariencia.	Alcance 2: Los componentes electrónicos que se utilizarán son de muy buena calidad lo que garantiza durabilidad a muy buen precio y su consumo eléctrico es bajo, por lo que reducirá costos en comparación a los gastos en papelería.	Alcance 3: Diseñar un código de programación en lenguaje BASIC, que indique a la memoria del prototipo que mensaje desplegar, y que pueda ser compilado y descargado por el IDE Bascom-AVR.

Producto 1: Un prototipo de una pantalla luminosa digital.	Producto 2: Los componentes son de calidad y a un costo accesible.	Producto 3: Código de programación en lenguaje BASIC.
--	--	---

ANEXO 2: HOJAS TECNICAS

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4K/8/16/32K Bytes of In-System Self-Programmable Flash program memory
 - 256/512/612/1K Bytes EEPROM
 - 512/1K/1K/2K Bytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C¹
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timers/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFNMLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips PC compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFNMLF and 32-pad QFNMLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (including 32 KHz RTC)



8-bit **AVR®**
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

ATmega48A
ATmega48PA
ATmega88A
ATmega88PA
ATmega168A
ATmega168PA
ATmega328
ATmega328P

Summary

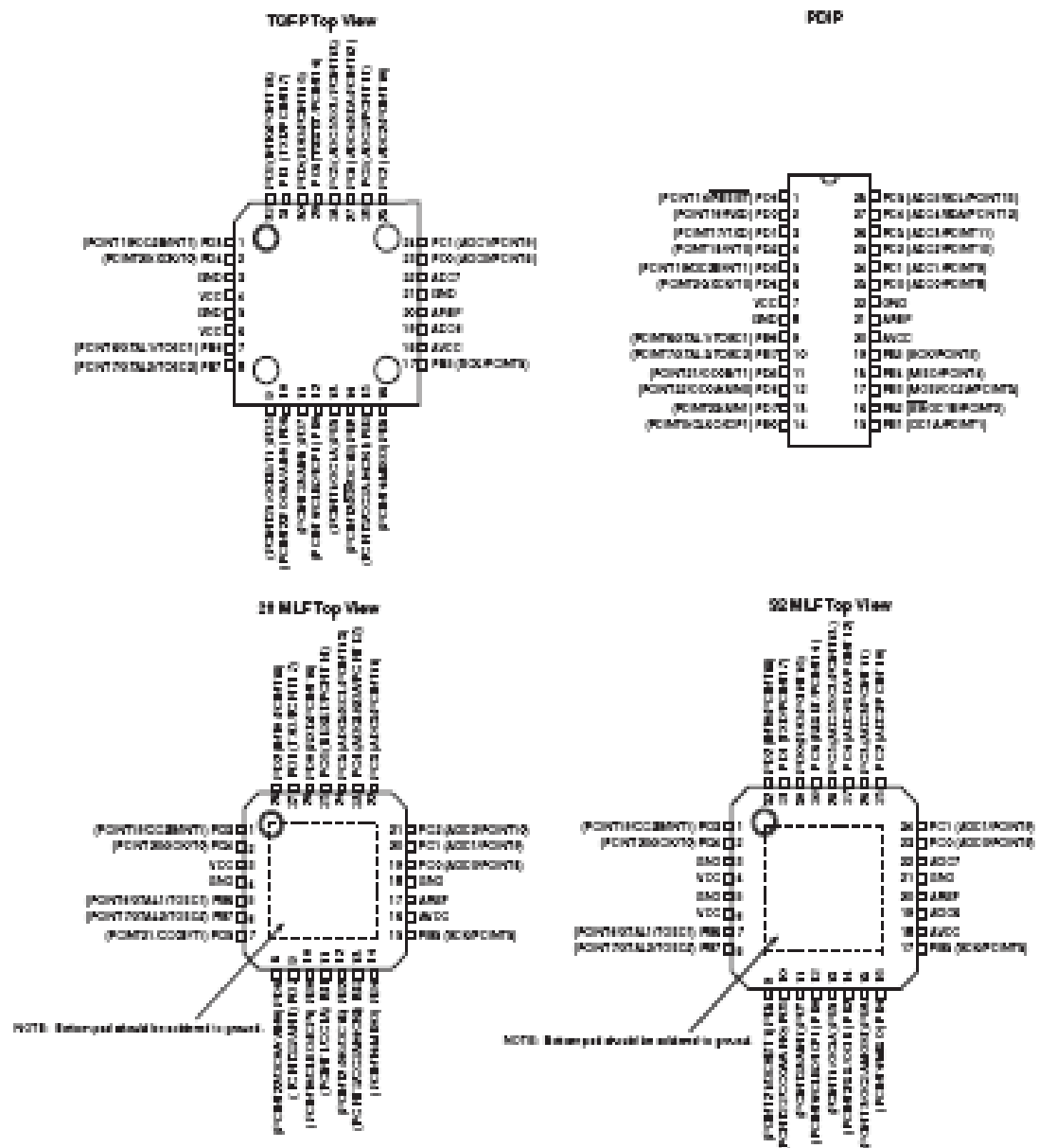
Rev. 827155-AVR-02/10



ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

1. Pin Configurations

Figure 1-1. Pinout ATmega48A/48PA/88A/88PA/168A/168PA/328/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the Inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ["Alternate Functions of Port B"](#) on page 28.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-12 on page 323](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C"](#) on page 28.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.



ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 89.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC6:0, and ADC7:6. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC} .

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.



LED DOT MATRIX

KLS9 M - 23 8 8 1 A EG - 1 1 - XXX

1 2 3 4 5 6 7 8 9 10 11

- 1 Brand:KLS ELECTRONIC
- 2 Digit Mode: M-Dot Matrix ML-Lamp Dot Matrix
- 3 Mode Number:
- 4 Column Number:
- 5 Row Number:
- 6 Model Number:
- 7 Drive Mode: A. C. E. - Row Cathode Column Anode B. D. F. - Row Anode Column Cathode
- 8 Color Code: See Part 1. 7
- 9 Reflector Surface Color: See Part 1. 8
- 10 Segment Epoxy Color: See Part 1. 9
- 11 Special Device: See Part 1. 10

DOT MATRIX DISPLAY



PACKAGE DIMENSION	INTERNAL CIRCUIT DIAGRAM
NFM-20571 Series	NFM-20571ARGB NFM-20571BRGB
NFM-23581 Series	NFM-23581Ax NFM-23581Bx
NFM-23581 Series	NFM-23581Cx NFM-23581Dx

Electrical-Optical Characteristics (Ta=25°C)

PARAMETER	SYMBOL	DEVICES	CODE	TYP	MAX	UNIT	TEST CONDITIONS
Forward Voltage	V_f	RED	H	2.25	2.60	V	$I_f=20\text{mA}$
		HI-RED	S	1.85	2.20		
		SUPER-RED	D	1.85	2.20		
		ORANGE	E	2.10	2.50		
		GREEN	G	2.20	2.50		
		YELLOW	Y	2.10	2.50		
Peak Emission Wavelength	λ_p	RED	H	700		nm	$I_f=20\text{mA}$
		HI-RED	S	660			
		SUPER-RED	D	660			
		ORANGE	E	635			
		GREEN	G	568			
		YELLOW	Y	585			
Spectral Line Half-Width	$\Delta\lambda$	RED	H	90		nm	$I_f=20\text{mA}$
		HI-RED	S	20			
		SUPER-RED	D	20			
		ORANGE	E	35			
		GREEN	G	30			
		YELLOW	Y	35			
Reverse Current	I_R	RED	H		20	μA	$V_R=5\text{V}$
		HI-RED	S				
		SUPER-RED	D				
		ORANGE	E				
		GREEN	G				
		YELLOW	Y				
Average Luminous Intensity	I_v	RED	H	500		μcd	$I_f=10\text{mA}$
		HI-RED	S	3500			
		SUPER-RED	D	6000			
		ORANGE	E	2500			
		GREEN	G	2500			
		YELLOW	Y	2000			
Segment-to-Segment Luminous Intensity Ratio	$I_{f,M}$		ALL MODELS		1.5:1		$I_f=20\text{mA}$

Absolute Maximum Rating(Ta=25°C)

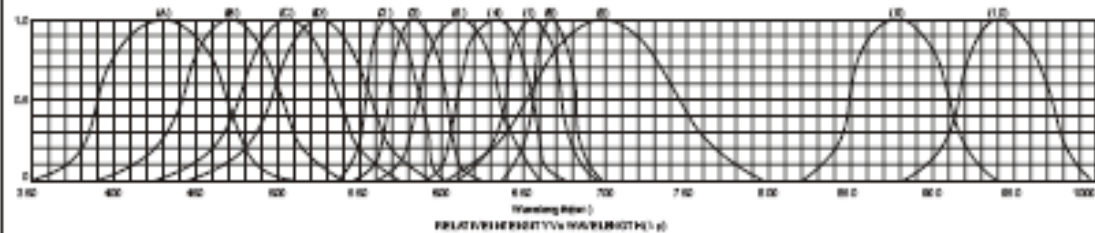
PARAMETER	RED (Gap)	HI-RED (GaAlAs/GaAs)	SUPER-RED (GaAlAs/GaAs)	ORANGE (GaAsP/Gap)	GREEN (Gap)	YELLOW (GaAsP/Gap)	UNIT
Reverse voltage V_R	5	5	5	5	5	5	V
Forward current I_f	15	25	25	30	30	30	mA
Peak forward current I_{fM}	50	150	150	150	150	150	mA
Power dissipation P_D	40	60	60	80	80	80	mW
Operating temperature T_{om}	-40~+80	-40~+80	-40~+80	-40~+80	-40~+80	-40~+80	
Storage temperature T_{st}	-40~+85	-40~+85	-40~+85	-40~+85	-40~+85	-40~+85	
Lead soldering temperature	1.60mm from body-maximum 3 second for 260±5						

Remark: Duty cycle: 1:10, Pulse width: 0.1ms

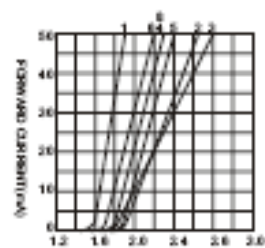
GENERAL INFORMATION



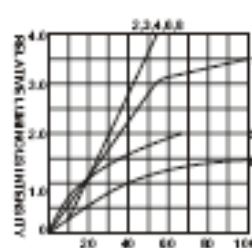
Typical Electrical-Optical Characteristics Curves



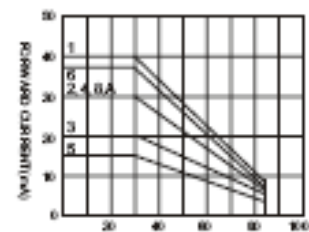
- | | |
|---|--|
| (1) - GaAlP/GaAs 655nm/Red | (9) - GaAlAs 690nm |
| (2) - GaP 565nm/Yellow Green | (10) - GaAs/GaP & GaAlAs/GaAs 940nm |
| (3) - GaAlP/GaP 585nm/Yellow | (A) - GaInSbC 430nm/Blue |
| (4) - GaAlP/GaP 635nm/Orange & Hi-Off Red | (B) - InGaAlSbC 470nm/Blue |
| (5) - GaP 700nm/Bright Red | (C) - InGaInSbC 505nm/Super Pure Green |
| (6) - GaAlAs/GaAs 660nm/Super Red | (D) - InGaInSbC 525nm/Super Bluish Green |
| (8) - GaAlP/GaP 610nm/Super Red | |



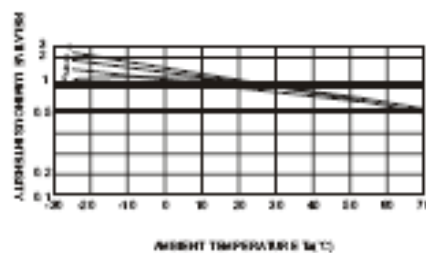
FORWARD VOLTAGE (V)
FORWARD CURRENT (mA)
FORWARD VOLTAGE



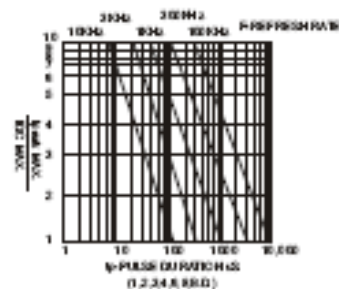
FORWARD CURRENT (mA)
RELATIVE LUMINOUS INTENSITY (%)
RELATIVE LUMINOUS INTENSITY (%)



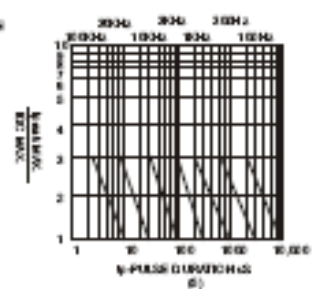
AMBIENT TEMPERATURE (°C)
FORWARD CURRENT (mA)
FORWARD CURRENT



AMBIENT TEMPERATURE (°C)



PULSE DURATION (s)



PULSE DURATION (s)

NOTE: 25°C: Free air temperature, not case or heat sink temperature

TPIC6B596 POWER LOGIC 8-BIT SHIFT REGISTER

SLUS255A – MARCH 2000 – REVISED MAY 2005

- Low $r_{DS(on)}$. . . 5 Ω
- Avalanche Energy . . . 30 mJ
- Eight Power DMOS-Transistor Outputs of 150-mA Continuous Current
- 500-mA Typical Current-Limiting Capability
- Output Clamp Voltage . . . 50 V
- Enhanced Cascading for Multiple Stages
- All Registers Cleared With Single Input
- Low Power Consumption

description

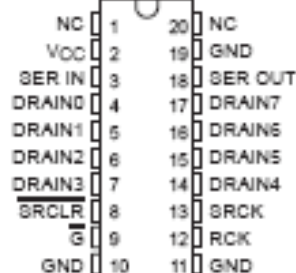
The TPIC6B596 is a monolithic, high-voltage, medium-current power 8-bit shift register designed for use in systems that require relatively high load power. The device contains a built-in voltage clamp on the outputs for inductive transient protection. Power driver applications include relays, solenoids, and other medium-current or high-voltage loads.

This device contains an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. Data transfers through both the shift and storage registers on the rising edge of the shift-register clock (SRCK) and the register clock (RCK), respectively. The storage register transfers data to the output buffer when shift-register clear (SRCLR) is high. When SRCLR is low, all registers in the device are cleared. When output enable ($\overline{G}$) is held high, all data in the output buffers is held low and all drain outputs are off. When $\overline{G}$ is held low, data from the storage register is transparent to the output buffers. When data in the output buffers is low, the DMOS-transistor outputs are off. When data is high, the DMOS-transistor outputs have sink-current capability. The serial output (SER OUT) is clocked out of the device on the falling edge of SRCK to provide additional hold time for cascaded applications. This will provide improved performance for applications where clock signals may be skewed, devices are not located near one another, or the system must tolerate electromagnetic interference.

Outputs are low-side, open-drain DMOS transistors with output ratings of 50 V and 150-mA continuous sink-current capability. Each output provides a 500-mA typical current limit at $T_C = 25^\circ\text{C}$. The current limit decreases as the junction temperature increases for additional device protection.

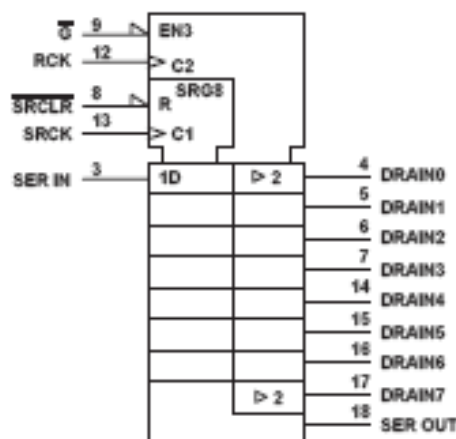
The TPIC6B596 is characterized for operation over the operating case temperature range of -40°C to 125°C .

DW OR N PACKAGE
(TOP VIEW)



NC = No internal connection

logic symbol



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.



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

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



TEXAS
INSTRUMENTS

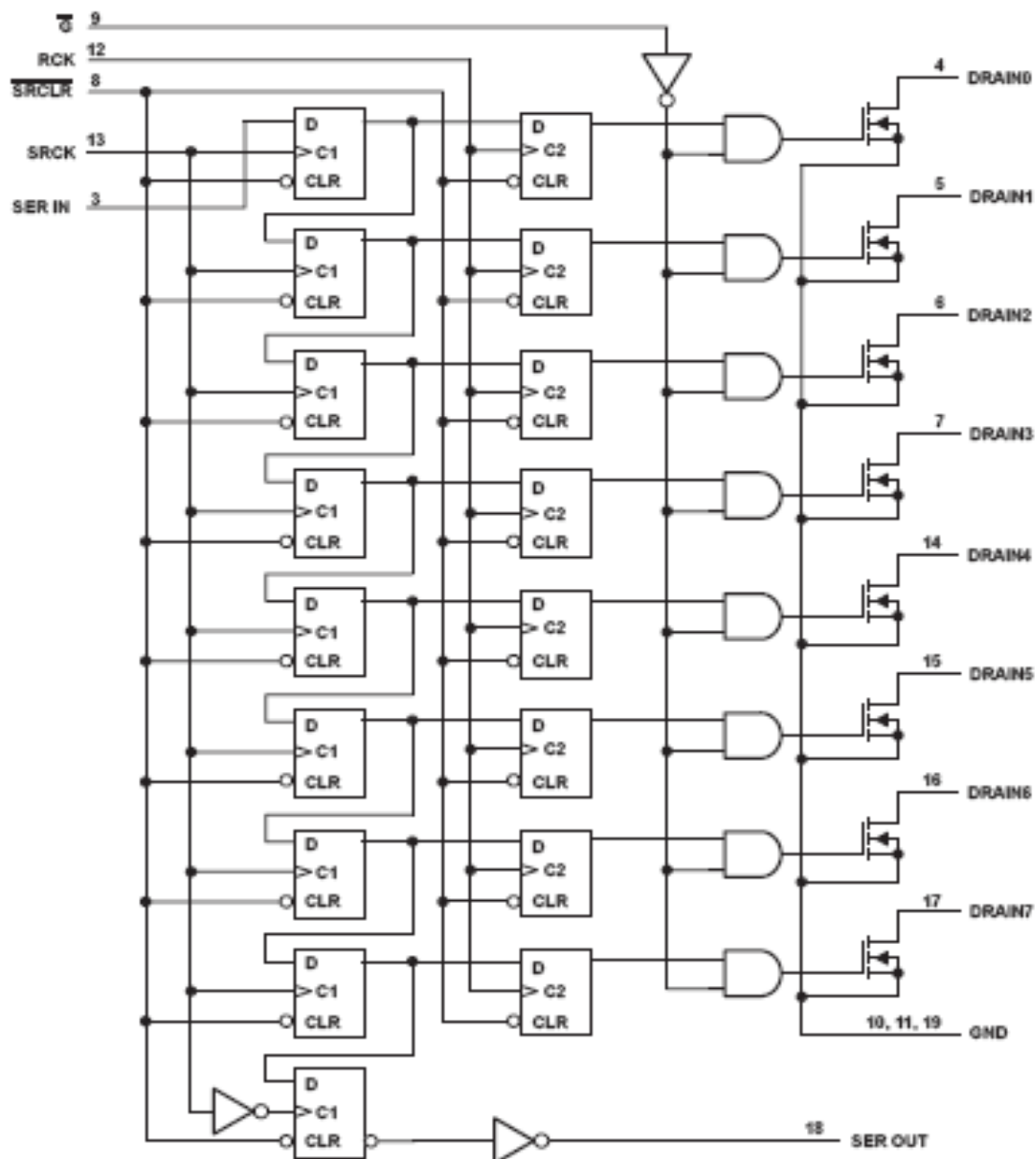
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2000 – 2005, Texas Instruments Incorporated

TPIC6B596 **POWER LOGIC 8-BIT SHIFT REGISTER**

SLIS005A – MARCH 2000 – REVISED MAY 2005

logic diagram (positive logic)



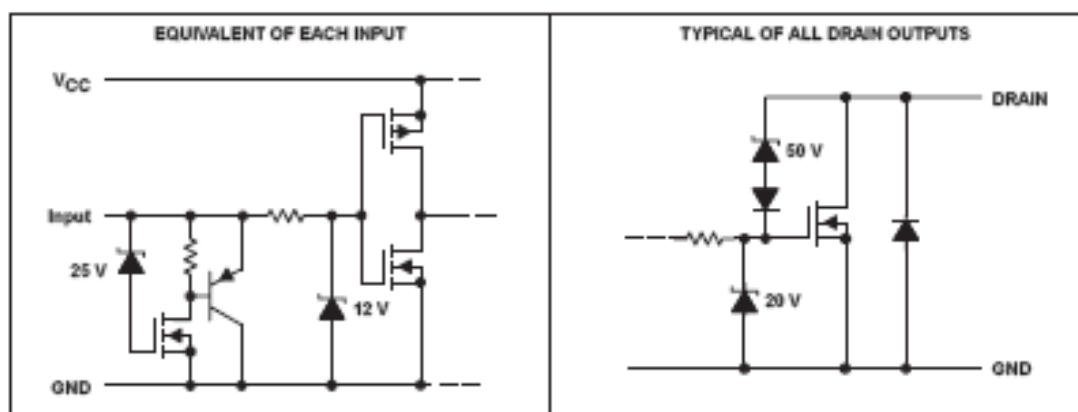
**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655393 • DALLAS, TEXAS 75285

TPIC6B596 POWER LOGIC 8-BIT SHIFT REGISTER

SLUS255A – MARCH 2000 – REVISED MAY 2005

schematic of inputs and outputs



absolute maximum ratings over recommended operating case temperature range (unless otherwise noted)[†]

Logic supply voltage, V_{CC} (see Note 1)	7 V
Logic input voltage range, V_I	-0.3 V to 7 V
Power DMOS drain-to-source voltage, V_{DS} (see Note 2)	50 V
Continuous source-to-drain diode anode current	500 mA
Pulsed source-to-drain diode anode current (see Note 3)	1 A
Pulsed drain current, each output, all outputs on, I_D , $T_C = 25^\circ\text{C}$ (see Note 3)	500 mA
Continuous drain current, each output, all outputs on, I_D , $T_C = 25^\circ\text{C}$	150 mA
Peak drain current single output, I_{DM} , $T_C = 25^\circ\text{C}$ (see Note 3)	500 mA
Single-pulse avalanche energy, E_{AS} (see Figure 4)	30 mJ
Avalanche current, I_{AS} (see Note 4)	500 mA
Continuous total dissipation	See Dissipation Rating Table
Operating virtual junction temperature range, T_J	-40°C to 150°C
Operating case temperature range, T_C	-40°C to 125°C
Storage temperature range	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

[†] Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. All voltage values are with respect to GND.
2. Each power DMOS source is internally connected to GND.
3. Pulse duration $\leq 100 \mu\text{s}$ and duty cycle $\leq 2\%$.
4. DRAIN supply voltage = 15 V, starting junction temperature (T_{JS}) = 25°C, $L = 200 \text{ mH}$, $I_{AS} = 0.5 \text{ A}$ (see Figure 4).

DISSIPATION RATING TABLE

PACKAGE	$T_C \leq 25^\circ\text{C}$ POWER RATING	DERATING FACTOR ABOVE $T_C = 25^\circ\text{C}$	$T_C = 125^\circ\text{C}$ POWER RATING
DW	1389 mW	11.1 mW/°C	278 mW
N	1050 mW	10.5 mW/°C	263 mW

TPIC6B596

POWER LOGIC 8-BIT SHIFT REGISTER

SL15095A – MARCH 2000 – REVISED MAY 2005

recommended operating conditions

	MIN	MAX	UNIT
Logic supply voltage, V_{CC}	4.5	5.5	V
High-level input voltage, V_{IH}	0.85 V_{CC}		V
Low-level input voltage, V_{IL}		0.15 V_{CC}	V
Pulsed drain output current, $T_C = 25^\circ\text{C}$, $V_{CC} = 5\text{ V}$ (see Notes 3 and 5)	-500	500	mA
Setup time, SER IN high before SRCK $\uparrow$, t_{SU} (see Figure 2)	15		ns
Hold time, SER IN high after SRCK $\uparrow$, t_H (see Figure 2)	15		ns
Pulse duration, t_W (see Figure 2)	40		ns
Operating case temperature, T_C	-40	125	$^\circ\text{C}$

NOTES: 3. Pulse duration $\leq 100\text{ }\mu\text{s}$ and duty cycle $\leq 2\%$.
5. Technique should limit $T_J - T_C$ to 10°C maximum.

electrical characteristics, $V_{CC} = 5\text{ V}$, $T_C = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$V_{(BR)DSX}$ Drain-to-source breakdown voltage	$I_D = 1\text{ mA}$	50			V
V_{SD} Source-to-drain diode forward voltage	$I_F = 100\text{ mA}$		0.85	1	V
V_{OH} High-level output voltage, SER OUT	$I_{OH} = -20\text{ }\mu\text{A}$, $V_{CC} = 4.5\text{ V}$	4.4	4.49		V
	$I_{OH} = -4\text{ mA}$, $V_{CC} = 4.5\text{ V}$	4	4.2		
V_{OL} Low-level output voltage, SER OUT	$I_{OL} = 20\text{ }\mu\text{A}$, $V_{CC} = 4.5\text{ V}$		0.005	0.1	V
	$I_{OL} = 4\text{ mA}$, $V_{CC} = 4.5\text{ V}$		0.3	0.5	
I_{IH} High-level input current	$V_{CC} = 5.5\text{ V}$, $V_I = V_{CC}$			1	μA
I_{IL} Low-level input current	$V_{CC} = 5.5\text{ V}$, $V_I = 0$			-1	μA
I_{CC} Logic supply current	$V_{CC} = 5.5\text{ V}$	All outputs off	20	100	μA
		All outputs on	150	300	
$I_{CC(FREQ)}$ Logic supply current at frequency	$f_{SRCK} = 5\text{ MHz}$, $C_L = 30\text{ pF}$, All outputs off, See Figures 2 and 6		0.4	5	mA
I_N Nominal current	$V_{DS(on)} = 0.5\text{ V}$, $I_N = I_D$, $T_C = 85^\circ\text{C}$, See Notes 5, 6, and 7		90		mA
I_{DSX} Off-state drain current	$V_{DS} = 40\text{ V}$, $V_{CC} = 5.5\text{ V}$		0.1	5	μA
	$V_{DS} = 40\text{ V}$, $V_{CC} = 5.5\text{ V}$, $T_C = 125^\circ\text{C}$		0.15	8	
$r_{DS(on)}$ Static drain-source on-state resistance	$I_D = 100\text{ mA}$, $V_{CC} = 4.5\text{ V}$		4.2	5.7	Ω
	$I_D = 100\text{ mA}$, $T_C = 125^\circ\text{C}$, $V_{CC} = 4.5\text{ V}$, See Notes 5 and 6 and Figures 7 and 8		6.8	9.5	
	$I_D = 350\text{ mA}$, $V_{CC} = 4.5\text{ V}$		5.5	8	

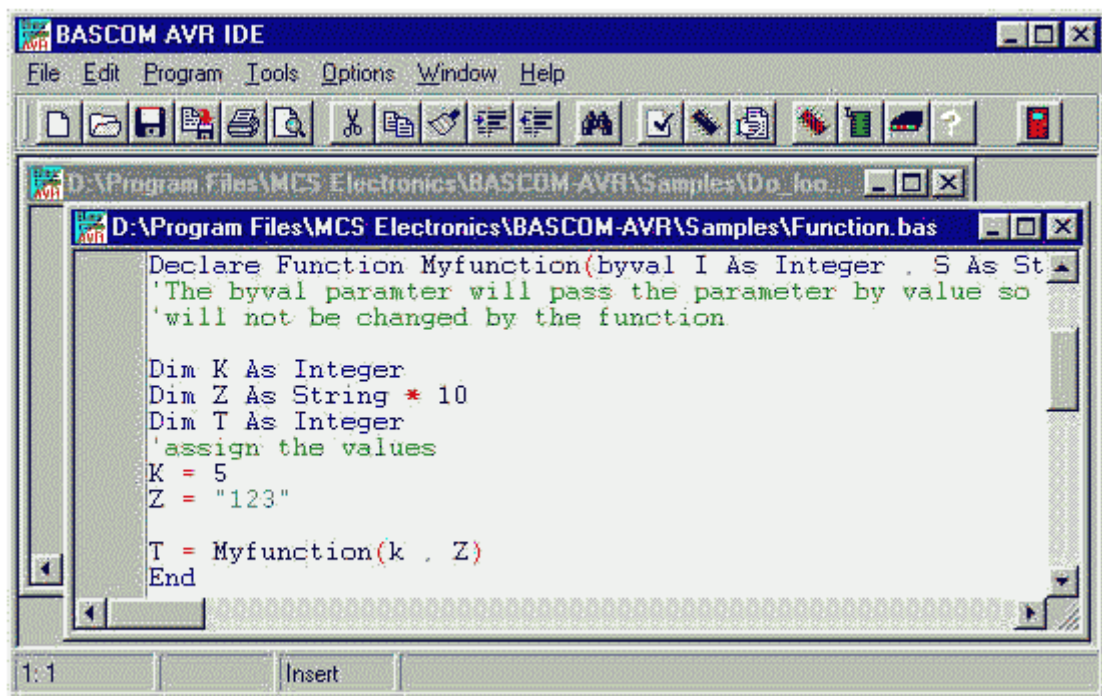
NOTES: 5. Technique should limit $T_J - T_C$ to 10°C maximum.
6. These parameters are measured with voltage-sensing contacts separate from the current-carrying contacts.
7. Nominal current is defined for a consistent comparison between devices from different sources. It is the current that produces a voltage drop of 0.5 V at $T_C = 85^\circ\text{C}$.

US\$112.00

Atmel AVR BASIC Compiler (BASCOM)

BSCAVR

BASCOM-AVR© is an Integrated Development Environment that includes a BASIC Compiler for the Atmel AVR microcontroller family, Editor, AVR Simulator and In-System Programming support for a range of 3rd party hardware. It is designed to run on W95/W98/NT/W2000/XP and Vista.



BASCOM-AVR Key Benefits

- Structured BASIC with labels.
- Structured programming with IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- Fast machine code instead of interpreted code.
- Variables and labels can be as long as 32 characters.

- Bit, Byte, Integer, Word, Long, Single and String variables.
- Large set of Trig Floating point functions. Date & Time calculation functions.
- Compiled programs work with all AVR microprocessors that have internal memory.
- Statements are highly compatible with Microsoft's VB/QB.
- Special commands for LCD-displays , I2C chips and 1WIRE chips, PC keyboard, matrix keyboard, RC5 reception, software UART, SPI , graphical LCD, send IR RC5, RC6 or Sony code. TCP/IP with W3100A chip.
- Local variables, user functions, library support.
- Integrated terminal emulator with download option.
- Integrated simulator for testing.
- Integrated ISP programmer (application note AVR910.ASM).
- Many other programmers supported via the Universal Interface.
- Editor with statement highlighting.
- Context sensitive help.
- DEMO version compiles 4KB of code. Well suited for the AT90S2313.
- Special tcp/ip library, AT mouse simulator, AT keyboard simulator available as add-ons.

The following statements are supported (actually there are many more look in the helpfile):

Decision and Structures

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, STEP, EXIT FOR, ON .. GOTO/GOSUB, SELECT, CASE.

Input and Output

PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIFTLIN, SHIFTLIN, GETATKBD, SPC, SERIN, SEROUT

Numeric Functions

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

I2C Interface

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.

1WIRE Interface

1WWRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT.

SPI Interface

SPIINIT, SPIIN, SPIOUT, SPIMOVE.

Atmel AVR Interrupt Programming

ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, COUNTER_x, CAPTURE_x, INTERRUPTS, CONFIG, START, LOAD.

Bit Manipulation

SET, RESET, ROTATE, SHIFT, BITWAIT, TOGGLE.

Variables

DIM, BIT , BYTE , INTEGER , WORD, LONG, SINGLE, STRING , DEFBIT, DEFBYTE, DEFINT, DEFWORD.

Miscellaneous

REM, ' , SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB, POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC, MAKEBCD, INP,OUT, ALIAS, DIM , ERASE, DATA, READ, RESTORE, INCR, DECR, PEEK, POKE, CPEEK, FUNCTION, READMAGCARD, BIN2GREY, GREY2BIN, CRC8, CRC16, CHECKSUM.

Compiler Directives

\$INCLUDE, \$BAUD and \$CRYSTAL, \$SERIALINPUT, \$SERIALOUTPUT, \$RAMSIZE, \$RAMSTART, \$DEFAULT XRAM, \$ASM-\$END ASM, \$LCD, \$EXTERNAL, \$LIB.

String manipulation

STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

..... and many other functions, statements and directives.

To build a program takes just a few steps :

- Write the program in BASIC
- Compile it to fast machine binary code
- Test the result with the integrated simulator.
- Program the Atmel AVR micro with one of the supported programmers.
(Programming hardware must be purchased separately).

Integrated Editor

The program can be written in a comfortable MDI color coded editor. Besides the normal editing features, the editor supports Undo, Redo, Bookmarks and block indentation.

Integrated Atmel AVR Simulator

The simulator lets you test your program before writing it to the AVR microcontroller. You can watch variables, step through the program one line at a time, run to a specific line, or alter variables. To watch a variable's value you simply point the mouse cursor over it.

A powerful feature is the built-in hardware simulator, to simulate an LCD display and the ports. The LCD simulator can also simulate custom build LCD characters.

Download a [demo version of the BASCOM AVR BASIC Compiler](#). (compiles 4k of code) (download file size: 20.6MB)

Ships	from:	USA
Leadtime: Ex-stock		

You can purchase the full version of this compiler as a web-download from our website (36.2MB), a registration key to activate the software will be emailed to you within 24 hours. Alternatively you may order the compiler on CD-ROM

ANEXO 3: FIRMWARE O CODIGO DEL PROYECTO

```

' * Titulo      : Dot Matrix Display text.bas
' * Version    : 1.0
' * Microcontrolador : ATMEGA 8 @ 20Mhz
' * Compilador   : BASCOM AVR
' * Hardware    : Matrices NFH23581BG / TPIC6B595
' * Descripcion  : Despliega tipo Scroll los mensajes almacenados en las
' *              variables texto_msj1, texto_msj2, texto_msj3 y texto_msj4
' *              cada mensajes puede ser de 50 caracteres alfanumericos,
' *              mayusculas, minusculas, numeros y caracteres especiales.
' *
' *
' * *****

' * Directivas
$regfile = "M8def.dat"
$crystal = 10000000

' * Configuraciones
Config Portb = Output
Config Portd = Output

' * Declaracion de variables & constantes
Dim Mensaje_1 As String * 60
Dim Mensaje_2 As String * 60
Dim Mensaje_3 As String * 60
Dim Mensaje_4 As String * 60
Dim Str_text As String * 85
Dim Mensaje As String * 60
Dim Cuadro_temp(24) As Byte
Dim Str_temp As String * 2 , Reeks(450) As Byte
Dim Str_len As Word , Col_len As Word , R As Word
Dim Tempmsb As Byte , Temppls As Byte , Tempstr As String * 2
Dim Char As Byte , Digit As Byte , L As Byte
Dim Re As Word , Temp(2) As Byte , Temp2 As Byte
Dim Scroll_speed As Byte
Dim B As Byte : Dim X As Byte
Dim I As Word : Dim V As Byte
Dim C As Word : Dim W As Word
Dim T As Byte : Dim Q As Byte
Dim O As Byte : Dim Velocidad As Byte
Dim Str_space As String * 6
Dim Str_text_2 As String * 8
Dim Total_columnas As Byte
Dim Velocidad_texto As Byte
Declare Sub Scroll(mensaje As String)

' * Inicializaciones

```

```

Portd = 0
Portb = 0
Total_columnas = 24
Mensaje_1 = "Hola Mundo UTEC 2010"
Mensaje_2 = "abcdefghijklmnopqrstuvwxyz_ABCDEFJHIJKLMNOPQRSTUVWXYZ"
Mensaje_3 = "mensajes hasta 60 caracteres"
Mensaje_4 = "by ATMEGA powered"
Velocidad_texto = 18
' * Gosub Prueba

' * Programa Principal
Do
'//*****
  Call Scroll(mensaje_1)
  Waitms 500
  Call Scroll(mensaje_2)
  Waitms 500
  Call Scroll(mensaje_3)
  Waitms 500
  Call Scroll(mensaje_4)
  Waitms 500
Loop
End
' * Fin del programa principal

'//Subrutina para el scroll del mensaje
*****
Sub Scroll(mensaje As String)
Str_text = " " + Mensaje
Str_len = Len(str_text)
Col_len = Str_len * 6
C = 1
For Digit = 1 To Str_len
  Str_temp = Mid(str_text , Digit , 1)
  Char = Asc(str_temp)
  Re = Char * 5
  For I = 1 To 5
    Reeks(c) = Lookup(re , Datos_texto)
    Re = Re + 1
    C = C + 1
  Next I
  Reeks(c) = 0
  C = C + 1
Next Digit
C = 0
For I = 0 To Col_len
  C = I
  For B = 0 To Total_columnas
    Cuadro_temp(b) = Reeks(c)
    W = W + 1
  Next B
  Incr C

```

```

        Next B
        Velocidad = Velocidad_texto
        Portb = 0
        Portd = 0
        Gosub Desplegar

    Next I
    For C = 0 To 450
        Reeks(c) = 0
    Next C
End Sub

'//Subrutina auxiliar que despliega un frame de 32 columnas
'*****

Desplegar:
    For V = 0 To Velocidad
        Set Portb.3
        For X = 1 To Total_columnas          '***numero de framesd
            Toggle Portb.1
            Reset Portb.3
            Toggle Portb.1

            Toggle Portb.2                    'rck pin7

            Toggle Portb.2
            Portd = Cuadro_temp(x)
            Waitus 150
            Portd = 0
            Portb = 0
        Next X
    Next V
Return

'//Tabla de patrones de caracteres alfanumericos
'//para scroll de mensajes.
'//*****

Datos_texto:
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '
Data 0 , 0 , 0 , 0 , 0 , 0      '

```

```
'space
'!'
'''
'#
'$
'%
'&
"
'('
')'
'*
'+
','
'_
':'
'/
'0
'1
'2
'3
'4
'5
'6
'7
'8
'9
':.'
';'
'<
'='
'>
```

Data 2 , 1 , 81 , 9 , 6	'?
Data 50 , 73 , 121 , 65 , 62	'@
Data 124 , 18 , 17 , 18 , 124	'A
Data 65 , 127 , 73 , 73 , 62	'B
Data 62 , 65 , 65 , 65 , 34	'C
Data 65 , 127 , 65 , 65 , 62	'D
Data 127 , 73 , 73 , 65 , 65	'E
Data 127 , 9 , 9 , 1 , 1	'F
Data 62 , 65 , 73 , 73 , 58	'G
Data 127 , 8 , 8 , 8 , 127	'H
Data 0 , 65 , 127 , 65 , 0	'I
Data 32 , 64 , 65 , 63 , 1	'J
Data 127 , 8 , 20 , 34 , 65	'K
Data 127 , 64 , 64 , 64 , 64	'L
Data 127 , 2 , 12 , 2 , 127	'M
Data 127 , 2 , 4 , 8 , 127	'N
Data 62 , 65 , 65 , 65 , 62	'O
Data 127 , 9 , 9 , 9 , 6	'P
Data 62 , 65 , 65 , 33 , 94	'Q
Data 127 , 9 , 25 , 41 , 70	'R
Data 38 , 73 , 73 , 73 , 50	'S
Data 1 , 1 , 127 , 1 , 1	'T
Data 63 , 64 , 64 , 64 , 63	'U
Data 7 , 24 , 96 , 24 , 7	'V
Data 127 , 32 , 24 , 32 , 127	'W
Data 99 , 20 , 8 , 20 , 99	'X
Data 3 , 4 , 120 , 4 , 3	'Y
Data 97 , 81 , 73 , 69 , 67	'Z
Data 0 , 127 , 65 , 65 , 0	'[
Data 2 , 4 , 8 , 16 , 32	'\
Data 0 , 65 , 65 , 127 , 0	']
Data 4 , 2 , 1 , 2 , 4	'^
Data 64 , 64 , 64 , 64 , 64	'_
Data 0 , 0 , 7 , 11 , 0	'"
Data 32 , 84 , 84 , 84 , 56	'a
Data 127 , 40 , 68 , 68 , 56	'b
Data 0 , 56 , 68 , 68 , 68	'c
Data 56 , 68 , 68 , 40 , 127	'd
Data 56 , 84 , 84 , 84 , 24	'e
Data 8 , 126 , 9 , 9 , 2	'f
Data 8 , 84 , 84 , 84 , 60	'g
Data 127 , 16 , 8 , 8 , 112	'h
Data 0 , 68 , 125 , 0 , 0	'i
Data 32 , 64 , 68 , 61 , 0	'j
Data 0 , 127 , 16 , 40 , 68	'k
Data 0 , 65 , 127 , 64 , 0	'l
Data 124 , 4 , 120 , 4 , 120	'm
Data 124 , 8 , 4 , 4 , 120	'n
Data 56 , 68 , 68 , 68 , 56	'o
Data 124 , 20 , 20 , 20 , 8	'p
Data 8 , 20 , 20 , 20 , 124	'q
Data 124 , 8 , 4 , 4 , 8	'r

Data 72 , 84 , 84 , 84 , 36	's
Data 0 , 4 , 63 , 68 , 68	't
Data 60 , 64 , 64 , 32 , 124	'u
Data 28 , 32 , 64 , 32 , 28	'v
Data 60 , 64 , 60 , 64 , 60	'w
Data 68 , 40 , 16 , 40 , 68	'x
Data 12 , 80 , 80 , 80 , 60	'y
Data 68 , 100 , 84 , 76 , 68	'z
Data 0 , 8 , 54 , 65 , 0	'{'
Data 0 , 0 , 119 , 0 , 0	' '
Data 0 , 65 , 54 , 8 , 0	'}'
Data 2 , 1 , 2 , 4 , 2	'~'
Data 42 , 85 , 42 , 85 , 42	

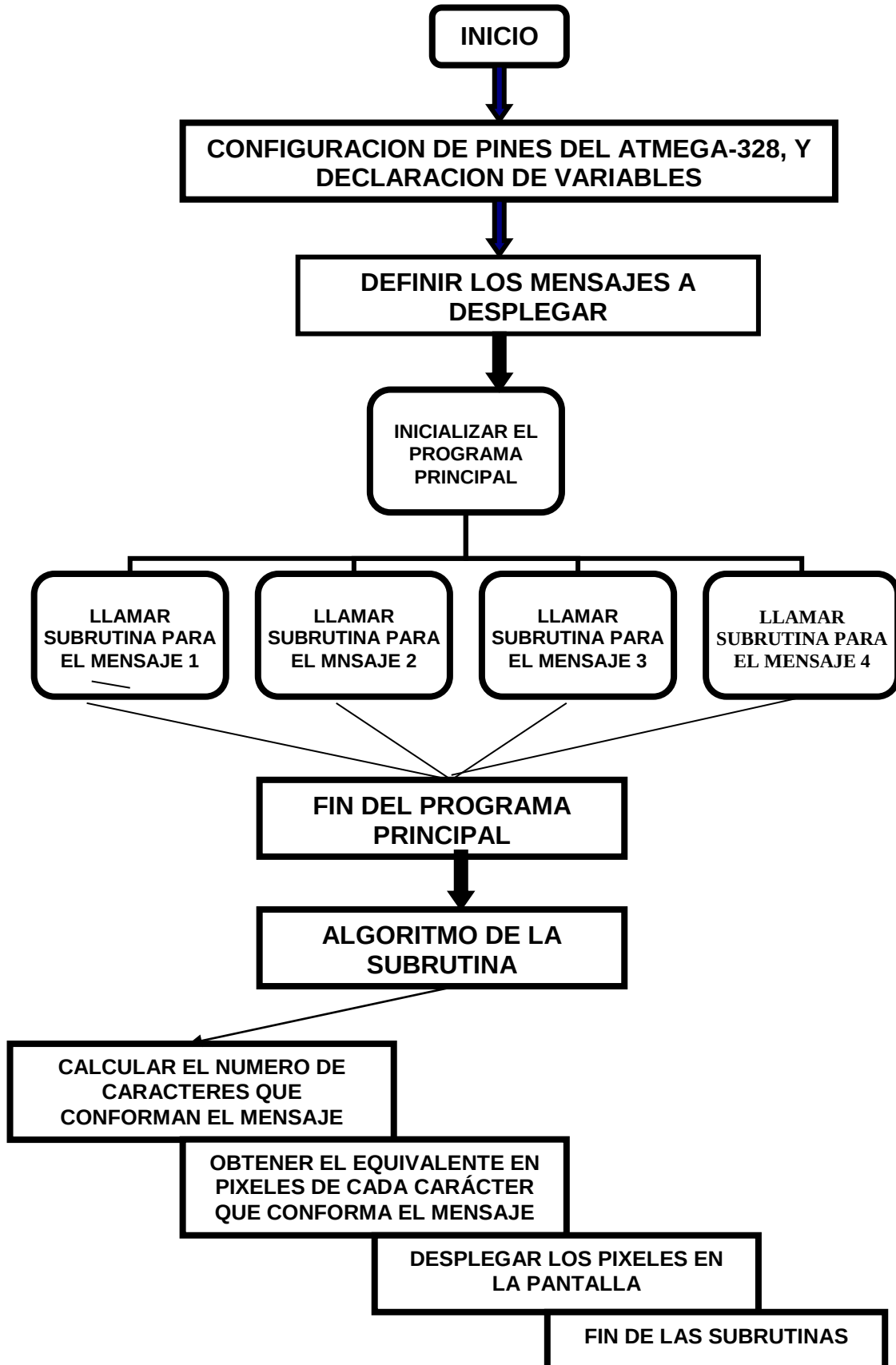
ANEXO 4: ALGORITMO DEL PROYECTO

ALGORITMO

- 1. Iniciar el programa principal**
- 2. configuración de salidas**
- 3. declarar todas las variables a utilizar**
- 4. inicializar el programa**
- 5. definir los mensajes a desplegar**
- 6. inicializar el programa principal**
- 7. llamar la subrutina scroll para el despliegue del mensaje 1**
- 8. esperar 500 milisegundos**
- 9. llamar la subrutina scroll para el despliegue del mensaje 2**
- 10. esperar 500 milisegundos**
- 11. llamar la subrutina scroll para el despliegue del mensaje 3**
- 12. esperar 500 milisegundos**
- 13. llamar la subrutina scroll para el despliegue del mensaje 4**
- 14. esperar 500 milisegundos**
- 15. fin del programa principal.**
- 16. Algoritmo de la subrutina**
- 17. calcular el numero de caracteres que conforman el mensaje**
- 18. obtener el numero en píxeles de cada carácter que conforma el mensaje**
- 19. desplegar los píxeles en la pantalla**
- 20. fin de las subrutinas.**

ANEXO 5: FLUJOGRAMA DEL PROYECTO

FLUJOGRAMA



ANEXO 6: GUIA DE USUARIO PARA LA PROGRAMACION DEL PROTOTIPO

GUIA DE USUARIO PARA LA PROGRAMACION Y REPROGRAMACION DEL PROTOTIPO

Para realizar el procedimiento de programación y reprogramación del CPU del circuito, el usuario debe cumplir los siguientes requisitos.

Para la realización de este procedimiento el usuario debe tener los conocimientos básicos necesarios, debe ser alguien que conozca sobre la programación de microcontroladores, en éste caso de la familia AVR del fabricante ATMEL.

Y para iniciar el proceso debe contarse con las siguientes herramientas:

- Un Breadboard
- Ensamblar el circuito en la Bradford para la descarga del firmware hacia el microcontrolador.
- Una interfaz que permita la comunicación entra la PC y el circuito en el que se conecta el microcontrolador, puede ser desde el puerto paralelo.
- Tener previamente instalado el software IDE BASCOM-AVR.
- Digitar el programa que se incluye en el documento escrito en lenguaje Basic.

Para iniciar el proceso de programación o reprogramación debe apagarse el prototipo y extraer el microcontrolador.

Luego de extraerlo se ubica en el circuito que debe ensamblarse el la breadbord. En este circuito se indican las entradas y las salidas del microcontrolador para su correcto funcionamiento.

Tal como lo muestra la imagen siguiente

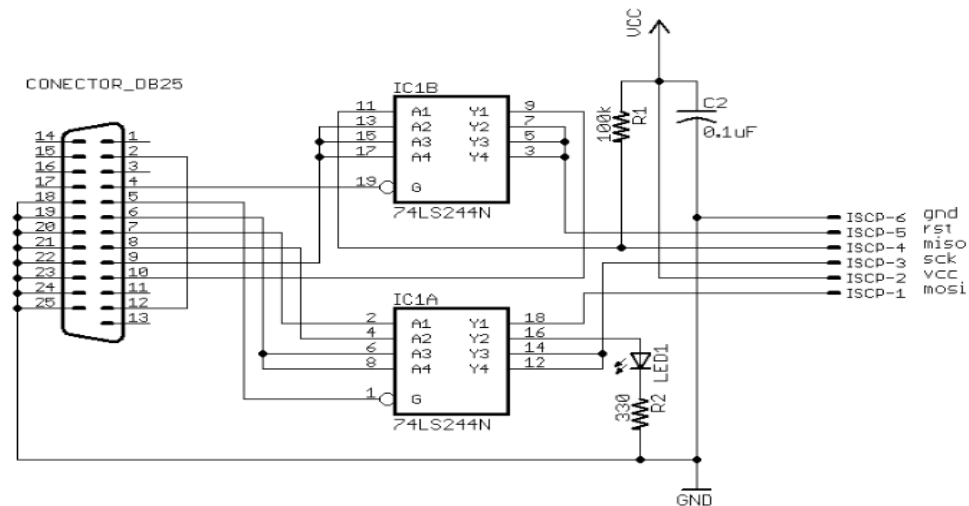


Fig. 1: ensamblado en breadbord

Debe tomarse en cuenta que el circuito debe ser ensamblado exactamente como lo muestra la imagen anterior para evitar errores, fallas o lo peor quemar el micro.

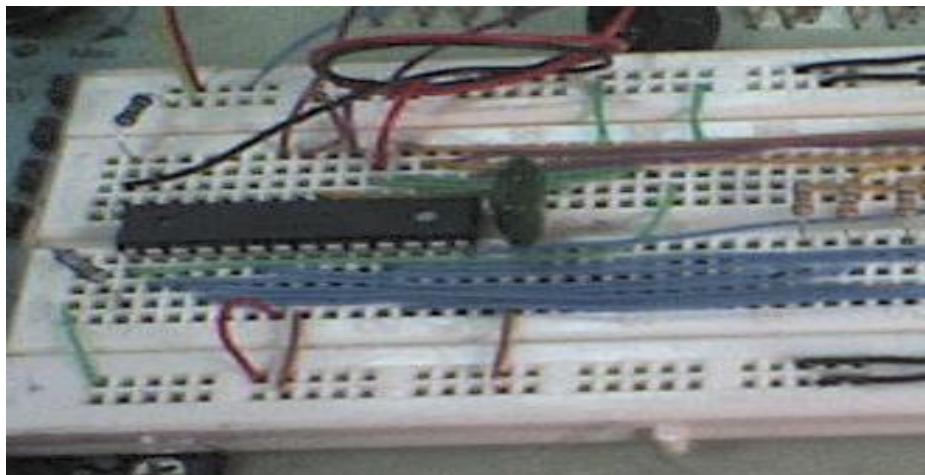


Fig. 2: microcontrolador ensamblado en breadbord.

Ya teniendo el micro correctamente ensamblado en el breadbord se conecta la corriente y se procede al área del firmware.

Para ello el usuario debe tener previamente instalado el IDE Bascom-AVR y escribir el código de programación que se incluye en el documento.

Para iniciar, se abre el programa y en el área de escritura se digita el programa, como lo muestra la siguiente imagen.

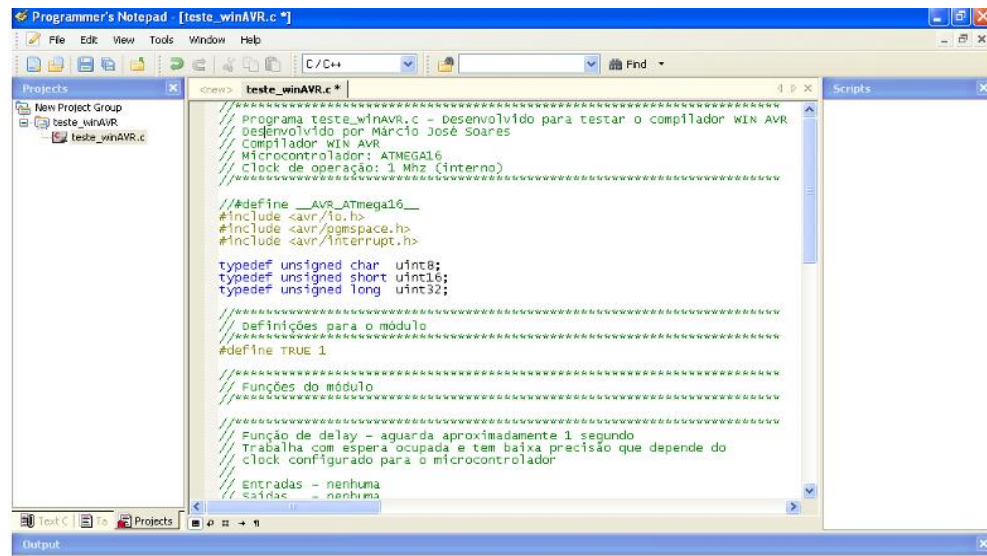


Fig. 3: escritura del código en el IDE Bascom

Dentro del programa, en las variables que indican los mensajes a desplegar se hace referencia a los cuatro mensajes consecutivos que se pueden mostrar en esta pantalla.

Los cuales se muestran en la siguiente imagen:

```
' * Inicializaciones
Portd = 0
Portb = 0
Total_columns = 24
Mensaje_1 = "Hola Mundo UTEC 2010"
Mensaje_2 =
"abcdefghijklmnpqrstuvwxyz_ABCDEFJHIJKLMNOPQRSTUVWXYZ"
Mensaje_3 = "mensajes hasta 60 caracteres"
Mensaje_4 = "by ATMEGA powered"
Velocidad_texto = 18
' * Gosub Prueba
```

Fig. 4: etiquetas que indican el mensaje a desplegar.

Estas cuatro líneas son las que se deben modificar, en ellas se indican los mensajes que la pantalla desplegara. Y para iniciar el proceso de programación debe conectarse la interfaz entre el microcontrolador y el puerto paralelo de la PC.

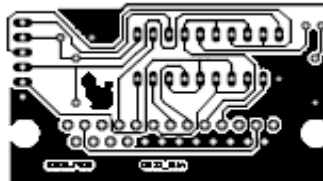
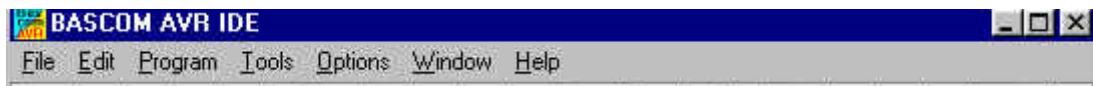


Fig. 5: interfaz de comunicación entre el microcontrolador y la PC.

Esta imagen es un ejemplo de la interfaz que se puede utilizar como puerto de comunicación entre el microcontrolador y la PC, esta interfaz se conecta en el circuito de la figura 1 y al puerto paralelo de la pc.

Ya teniendo comunicación entre el microcontrolador y la PC, y el programa listo, se compila, esto en la opción compilar que esta ubicada en la barra de herramientas del programa.



En esta barra de herramientas en la opción **Program** se debe seleccionar la opción **compiler**.

Esta opción **compilar**, se encarga de convertir el código escrito en lenguaje Basic a lenguaje maquina para que pueda ser entendido y ejecutado por el microcontrolador.

Después de compilarlo se da clic en la opción programar



Y al dar clic en el icono que muestra la imagen anterior se despliega la imagen siguiente.

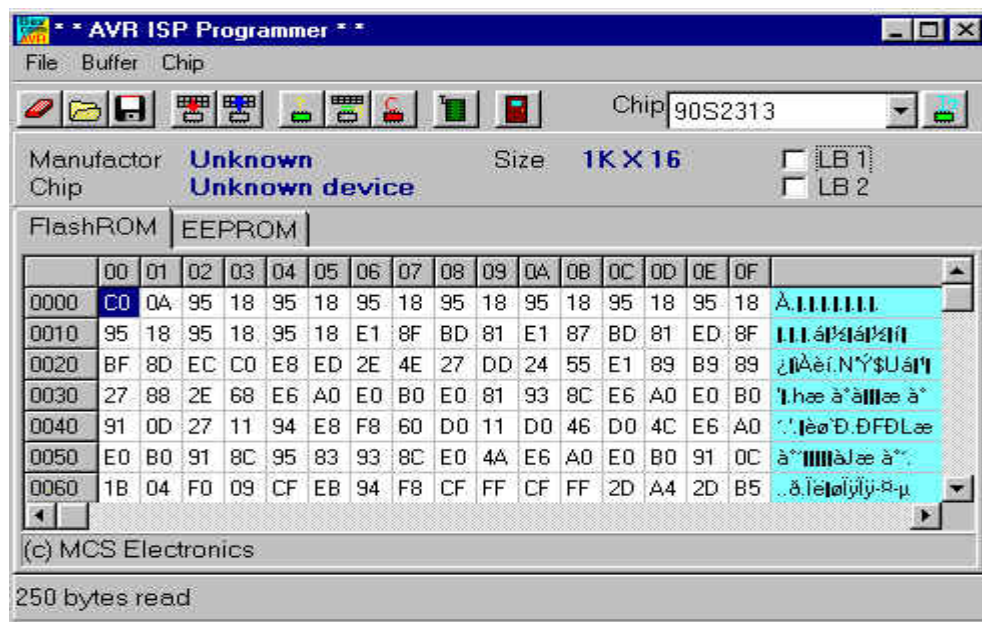
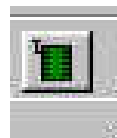


Fig 6: ventana d programación dentro del IDE Bascom

En esta área en la casilla **Chip** se elige el modelo del microcontrolador que se esta utilizando, en este caso es el **Atmega328**, y luego se da clic en programar.



Después de esto el microcontrolador ya esta programado o reprogramado. Y debe extraerse del circuito de programación y ubicarlo nuevamente en el prototipo de la pantalla.

Referencias

Apponic. (2009). *Bascom avr*. Recuperado de [http://download.bascom-avr/2/161.html](http://download.bascom-avr/download.bascom-avr/2/161.html)

Electan, C. (2008). *Avr solutions*. Recuperado de <http://www.electan.com/catalog/arduino-atmega328-microprocesador-p-2925.html>

Fundación Wikimedia Inc. (2007). *Sistema embebido*. Recuperado de http://wikipedia.org/wiki/Sistema_embebido.