

ESCUELA POLITECNICA NACIONAL

ESCUELA DE FORMACION DE TECNOLOGOS

**CONSTRUCCION DE UN SISTEMA ELECTRONICO PUBLICITARIO
CON TECNOLOGIA ATMEL PARA LA CORPORACION
ELECTRONICA TELEVID**

**PROYECTO PREVIO A LA OBTENCION DEL TITULO DE TECNOLOGO EN
ELECTRONICA Y TELECOMUNICACIONES**

LOACHAMIN GUALOTUÑA LUIS MIGUEL

luismi_lmlg@hotmail.com

DIRECTOR: ING. PATRICIO CARRASCO

pcarrasco@espe.edu.ec

Quito, Julio, 2011

DECLARACIÓN

Yo, Loachamin Gualotuña Luis Miguel, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

Loachamin Gualotuña Luis Miguel

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Loachamin Gualotuña Luis Miguel, bajo mi supervisión.

Ing. PATRICIO CARRASCO

DIRECTOR DE PROYECTO

AGRADECIMIENTOS

A mis padres que me brindaron su apoyo permitiéndome culminar con éxito mis estudios y a aquellas personas que de una u otra manera me ayudaron a la consecución de este proyecto.

A la Corporación TELEVID por permitirme cumplir una de mis metas en sus instalaciones y brindarme todas las facilidades para la culminación del proyecto.

RESUMEN

En el presente proyecto se construyó un sistema electrónico publicitario con tecnología Atmel para la Corporación Electrónica Televid.

Consta de dos capítulos y además se presentan los antecedentes, las conclusiones y recomendaciones extraídas del proyecto, así como las fuentes bibliográficas con los anexos.

En el primer capítulo se presentan los fundamentos teóricos, se definen las características principales de la tecnología Atmel y los recursos que poseen sus integrados. Se define los dispositivos electrónicos utilizados con su respectivo funcionamiento, los pasos necesarios para la creación del código de los caracteres a presentar en el rótulo. Se realiza un resumen del software Bascom Avr, con las principales instrucciones de programación utilizadas en el desarrollo del programa. También se realiza un resumen del software Eagle, indicando sus principales entornos de trabajo.

En el segundo capítulo se presenta la construcción del sistema publicitario. Este capítulo inicia con la realización de los diagramas de bloques que conforman el sistema publicitario. A continuación se desarrolla el proceso de construcción de cada bloque explicando el respectivo funcionamiento de cada uno, para luego implantar todo el sistema publicitario. También se presenta el respectivo software que controla el sistema publicitario, indicando primero el diagrama lógico del programa. Este capítulo finaliza con la presentación de las pruebas hechas en el sistema publicitario y el análisis técnico económico del sistema.

ANTECEDENTES

Los rótulos son el medio utilizado para promocionar un producto o un servicio y la finalidad de estos es lograr que una persona conozca que existe un producto, pudiendo adquirirlo o conozca que se ofrece un determinado servicio para ser utilizado o contratado cuando se requiera.

En las calles los viandantes y el tráfico rodado pueden observar carteles, gigantografías, rótulos con frases fijas, rótulos contruidos con tubos de neón y rótulos electrónicos promocionando diferentes productos o servicios, de los cuales varios no son vistos, haciendo que la persona que invirtió en el medio publicitario haya realizado una mala inversión gastando dinero, tiempo y se encuentre desperdiciando un espacio publicitario.

Un rótulo logra ser visto, mientras mayor tamaño posea o más llamativo sea presentado, si se elige la primera opción para presentar la publicidad el rótulo tiene que estar ubicado en un lugar alto donde no pueda molestar a las personas para que logre ser visto, por otro lado se invierte dinero en mantener el espacio publicitario y también si se desea presentar la publicidad en las noches, el rótulo tiene que ser iluminado. La otra opción es que la publicidad sea presentada de una manera llamativa lo que es posible realizar con un rótulo electrónico, en donde la publicidad es presentada en mensajes rotativos y la variada forma de presentación de los mismos llaman la atención del viandante, debido a los efectos que pueden tener hace que una persona observe el rótulo y lea su contenido, además estos rótulos son pequeños y sus mensajes son visibles tanto en el día como en la noche.

La cantidad de información que posea un rótulo y lo novedoso de esta, es un parámetro muy importante al momento de promocionar un producto o un servicio, y para esto los proveedores de productos o servicios siempre están al día con lo más novedoso y promocionar un producto o servicio, puede requerir una gran inversión si se desea utilizar un cartel o rótulo fijo, por otro lado en un rótulo electrónico la publicidad puede ser cambiada el momento que se requiera y puede presentar bastante y variada información.

INDICE

CAPITULO I	1
1. MARCO TEORICO	1
1.1. MICROCONTROLADOR ATMEL	1
1.1.1. ARQUITECTURA DEL MICROPROCESADOR.....	3
1.1.1.1. Descripción de los Componentes del Microprocesador.....	3
1.1.1.2. Memoria Flash.....	4
1.1.1.3. Registros de Propósito General	4
1.1.1.4. Unidad Aritmético Lógica (ALU)	4
1.1.1.5. El Registro de Estado	4
1.1.1.6. Sram	4
1.1.1.7. Eeprom	5
1.1.2. INTERRUPCIONES Y RECURSOS ESPECIALES	5
1.1.2.1. I2C.....	5
1.1.2.2. SPI.....	6
1.1.2.3. Temporizadores o "Timers".....	6
1.1.2.4. Perro Guardián o "Watchdog"	6
1.1.3. TIPOS DE ENCAPSULADOS.....	7
1.1.4. MICROCONTROLADOR ATMEGA 644.....	8
1.1.4.1. Puertos I/O (Entrada /Salida).....	9
1.2. MEMORIA DE ALMACENAMIENTO DE DATOS Y RELOJ DE TIEMPO REAL	9
1.2.1. CONTROL DE INTERFACE INTEGRADA I2C.....	10
1.2.1.1. Descripción de las Señales	10
1.2.1.2. Protocolo de Comunicación del Bus I2C.....	11
1.2.1.3. Proceso de Escritura en un Dispositivo Esclavo.....	13
1.2.1.4. Proceso de Lectura desde un Dispositivo Esclavo.....	14
1.2.2. RELOJ TIEMPO REAL DS1307.....	15
1.2.3. MEMORIA 24LS256.....	17
1.3. CREACION DEL CODIGO BINARIO	20
1.3.1. CREACION DE UN CARACTER UTILIZANDO EXCEL	20
1.3.2. ORGANIZACION DEL CODIGO CREADO.....	23
1.3.3. DIRECCIONAMIENTO INTERNO DE LA MEMORIA.....	26
1.4. TECLADO	27
1.4.1. TECLADO MATRICIAL 4 x 4.....	27
1.5. MATRIZ DE LED´S	28
1.6. OTROS DISPOSITIVOS ELECTRÓNICOS	29
1.6.1. DISPLAY LCD 16 x 2.....	29

1.6.2.	INTEGRADO ULN 2803	31
1.6.3.	INTEGRADO 74LS244.....	31
1.6.4.	INTEGRADO 74LS595.....	32
1.6.5.	DIODOS LED.....	33
1.6.6.	FUENTE DE TENSION	35
1.6.7.	REGULADORES DE TENSION	38
1.7.	LENGUAJE DE PROGRAMACIÓN BASCOM AVR.....	39
1.7.1.	PRINCIPALES SENTENCIAS DE BASCOM	41
1.7.1.1.	Directivas del Compilador.....	41
1.7.1.1.1.	<i>\$regfile</i>	41
1.7.1.1.2.	<i>\$crystal</i>	41
1.7.1.2.	Configuraciones Iniciales	41
1.7.1.2.1.	<i>Config</i>	41
1.7.1.2.2.	<i>DDRx, PORTx, PINx</i>	42
1.7.1.2.3.	<i>Alias</i>	43
1.7.1.3.	Tipos de Datos	43
1.7.1.3.1.	<i>Dim</i>	43
1.7.1.4.	Manipulación de Bits.....	44
1.7.1.4.1.	<i>Reset</i>	44
1.7.1.4.2.	<i>Toggle</i>	45
1.7.1.5.	Manipulación de Strings.....	45
1.7.1.5.1.	<i>INSTR</i>	45
1.7.1.5.2.	<i>LOOKUPSTR</i>	46
1.7.1.5.3.	<i>Mid</i>	46
1.7.1.6.	Instrucciones de Uso General	47
1.7.1.6.1.	<i>Wait</i>	47
1.7.1.6.2.	<i>Incr</i>	47
1.7.1.6.3.	<i>Decr</i>	48
1.7.1.6.4.	<i>Lookup</i>	48
1.7.1.6.5.	<i>Lookdown</i>	48
1.7.2.	SIMBOLOS OPERADORES	49
1.7.2.1.	Representación de Lógica Digital.....	50
1.7.3.	DECISION Y ESTRUCTURAS.....	50
1.7.3.1.	Do – Loop.....	50
1.7.3.2.	If – Them – Else	50
1.7.3.3.	For – Next.....	51
1.7.3.4.	Select – Case	51
1.7.3.5.	Gosub	51

1.7.4.	LCD (DISPLAY DE CRISTAL LIQUIDO).....	52
1.7.4.1.	Configuración Mediante Comandos.....	52
1.7.4.1.1.	Config Lcd.....	52
1.7.4.1.2.	Config lcdbus.....	52
1.7.4.1.3.	Config Lcdpin.....	53
1.7.4.2.	Mediante Cuadro de Dialogo.....	53
1.7.4.3.	Instrucciones Para el Control del Lcd.....	54
1.7.4.3.1.	Locate x,y.....	54
1.7.4.3.2.	Lcd " ".....	55
1.7.4.4.	Lcd Designer.....	55
1.7.5.	SIMULADOR.....	56
1.7.5.1.	Asignando una Ruta a Seguir en el Simulador.....	57
1.7.6.	ESTRUCTURA DE UN PROGRAMA.....	59
1.7.7.	PROGISP 1.6.7.....	60
1.8.	EAGLE.....	62
1.8.1.	EDITOR DE ESQUEMAS.....	63
1.8.2.	EDITOR DE LINEAS DE CONEXION.....	63
1.8.3.	EDITOR DE LIBRERIA.....	64
CAPITULO II.....		66
2.	CONSTRUCCION DEL SISTEMA PUBLICITARIO.....	66
2.1.	DIAGRAMA DE BLOQUES DEL SISTEMA PUBLICITARIO.....	66
2.2.	CONSTRUCCIÓN DEL SISTEMA PUBLICITARIO.....	67
2.2.1.	CONTROL DE DATOS Y ALMACENAMIENTO.....	67
2.2.1.1.	Microcontrolador ATmega 644.....	67
2.2.1.1.1.	Configuraciones y Test del Rótulo.....	69
2.2.1.1.2.	Salida de Publicidad.....	70
2.2.1.1.3.	Menús de Usuario.....	70
2.2.1.1.3.1.	Numero de Mensajes.....	71
2.2.1.1.3.2.	Características de Presentación de Mensajes y Líneas.....	72
2.2.1.2.	Integrado 24LC256.....	73
2.2.1.2.1.	Lectura y Escritura de un dato en la memoria 24LC256.....	75
2.2.1.3.	Integrado De Reloj DS1307.....	77
2.2.1.4.	Cerebro de Control.....	79
2.2.2.	INGRESO Y VERIFICACION DE INFORMACION.....	82
2.2.2.1.	Tarjeta de Teclado.....	82
2.2.2.1.1.	Descripción de los Componentes del Teclado.....	83
2.2.2.2.	Habilitación de Teclas de Menús y Visualización de Mensajes.....	85
2.2.2.2.1.	Primera Sección.....	85

2.2.2.2.2.	<i>Segunda Sección</i>	86
2.2.2.3.	Menús de Trabajo	86
2.2.2.3.1.	<i>Menú de Configuraciones Iniciales</i>	86
2.2.2.3.2.	<i>Menú para Configuraciones de Mensajes</i>	88
2.2.2.3.3.	<i>Menú Principal</i>	88
2.2.2.3.4.	Edición de Mensajes	89
2.2.2.3.5.	<i>Menú Efectos</i>	91
2.2.2.3.6.	<i>Menú Insertar Reloj</i>	92
2.2.2.3.7.	<i>Menú Nivel de Brillo</i>	94
2.2.3.	VISUALIZACION DE DATOS	95
2.2.3.1.	Tipos de Letra	95
2.2.3.2.	Efectos de Presentación de Líneas	100
2.2.3.2.1.	<i>Efecto 1 - Texto Corrido</i>	100
2.2.3.2.2.	<i>Efecto 2 - Avanza Detiene</i>	101
2.2.3.2.3.	<i>Efecto 3 - Avanza Flash</i>	102
2.2.3.2.4.	<i>Efecto 4 - Presenta Texto</i>	103
2.2.3.2.5.	<i>Efecto 5 – Destellante</i>	104
2.2.3.2.6.	<i>Efecto 6 - Abrir Texto 1</i>	104
2.2.3.2.7.	<i>Efecto 7 - Abrir Texto 2</i>	105
2.2.3.2.8.	<i>Efecto 8 - Abrir Texto 3</i>	106
2.2.3.2.9.	<i>Efecto 9 - Abrir Texto 4</i>	106
2.2.3.2.10.	<i>Efecto 10 - Ascender Texto</i>	107
2.2.3.2.11.	<i>Efecto 11 - Descender Texto</i>	107
2.2.3.2.12.	<i>Efecto 12 – Deletreando</i>	108
2.2.3.2.13.	<i>Efecto 13 - Abriendo Letra</i>	108
2.2.3.3.	Interacciones Entre Efectos	109
2.2.3.3.1.	<i>Interacción Entre Efecto 1 y Efecto1</i>	109
2.2.3.3.2.	<i>Interacción Entre Efecto 1, 2 y Efecto 2, 3</i>	109
2.2.3.3.3.	<i>Interacción Entre Efecto 2, 3 y Efecto 1</i>	110
2.2.3.3.4.	<i>Interacción Entre Efecto 1, 2 ,3 y Efecto 4 Al 13</i>	110
2.2.3.3.5.	<i>Interacción Entre Efecto 4 Al 13 y Efecto 4 Al 13</i>	110
2.2.3.3.6.	<i>Interacción Entre Efecto 4 Al 13 y Efecto 1 Al 3</i>	110
2.2.3.4.	Máximo Número de Caracteres por Tipo de Efecto	110
2.2.3.5.	Tarjeta de Led's	111
2.2.4.	FUENTE	116
2.3.	IMPLANTACION DEL SISTEMA	118
2.3.1.	TARJETA DE CEREBRO	118
2.3.2.	TARJETA DE LED'S	120

2.3.3.	ARMADO DEL ROTULO	122
2.4.	DESCRIPCION DEL SOFTWARE.....	124
2.4.1.	DESCRIPCION DE LA FIGURA 2.76	124
2.4.2.	DESCRIPCION DE LA FIGURA 2.77	126
2.4.3.	DESCRIPCION DE LA FIGURA 2.78	128
2.4.4.	DESCRIPCION DE LA FIGURA 2.79	130
2.4.5.	DESCRIPCION DE LA FIGURA 2.80	131
2.4.6.	DESCRIPCION DE LA FIGURA 2.81	132
2.4.7.	DESCRIPCION DE LA FIGURA 2.82	133
2.4.8.	DESCRIPCION DE LA FIGURA 2.83	133
2.4.9.	DESCRIPCION DE LA FIGURA 2.84	135
2.4.10.	SOFTWARE PARA EL SISTEMA.....	137
2.5.	PRUEBAS Y RESULTADOS.....	207
2.5.1.	CEREBRO DE CONTROL	207
2.5.2.	TARJETA DE LED'S.....	208
2.6.	ANALISIS TECNICO Y ECONOMICO	209
2.6.1.	ANALISIS TECNICO	209
2.6.2.	ANALISIS ECONOMICO	210
	CONCLUSIONES.....	214
	RECOMENDACIONES	215
	BIBLIOGRAFÍA	216
	ANEXOS	218
	ANEXO A.....	219
	ANEXO B.....	222
	ANEXO C.....	224
	ANEXO D.....	229

CAPITULO I

1. MARCO TEORICO

El presente capítulo indica los dispositivos electrónicos que son utilizados en la construcción del sistema electrónico publicitario, las características principales de cada uno, también se describe al software Bascom Avr, sus instrucciones de programación y al software Eagle utilizado para diseñar circuitos electrónicos.

1.1. MICROCONTROLADOR ATMEL

Un microcontrolador es un chip que posee en su interior a un microprocesador, memoria de programa, memoria de datos y puertos para comunicarse con el exterior, dispone normalmente de los siguientes componentes:

- Procesador o CPU (Unidad Central de Proceso).
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/EPROM/EEPROM/FLASH.
- Líneas de E/S para comunicarse con el exterior.



Figura 1.1: Microcontrolador Atmel

El microcontrolador Atmel Avr (figura 1.1) pertenece a una familia de 8 bits RISC¹ (Reduced Instruction Set Computer), el cual posee una bien definida estructura de entradas y salidas que limitan la necesidad de componentes externos, teniendo además un amplio rango de microcontroladores disponibles desde integrados de 8 pines/1k Flash hasta integrados de 100 pines/256k Flash y su alto grado de

¹ <http://es.wikipedia.org/wiki/AVR>

integración hacen de estos la mejor opción para elegir, un microcontrolador, en la figura 1.1 se observa al microcontrolador Atmel con distintas presentaciones.

Los microcontroladores AVR permiten la ejecución de instrucciones mediante la metodología 'pipeline' con dos etapas (cargar y ejecutar).

En estas etapas el repertorio de instrucciones de máquina es muy reducido y las instrucciones son simples y generalmente les permite ejecutar la mayoría de las instrucciones en un ciclo de reloj, lo que los hace relativamente rápidos entre los microcontroladores de 8 bits, la sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

Los microcontroladores AVR trabajan con sus direcciones y datos, utilizando la arquitectura Harvard², representada en la figura 1.2, la cual dispone de dos memorias independientes, una que contiene sólo instrucciones y otra sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

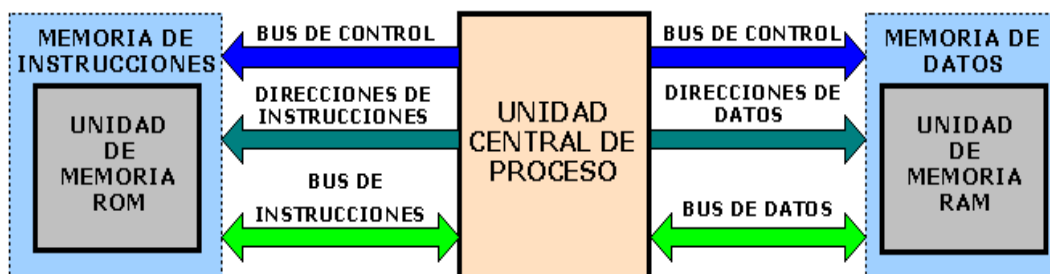


Figura 1.2: Arquitectura Harvard

Los microcontroladores de AVR, es la familia más grande de los procesadores que comparten una sola arquitectura base.

Además se puede utilizar las mismas herramientas de desarrollo para todos los microcontroladores AVR.

² <http://perso.wanadoo.es/pictob/micropic.htm>

1.1.1. ARQUITECTURA DEL MICROPROCESADOR

La función principal del núcleo del microcontrolador (CPU), es asegurar una correcta ejecución del programa. La CPU, por lo tanto, debe acceder a memorias, realizar cálculos, controlar periféricos, y manejar interrupciones.

1.1.1.1. Descripción de los Componentes del Microprocesador

En la figura 1.3³ se puede observar la arquitectura interna del microprocesador. El AVR usa una arquitectura Harvard, con memorias y buses separados para el programa y los datos, a fin de maximizar el desempeño y el paralelismo

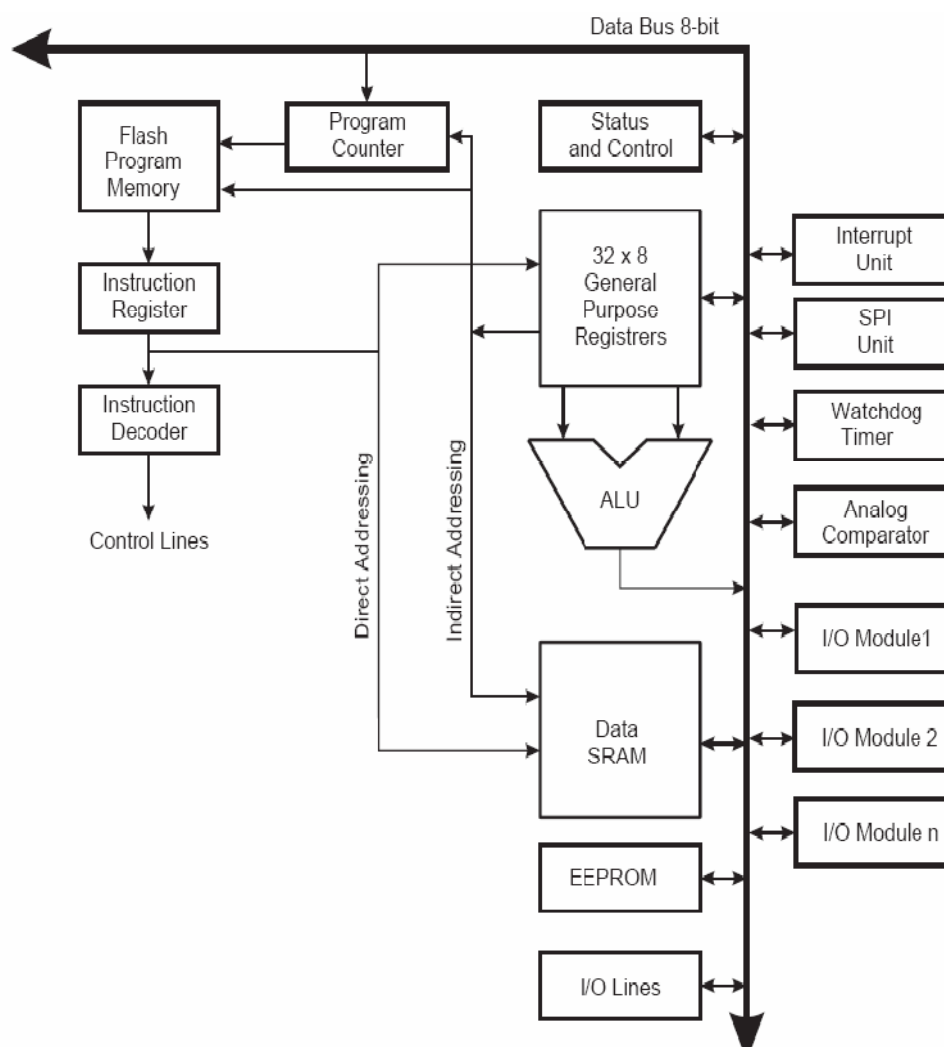


Figura 1.3: Arquitectura de la CPU

³ <http://www.webelectronica.com.ar/>

1.1.1.2. Memoria Flash

La memoria de programa es la memoria flash reprogramable en el sistema, aquí es donde se almacenan todas las instrucciones del programa.

1.1.1.3. Registros de Propósito General

El microprocesador contiene 32 registros de trabajo de propósito general de 8 bits con un tiempo de acceso de un solo ciclo de reloj. Esto permite la operación de la Unidad Aritmético Lógica (ALU) en un sólo ciclo.

1.1.1.4. Unidad Aritmético Lógica (ALU)

Realiza las operaciones aritméticas. En una típica operación de la ALU, se toman 2 operandos del Archivo de Registros, se ejecuta la operación, y el resultado se almacena nuevamente en el Archivo de Registros en un ciclo de reloj. Luego de una operación aritmética, el Registro de Estado se actualiza para reflejar la información sobre el resultado de la operación.

1.1.1.5. El Registro de Estado

El Registro de Estado contiene información sobre el resultado de la instrucción más recientemente ejecutada. Esta información se puede usar para alterar el flujo del programa a fin de ejecutar operaciones condicionales.

1.1.1.6. Sram

Es donde residen los Registros Específicos (SFR) del Avr, tienen direcciones de 0 a 31. Casi todos estos registros son utilizados por el compilador o pueden ser utilizados en el futuro. Cada registro es utilizado dependiendo de los enunciados programados a usar.

1.1.1.7. Eeprom

Son memorias de sólo lectura, programables y borrables eléctricamente, EEPROM (Electrically Erasable Programmable Read Only Memory), este tipo de memoria guarda un dato de manera permanente, pudiendo conservar su contenido cuando no hay energía.

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

1.1.2. INTERRUPCIONES Y RECURSOS ESPECIALES

Las interrupciones son utilizadas para realizar el control de eventos y pueden presentarse en cualquier momento sin importar la línea que se encuentre ejecutando el programa, realizando tareas específicas⁴.

Cuando una interrupción es detectada el programa se detiene para ingresar a una subrutina y atender la causa de la interrupción, al finalizar esta, el programa regresa al lugar donde estaba, antes de ingresar a la interrupción, para continuar con el funcionamiento normal del programa. El microcontrolador posee interrupciones internas, externas y un nivel configurable de estas, que son descritas a continuación.

1.1.2.1. I2C

Es una interfaz de comunicación serie de dos hilos, utilizada para comunicarse con otros dispositivos. Cada dispositivo conectado al bus tiene un código de dirección seleccionable mediante software. Habiendo permanentemente una relación Master/Slave (Maestro/Esclavo) entre el microcontrolador y los dispositivos conectados

⁴ <http://perso.wanadoo.es/pictob/microcr.htm>

1.1.2.2. SPI

(Serial Peripheral Interface Bus) Es un bus de datos serial síncrono que opera en modo full dúplex. Permite la comunicación entre master/slave utilizando un bus de cuatro líneas.

1.1.2.3. Temporizadores o "Timers"

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

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso. Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patillas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

1.1.2.4. Perro Guardián o "Watchdog"

El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reseteo automáticamente en el sistema, esto sucede cuando el microcontrolador se bloquea por un fallo del software u otra causa, para luego reiniciarse el sistema.

1.1.3. TIPOS DE ENCAPSULADOS

El encapsulado es la protección que tienen los chips para evitar su daño y facilitar su conexión con el exterior, su presentación es muy variada, en la tabla 1.1 es posible observar cuatro tipos de encapsulados. El encapsulado para proteger al microprocesador que posee debe cumplir con las siguientes funciones⁵:

Excluir las influencias ambientales: Evita que la humedad, el polvo en el aire, las vibraciones, los golpes e iluminación causen un mal funcionamiento.

Conectividad eléctrica: Los encapsulados permiten la fijación de conductores metálicos denominados pines, permitiendo que las señales sean enviadas a y desde el dispositivo semiconductor.

Disipar el calor: Los encapsulados pueden liberar el calor generado.

Mejorar el manejo y montaje: Colocar el chip en una cápsula hace que sea más fácil manejar y de montar en placas de circuitos impresos.





BGA (Ball Grid Array)	
DIP (Dual In-Line Package)	
MLF (Micro Lead Frame)	
TSOP (Thin Small Out-Line Package)	

Tabla 1.1: Encapsulados

⁵ <http://ayudaelectronica.com/tipos-de-encapsulados/>

1.1.4. MICROCONTROLADOR ATMEGA 644

El ATmega 644 es el microcontrolador seleccionado para realizar las operaciones de control y envío de datos requeridos por el proyecto. Es un microcontrolador Cmos de 8 bits de baja potencia.

En la figura 1.4⁶ se indica la distribución de pines que tiene el ATmega 644 en un encapsulado DIP.

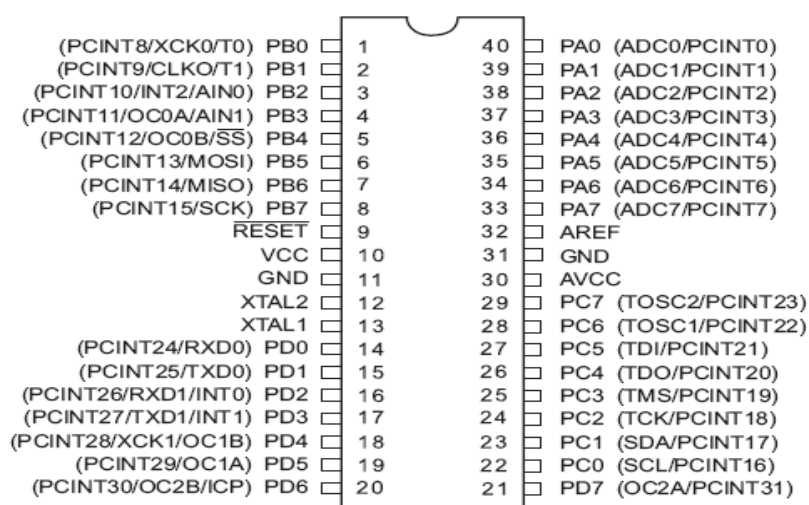


Figura 1.4: ATmega 644

Características:

La principal característica de este integrado es la capacidad de almacenamiento de sus memorias cuyos valores se presentan en la tabla 1.2.

MEMORIA	CAPACIDAD
Flash	64 kB
EEPROM	2048 B
SRAM	4096 B

Tabla 1.2: Capacidad de Almacenamiento

El ATmega 644 puede trabajar a una frecuencia máxima (20 Mhz), mediante el uso de un oscilador externo.

⁶<http://www.avrfreaks.net/index.php?module=photoshare&func=viewimage&thumbnail=&iid=414>

1.1.4.1. Puertos I/O (Entrada /Salida)

El ATmega 644 tiene 4 puertos paralelos (A, B, C, D) de 8 bits cada uno configurables como entradas o salidas.

Además los pines pueden ser configurados para realizar otras funciones especiales, que realizan la detección de interrupciones o la transmisión y recepción de datos, la tabla 1.3 indica algunas funciones especiales del microcontrolador con sus respectivos pines.

# Pin	Pin	Funciones Especiales
6	PB5	MOSI: Utilizado en la comunicación SPI
7	PB6	MISO: Utilizado en la comunicación SPI
8	PB7	SCK: Utilizado en la comunicación SPI
14	PD0	RX: Comunicación Serial (Pin de Recepción)
15	PD1	TX: Comunicación Serial (Pin de Transmisión)
16	PD2	INT0: Pin de interrupción Externa
17	PD3	INT1: Pin de interrupción Externa
19	PD5	OC1A: Pin generador de PWM
22	PC0	SCL: Utilizado en la comunicación I2C
23	PC1	SDA: Utilizado en la comunicación I2C

Tabla 1.3: Funciones especiales

1.2.MEMORIA DE ALMACENAMIENTO DE DATOS Y RELOJ DE TIEMPO REAL

La memoria de almacenamiento de datos y el integrado de reloj, son dispositivos que permiten la lectura y escritura de sus datos mediante la comunicación I2C por lo tanto es indispensable conocer como es utilizado este protocolo, para poder realizar una comunicación exitosa.

1.2.1. CONTROL DE INTERFACE INTEGRADA I2C

El protocolo I2C, es un estándar que fue diseñado por Philips en 1980, usa solo dos líneas llamadas SCL (serial clock) y SDA (serial data), facilita la comunicación entre microcontroladores, memorias, reloj de tiempo real, sensores de temperatura y otros dispositivos.

La comunicación I2C se establece entre el maestro (master) y el esclavo (slave), representado en la figura 1.5⁷, el maestro usualmente es un microcontrolador mientras que el esclavo es el dispositivo conectado al bus, El protocolo I2C establece la comunicación hasta un máximo de 128 esclavos conectados al mismo bus.

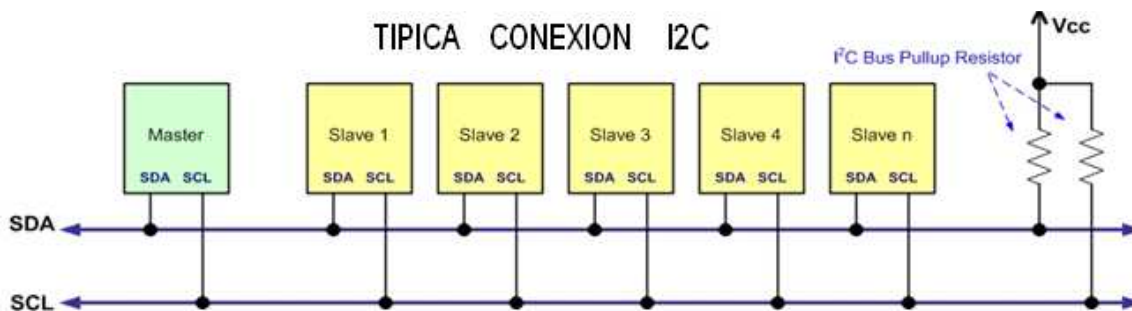


Figura 1.5: Comunicación I2C

La comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos

1.2.1.1. Descripción de las Señales

Todos los dispositivos conectados al bus I2C⁸ son controlados mediante las líneas:

- **SCL** (System Clock): Es la línea de los pulsos de reloj que sincronizan el sistema.

⁷ <http://www.ermicro.com/blog/?p=744>

⁸ http://axxon.com.ar/rob/Comunicacion_busI2C.htm

- **SDA** (System Data): Es la línea por la que se mueven los datos entre los dispositivos.
- **GND** (Ground): Masa común de la interconexión entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL se deben polarizar en un nivel lógico alto (conectando a la alimentación por medio de resistores "pull-up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

El valor de los resistores de polarización (pull up), puede ir desde $1.8\text{ K } \Omega$ hasta $47\text{ K } \Omega$. Un valor menor de resistencia incrementa el consumo de los integrados pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales.

1.2.1.2. Protocolo de Comunicación del Bus I2C

Para que un dispositivo Maestro pueda establecer la comunicación, el bus I2C debe estar libre, este estado se cumple cuando las señales SDA y SCL se encuentran en estado lógico alto.

En este estado cualquier dispositivo maestro puede ocupar el bus, primero establece una la condición de **inicio** (start), poniendo en estado bajo la línea de datos (SDA), pero dejando en alto la línea de reloj (SCL), indicado en la figura 1.6⁹

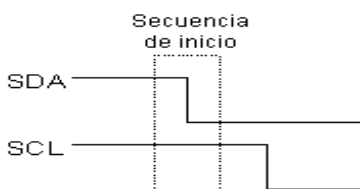


Figura 1.6: Condición de Inicio I2C

⁹ http://axxon.com.ar/rob/Comunicacion_busI2C.htm

Luego de la condición de inicio, el primer byte que se transmite (figura 1.7) contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura).

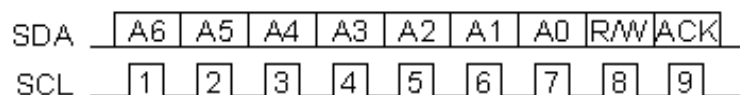


Figura 1.7: Dirección del Esclavo

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits está presente en el bus, éste contesta con un bit en bajo, ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro. Este bit de reconocimiento (ACK) en bajo le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse. Aquí la comunicación se establece en firme y comienza el intercambio de información entre los dispositivos. Durante el intercambio de información los datos se transfieren en secuencias de 8 bits. Cada bit transmitido se encuentra acompañado por un pulso de reloj pues la comunicación es sincrónica.

En la figura 1.8 se puede observar el diagrama de tiempo entre las señales SDA y SCL representando los sincronismos existentes entre estas señales.

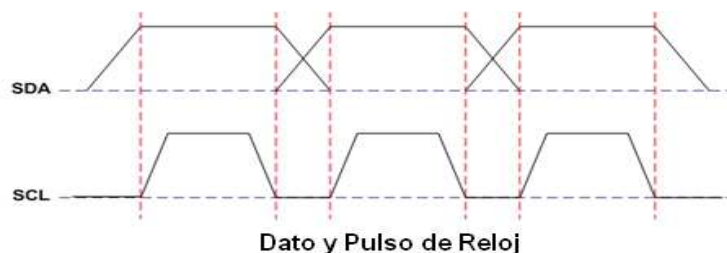


Figura 1.8: Diagrama de Tiempo

Terminado el proceso de lectura o escritura el dispositivo maestro puede dejar libre el bus generando una condición de **stop** (parada o detención).

1.2.1.3. Proceso de Escritura en un Dispositivo Esclavo

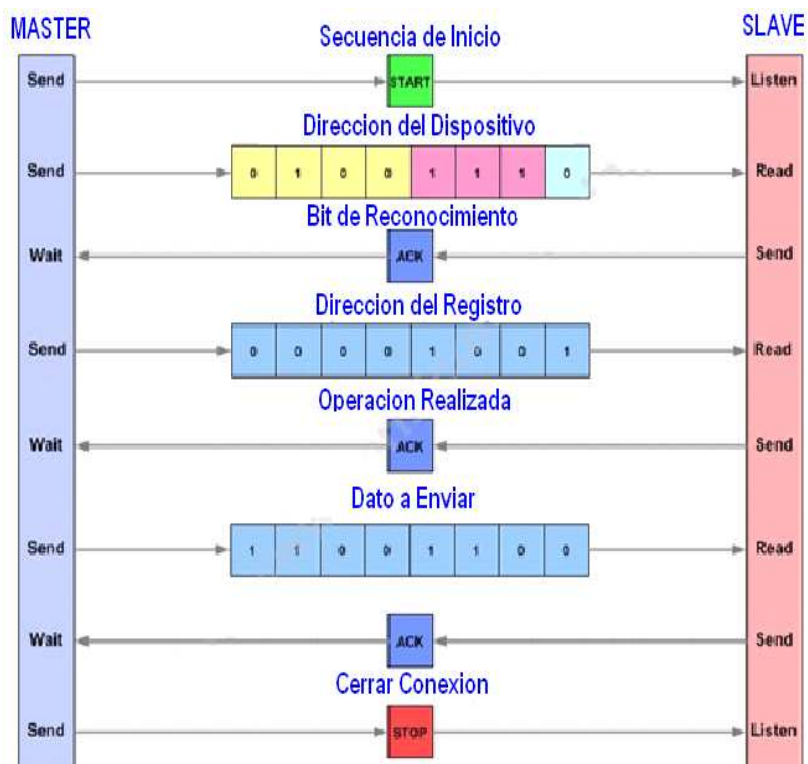


Figura 1.9: Proceso de Escritura

Cuando el maestro desea escribir en un dispositivo conectado al bus I2C debe realizar el proceso representado en la figura 1.9¹⁰ que es descrito a continuación:

1. Enviar una secuencia de inicio
2. Enviar la dirección de dispositivo (7 bits) con el bit de lectura/escritura (R/W) en bajo.
3. Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits está presente en el bus, éste contesta con un bit de reconocimiento en bajo (ACK), ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro
4. Enviar el número de registro interno en el que se desea escribir
5. Enviar el byte de dato
6. [Opcionalmente, enviar más bytes de dato]
7. Enviar la secuencia de parada

¹⁰ <http://www.ermicro.com/blog/?p=1239>

1.2.1.4. Proceso de Lectura desde un Dispositivo Esclavo

La secuencia de lectura de bytes de un dispositivo esclavo representada en la figura 1.10¹¹, es un poco complicada pues inicia como una secuencia de escritura pero no dificulta el entender cómo funciona.

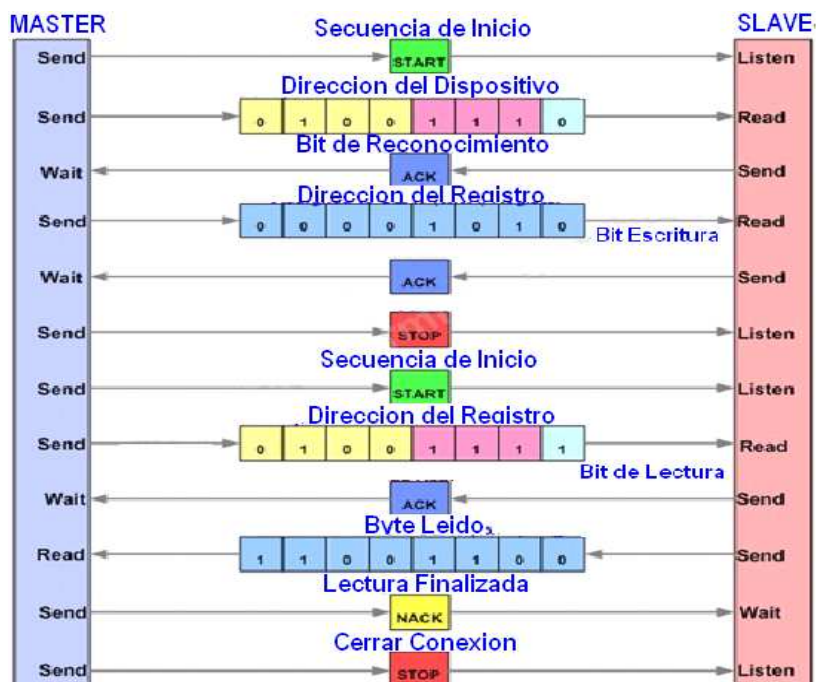


Figura 1.10: Proceso de Lectura

Los pasos a seguir para realizar la lectura son los siguientes:

1. Enviar una secuencia de inicio
2. Enviar la dirección de dispositivo con el bit de lectura/escritura (R/W) en bajo.
3. Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits está presente en el bus, éste contesta con un bit de reconocimiento en bajo (ACK), ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro
4. Enviar el número de registro interno en el que se desea leer

¹¹ <http://www.ermicro.com/blog/?p=1239>

El pin3 opcionalmente tiene la conexión para una batería de respaldo de 3 volt, asegurando que se mantendrá el tiempo a la fecha aunque esté desconectada la fuente de tensión del circuito principal. El circuito integrado detecta automáticamente que se ha removido la energía en el circuito principal y se conecta a la batería de respaldo cuando es requerido

El pin 4 estará conectado a tierra y el pin 5 a positivo a una fuente de 5 voltios. El pin 7 es una salida de colector abierto, que puede ser programado para hacer "flash" cada 1Hz. El circuito integrado también tiene 56 bytes de memoria RAM para propósito general. La dirección y los datos son transferidos serialmente por I2C utilizando el pin 6 (SCL) y el pin 5 (SDA).

La dirección de escritura del integrado es 1101000x, donde x es el bit que especifica la acción a realizar: 0 Escritura, 1 Lectura

Los datos de tiempo/fecha están en formato BCD lo cual se debe tener muy en cuenta para guardar los datos, leerlos o realizar su conversión de ser necesario. Las direcciones para guardar los datos de hora y fecha se observan en la siguiente figura¹³ con sus respectivos registros.

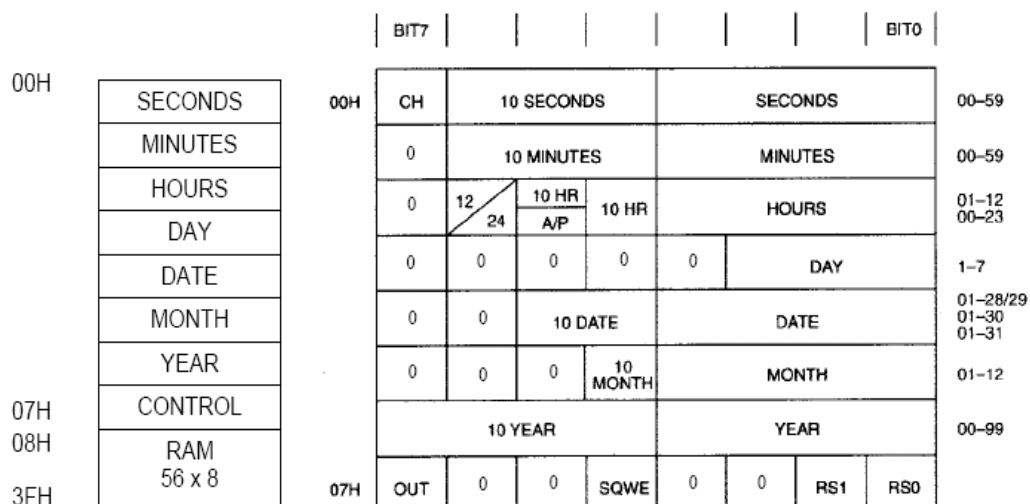


Figura 1.13: Direcciones de Horario

¹³ Ver Anexo 4

El registro de control presentado en la figura 1.14 es usado para el control de operación del pin SQW/OUT (pin 7).

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Figura 1.14: Registro de Control

OUT (Output Control): Este bit controla el nivel de salida del pin 7, cuando SQWE esta deshabilitado. Si SQWE es cero el nivel lógico de salida es 1.

SQWE (Square Wave Enable): Cuando este bit es 1 lógico habilita la oscilación de salida. La frecuencia de oscilación depende del valor de los registros RS1 y RS0.

RS (Rate Select): Estos bit controlan la frecuencia de SQWE, la siguiente tabla indica las frecuencias a ser seleccionadas.

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

Tabla 1.4: Rate Select

1.2.3.MEMORIA 24LS256

La memoria 24LS256 mostrado en la figura 1.15, es un dispositivo que permite almacenar 256Kbits de información, ha sido desarrollada por el avance de las aplicaciones, tales como las comunicaciones personales o la adquisición de datos, que requieren trabajar a baja potencia.

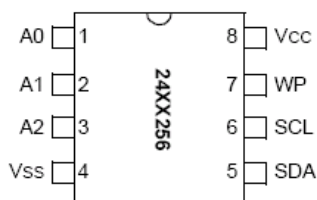


Figura 1.15: Memoria 24LS256

Esta memoria permite leer la información almacenada de forma secuencial o de forma aleatoria.

Las especificaciones de la memoria son las siguientes:

- Interface serial de comunicación de dos hilos (I2C)
- 1000000 de ciclos de grabación y borrado
- Permite la conexión de dispositivos similares en cascada
- Tiempo de retención de la información, 200 años
- Opera en un amplio rango de voltaje (1.8 V a 5.5 V)

La figura 1.16 indica la forma de conexión entre la memoria y es microcontrolador y la respectiva descripción de sus pines son presentados en la tabla 1. 5.

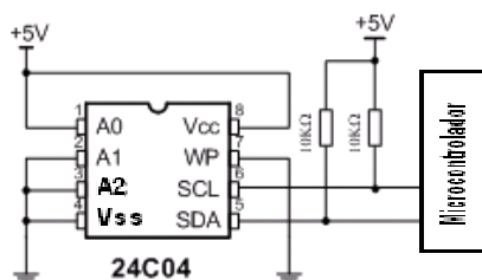


Figura 1.16: Diagrama de Conexión

A0,A1,A2	Estas entradas determinan la dirección que tendrá la memoria, son utilizadas para trabajar con varias memorias en cascada, pudiendose conectar como máximo 8 memorias en cascada.
Vss	Gnd
SDA	Pin bi-direccional usado para transferir direcciones y datos.
SCL	Esta entrada es usada para sincronizar el dato transferido desde y hasta la memoria.
WP	(Write Project) Habilita la operación normal de la memoria, permitiendo leer y escribir en todo su rango (0000-7FFF).
Vcc	Vcc

Tabla 1.5: Descripción de Pines

La selección de la dirección de inicio en la memoria a trabajar se indica en la figura 1.17¹⁴, la cual es controlada mediante el siguiente formato de byte.

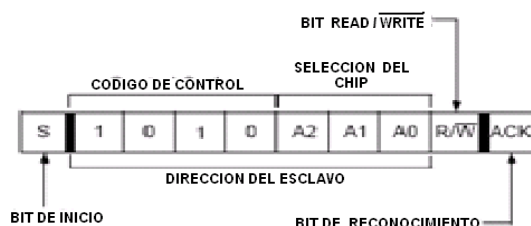


Figura 1.17: Dirección de inicio

El formato del byte de control consiste en iniciar enviando una condición de inicio, para luego enviar el código de control 1010 para el 24LC256.

Los siguientes 3 bit seleccionan la memoria en la cual se trabajará en caso de haber mas en cascada en el mismo bus.

El bit R/W (Read/Write) es seleccionado dependiendo de la aplicación a realizar, seguido de su bit de reconocimiento ACK.

Una vez seleccionada la memoria a trabajar, se elige la dirección en donde se debe leer o escribir un dato. (figura 1.18) La dirección es seleccionada utilizando dos bytes

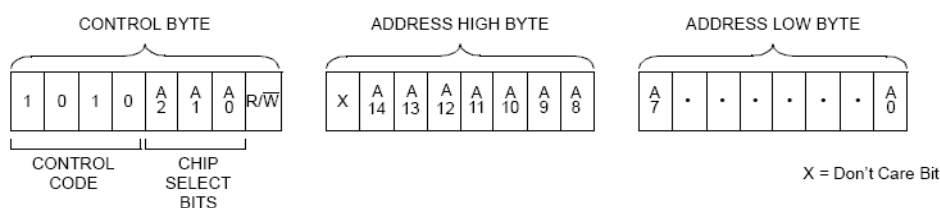


Figura 1.18: Selección de localidad en la memoria

La memoria tiene una capacidad de 256Kbits y para escribir en una localidad de esta su dirección puede variar entre el rango de (0000-7FFF), por lo que se necesita 2 bytes de direccionamiento, la manera de seleccionar una localidad es enviar primero el byte mas significativo de la dirección seguido del byte menos significativo, para luego leer o escribir el dato en la localidad seleccionada.

¹⁴ Ver Anexo 3

1.3.CREACION DEL CODIGO BINARIO

En un arreglo de led's es posible observar gráficos y letras de todo tipo o forma pero para que cada tipo de letra o gráfico sea visualizado es necesario crear su respectivo código binario, el código puede ser pequeño o muy extenso dependiendo del número de pixeles de cada uno.

Para crear el código binario es necesario determinar:

- El tamaño de la matriz en la cual será presentado un carácter.
- La forma que tendrá el carácter.
- El orden en la cual se organizara la información a ser creada.

Además la información creada debe ser organizada para ser guardada en el dispositivo de almacenamiento.

1.3.1. CREACION DE UN CARACTER UTILIZANDO EXCEL

Se ha elegido el Software Excel para realizar el código binario, porque facilita el desarrollo del código, debido a que presenta las siguientes características.

- Permite realizar tablas con gran facilidad
- Permite crear celdas con formato condicional.
- Permite desarrollar formulas para organizar la información.

La figura 1.19 indica, el tamaño de la matriz determinada según el número de filas a controlar y el tamaño del carácter a presentar, la cual es de 6 columnas por 7 filas, donde se realizará el carácter "G", y se ha determinado el orden de organización de la información del carácter.

		COLUMNAS					
		1	2	3	4	5	
FILAS	1	x	x	x	x	x	MSB
	2	x	x	x	x	x	
	3	x	x	x	x	x	
	4	x	x	x	x	x	
	5	x	x	x	x	x	
	6	x	x	x	x	x	
	7	x	x	x	x	x	LSB
		BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	

Figura 1.19: Condiciones iniciales de la Matriz

En la tabla se han determinado el orden de filas, columnas, la posición en la cual se almacenarán los bytes que formarán el código y el respectivo orden que tendrá cada byte, con una posición definida de los bits LSB y MSB.

Para visualizar la forma que tendrá un carácter, cada celda es declarada con un formato condicional asignando un color determinado a cada celda en caso de que contenga el valor de cero o el valor de uno, ya que estos son los únicos valores que puede poseer cada bit que forma parte de la matriz, estableciendo que “X” será igual a uno para encender un determinado led de la matriz.

La creación de una celda condicional en Excel se realiza de la siguiente manera:
1.- En Excel se selecciona una celda para luego ir al menú inicio y dar un click sobre la viñeta, formato condicional tal como lo indica la figura 1.20.

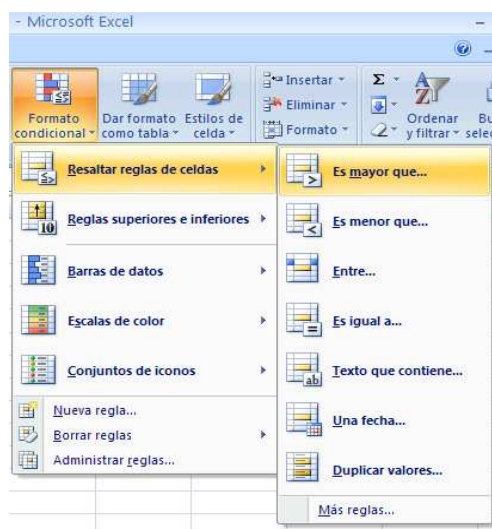


Figura 1.20: Formato Condicional

2.- Se elige la opción “es mayor que” que despliega la ventana indicada en la figura 1.21 en donde determino el valor cero, indicando que para números mayores que este la celda tendrá un color definido, seleccionando esto en formato personalizado el cual despliega la ventana indicada en la figura 1.22, en la cual se selecciona el color de la celda.

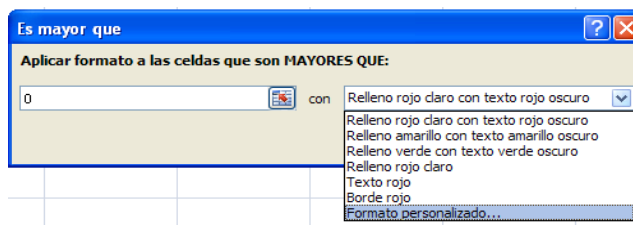


Figura 1.21: Condición “es mayor que”

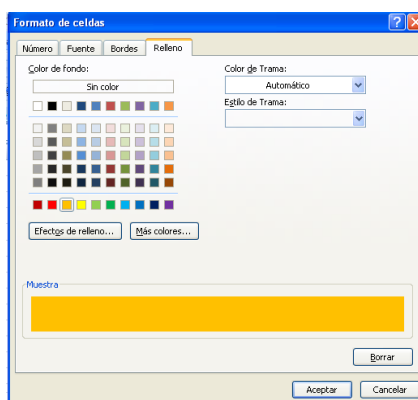


Figura 1.22: Personalización del Celda

3.- Para seleccionar el color que tendrá una celda cuando tenga el valor de cero se realizan los mismos pasos anteriores pero eligiendo esta vez la opción “es menor que” para asignarle el valor de 1 y determinar el color que tendrá la celda. La figura 1.23 indica el aspecto que tiene la celda en la cual se ha establecido las condiciones anteriores.

		COLUMNAS					
		1	2	3	4	5	
FILAS	1	0					MSB
	2						
	3						
	4						
	5						
	6						
	7						LSB
		BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	

Figura 1.23: Celda Condicionada

4.-Luego se arrastra con el cursor la celda en la cual se trabajó, para que las características de esta celda se copien a las demás de la matriz como lo indica la figura 1.24.

		COLUMNAS					
		1	2	3	4	5	
FILAS	1	0	0	0	0	0	MSB
	2	0	0	0	0	0	
	3	0	0	0	0	0	
	4	0	0	0	0	0	
	5	0	0	0	0	0	
	6	0	0	0	0	0	
	7	0	0	0	0	0	LSB
	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5		

Figura 1.24: Matriz Condicionada

5.-A continuación se forma el carácter “G”, asignando los valores de uno o cero sobre la matriz tal como lo indica la figura 1.25.

		COLUMNAS					
		1	2	3	4	5	
FILAS	1	0	1	1	1	0	MSB
	2	1	1	0	0	1	
	3	1	1	0	0	0	
	4	1	1	0	1	0	
	5	1	1	0	0	1	
	6	1	1	0	0	1	
	7	0	1	1	1	0	LSB
	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5		

Figura 1.25: Matriz con valores lógicos

1.3.2. ORGANIZACION DEL CODIGO CREADO

El carácter que se visualiza en la matriz de la figura 1.25, solo posee las combinaciones de ceros y unos que la forman.

Para que esta información pueda ser útil, debe ser organizada en una celda poniendo en orden todos los bits de cada byte.

Para lograr esto se organiza junto a la matriz creada los datos necesarios que permitan utilizar el código, para grabar toda esta información en el microcontrolador utilizando otro software como Micro C, Bascom Avr, etc.

En la figura 1.26 se han creado dos datos adicionales correspondientes al orden del número de cada byte y el dato cero, que representa el octavo bit de todos los bytes definidos.

Realizado esto se selecciona la celda donde se organizará el valor de cada byte.

		COLUMNAS					
		1	2	3	4	5	
FILAS	1	0	1	1	1	0	MSB
	2	1	1	0	0	1	
	3	1	1	0	0	0	
	4	1	1	0	1	0	
	5	1	1	0	0	1	
	6	1	1	0	0	1	
	7	0	1	1	1	0	LSB
		BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	

0	Byte1 =	
	Byte2 =	
	Byte3 =	
	Byte4 =	
	Byte5 =	

Figura 1.26: Creación Datos Adicionales

Para organizar la información se utiliza la función de Excel “concatenar” que permite poner el texto de una celda a continuación de otro, esta función se encuentra en el menú de formulas en opciones de texto, la cual es mostrada en la figura 1.27.

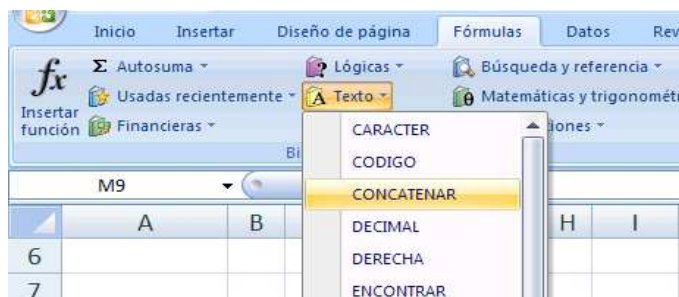


Figura 1.27: Selección Función Concatenar

Dando click sobre esta función se despliega la ventana indicada en la figura 1.28 en la cual se elige primero la celda que contiene escrito “Byte1=” luego las celdas correspondientes a cada bit, desde MSB hasta el LSB de un mismo byte.

También es posible visualizar el resultado de la función en la parte inferior de la ventana para verificar valores, para luego aceptar la concatenación creada.

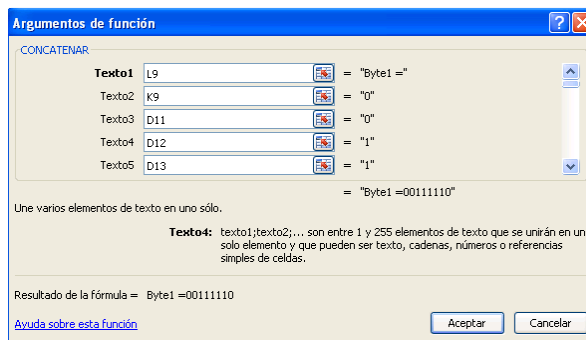


Figura 1.28: Selección de Celdas

La figura 1.29 permite observar la organización de cada byte después de elegir los datos de cada celda.

		COLUMNAS								
		1	2	3	4	5				
FILAS	1	0	1	1	1	0	MSB	0	Byte1=	Byte1=00111110
	2	1	1	0	0	1			Byte2=	Byte2=01111111
	3	1	1	0	0	0			Byte3=	Byte3=01000001
	4	1	1	0	1	0			Byte4=	Byte4=01001001
	5	1	1	0	0	1			Byte5=	Byte5=00100110
	6	1	1	0	0	1				
	7	0	1	1	1	0	LSB			
		BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5				

Figura 1.29: Organización de bytes

Toda la información organizada puede ser copiada e utilizada para ser grabada. Creando el código de esta manera permite variar el contenido de una o varias celdas obteniendo una actualización al momento de sus datos, por ejemplo la figura 1.30, permite observar la creación del carácter "S", en donde el código entregado puede ser utilizado por el software Bascom Avr.

		COLUMNAS								
		1	2	3	4	5				
FILAS	1	0	1	1	1	0	MSB	1	Data=&B	Data=&B10110010
	2	1	1	0	0	1			Data=&B	Data=&B11111001
	3	1	1	0	0	0			Data=&B	Data=&B11001001
	4	0	1	1	1	0			Data=&B	Data=&B11001111
	5	0	0	0	1	1			Data=&B	Data=&B10100110
	6	1	0	0	1	1				
	7	0	1	1	1	0	LSB			
		BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5				

Figura 1.30: Código utilizado en Bascom Avr

1.3.3.DIRECCIONAMIENTO INTERNO DE LA MEMORIA

El direccionamiento interno de una memoria consiste en asignar secciones específicas de la memoria para guardar un dato o un grupo de datos que tienen una relación en común. Por ejemplo en la figura 1.31 se indica la creación de un tipo de letra, en la cual cada carácter creado está conformado por 5 bytes, en donde se indica cual es la posición inicial de cada carácter especificando que el carácter "A" inicia en la dirección cero de la memoria.

					DIRECCIONAMIENTO			
0	1	1	1	0	0	Byte1=	Byte1=00111111	0
1	1	0	0	1		Byte2=	Byte2=01111111	
1	1	0	0	1		Byte3=	Byte3=01001100	
1	1	1	1	1		Byte4=	Byte4=01001100	
1	1	1	1	1		Byte5=	Byte5=00111111	
1	1	0	0	1				
1	1	0	0	1				
1	1	1	1	0	0	Byte1=	Byte1=01111111	5
1	1	0	0	1		Byte2=	Byte2=01111111	
1	1	0	0	1		Byte3=	Byte3=01001001	
1	1	1	1	0		Byte4=	Byte4=01001001	
1	1	0	0	1		Byte5=	Byte5=00110110	

Figura 1.31: Código Tipo Letra

Los datos son organizados internamente en la memoria 24LC256, pudiendo visualizar su orden en el mapa de memoria de la figura 1.32, en donde los bytes de cada letra son guardados en una dirección específica y ordenados siguiendo una lógica determinada, para poder modificarlos o facilitar la lectura de los mismos.

MEMORIA 24LC256		
	DIRECCION	
Byte1	0	00111111
Byte2	1	01111111
Byte3	2	01001100
Byte4	3	01001100
Byte5	4	00111111
	5	01111111
	6	01111111
	7	01001001
	8	01001001
	9	00110110
	10-14	
	:	
	125-129	Z
	130...	
	...	
	...400	
	401...	
	...32768	

TIPO DE LETRA 1 (A-Z)

TIPO DE LETRA 2

ESPACIO LIBRE

Figura 1.32: Organización de datos en memoria

1.4.TECLADO

El teclado es un elemento muy utilizado que permite ingresar datos hacia el microcontrolador de una manera fácil y a su vez económica.

El teclado tiene que ser creado con el número de teclas necesarias que permitan ingresar toda la información requerida para realizar una función específica, o de ser el caso una tecla debe cumplir con varias funciones, dependiendo del menú que se encuentre ejecutando el usuario. Por ejemplo el teclado de un celular permite escribir gran cantidad de información y realizar otras funciones con un número muy reducido de teclas.

1.4.1.TECLADO MATRICIAL 4 x 4

El teclado matricial 4 x 4 es un dispositivo de entrada de datos que consta de una matriz de pulsadores o teclas dispuestos e interconectados en filas y columnas (figura 1.33)¹⁵, realiza la función de calculadora, marcador telefónico, cerradura electrónica y una infinidad de aplicaciones.

Para detectar si hay una tecla pulsada se aplica el siguiente procedimiento:

Se hace un barrido por cada una de las filas, inicialmente se aplica a la primera fila un nivel alto y a las otras tres un nivel bajo si en este instante se pulsa alguna tecla de esta fila entonces el nivel alto aparecerá en la columna en la que se encontraba la tecla, seguidamente se pasa a verificar la segunda fila enviando un nivel alto a esta y a las otras un nivel bajo y así sucesivamente.

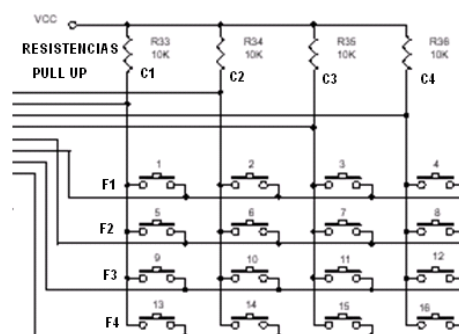


Figura 1.33: Teclado Matricial 4 x 4

¹⁵ http://www.disca.upv.es/aperles/web51/pdf/placas/sch/modulo_07.pdf

Cuando se detecta que una tecla ha sido presionada, se debe verificar si verdaderamente se ha realizado esta acción, para evitar que el microcontrolador realice una tarea que no fue seleccionada.

1.5.MATRIZ DE LED'S¹⁶

Una matriz de led's (figura 1.34) consiste en un arreglo de led's que pueden ser encendidos y apagados individualmente desde un microcontrolador.

La matriz es utilizada como una pantalla en la cual es posible presentar gráficos y textos, tanto estáticos como en movimiento.

La matriz está formada por una serie de filas y columnas en donde la intersección entre ambas contiene un led.

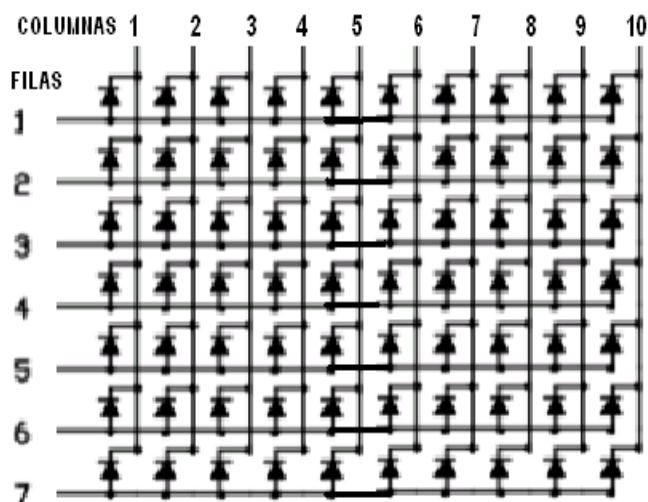


Figura 1.34: Matriz de Led's

El envío de datos en una matriz de led's se realiza mediante el uso de buffers, cuando esta posee varias columnas, pues permite utilizar menor número de pines del microcontrolador.

¹⁶ http://usuarios.multimania.es/carlosyaco/microcontroaldor/matriz_led's.htm

1.6. OTROS DISPOSITIVOS ELECTRÓNICOS

1.6.1. DISPLAY LCD 16 x 2

Las pantallas de cristal líquido LCD o display LCD para mensajes (*Liquid Cristal Display*), es una pantalla delgada y plana colocada delante de una fuente de luz. Las pantallas LCD tiene la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil.

La pantalla consta de una matriz de caracteres (normalmente de 5x7 o 5x8 puntos) distribuidos en una, dos, tres o cuatro líneas de 16 hasta 40 caracteres por línea, en la figura 1.35 se muestra una pantalla compuesta por 2 líneas de 16 caracteres.



Figura 1.35: LCD 16 x 2

El funcionamiento del LCD, se basa en el envío de la información desde un microcontrolador, durante un tiempo determinado por el fabricante. Pero cuando se trata de programación en alto nivel, este proceso es realizado internamente por librerías que son previamente definidas en el software.

Las características generales de un módulo **LCD 16x2** son las siguientes:¹⁷

- Consumo muy reducido, del orden de 7.5mW
- Pantalla de caracteres ASCII, además de los caracteres japoneses, caracteres griegos y símbolos matemáticos.
- Desplazamiento de los caracteres hacia la izquierda o a la derecha

¹⁷ http://www.caveo.com.ar/lcd_16x2_sinc_big.htm

- Memoria de 40 caracteres por línea de pantalla, visualizándose 16 caracteres por línea
- Movimiento del cursor y cambio de su aspecto.
- Permite que el usuario pueda programar sus caracteres creados.
- Pueden ser gobernados de 2 formas principales:
 - Conexión con bus de 4 bits
 - Conexión con bus de 8 bits

En la figura 1.36 se puede ver la conexión en un LCD 16x2 con un bus de 4 hilos seguida de la descripción de sus pines¹⁸.

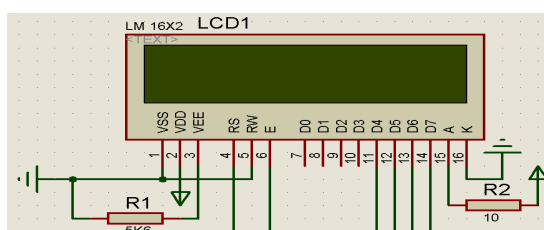


Figura 1.36: Conexión de un LCD

PIN	SIMBOLO	DESCRIPCION
1	VSS	Patilla de tierra de alimentación
2	VDD	Patilla de alimentación de 5 V
3	VEE	Patilla de contraste del cristal líquido. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y +5V que permite regular el contraste del cristal líquido.
4	RS	Selección del registro de control/registro de datos:
5	R/W	Señal de lectura/escritura con R/W=0, el módulo LCD es escrito
6	E	Señal de activación del módulo LCD: E=1 Módulo conectado
7-14	D0-D7	Bus de datos bi-direccional.
15	A	Patilla de alimentación del backlight (5 V)
16	K	Patilla de tierra del backlight

¹⁸ Carlos Reyes, Microcontroladores Pic, 3^{ra} Edición

1.6.2. INTEGRADO ULN 2803

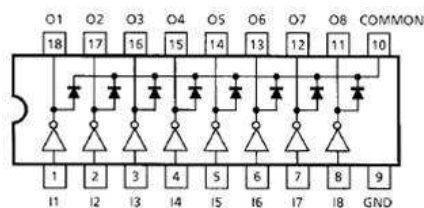


Figura 1.37: ULN 2803

El ULN2803 es un integrado driver que empaqueta 8 transistores de arreglo Darlington y sus respectivos diodos, presentado en la figura 1.37¹⁹, se utiliza principalmente como interface, para acondicionar pulsos o señales digitales de baja intensidad (como las obtenidas de las puertas lógicas CMOS, TTL, etc.) de tal manera que puedan mover componentes que requieren altas corrientes o voltajes. Cada transistor Darlington tiene una entrada IN y una salida OUT, que se encuentran enfrentadas en filas de pines opuestas en el circuito integrado, la tierra (GND) es común a la alimentación utilizada para la carga.

Lo que en realidad hace es tomar la señal eléctrica generada por los elementos digitales y aumentar su tensión y corriente por medio de transistores de potencia. Este circuito integrado, que permite extraer 500 mA por pin de salida, aplicando tensiones a la carga de hasta 50v. Además tiene un pin COMMON que permite el acceso a los diodos incluidos en el chip, cuya tarea es proteger los transistores del mismo frente de picos de sobretensión generados por cargas de tipo inductivo, como motores o bobinas.

1.6.3. INTEGRADO 74LS244

Eléctricamente, el microcontrolador entrega señales TTL y como tal, teóricamente, se le puede conectar cualquier dispositivo que cumpla con los niveles de voltaje específicos de la lógica TTL, sin embargo está muy limitado en cuanto a su capacidad de manejo de corriente.

¹⁹ http://picaxe.electronicasimple.com/2008/11/uln2803-qu-es-cual-es-su-funcin_11.html

El integrado **74LS244** mostrado en la figura 1.38, mejora la capacidad de manejo de corriente, y al mismo tiempo es etapa separadora, entre el microcontrolador y el hardware que este controla protegiéndolo de cortocircuitos.

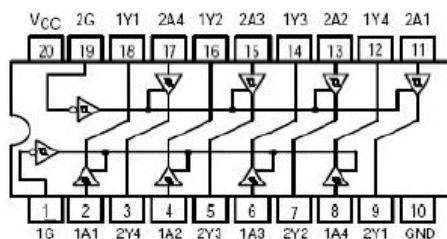


Figura 1.38: 74LS244

Estos circuitos contienen 8 compuertas buffer de 3 estados. Las señales 1G y 2G (Output Enable) habilitan o deshabilitan las salidas de las compuertas. Como su nombre lo indica, la lógica de 3 estados maneja un tercer estado: Alta impedancia.

Esto es, la salida de estas compuertas puede estar en estado alto (H), en estado bajo (L) o en Alta impedancia (Z). El estado de Alta impedancia simula la desconexión de la salida de la compuerta y dicha salida se considera deshabilitada

1.6.4. INTEGRADO 74LS595

El integrado 74LS595 es un registro de 8 bits de entrada serial y salida paralela, en la figura 1.39 se puede apreciar la distribución de pines.

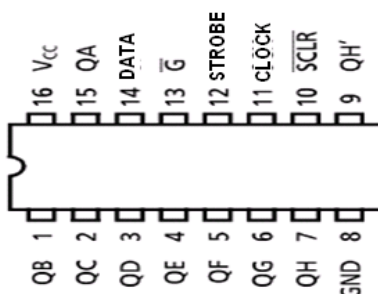


Figura 1.39: 74LS595

La tabla 1.6 describe la función de cada uno de sus pines

Pin	Descripción
QA...QH	Salida de datos paralela
DATA	Pin de entrada de datos
G	Habilita el integrado
STROBE	Pin de actualización de datos
CLOCK	Transfiere los datos de manera serial
SCLR	Blanquea en registro de salida en cero lógico
QH'	Salida de datos

Tabla 1.6: Descripción de pines del 74LS 595

Habilitado el integrado, la transferencia de datos se realiza de la siguiente manera:

1. El dato a transmitir se encuentra en el DATA
2. El pin SCLR es llevado a uno lógico
3. El pin STROBE es llevado a cero lógico.
4. El dato es guardado en el registro serial al detectar una transición positiva en CLOCK
5. Para actualizar los datos a la salida paralela, el pin STROBE es llevado a uno lógico.

1.6.5. DIODOS LED²⁰

El diodo LED es un dispositivo que proporciona un nuevo tipo de iluminación, de mayor calidad y con un consumo reducido de energía.

Gracias a la tecnología LED, los sistemas de iluminación son, al menos, diez veces más duraderos y resistentes que los sistemas de focos convencionales, y se consigue un ahorro energético mínimo del 33%, y de hasta un 85%.

²⁰ http://www.iearobotics.com/personal/ricardo/articulos/diodos_led/index.html

Son muchas las ventajas que ofrece la tecnología LED, enumerando estas a continuación de la forma más sencilla:

- **Bajo consumo:** Un LED se alimenta a baja tensión, la iluminación que se consigue (Lúmenes) por cada Watio de consumo es muy inferior al que ofrecen las lámparas convencionales.
- **Baja temperatura:** El reducido consumo del LED produce muy poco calor.
- **Amplia banda espectral:** El LED es un dispositivo de longitud de onda fija pero que puede trabajar en una amplia banda del espectro, que permite iluminar con una longitud de onda específica, o lo que es lo mismo en un determinado color.
- **Mayor rapidez de respuesta:** El LED tiene una respuesta de funcionamiento mucho más rápida que el halógeno y el fluorescente, del orden de algunos microsegundos.
- **Luz más brillante:** En las mismas condiciones de luminosidad que sus rivales, la luz que emite el LED es mucho más nítida y brillante.
- **Mayor duración y fiabilidad:** La vida de un LED es muy larga en comparación con los demás sistemas de iluminación. Se le calcula una vida útil de 50.000 horas.

Además los led's no presentan iguales características, pues algunos pueden ser vistos desde diferente ángulo, otros iluminan más pudiendo ser vistos perfectamente en el día y estos deben ser elegidos según las consideraciones de trabajo.

Las principales consideraciones a la hora de elegir un LED son las siguientes:

- **Aplicación:** La aplicación define el tipo de LED a usar, ya que existen led's comunes de baja potencia o de potencia para aplicaciones donde se necesita más luz por LED, por ejemplo si el uso de los led's va a ser de iluminación, decoración o para algún señalamiento (Letreros, pantallas).
- **Angulo:** Según la aplicación se determinará el ángulo que se necesita, se tiene que tener presente que entre más cerrado sea el ángulo del LED más

concentrado es el haz de luz que este emite, por lo tanto el LED será visto a mayor distancia.

- **Tipo de lente:** Es el encapsulado del LED puede ser: lente transparente y lente difusa con encapsulado opaco.
- **Intensidad:** Por lo general la potencia lumínica en led's que no son de potencia se manejan unidades de mcd (milicandelas), cuando los led's son de potencia se manejan unidades de lúmenes (lm).

1.6.6. FUENTE DE TENSION

Todos los circuitos electrónicos requieren de una o más fuentes de tensión estable de continua. Las fuentes sencillas construidas con un transformador, un rectificador y un filtro no proporcionan una calidad suficiente porque sus tensiones de salida cambian con la corriente que circula por la carga y con la tensión de la línea, y además presentan una cantidad significativa de rizado a la frecuencia de la red. Por ello no son generalmente adecuadas para la mayoría de las aplicaciones.

En la figura 1.40, se presenta una fuente de 3 amperios que transforma un voltaje de 110 voltios AC a 12 voltios DC



Figura 1.40: Fuente de Tensión

La fuente está internamente constituida por un transformador, que se encarga de transformar la tensión de corriente alterna que tiene en la entrada otra diferente a la salida. El transformador está constituido por un núcleo magnético formado por un grupo de chapas magnéticas y dos circuitos eléctricos o bobinados.

El circuito que recibe la energía eléctrica se llama primario y el que la suministra transformada, se llama secundario. Ambos bobinados están enrollados sobre el núcleo y son eléctricamente independientes, como se observa en la figura 1.41²¹

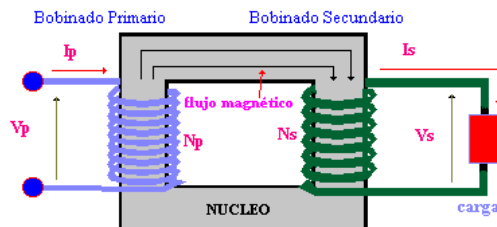


Figura 1.41: Esquema de un transformador simplificado.

La relación de transformación es la siguiente:

Potencia primaria $V_1 \cdot I_1$

Potencia secundaria $V_2 \cdot I_2$

Igualemos términos $V_1 \cdot I_1 = V_2 \cdot I_2$

$$\frac{V_1}{V_2} = \frac{I_2}{I_1} = m$$

Como la fuerza electromotriz inducida en una bobina es proporcional al número de espiras de la misma y la resistencia de una bobina es proporcional a la longitud de su hilo, y por tanto, de su número de espiras, tenemos que la relación de transformación de un transformador es función de estos cuatro parámetros:

$$\frac{V_1}{V_2} = \frac{I_2}{I_1} = \frac{N_1}{N_2} = \sqrt{\frac{Z_1}{Z_2}} = m$$

Donde: Z_1 y Z_2 representa respectivamente la impedancia de los bobinados.

En la fuente el voltaje en el secundario es rectificado utilizando un puente de diodos, como se presenta en la figura 1.42. Dada la figura 1.42 Supongamos que el punto A sea positivo (el punto B será negativo). La corriente eléctrica, al llegar al punto 1, polarizará al diodo D1, directamente y al D4 inversamente, con lo que la corriente circulará a través de D1, siguiendo por el punto 2 hacia la carga, de C a D, (por D2 no puede circular ya que se encuentra con el cátodo), llegando al

²¹ <http://www.alu.us.es/a/amaluqsen/Transformadores.doc>

punto 4 y siguiendo por D3, hasta el punto B y cerrando el circuito por el bobinado del transformador.

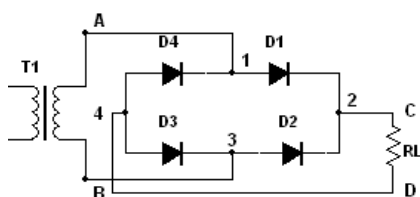


Figura 1.42: Rectificador de onda completa tipo puente de diodos.

Al siguiente semiciclo, el punto A es negativo, pero el B positivo. La corriente eléctrica saldrá por el punto B, llegará al punto 3; irá por D2 hasta el punto 2, luego por la carga en sentido C a D; llegará al punto 4, luego por D4, punto 1, punto A y se cerrará el circuito por el secundario del transformador, obteniendo en la carga una tensión unidireccional tal como se observa en la figura 1.43²²

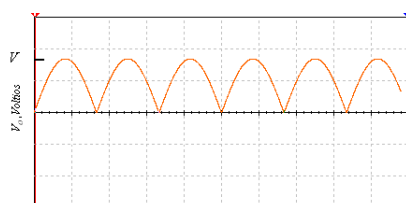


Figura 1.43: Forma de onda de salida para un rectificador de onda completa

Después de que el voltaje en el secundario es rectificado, la fuente solo posee un condensador cerámico y un electrolítico (figura 1.44) para mantener constante el voltaje, el filtrado de este circuito es ilustrado en la figura 1.45

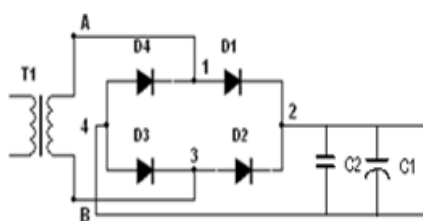


Figura 1.44: Rectificador de onda completa con filtro capacitivo.

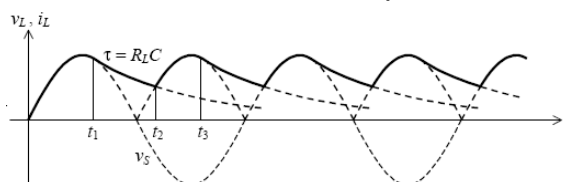


Figura 1.45: Salida del rectificador de onda completa con filtro capacitivo.

²²<http://www.juntadeandalucia.es/averroes/iesbahiadecadiz/pdf/electr/cesarsanchez/modulo8/tema8.pdf>

1.6.7. REGULADORES DE TENSION

Los reguladores de tensión comprenden una gran variedad de circuitos integrados. Proporcionan una tensión de salida fija positiva (reguladores de tensión positiva), fija negativa (reguladores de tensión negativos) o ajustable a través de potenciómetros.

Los reguladores de tensión fija poseen tres terminales (figura 1.46), tienen un terminal para la entrada no regulada (IN), la salida regulada (OUT) y tierra (COMMON) y están ajustados para proporcionar una tensión de salida constante.

Dentro de esta categoría se encuentra la serie 78XX (positivos) o 79XX (negativos). Los dos últimos dígitos indicados por XX indican la tensión de salida y pueden ser 05,06 ,08 ,12 ,15 ,18 ,24V.

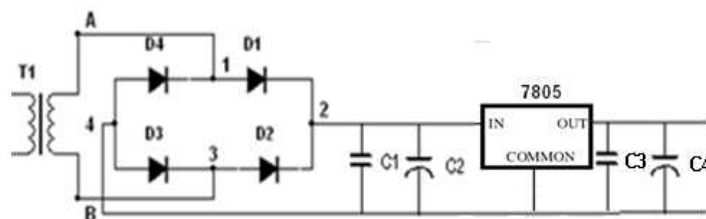


Figura 1.46: Regulador


Los reguladores de tensión poseen protección contra sobrecargas térmicas y contra cortocircuitos, que desconectan el regulador en caso de que su temperatura de juntura supere los 125°C, para que un regulador alcance su corriente máxima es necesario dotarlo de un disipador de calor adecuado, sin él solo obtendremos una fracción de esta corriente antes de que el regulador alcance su temperatura máxima y se desconecte.

Cuando un CI está conectado a unos cuantos centímetros del filtro condensador de la fuente de alimentación no regulada, la inductancia del cable de conexión puede producir oscilaciones dentro del CI. Por esta razón, los fabricantes recomiendan utilizar un condensador de desacoplo, para mejorar la respuesta transitoria del voltaje de salida.

1.7.LENGUAJE DE PROGRAMACIÓN BASCOM AVR

La herramienta BASCOM AVR desarrollada por la empresa MCS Electronics, sirve para realizar programas de alto nivel para microcontroladores AVR. Ofrece una completa solución para editar, compilar, simular y programar. Posee un compilador y un ensamblador que traduce las instrucciones estructuradas en lenguaje de máquina.²³

Para iniciar programando en Bascom Avr es posible descargar su versión demo, en la página principal de MCS Electronics.

Después de haber instalado Bascom dar doble click sobre el icono  que permite abrir el programa (figura 1.47).

La siguiente pantalla es apreciada:

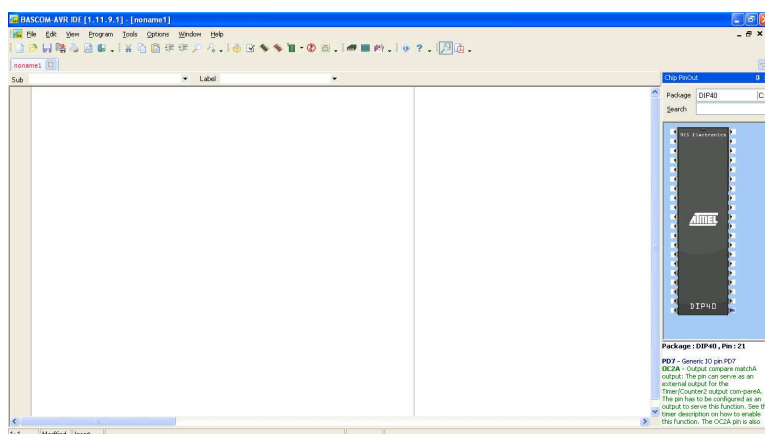



Figura 1.47: Ambiente del Bascom Avr

Dentro de ella podemos ver claramente la barra de herramientas, el menú y el área de trabajo. A continuación se explicara los iconos o atajos más importantes para manejar la herramienta BASCOM AVR.

ICONO	DESCRIPCION
	New: Crea un nuevo archivo

²³ Ramiro Valencia B, “Aplicaciones Electrónicas con Microcontroladores”



Find Text: Busca un determinado texto en el programa



Simulate Program : Permite simular el programa creado, una vez que se ha compilado



Compile Program: Compila el proyecto que se ha creado para obtener el archivo .hex que será grabado en el microcontrolador.

Al compilar un programa presionando el icono de la barra de herramientas o F7, si no existe ningún error se observa el cuadro de confirmación mostrado en la siguiente figura



Figura 1.48: Cuadro de confirmación de compilación

En el cual es posible visualizar el porcentaje de memoria que se encuentra utilizado del microcontrolador.



Pin Layout: Dando click sobre este icono se visualiza la ventana mostrada en la figura 1.49, que indica el integrado utilizado y también las características de cada pin cuando este es seleccionado con el mouse.

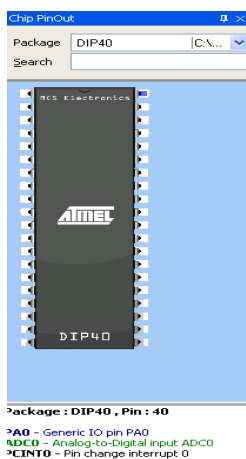


Figura 1.49: Integrado Utilizado

1.7.1. PRINCIPALES SENTENCIAS DE BASCOM

1.7.1.1. Directivas del Compilador

Son las instrucciones iniciales que el programa requiere para determinar las características del integrado, estas son las siguientes:

1.7.1.1.1. *\$regfile*

Esta instrucción siempre va al inicio del proyecto a realizar, determina el microcontrolador que será utilizado.

Por ejemplo si vamos a utilizar:

ATMEGA 48 → **\$regfile="m48def.dat"**

ATMEGA 16 → **\$regfile="m16def.dat"**

ATMEGA 8 → **\$regfile="m8def.dat"**

1.7.1.1.2. *\$crystal*

Esta instrucción permite determinar la frecuencia de oscilación con la que va a funcionar el microcontrolador.

Ejemplo:

\$crystal=1000000 para 1Mhz

\$crystal=8000000 para 8Mhz

1.7.1.2. Configuraciones Iniciales

Inicializan un pin o grupo de pines para que realicen una tarea específica.

1.7.1.2.1. *Config*

Esta instrucción especifica la configuración de un pin, un puerto o un dispositivo, ya que pueden ser configurados como entradas o salida de datos.

Ejemplo:

Config portb = output Declara todo el puerto B como salida

Config pina.0 = input Pin A.0 como entrada

1.7.1.2.2. *DDR_x, PORT_x, PIN_x*

DDR, PORT Y PIN son registros que nos permiten utilizar el puerto como entrada o salida de datos.

DDR : Configura el pin como entrada o salida de datos.

PORT : Es el registro de salida de datos.

Pin : Es el registro de entrada de datos.

La tabla 1.7 muestra las combinaciones, para que los pines funcionen en configuración especial.

DDR _x	PORT _x	I/O	Pull up	Comentario
0	0	Entrada	No	Tercer estado (Alta impedancia)
0	1	Entrada	Si	
1	0	Salida	No	Salida Push-Pull en Cero
1	1	Salida	No	Salida Push-Pull en Uno

Tabla 1.7: Configuración Especial de Pines

Ejemplo:

Ddrb.0 = 1 : Portb.0 = 0 : Salida_1 Alias Portb.0 Puerto definido como salida

Ddrd.7 = 0 : Portd.7 = 1 : Entrada_1 Alias Pind.7 Puerto definido como entrada

Es importante recalcar que cuando se configura un puerto como salida, se debe ocupar la palabra PORT y si se lo configura como entrada se usa la palabra PIN.

Ejemplo

Dim Dato_1 As Byte Declara Dato_1 como byte.

Dim Dato_2(10) As Byte Declara Dato_2 como una matriz de bytes de 10 elementos.

Dim Dato_3 As String *10 Declara Dato_3 como una String de 10 elementos

Dim Dato_4 As Word At \$100 Declara Dato_4 como una Word ubicados en la dirección 100 de la memoria.

Dim Dato_5(2) As Byte At \$100 Overlay Declara Dato_5 en la misma dirección de la memoria de Dato_4.

Cuando se declara variables en una dirección específica de la memoria se tiene control sobre esta, para poder dividir los datos en una cadena de caracteres o grupo de bytes declarando otra variable en la misma dirección, esto se ilustra en la tabla 1.9, asumiendo que **Dato_4** tiene cargado el valor hexadecimal 27B5

Dato_4		Dato_4 tiene el valor 27B5
27	B5	
Dato_5		El byte menos significativo de Dato_5 es B5
27	B5	El byte más significativo de Dato_5 es 27

Tabla 1.9: Variables declaradas en una misma dirección

1.7.1.4. Manipulación de Bits

1.7.1.4.1. Reset

Con este comando se lleva un pin del microcontrolador al estado de 0 lógico.

Ejemplo:

Ddrb.0 = 1 : Portb.0 = 0 : Foco Alias Portb.0

Reset Foco

1.7.1.4.2. Toggle

Este comando sirve para complementar el estado anterior de alguna variable o pin de algún puerto.

Ejemplo:

Ddrb.0 = 1 : Portb.0 = 0 : Foco Alias Portb.0

Foco =1

Toggle Foco Complementa Foco → Foco=0

Toggle Foco Complementa Foco → Foco=1

1.7.1.5. Manipulación de Strings

1.7.1.5.1. INSTR

Retorna la posición de una substring en una string

var = **INSTR**(Start, string , substr)

var = **INSTR**(string , substr)

Var: Variable numérica donde será asignada la posición de la substring en la string. Retorna cero cuando no es encontrada.

Start: Parámetro opcional que permite asignar la posición de inicio donde se desea que busque la substring. Por default cuando no es usada empieza la búsqueda desde la primera posición.

String: String principal.

Substr: La string a encontrar.

Ejemplo:

Dim **bit_posicion** as **Byte**

Dim **s_principal** as **string x 30**

Dim **s_buscar** as **string x 2**

s_principal = "This is a test" : s_buscar = "a"

bit_posicion = **Instr** (1, s_buscar _principal, z) → bit_posicion = 9

1.7.1.5.2. LOOKUPSTR

Retorna una string que se encuentra dentro de una tabla de datos, dado la posición de la tabla.

var =LOOKUPSTR(value, label)

var: La string que es retornada

value: Es una variable que lleva el índice (posición de la variable dentro de la tabla). Si el índice es cero retorna el primer elemento de la tabla.

Label: Etiqueta donde se encuentran los datos

Ejemplo:

Dim **s** as string, **idx** as Byte

idx = 0

s = **LookupStr**(idx,Sdata) → “**s**” se carga con "**This**"

Sdata:

Data "This", "is" , "a test"

1.7.1.5.3. Mid

La función MID retorna parte de una string a una sub string.

var = **MID**(var1 ,st [, l])

La función MID reemplaza parte de una variable de una string por otra string.

MID(var ,st [, l]) = var1

var : La string que es asignada.

var1: La string fuente.

st : Es la posición de inicio.

l : Define el número de caracteres a conseguir o modificar

Ejemplo:

Dim **S_pr** As String x 15, **Z** As String * 15

S_pr = "ABCDEFGH"

Z="12345"

z = **Mid**(**S_pr**,2,3) \rightarrow **z**=BCD

Mid(**S_pr**,2,2) = **z** \rightarrow **S_pr** =A12DEFG

Z se carga con 3 caracteres de **S_pr** seleccionados desde la segunda posición

A **S_pr** se adicionan los primeros 2 caracteres de **Z** desde la segunda posición

Además se debe tener en cuenta que la posición inicial de una string siempre empieza en 1.

1.7.1.6. Instrucciones de Uso General

1.7.1.6.1. *Wait*

Esta instrucción permite realizar una pausa, ya sea en segundos, milisegundos y microsegundos respectivamente.

Ejemplo:

Wait 3 Espera 3 segundos

Waitms 700 Espera 700 milisegundos

Waitus 500 Espera 500 microsegundos

1.7.1.6.2. *Incr*

Incrementa el valor de una variable

Ejemplo:

Dim **A** As byte

Incr A

1.7.1.6.3. Decr

Decrementa el valor de una variable

Ejemplo:

Dim **A** As byte

Decr A

1.7.1.6.4. Lookup

Retorna un valor de una tabla de datos dado el índice.

var = **Lookup**(value, label)

Var: Valor retornado

Value: Valor que tiene el índice de la tabla

Label: Etiqueta donde se encuentran los datos

El número máximo de elementos que puede tener la tabla es 65535.

Ejemplo:

Dim **s** As byte, **idx** as Byte

idx = 2

s = **Lookup** (idx,Sdata) ⇒ “**s**” se carga con **20**

Sdata:

Data 5,10,20,36

1.7.1.6.5. Lookdown

Retorna la posición que tiene una variable dentro de una tabla (índice)

var = **Lookdown**(value, label, entries)

Var: Guarda el valor del índice retornado.

Value: Es la variable a ser buscada

Label: Es la etiqueta donde se encuentran los datos

Entries: Es el número de entradas que debe ser buscado el dato.

Ejemplo:

Dim **idx** As Byte, **Entries** As Byte

Dim **isearch** As String

isearch="C"

Idx = **Lookdown**(isearch , Label , Entries) → Idx =2

Label:

Data "A","B","C","D","E"

Si el dato a buscar no es encontrado el valor de -1 es retornado.

1.7.2.SIMBOLOS OPERADORES

Dentro de los operadores, pueden utilizarse los matemáticos, de relación y lógicos.

Además se debe tomar en cuenta que BASCOM permite realizar operaciones únicamente con dos variables a la vez. A continuación podremos observar los operadores más comunes.

Operadores Matemáticos

Suma: $a=b+c$

Resta: $a=b-c$

Multiplicación: $a=b*c$

División: $a \text{ MOD } b$

Operadores de relación

= Igual $X=Y$ <> No es igual $X<>Y$

< Menor que $X<y$ > Mayor que $X>Y$

<= Menor Igual $X<=Y$ >= Mayor Igual $X>=Y$

Operadores lógicos

NOT	Complemento (Negación)	AND	Conjunción (Y)
OR	Disyunción(O)	XOR	Or Exclusiva

1.7.2.1. Representación de Lógica Digital

Para la representación de un número binario o hexadecimal, dentro de BASCOM AVR, es necesario anteponer el símbolo "&". En el caso de números decimales, no es necesario anteponer ningún símbolo.

Ejemplo:

Porta= &HC4	Número Hexadecimal
Porta= &b10000011	Número binario
Porta= 396	Número decimal

1.7.3. DECISION Y ESTRUCTURAS²⁴

1.7.3.1. Do – Loop

Esta sentencia crea un lazo cerrado, en el cual se ejecuta un conjunto de instrucciones de forma indefinida.

1.7.3.2. If – Them – Else

Son sentencias condicionales, que permiten condicionar la ejecución de instrucciones, basados en la evaluación entre dos o más variables usando los operadores lógicos.

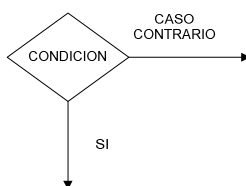


Figura 1.50: Condición Lógica IF-ELSE

²⁴ Ramiro Valencia B, "Aplicaciones Electrónicas con Microcontroladores"

1.7.3.3. For – Next

Es una sentencia de repetición, dentro de esta sentencia se ejecutan un grupo de instrucciones hasta que se cumpla la condición que finaliza el lazo, esta condición de fin está dada por una variable que se decrementa o incrementa en pasos previamente establecidos.

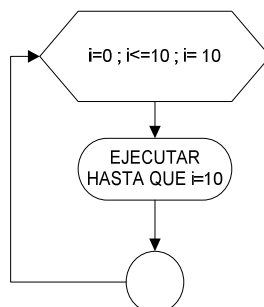


Figura 1.51: Condición de repetición FOR – NEXT

1.7.3.4. Select – Case

Son sentencias que se pueden ejecutar, dependiendo del valor de una variable de selección. Dentro de este esquema se puede tener un conjunto de casos que pueden ser ejecutados, dependiendo del valor de la variable que selecciona el caso.

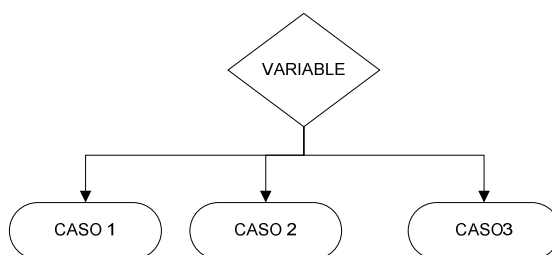


Figura 1.52: Condición de selección SELECT – CASE

1.7.3.5. Gosub

Esta sentencia obliga al programa a saltar a una subrutina, en donde ejecuta las instrucciones definidas para luego regresar y continuar con el programa

Ejemplo:

Do

Gosub Incrementar

Loop

Incrementar:

A=A+1

If A > 50 **And** A < 60 Then **Goto** No_ejecutar

B=B+5

No_ejecutar:

Return Esta sentencia determina el fin de la subrutina

1.7.4.LCD (DISPLAY DE CRISTAL LIQUIDO)

En el caso del BASCOM AVR, podemos controlar al LCD de dos maneras: por comandos o por configuración en cuadro de dialogo.

1.7.4.1. Configuración Mediante Comandos²⁵

1.7.4.1.1. *Config Lcd*

Sirve para configurar el LCD a ser utilizado, puede ser de 16 caracteres por dos líneas (16x2), de 20 caracteres por 4 líneas (20x4), etc.

Ejemplo:

Config Lcd=16 x 2 (Display de 16 x 2)

1.7.4.1.2. *Config lcdbus*

Esta instrucción sirve para configurar cual será el modo de envío de datos, ya que puede ser hecho mediante un bus de datos de 4 pines o de un bus de datos de 8 pines.

²⁵ Manual BASCOM-AVR <http://www.mcselec.com>

Ejemplo:

Config Icdbus=4 (4 pines de datos)

1.7.4.1.3. Config Lcdpin

Sirve para configurar los pines por los cuales se enviará la información a ser mostrada en el LCD.

Ejemplo:

Config Lcdpin= Pin, **Db4**=Porta.4, **Db5**=Porta.5, **Db6**= Porta.6, **Db7**=Porta.7, **E**=Portc.7, **Rs**=Portc.6

Esta configuración controla un bus de datos de 4 pines, y permite utilizar en otras aplicaciones los pines: Db0, Db1, Db2 y Db3 utilizados en una configuración de un bus de datos de 8 pines.

1.7.4.2. Mediante Cuadro de Dialogo

BACOM AVR, permite interactuar con el hardware, mediante cuadros de diálogo, en los cuales podemos elegir las configuraciones que tendrá LCD a ser utilizado.

Ingresando mediante el menú de opciones seleccionamos LCD, para visualizar sus configuraciones y pines de control, tal como se observa en la figura 1.53.

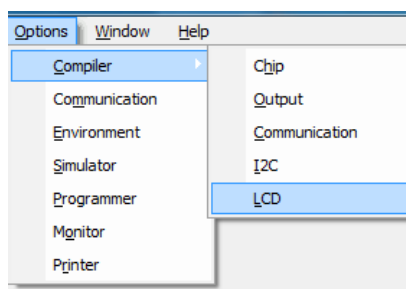


Figura 1.53: Cuadro de dialogo del BASCOM AVR para configurar un LCD

La figura 1.54 muestra cómo podemos configurar los pines y el tipo de LCD que utilizaremos.

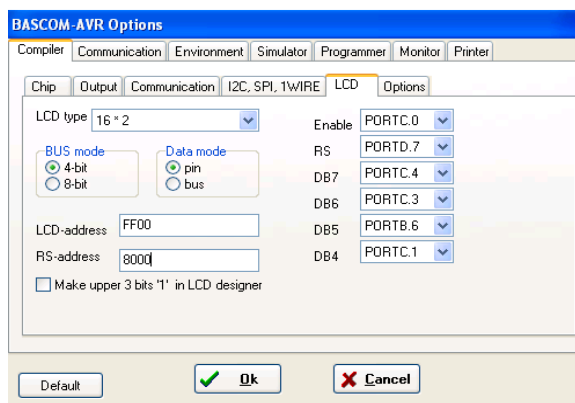


Figura 1.54: Cuadro de dialogo de configuraciones del LCD

Dentro de este cuadro de diálogo seleccionamos el bus de datos a utilizar y el tipo de LCD.

En **Data mode** seleccionamos **pin**, para definir los pines que serán utilizados para la transmisión de datos, Enable y Rs.

Una ventaja que brinda Bascom Avr al controlar el LCD es que se puede seleccionar, cualquier pin para el control de datos, no limitando su control a un puerto definido.

Es recomendable realizar todas las configuraciones de los dispositivos mediante código y mediante cuadros de diálogo; de esta manera nos aseguramos que la información de las interfaces de salida, no se pierda en ningún momento.

1.7.4.3. Instrucciones Para el Control del Lcd

1.7.4.3.1. Locate x,y

Sirve para ubicar el cursor en la línea y columna adecuada, para poder empezar a escribir en el LCD

Ejemplo:

Locate 1,1 (Localización del cursor en la fila 1, columna 1)

1.7.4.3.2. Lcd ” ”

Sirve para escribir cualquier frase en el LCD, la información es observada siempre a continuación de la posición en la que está el cursor.

Ejemplo:

Lcd= "HOLA"

1.7.4.4. Lcd Designer

Esta opción permite diseñar caracteres especiales para ser mostrados en el LCD, se selecciona esta opción ingresando en el menú de herramientas (figura 1.55).



Figura 1.55: Selección LCD Designer

Luego se observa la ventana del diseñador de caracteres (figura 1.56).

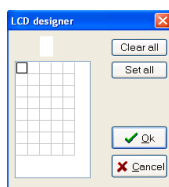


Figura 1.56: LCD Designer

Esta ventana (figura 1.57) es una pequeña matriz de 5 x 7 pixeles que permite dar un **click** sobre los cuadros para formar la figura deseada.

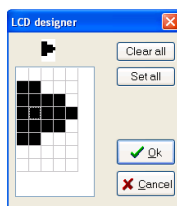


Figura 1.57: Creación de un Carácter

Finalizado la figura se da un click en **OK**, a continuación un código parecido al siguiente es creado:

```
Deflcdchar ?,32,24,30,31,30,24,32,32
```


El carácter "?" debe ser reemplazado por un número entre 0 y 7.

```
Deflcdchar 1,32,24,30,31,30,24,32,32
```

La figura creada es observada en el LCD mediante la siguiente instrucción:

```
Lcd Chr(1) El número es el que fue reemplazado por ?.
```

1.7.5.SIMULADOR

El simulador que tiene Bascom Avr permite verificar el funcionamiento de un programa mediante la verificación de su lógica de funcionamiento, se accede al simulador dando un click sobre el icono  o presionando F2, compilando previamente el programa. Accediendo al compilador la figura 1.58 es observada.

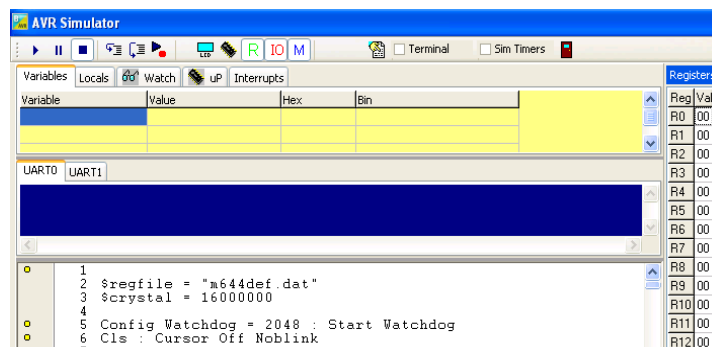


Figura 1.58: Simulador

Los iconos utilizados para verificar el funcionamiento del programa son los siguientes:





ICONO DESCRIPCION



Da inicio a la simulación de un programa



Pausa una simulación

-  Finaliza la simulación
-  Permite simular un programa línea por línea de código.
-  Permite saltar una subrutina
-  Permite observar los datos escritos que serán visualizados en un LCD

1.7.5.1. Asignando una Ruta a Seguir en el Simulador

Las instrucciones programadas, pueden ser verificadas avanzando línea por línea en su código creado. Al iniciar la simulación se observa una flecha que indica el número de línea que se encuentra ejecutando el simulador, como lo indica la siguiente figura 1.59.

```

1627 Seleccionar_efectos:
1628   Cls
1629   Select Case Tipo_efecto
1630     Case 1 : Lcd " TEXTO CORRIDO "
1631     Case 2 : Lcd " AVANZA DETIENE "
1632     Case 3 : Lcd " AVANZA FLASH "
1633     Case 4 : Lcd " PRESENTA TEXTO "
1634     Case 5 : Lcd " DESTELLANTE "
1635     Case 6 :
1636       Lcd Chr(3) : Lcd " ABRIR TEXTO " : Lcd Chr(3)
1637     Case 7 :
1638       Lcd Chr(4) : Lcd " ABRIR TEXTO " : Lcd Chr(4)
1639     Case 8 :
1640       Lcd Chr(5) : Lcd " ABRIR TEXTO " : Lcd Chr(5)
1641     Case 9 :
1642       Lcd Chr(6) : Lcd Chr(7) : Lcd "ABRIR TEXTO" : Lcd Chr(6) : Lcd Chr(7)
1643     Case 10 :
1644       Lcd Chr(4) : Lcd "ASCENDER TEXTO" : Lcd Chr(4)
1645     Case 11 :
1646       Lcd Chr(5) : Lcd "DESCENDER TEXT" : Lcd Chr(5)
1647     Case 12 :
1648       Lcd Chr(7) : Lcd " DELETREANDO " : Lcd Chr(7) : Lcd Chr(7)
1649     Case 13 : Lcd "ABRIENDO LETRAS "
1650   End Select
1651   Locate 2 , 1 : Lcd Character
1652 Return

```

Figura 1.59: Simulador

Se asigna la ruta a seguir por el simulador, seleccionando una variable (figura 1.60) y dando un valor que cumpla con las condiciones del programa (figura 1.61).

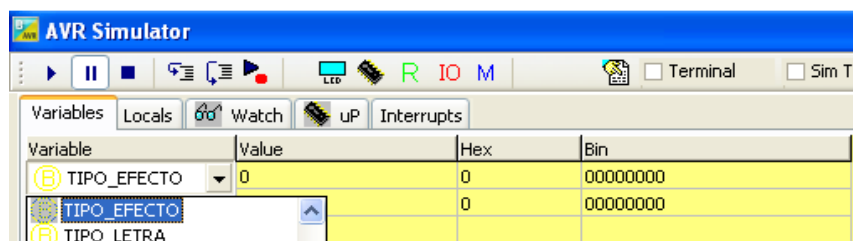


Figura 1.60: Selección de variables

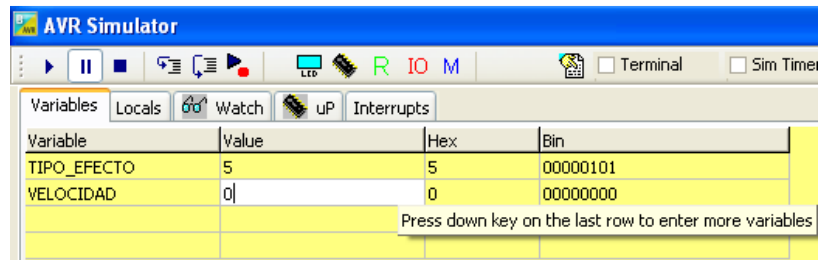


Figura 1.61: Asignación de Valores

Continuando con la simulación se puede comprobar su lógica, observando en el simulador del LCD los valores de las variables o las frases escritas y el salto de posición que realiza la flecha tal como lo indican la figura 1.62 y la figura 1.63.

```

1627 Seleccionar_efectos:
1628     Cls
1629     Select Case Tipo_efecto
1630         Case 1 : Lcd " TEXTO CORRIDO "
1631         Case 2 : Lcd " AVANZA DETIENE "
1632         Case 3 : Lcd " AVANZA FLASH "
1633         Case 4 : Lcd " PRESENTA TEXTO "
1634         Case 5 : Lcd " DESTELLANTE "
1635         Case 6 :
1636             Lcd Chr(3) : Lcd " ABRIR
1637         Case 7 :
1638             Lcd Chr(4) : Lcd " ABRIR
1639         Case 8 :
1640             Lcd Chr(5) : Lcd " ABRIR
1641         Case 9 :
1642             Lcd Chr(6) : Lcd Chr(7) :
1643         Case 10 :
1644             Lcd Chr(4) : Lcd "ASCENDE
1645         Case 11 :
1646             Lcd Chr(5) : Lcd "DESCEND
1647         Case 12 :
1648             Lcd Chr(7) : Lcd " DELETR
1649         Case 13 : Lcd "ABRIENDO LETRAS "
1650     End Select
1651     Locate 2 , 1 : Lcd Character
1652 Return

```

Figura 1.62: Salto de Posición

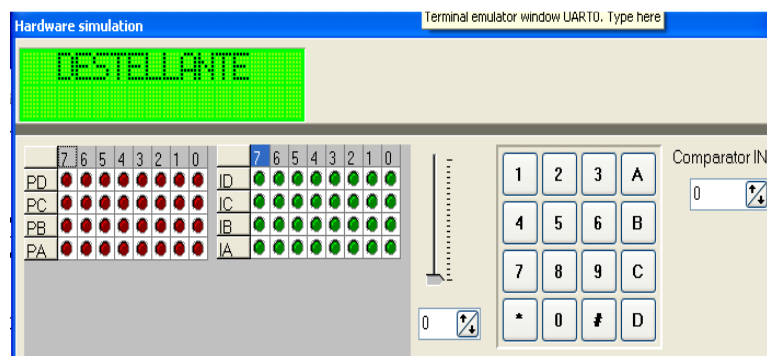


Figura 1.63: Simulador LCD

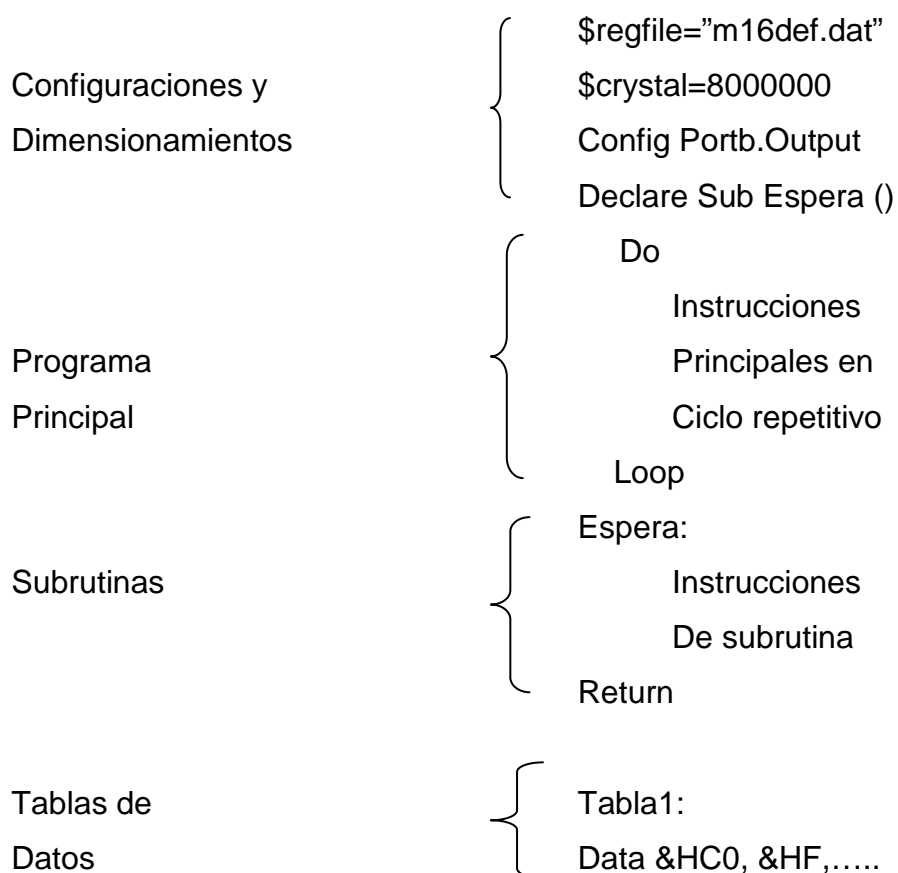
1.7.6. ESTRUCTURA DE UN PROGRAMA ²⁶

Cuando se estructura un programa en alto nivel, es necesario llevar un orden y vinculación de las instrucciones que se realizan.

Es primordial que se tengan estructuradas 4 partes dentro de un programa en lenguaje de alto nivel.

1. Configuraciones y Dimensionamientos de variables y subrutinas
2. Programa principal
3. Subrutinas
4. Tablas de datos

El siguiente es un ejemplo de cómo se puede estructurar un programa en alto nivel, con tipos de instrucciones que se pueden realizar en su respectivo orden.



²⁶ Ramiro Valencia B, "Aplicaciones Electrónicas con Microcontroladores"

1.7.7. PROGISP 1.6.7

Al compilar un programa en Bascom Avr se crea el archivo hexadecimal .hex el cual posee todas instrucciones que el microcontrolador necesita para funcionar, este archivo es guardado en el microcontrolador utilizando el grabador Progisip.

El grabador posee las siguientes características:

1. Se comunica con el computador mediante un puerto USB
2. Posee un jumper el cual permite alimentar al microcontrolador con el voltaje del computador o con una fuente externa.
3. Posee un jumper que selecciona la velocidad de grabación.
4. Para guardar el archivo hexadecimal el grabador posee 6 pines de conexión

En la figura 1.64 se puede observar los pines de conexión entre grabador y el microcontrolador.

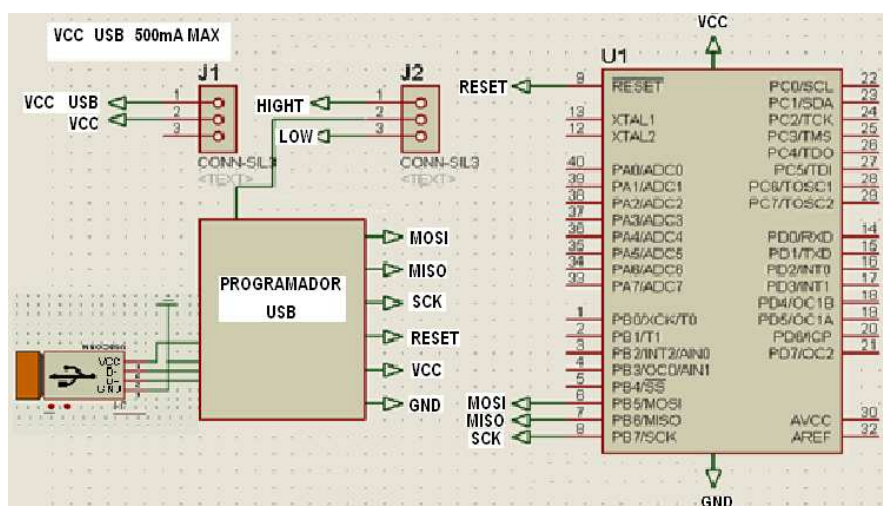


Figura 1.64: Programador USB Progisip 167

Los pines de conexión son: miso, mosi, sck, reset, vcc y gnd, del grabador se conectan directamente a los pines del micro que poseen la misma nomenclatura.

El grabador posee además un software amigable que permite seleccionar el microcontrolador a grabar, y sus respectivos “fuse bits”, los cuales permiten:

- Seleccionar el nivel de voltaje de funcionamiento
- El oscilador con el cual trabajará el microcontrolador
- Definir si el oscilador es interno o externo

También posee opciones para la programación que permite:

- Leer el contenido del microcontrolador
- Borrar el contenido del microcontrolador
- Verificar si la grabación se realizó correctamente
- Cargar automáticamente el archivo hexadecimal
- Proteger al archivo hexadecimal contra lectura

La figura 1.65 permite observar las configuraciones en el software de los “fuse bits” y sus opciones de programación para un integrado ATmega 644.

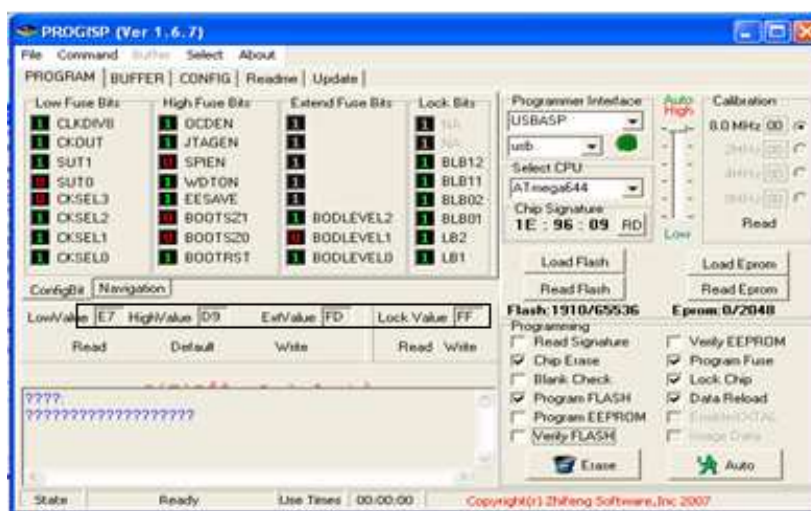


Figura 1.65: Pantalla de grabación del programador USB

En esta configuración al microcontrolador se asigna un oscilador externo de trabajo de 16 Mhz mediante la selección de los casilleros CKSELX y se ha protegido el código del chip contra lectura mediante el uso de Lock Chip ubicado en las opciones de programación.

Se determina todas las configuraciones de los “fuse bits” mediante el uso de la respectiva hoja de datos del microcontrolador la cual puede ser descargada desde www.atmel.com , por ejemplo la tabla 1.10, indica las configuraciones de los fuse bits de CKSEL para determinar el tipo de oscilador que utilizara un microcontrolador ATmega 644.

Frequency Range ⁽¹⁾ (MHz)	CKSEL3..1	Recommended Range for Capacitors C1 and C2 (pF)
0.4 - 0.9	100 ⁽²⁾	–
0.9 - 3.0	101	12 - 22
3.0 - 8.0	110	12 - 22
8.0 - 16.0	111	12 - 22

Tabla 1.10: Configuración de fuses de oscilador

Esta tabla determina que para un oscilador de 8 Mhz a 16 Mhz los pines menos significativos de CKSEL deben estar en 1, recomendando también los valores de los condensadores a utilizar con el oscilador.

1.8.EAGLE

El Eagle Layout Editor es una poderosa herramienta que permite diseñar circuitos impresos de forma fácil.

El nombre Eagle es un acrónimo de “Easily Applicable Graphical Layout Editor”, el programa consiste de tres principales módulos editor de trazado, editor esquemático y auto ruteado, pudiendo editar con estos módulos los archivos que formarán parte de un circuito impreso.

En la tabla 1.11²⁷ se muestra un listado con los tipos de archivos más importantes que pueden ser editados con Eagle.

²⁷ http://picmania.garcia-cuervo.net/eagle_tutlbr_i_library.php

Tipo	Ventana	Nombre
Placa	Editor de líneas de conexión	*.bdr
Esquema	Editor de esquemas	*.sch
Librería	Editor de librerías	*.lbr

Tabla 1.11: Archivos de Eagle

1.8.1. EDITOR DE ESQUEMAS

En esta ventana se realiza el diagrama esquemático del circuito con todas las conexiones que requiere el diseño.

La figura 1.66 permite observar la ventana del editor de esquemas con un circuito interno.

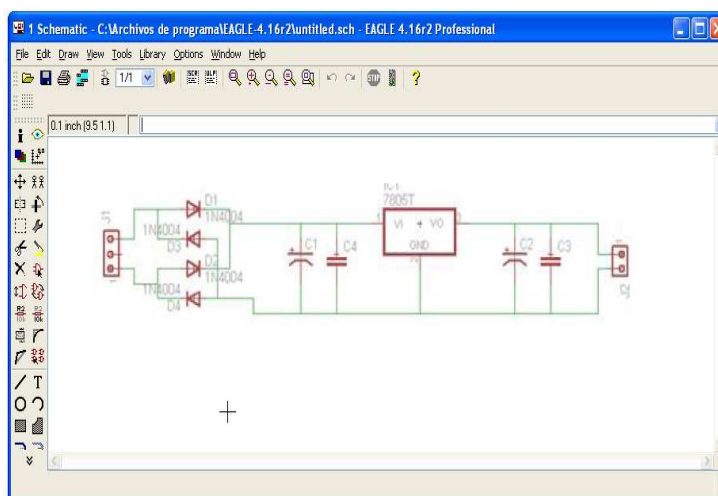


Figura 1.66: Editor de Esquemas

1.8.2. EDITOR DE LINEAS DE CONEXION

En este archivo los componentes del circuito, creados en el diagrama esquemático son observados conservando su forma real y ocupando un espacio definido.

El editor de líneas permite crear las pistas de conexión entre los elementos que forman parte del circuito estableciendo un tamaño y forma definida para que finalizado su diseño pueda este ser imprimido, transferido a una baquelita y quemado.

La figura 1.67 muestra el circuito del diagrama esquemático finalizado en el editor de líneas con sus respectivas conexiones.

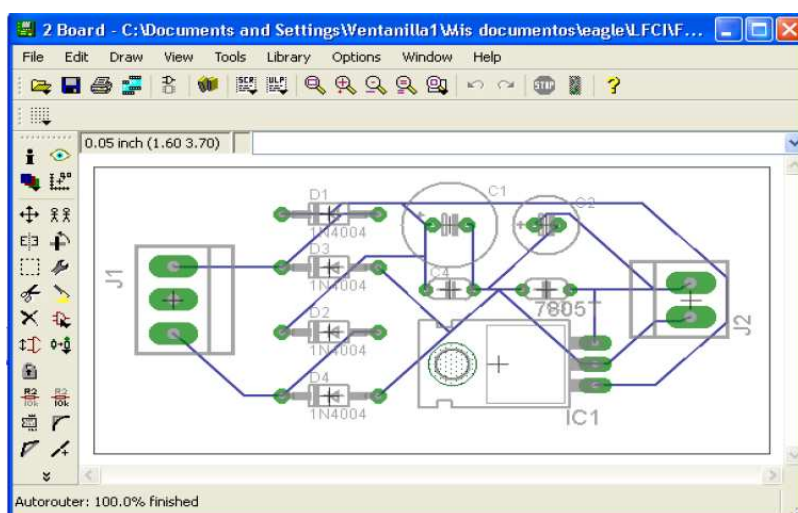


Figura 1.67: Editor de líneas de conexión

1.8.3. EDITOR DE LIBRERIA

Eagle facilita el diseño de un circuito, debido a que permite crear librerías propias para elementos que no sean encontrados dentro de sus librerías disponibles o que necesiten ser modificados en alguna de sus características, permitiendo crear un componente con conexiones específicas.

Una librería está compuesta de uno o varios componentes electrónicos. Cada elemento tiene tres archivos, uno a utilizar en el editor esquemático, otro para el editor de líneas y uno que será presentado en el panel de control con sus características.

Todos estos archivos poseen una conexión común entre pines de entrada, salida o alimentación para formar un elemento.

En la figura 1.68 se puede observar el panel de control con la librería de un elemento presentando su diagrama esquemático y el paquete que será utilizado en el editor de líneas de conexión.

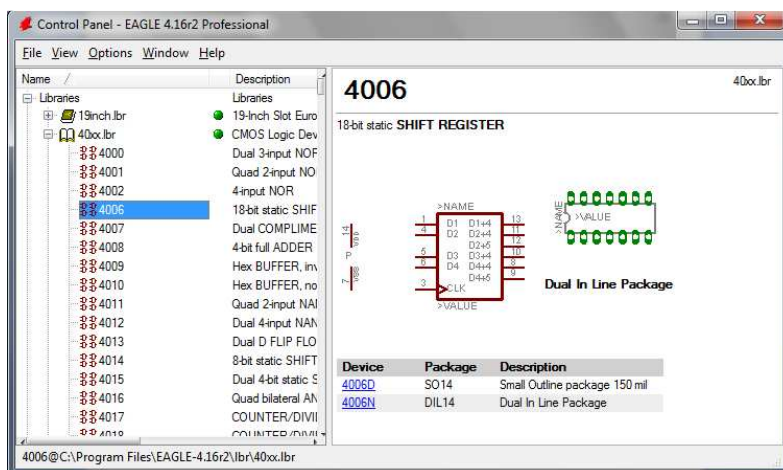


Figura 1.68: Panel de Control

CAPITULO II

2. CONSTRUCCION DEL SISTEMA PUBLICITARIO

El presente capítulo describe los bloques que conforman el sistema electrónico publicitario, indicando la manera que cada bloque interactúa para que un mensaje pueda ser visualizado en el rótulo.

2.1. DIAGRAMA DE BLOQUES DEL SISTEMA PUBLICITARIO

La figura 2.1 indica el diagrama que constituye el sistema electrónico publicitario con los bloques que lo conforman.

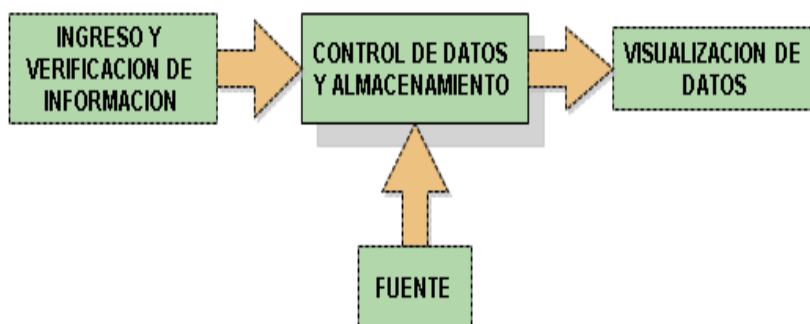


Figura 2.1: Diagrama de bloques del sistema electrónico publicitario

El sistema electrónico publicitario está compuesto de cuatro bloques principales.

Un bloque provee voltaje a todo el sistema publicitario y tres bloques controlan el ingreso de datos, la escritura o lectura de los mismos y el proceso de control de la información para la presentación de los datos.

El ingreso y verificación de la información está conformado por dos componentes, el teclado y el LCD. El teclado permite realizar una aplicación definida mediante la selección de un menú para el ingreso de diferentes tipos de datos. El LCD es la interface que permite que el usuario pueda observar la información que se encuentra escribiendo o el parámetro que está modificando.

El control de datos y almacenamiento está conformado por el microcontrolador ATmega 644, una memoria 24LC256 y el integrado de reloj DS1307. El ATmega 644 es el eje central del sistema electrónico publicitario debido a que posee el programa principal para controlar los periféricos que forman parte del circuito, controlando el ingreso y la salida de datos. El microcontrolador escribe o lee la información que será utilizada para el control del sistema publicitario en la memoria 24LC256 o en el integrado de reloj DS1307, utilizando la comunicación I2C, guardando los datos en el integrado adecuado.

La salida de datos está conformada por dos matrices de led's en donde es posible observar diferentes mensajes con diversas formas de presentación de texto y efectos.

2.2. CONSTRUCCIÓN DEL SISTEMA PUBLICITARIO

2.2.1. CONTROL DE DATOS Y ALMACENAMIENTO

2.2.1.1. Microcontrolador ATmega 644

El microcontrolador ATmega 644 controla todo el proceso que gobierna al sistema publicitario utilizando sus pines para realizar funciones definidas. La figura 2.2 muestra el nombre o alias que tiene cada pin utilizado del microcontrolador.

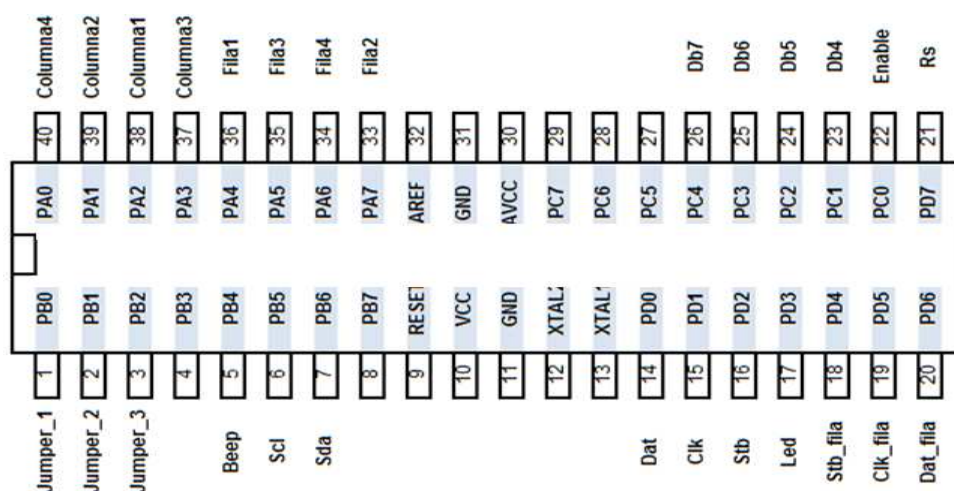


Figura 2.2: ATmega 644

En la tabla 2.1 se describe los pines del microcontrolador agrupados según el tipo de función que realizan, con su respectivo alias y descripción.

FUNCION	ALIAS	DESCRIPCION
Configuraciones Iniciales	Jumper_1	Son pines utilizados para guardar un registro interno del rótulo y también son utilizados para realizar dos test de prueba.
	Jumper_2	
	Jumper_3	
Verificación	Beep	Activa una chicharra, indicando que una determinada acción está siendo realizada.
	Led	Cambia el estado de un led, informando que el microcontrolador se encuentra trabajando.
Comunicación I2C	Scl	Pines utilizados para escribir o leer datos en la memoria externa y en el integrado de reloj.
	Sda	
Control de tarjeta de led's	Dat	Pines que envían diferentes datos que permiten controlar las columnas de la matriz de led's, del rótulo.
	Clk	
	Stb	
	Dat_fila	Pines que envían diferentes datos que permiten controlar las filas de la matriz de led's, del rótulo.
	Clk_fila	
	Stb_fila	
Teclado	Columna 4,3,2,1	Son pines que permiten controlar las filas y las columnas de un teclado 4 x 4, por el cual se ingresa o modifica diferentes datos.
	Fila 4,3,2,1	
LCD	Db7,6,5,4	Pines utilizados para el control del LCD
	Enable,Rs	

Tabla 2.1: Funciones de Pines

El programa principal, está constituido por tres secciones principales, en la primera sección el programa ingresa solo si se cumplen condiciones de habilitación, la segunda sección permite mostrar un mensaje inicial de la Corporación Electrónica Televid y la tercera sección permite acceder a los diferentes menús para configurar distintos mensajes.

La figura 2.3 muestra el diagrama de las secciones principales del programa.



Figura 2.3: Estructura principal del programa

2.2.1.1.1. *Configuraciones y Test del Rótulo*

Esta sección del programa permite realizar diferentes configuraciones y dos test de prueba en el sistema publicitario, el ingreso a esta sección es realizada siempre que el microcontrolador cuando inicie su trabajo, detecte que uno de los pines de configuración ha sido conectado a cero lógico. Cada jumper permite realizar las funciones presentadas a continuación:

1. El jumper_1 permite configurar, el número de mensajes que se podrán escribir en la memoria. Establecer si el mensaje inicial de la Corporación Electrónica Televid será visualizado con o sin número telefónico. Configurar la fecha de fabricación y el número de serie del rótulo.
2. El jumper_2 ingresa a un menú de test de teclas. Este menú verifica si una tecla es presionada, para enviar hacia el LCD, el número de tecla presionada, con esto se verifica el estado del teclado.
3. El jumper_3 realiza un test de los led's de las tarjetas. Este test enciende los led's fila por fila y una a la vez, progresivamente de derecha a izquierda para verificar que todos los led's funcionen correctamente.

2.2.1.1.2. Salida de Publicidad

Esta sección del programa permite presentar en el rótulo la frase “Corp. Electronica Televid CTE Quito – Ecuador”, esta publicidad se muestra siempre al iniciar el trabajo del rótulo y cuando ningún jumper es habilitado, pues si existe algún jumper habilitado el programa se encuentra en una subrutina de trabajo, que impide continuar con las demás secciones del programa.

La frase puede ser mostrada seguida del número telefónico, si utilizando el jumper_1, se configuró que el mensaje inicial, se muestre con el número telefónico. La frase visualizada es la siguiente “Corp. Electronica Televid CTE Quito – Ecuador 023001041”.

2.2.1.1.3. Menús de Usuario

El ingreso a esta sección del programa se realiza después de visualizar la publicidad. Esta sección tiene diferentes opciones para escribir un mensaje, modificarlo y seleccionar las características de presentación del mismo. Estas opciones son realizadas en diferentes menús de trabajo. En esta sección es posible realizar lo siguiente:

1. Escribir y modificar un mensaje.
2. Definir la forma de presentación de un mensaje, mediante la asignación de efectos a las frases que lo conforman.
3. Permite visualizar la hora, la fecha o las dos opciones al mismo tiempo, cada vez que un mensaje ha sido mostrado totalmente, después de un número de ciclos que pueden ser elegidos mediante un menú.
4. Se puede seleccionar el nivel de brillo de los led's, para la presentación de brillo de los mensajes entre cinco niveles diferentes.

También mientras un mensaje es visualizado en el rótulo es posible realizar:

1. La selección entre cinco diferentes tipos de velocidades de presentación.
2. Elegir el tipo de letra para presentar un mensaje.

2.2.1.1.3.1. Numero de Mensajes

En el rótulo se visualiza un mensaje a la vez y la cantidad de caracteres que puede poseer depende del número de mensajes totales que podrá presentar, mientras menor número de mensajes pueda presentar mayor es la capacidad de caracteres que puede tener el mensaje. Cada mensaje ha sido dividido en líneas de trabajo, en las que cada una está constituida por 16 caracteres.

La tabla 2.2 presenta la cantidad de mensajes que es posible definir con su respectivo número de líneas y caracteres.

CANTIDAD DE MENSAJES	NUMERO DE LINEAS	CARACTERES X MENSAJE	CANT. CARACTERES EN TOTAL
25	56	896	22400
50	28	448	22400
99	14	224	22176

Tabla 2.2: Número de mensajes

La línea de un mensaje posee una longitud de 16 caracteres, para facilitar la lectura y la presentación de su contenido en el LCD, debido a que en el LCD solo es posible presentar un máximo de 16 caracteres por línea, la figura 2.4 indica la forma de presentación de una línea de mensaje y la respectiva información que indica el número de línea visualizada del mensaje en el LCD.

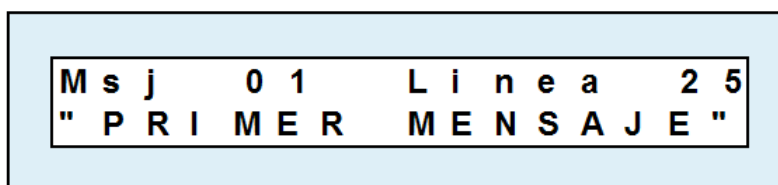


Figura 2.4: Presentación de un Mensaje

2.2.1.1.3.2. Características de Presentación de Mensajes y Líneas

La tabla 2.3 presenta las características que pueden ser asignadas a un mensaje y a sus respectivas líneas para su presentación en el rótulo.

M E N S A J E			L I N E A		
LETRA	1	NORMAL	E F E C T O S	1	TEXTO CORRIDO
	2	NEGRILLA		2	AVANZA DETIENE
	3	PEQUEÑO		3	AVANZA FLASH
	4	DOBLE		4	PRESENTA TEXTO
VELOCIDAD	POSEE 5 VALORES DE SELECCION			5	DESTELLANTE
				6	ABRIR TEXTO 1
				7	ABRIR TEXTO 2
				8	ABRIR TEXTO 3
				9	ABRIR TEXTO 4
BRILLO	POSEE 5 VALORES DE SELECCION			10	ASCENDER TEXTO
				11	DESCENDER TEXTO
				12	DELETREANDO
				13	ABRIENDO LETRAS

Tabla 2.3: Características de presentación

Los mensajes pueden ser configurados para ser presentados con distinto tipo de letra, brillo y con velocidades diferentes, en donde la velocidad determina el tiempo que será presentado un mensaje en el rótulo o la velocidad de recorrido del mismo.

Todas las líneas de cada mensaje son presentadas con igual tipo de letra, brillo y velocidad asignados en el mensaje, se diferencian entre cada una por la manera de presentación de su contenido, el cual depende del tipo de efecto que posea la línea.

También en el rótulo es posible presentar la hora y la fecha si se desea realizarlo, esta información es visualizada con las características que posea el mensaje presentado.

2.2.1.2. Integrado 24LC256

La memoria 24LC256 almacena información de los códigos de cada tipo de letra, los mensajes que serán presentados en el rótulo con sus respectivas características e información modificada en las configuraciones iniciales.

En la tabla 2.4 se presentan los datos de la memoria organizados en un mapa de memoria.

DIRECCION	DATOS ALMACENADOS	CANTIDAD BYTES
0	Inicio	2200
	Código de Letras (Normal, Negrita, Pequeña, Doble)	
2199	Fin	
2200	Inicio	800
	Código para otros tipos de textos	
2989	Fin	
2990	Constante Publicidad	1
2991	constante Cantidad de mensaje	1
2992	mensaje memoria	1
3000	Inicio	48
	Texto de Publicidad sin teléfono	
3047	Fin	
3048	Inicio	54
	Texto de Publicidad con teléfono	
3103	Fin	
3500	Inicio	100
	Códigos de Tipos de Letras de cada Mensaje	
3599	Fin	

Tabla 2.4: Mapa de memoria

DIRECCION	DATOS ALMACENADOS	CANTIDAD BYTES
3600	Inicio	100
	Códigos de Velocidad del Texto de cada Mensaje	
3699	Fin	
3700	Inicio	1400
	Códigos de Efectos de cada Línea de Texto	
5099	Fin	
5100	Inicio	100
	Códigos de Hora por cada Mensaje de Texto	
5199	Fin	
5200	Inicio	100
	Códigos de Fecha por cada Mensaje de Texto	
5299	Fin	
5300	Inicio	100
	Códigos de número de ciclos	
5399	Fin	
5400	Inicio	100
	Códigos de lectura tipo de brillo	
5499	Fin	
9200	Inicio	22800
	Texto a Presentar en Rótulo	
32767	Fin	

Tabla 2.4: Mapa de memoria

La información que se almacena en la memoria se encuentra organizada para facilitar la realización de cálculos que permitan leer o escribir distintos datos de la misma.

2.2.1.2.1. Lectura y Escritura de un dato en la memoria 24LC256

La lectura y escritura de un dato desde el microcontrolador a la memoria se realiza utilizando las subrutinas declaradas a continuación, indicando en primer lugar la inicialización de sus variables y su forma de direccionamiento.

Inicialización de valores

```
Const Cod_esc = 160 : Const Cod_lec = 161' Valores de escritura y lectura
Dim Direccion As Long At $100 , Direccion_h As Byte At $101 Overlay ,
Direccion_l As Byte At $100 Overlay
Dim Valor As Byte
Declare Sub Write_eeprom(byval Adr As Byte , Byval Value As Byte)
Declare Sub Read_eeprom(byval Adr As Byte , Value As Byte)
```

Un dato en la memoria puede estar entre la dirección 0 o en la 32767, por lo tanto se requiere de dos bytes para guardar el valor de la dirección.

Para seleccionar una localidad en la memoria primero se envía el byte más significativo y luego el de menor valor. Así que se declaró la variable **Direccion** como **long** que en el programa guarda el valor de la localidad a leer o escribir en la memoria y esta variable es dividida en sus bytes mediante la declaración de las variables **Direccion_h** y **Direccion_l** en las direcciones correspondientes al byte de mayor y de menor valor.

En la tabla 2.5 se visualiza como se dividen estos datos asumiendo que la variable **Direccion** tiene cargado el valor 27B5.

Direccion	
27B5	
Direccion_h	Direccion_l
27	B5

Tabla 2.5: División de variable para direccionamiento

Lectura de un byte

Sub Read_eeprom(**byval** Adr **As Byte** , Value **As Byte**)

```

I2cstart                                ' Genera el código de inicio
I2cwbyte Cod_esc                         ' Envío de dirección del dispositivo
I2cwbyte Direccion_h                     ' Dirección de la localidad de la memoria
I2cwbyte Direccion_l                     ' Dirección de la localidad de la memoria
I2cstart                                ' Genera código de inicio
I2cwbyte Cod_lec                         ' Cambio a modo de lectura
I2crbyte Valor , Nack                   ' Lectura de un dato
I2cstop

```

End Sub

Escritura de un byte

Sub Write_eeprom(**byval** Adr **As Byte** , **Byval** Value **As Byte**)

```

I2cstart                                ' Genera código de inicio
I2cwbyte Cod_esc                         ' Envío de dirección del dispositivo
I2cwbyte Direccion_h                     ' Dirección de la localidad de la memoria
I2cwbyte Direccion_l                     ' Dirección de la localidad de la memoria
I2cwbyte Valor                           ' Escritura de un dato
I2cstop

```

End Sub

En los siguientes ejemplos se indica la forma de envío de un dato hacia las subrutinas Write_eeprom y Read_eeprom para que un dato sea leído o escrito.

Escritura de un dato

```

Direccion=3700                            ' Dirección a ser escrita
Tipo_efecto=5                             ' Dato a escribir
Valor = Tipo_efecto                       ' Dato a ser escrito
Call Write_eeprom(1 , Valor)             ' Envío a la subrutina de escritura

```


Lectura de un dato

```

Direccion=3700           ' Dirección a ser leída
Call Read_eeprom(1 , Valor) ' Envió a la subrutina de lectura
Tipo_efecto = Valor     ' Dato leído

```

2.2.1.3. Integrado De Reloj DS1307

El integrado de reloj posee los datos de la hora y la fecha que son mostrados en la matriz de led's. La lectura y escritura de los datos se realizan utilizando tres librerías creadas para controlar el integrado, estas librerías se encuentran en el manual de Bascom Avr, las librerías utilizadas permiten realizar lo siguiente:

- Lectura de hora y fecha
- Escritura de hora
- Escritura de fecha

Inicialización de valores para las librerías

```

Config Scl = Portb.x : Config Sda = Portb.x
$lib "ds1307clock.lib"
Config Date = Mdy , Separator = / : Config Clock = User
Const Ds1307w = &HD0 : Const Ds1307r = &HD1
Dim _weekday As Byte , Hora As String * 8 , Fecha As String * 8

```

Librería para escritura de hora

```

Settime:           ' Permite escribir el valor de la hora
  _sec = Makebcd(_sec) : _min = Makebcd(_min) : _hour = Makebcd(_hour)
I2cstart         ' Genera el código de inicio
I2cwbyte Ds1307w ' Envía la dirección de escritura
I2cwbyte 0       ' Dirección de inicio de escritura
I2cwbyte _sec    ' Envía el dato del segundo

```

I2cwrite _min ' Envía el dato del minuto
I2cwrite _hour ' Envía el dato de la hora
I2cstop

Return

Librería para escritura de fecha

Setdate:

_day = **Makebcd**(_day) : _month = **Makebcd**(_month) : _year = **Makebcd**(
_year)
' Convierte de Decimal a Bcd
I2cstart ' Genera el código de inicio
I2cwrite Ds1307w ' Envía la dirección de escritura
I2cwrite 4 ' Dirección de inicio de escritura
I2cwrite _day ' Envía el dato del día
I2cwrite _month ' Envía el dato del mes
I2cwrite _year ' Envía el dato del año
I2cstop

Return

Librería para lectura de hora y fecha

Getdatetime:

I2cstart ' Genera el código de inicio
I2cwrite Ds1307w ' Envío de dirección de escritura
I2cwrite 0 ' Dirección de inicio de lectura
I2cstart ' Genera código de inicio
I2cwrite Ds1307r ' Cambio a modo de lectura
I2cwrite _sec , **Ack** ' Lectura de segundo
I2cwrite _min , **Ack** ' Lectura de minutos
I2cwrite _hour , **Ack** ' Lectura de hora
I2cwrite Weekday , **Ack** ' Lectura de los días de la semana
I2cwrite _day , **Ack** ' Numero de día
I2cwrite _month , **Ack** ' Numero de mes

```

I2crbyte _year , Nack           ' Año
I2cstop
_sec = Makedec(_sec) : _min = Makedec(_min) : _hour = Makedec(_hour)
_day = Makedec(_day) : _month = Makedec(_month) : _year = Makedec(
_year)           ' Convierte de Bcd a Decimal
Hora = Time$ : Fecha = Date$   ' Guarda los valores de hora y fecha

```

Return

2.2.1.4. Cerebro de Control

En el cerebro de control se encuentra el microcontrolador ATmega 644, el cual lee y escribe en la memoria 24LC256 y en el integrado de reloj, también controla al teclado, visualiza diferente tipo de información en el LCD, controla a una chicharra y a un led.

El ATmega 644 envía información hacia el rótulo utilizando los pines declarados como Stb, Clk, Dat, Stb_fila, Clk_fila, Dat_fila que controlan las columnas y filas del rótulo respectivamente.

En la figura 2.5 se presenta el diagrama esquemático de las conexiones de los diferentes integrados que conforman el cerebro de control, aquí es posible observar que la salida de datos hacia el rótulo primeramente pasa a través del amplificador 74LS244, esto es debido a que el microcontrolador es muy limitado en cuanto a su capacidad de corriente, el 74LS244 amplifica la señal enviada al rótulo y además provee protección hacia los pines de entrada.

En este diagrama también se observa que en la salida hacia el rótulo están presentes las conexiones para Vcc y Gnd esto es debido a que el voltaje del cerebro de control es proveído a través del rótulo.

2.2.2. INGRESO Y VERIFICACION DE INFORMACION

El presente bloque del sistema electrónico publicitario es la interfaz utilizada por el usuario para escribir o modificar el contenido de un mensaje y sus características. Los componentes presentes en este bloque son el teclado y el LCD.

Mediante el teclado es posible elegir un determinado menú para modificar los valores de configuración, escribir o modificar el contenido y las características de un mensaje. Por otra parte el LCD presenta diferentes mensajes que informan el menú que ha seleccionado una persona y la acción que se encuentra realizando.

2.2.2.1. Tarjeta de Teclado

Para el ingreso a los menús se utiliza un teclado matricial que posee 4 filas y 4 columnas, para esto se observó los requerimientos de la Corporación Electrónica TELEVID y se tuvo el siguiente modelo, presentado en la figura 2.8, el cual posee la distribución de pistas indicado en la figura 2.9²⁸.



Figura 2.8: Teclado

²⁸ Ver Anexo A

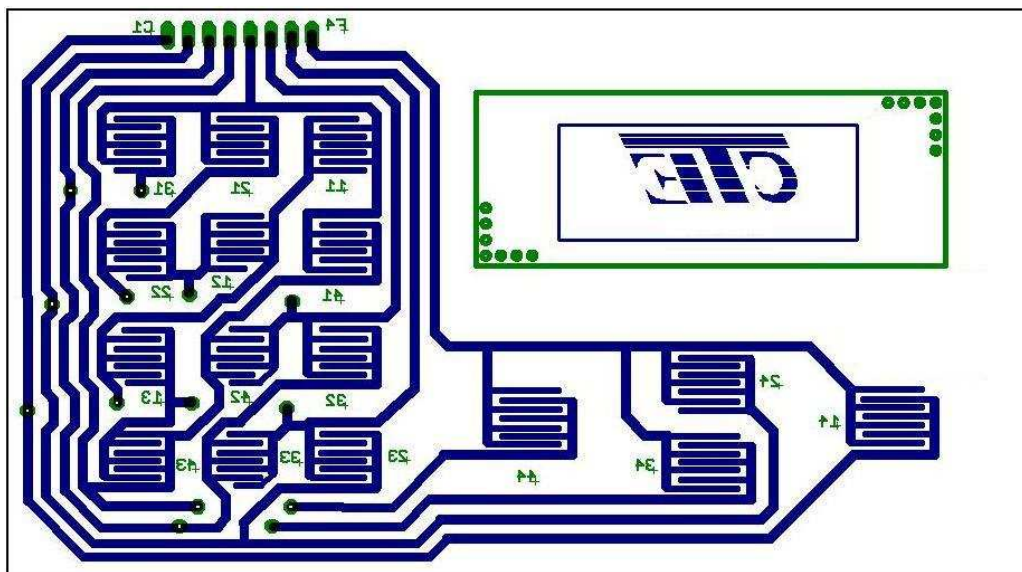


Figura 2.9: Ruteado del Teclado

2.2.2.1.1. Descripción de los Componentes del Teclado

El teclado del sistema publicitario tiene la distribución de teclas presentado en la tabla 2.6.

	COLUMNA 1	COLUMNA 2	COLUMNA 3	COLUMNA 4
FILA 1	Tecla 1	Tecla 2	Tecla 3	Tecla 4
FILA 2	Tecla 5	Tecla 6	Tecla 7	Tecla 8
FILA 3	Tecla 9	Tecla 10	Tecla 11	Tecla 12
FILA 4	Tecla 13	Tecla 14	Tecla 15	Tecla 16

Tabla 2.6: Distribución de teclas de la tarjeta de teclado.

La distribución de las teclas presentadas en la tabla 2.6, están organizadas en el teclado como lo indica la figura 2.10.

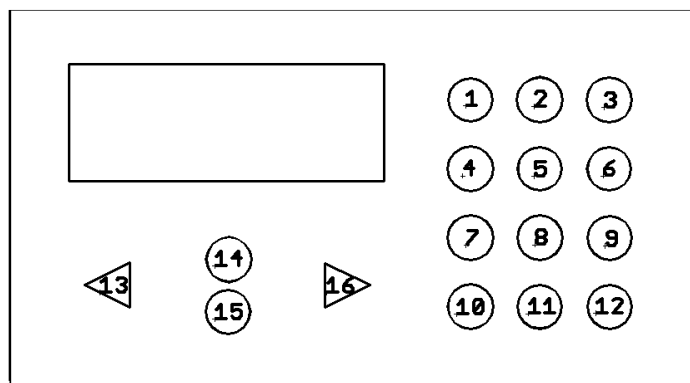


Figura 2.10: Enumeración de Teclas

Las teclas más utilizadas en la selección y modificación de datos en los menús, son presentadas en la siguiente tabla, con el respectivo nombre de cada tecla.

TECLAS	NOMBRE
14	MENU
15	ENTER
13	IZQUIERDA
16	DERECHA
10	SHIFT
12	TEXTO

Tabla 2.7: Teclas de acceso frecuente

La descripción de pines del microcontrolador que se conectan al teclado se presenta en la tabla 2.8.

CONEXION	PIN	CONEXION	PIN
Columna 1	Pin # 38 (PA2)	Fila 1	Pin # 36 (PA4)
Columna 2	Pin # 39 (PA1)	Fila 2	Pin # 33 (PA7)
Columna 3	Pin # 37 (PA3)	Fila 3	Pin # 35 (PA5)
Columna 4	Pin # 40 (PA0)	Fila 4	Pin # 34 (PA6)

Tabla 2.8: Pines del microcontrolador para uso del teclado.

2.2.2.2. **Habilitación de Teclas de Menús y Visualización de Mensajes**

Cuando se selecciona un menú, se habilitan o deshabilitan en el teclado diferentes teclas, en cambio otras realizan diferentes funciones, por esta razón es importante conocer las teclas habilitadas según el tipo de menú elegido, cual es la función que cada una realiza y conocer los mensajes presentes en el LCD, que informan los cambios que el usuario se encuentra realizando.

Los menús que posee el programa, se encuentran en dos secciones, la primera permite realizar configuraciones iniciales de trabajo y la segunda permite modificar los mensajes.

2.2.2.2.1. *Primera Sección*

El acceso a esta sección del programa se realiza, al momento de iniciar el trabajo del microcontrolador y cuando el programa detecta la conexión de un jumper a tierra. La tabla 2.9 indica las funciones que realiza el microcontrolador dependiendo del jumper conectado.

JUMPER	FUNCION
1	Ingreso de Fecha de Fabricación
	Ingreso del Número de Serie
	Ingreso de Fecha de Revisión 1
	Ingreso de Fecha de Revisión 2
	Ingreso de Número de Rutas
	Selección de Publicidad
2	Test de Teclado
3	Test de Tarjetas

Tabla 2.9: Configuraciones iniciales

2.2.2.2.2. Segunda Sección

Esta sección permite modificar el contenido de los mensajes y sus características mediante la selección de un menú definido.

El programa habilita varios submenús dependiendo del menú seleccionado para facilitar la modificación de un mensaje en su contenido o característica, informando mediante el envío de mensajes al LCD, los datos modificados y el menú en el cual se encuentra. Los menús que posee esta sección son los siguientes:

- Menú principal
- Menú para la edición de mensajes
- Menú efectos
- Menú insertar reloj
- Menú nivel de brillo

Cada menú posee varios mensajes que son presentados en el LCD, los cuales son actualizados dependiendo del dato modificado y la tecla presionada.

2.2.2.3. Menús de Trabajo

Son los menús que el programa posee para permitir la modificación de diferentes datos, en donde cada menú presenta en el LCD diversos mensajes creados para interactuar con el usuario indicando la función que realiza.

2.2.2.3.1. Menú de Configuraciones Iniciales

En la tabla 2.10 se indican los menús de esta sección de programa, con las respectivas teclas habilitadas por menú y los gráficos que representan los mensajes que el LCD presenta dependiendo del menú seleccionado.

Configuraciones iniciales	Ingreso de Fecha de Fabricación		
	Tecla	Función	Figura
	1	Selecciona el día	2.11
	2	Selecciona el mes	
	3	Selecciona el año	
	Ingreso del Número de Serie		
	Tecla	Función	Figura
	1	Selecciona el valor de la centena	2.12
	2	Selecciona el valor de la decena	
	3	Selecciona el valor de la unidad	
	Ingreso de Fecha de Revisión 1		
	1	Selecciona el valor de la centena	2.13
	2	Selecciona el valor de la decena	
	3	Selecciona el valor de la unidad	
	Ingreso de Fecha de Revisión 2		
	1	Selecciona el valor de la centena	2.14
	2	Selecciona el valor de la decena	
	3	Selecciona el valor de la unidad	
Ingreso de Número de Rutas			
10	Selecciona el número de cantidad de mensajes	2.15	
Selección de Publicidad			
12	Asigna la presentación o no del teléfono	2.16	

Tabla 2.10: Configuraciones Iniciales

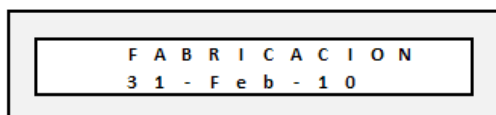


Figura 2.11: Fecha de Fabricación

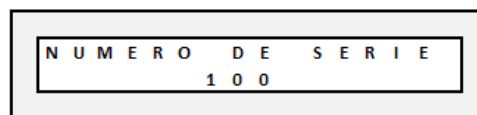


Figura 2.12: Número de Serie

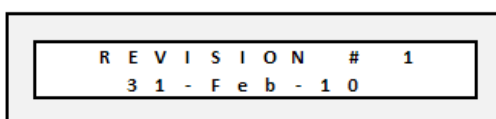


Figura 2.13: Revisión 1

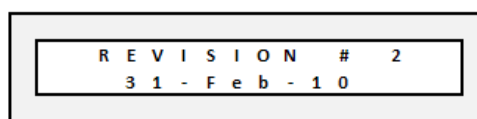


Figura 2.14: Revisión 2

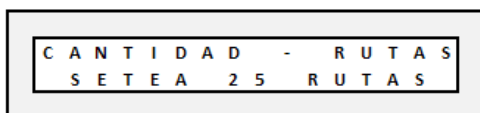


Figura 2.15: Numero de rutas

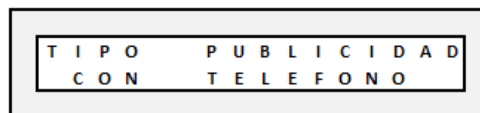


Figura 2.16: Publicidad

2.2.2.3.2. Menús para Configuraciones de Mensajes

Esta sección posee el menú principal que presenta el mensaje en el rótulo mientras verifica si las teclas de su menú han sido presionadas para habilitar los menús que permiten cambiar las características del mensaje.

2.2.2.3.3. Menú Principal

En la tabla 2.11 se presentan las respectivas teclas habilitadas de este menú y los gráficos que representan los mensajes en el LCD cuando se selecciona la tecla. En los gráficos la palabra "FRASE", representa al texto de una línea.

MENU PRINCIPAL		
Tecla	Función	Figura
10	Selecciona la velocidad de presentación de un mensaje	2.17
12	Selecciona el texto de presentación de un mensaje	2.18
13	Retrocede entre mensajes	2.19
14	(Tecla Enter)Selecciona diferentes menús del programa.	
15	(Tecla menú)Acepta el ingreso a un menú	
16	Avanza entre mensajes	

Tabla 2.11: Menú Principal

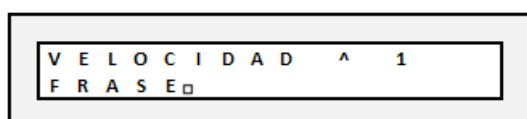


Figura 2.17: Fecha de Fabricación

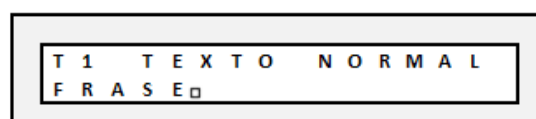


Figura 2.18: Número de Serie

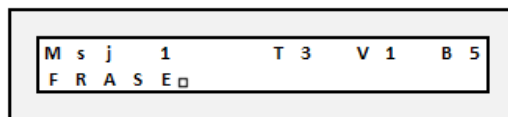


Figura 2.19: Frase en una Línea

2.2.2.3.4. Edición de Mensajes

Este menú es seleccionado en el menú principal mediante la tecla 14 (tecla menú), al observar en el LCD, el mensaje de la figura 2.20, para posteriormente seleccionar la línea a escribir con las teclas presentadas en la tabla 2.21.

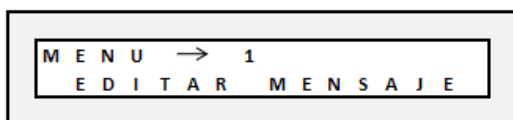


Figura 2.20: Mensaje Menú Edición

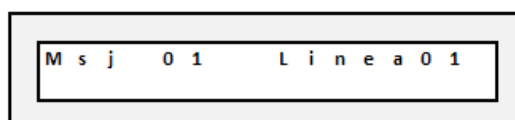


Figura 2.21: Selección Línea

MENU EDICION		
Tecla	Función	Figura
13	Permite seleccionar la línea a ser modificada	2.21
16		
15	Permite el acceso al menú de edición	

Tabla 2.12: Menú Edición

La tabla 2.13 presenta las respectivas teclas que permiten escribir o modificar un mensaje.

EDICION DE MENSAJES		
Tecla	Función	Figura
1	Selecciona los caracteres + - * / = 1	2.8
2	Selecciona los caracteres ABC2.:	2.8
3	Selecciona los caracteres DEF3,;	2.8
4	Selecciona los caracteres GHI4()	2.8

Tabla 2.13: Edición de Mensajes

EDICION DE MENSAJES		
Tecla	Función	Figura
5	Selecciona los caracteres JKL5[]	2.8
6	Selecciona los caracteres MNOÑ6<	2.8
7	Selecciona los caracteres PQRS7>	2.8
8	Selecciona los caracteres TUV8?!	2.8
9	Selecciona los caracteres WXYZ9´	2.8
10	Selecciona entre Mayúscula o Minúscula	2.8
11	Selecciona los caracteres _0@#\$%	2.8
12	Selecciona los caracteres ¡ ß visualizados como □ □	2.19
13	Avanza o retrocede para selección de una posición	2.22
16	en la línea editada.	
15	Guarda cambios y sale del menú	2.23

Tabla 2.13: Edición de Mensajes

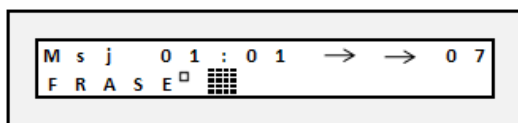


Figura 2.22: Posición a Editar

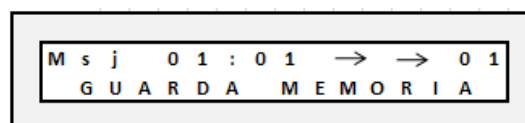


Figura 2.23: Acción Realizada

En este menú la tecla 12 escribe los caracteres que son utilizados para determinar el final de una línea (□) y el final del mensaje (□).

El terminador de línea es utilizado para calcular la cantidad de caracteres que posee una línea, que es un dato que permite determinar si una frase puede o no ser vista de manera completa en el rótulo, dependiendo de la cantidad de caracteres que pueden ser presentados por tipo de letra (Tabla 2.22) estableciendo si una línea puede o no poseer un determinado efecto.

El terminador final es utilizado para determinar el número de líneas y la cantidad de caracteres que posee el mensaje a ser presentado en el rótulo, estos datos son utilizados para determinar las posiciones de lectura de cada línea del mensaje.

Cuando no es encontrado estos caracteres en el mensaje, el terminador final es escrito en la memoria automáticamente el instante que el programa lee el número de caracteres del mensaje, y el terminador de línea es escrito al editar una línea.

2.2.2.3.5. Menú Efectos

El menú es seleccionado utilizando la tecla menú, cuando se observa en el LCD el mensaje de la figura 2.24, para posteriormente seleccionar la línea a escribir con las teclas presentadas en la tabla 2.14.

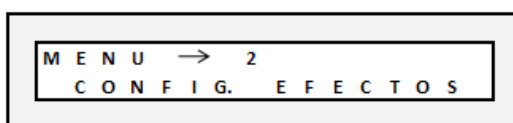


Figura 2.24: Mensaje Menú Efectos

MENU EFECTOS		
Tecla	Función	Figura
13	Permite seleccionar la línea a ser modificada	2.21
16		
15	Después de seleccionado la línea el menú permite acceder a la modificación de la misma	2.26

Tabla 2.14: Menú de Selección de Efectos

La tabla 2.15 presenta las teclas que permiten seleccionar una línea de mensaje y modificar un efecto.

EDICION DE EFECTOS		
Tecla	Función	Figura
13	Permite seleccionar la línea a ser modificada	
16	Permite seleccionar la línea a ser modificada	
15	Selección de efectos	2.25, 2.26
14	Salir del menú efectos	2.27

Tabla 2.15: Edición de Efectos

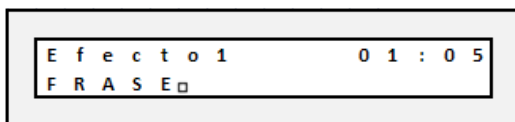
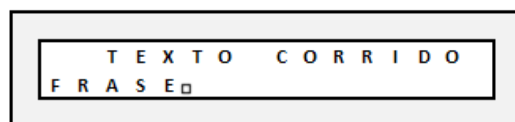


Figura 2.25: Información de línea

Figura 2.26: Efecto 1²⁹

Los mensajes de las figuras 2.25 y 2.26 son mostrados secuencialmente.

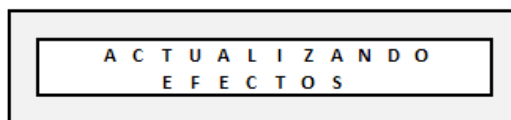


Figura 2.27: Salir de menú efectos

La figura 2.28 indica el mensaje que es mostrado en el LCD, cuando una línea no puede tener un determinado efecto debido a la cantidad de caracteres que pueden ser presentados en el rótulo. (Tabla 2.22)

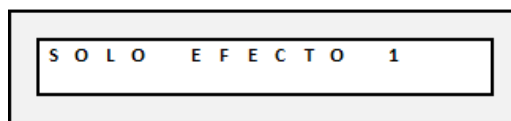


Figura 2.28: Mensaje

2.2.2.3.6. *Menú Insertar Reloj*

Este menú se selecciona en el menú principal mediante la tecla 14 (Tecla Menú), cuando se observa el mensaje de la figura 2.29.

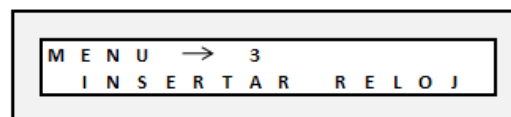


Figura 2.29: Mensaje Menú Reloj

En la tabla 2.16 se indican las respectivas teclas que permiten escribir o modificar el horario de un mensaje con los submenús que posee y los mensajes presentados en el LCD.

²⁹ Ver Anexo 2

Insertar Reloj	Insertar Hora – Fecha		
	Tecla	Función	Figura
	13	Elige si la hora es o no insertada en el mensaje	2.30
	16	Elige si la fecha es o no insertada en el mensaje	
	15	Sale del submenú y continúa al siguiente submenú	
	Seleccionar Número de Ciclos		
	Tecla	Función	Figura
	13	Decrementa el número de Ciclos	2.31
	16	Incrementa el número de Ciclos	
	15	Sale del submenú y continúa al siguiente submenú	
	Igualar Hora		
	13	Iguala hora	2.32
	16	Iguala minutos	
	15	Sale del menú si no se desea insertar la Fecha caso contrario continúa con el submenú de fecha.	
	Igualar Fecha		
	13	Iguala día	2.33
	16	Iguala mes	
	10	Iguala año	
	15	Sale del menú de horario	2.34

Tabla 2.16: Insertar Hora - Fecha

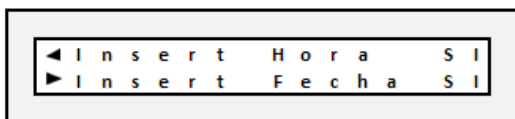


Figura 2.30: Submenú Insertar Reloj

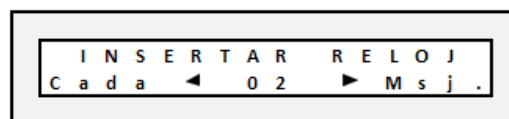


Figura 2.31: Submenú Insertar Ciclos

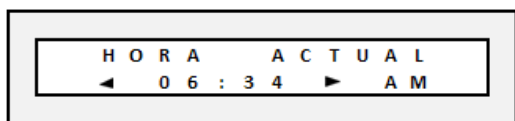


Figura 2.32: Submenú Insertar Hora

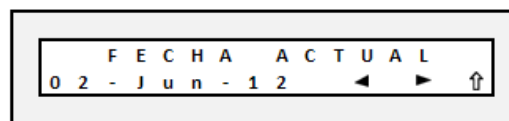


Figura 2.33: Submenú Insertar Fecha

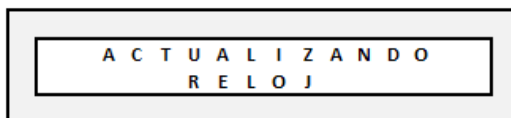


Figura 2.34: Salir del menú horario

Este menú presenta cada submenú en orden según la opción elegida, por ejemplo si en el submenú (Insertar Hora - Fecha), se eligió que no se presente la hora y la fecha, el programa finaliza el menú para posteriormente mostrar los cambios realizados en el rótulo, o si es seleccionado solo la hora o la fecha el programa presenta solo el menú de la hora o de la fecha.

En el submenú (Seleccionar Número de Ciclos) se elige el número de veces que un mensaje tiene que ser presentado para luego mostrar la hora o la fecha.

2.2.2.3.7. *Menú Nivel de Brillo*

Este menú es seleccionado en el menú principal, cuando se observa el mensaje de la figura 2.35

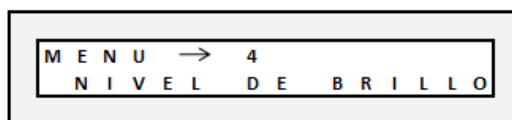


Figura 2.35: Mensaje Menú Brillo

La tabla 2.17 indica las respectivas teclas que permiten escribir o modificar el nivel de brillo de un mensaje.

MENU NIVEL DE BRILLO		
Tecla	Función	Figura
13	Disminuye el brillo de los led's	2.36
16	Aumenta el brillo de los led's	
15	Sale del menú	2.37

Tabla 2.17: Menú Nivel de Brillo

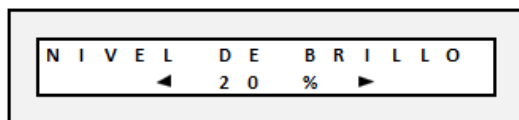


Figura 2.36: Menú nivel de brillo

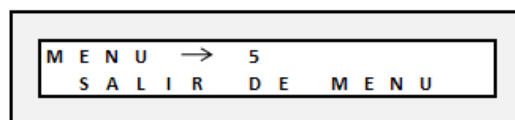


Figura 2.37: Saliendo del menú

2.2.3. VISUALIZACION DE DATOS

Presenta la manera que una línea de mensaje es visualizada en el rótulo, la línea escrita es mostrada en dos tarjetas de led's que conforman una matriz de 48 columnas por 7 filas, también se indica los datos que pueden ser visualizados y el respectivo tamaño de cada dato según su tipo de letra.

2.2.3.1. Tipos de Letra

Existen cuatro tipos de letra para poder presentar un mensaje. Cada tipo de letra está constituido por un conjunto de 88 caracteres y cada uno tiene su propio código y un tamaño definido para su presentación. La tabla 2.18 presenta el número de columnas que posee el código de los caracteres según el tipo de letra.

Tipo de letra	Numero de columnas
1 - Normal	5
2 - Negrilla	6
3 - Pequeña	6
4 - Doble	8

Tabla 2.18: Numero de columnas por carácter

El código de cada carácter es creado en Excel para posteriormente ser grabado en la memoria 24LC256. En la tabla 2.19 se presentan los gráficos de cada tipo de letra con su respectivo código, para el carácter "5", indicando además que todos los caracteres correspondientes a cada tipo de letra fueron creados en Excel y su código fue grabado en la memoria desarrollando un programa en

Bascom que escribe los códigos en la memoria ordenadamente, en las direcciones correspondientes al tipo de código.

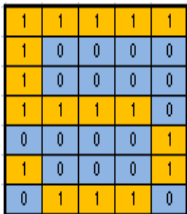
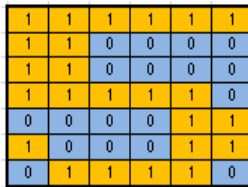
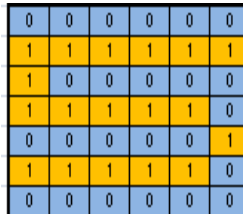
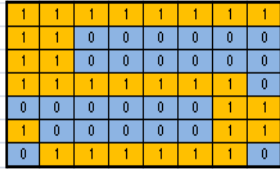
TIPO DE LETRA	CARACTER	CODIGO
NORMAL	 <pre> 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 </pre>	Data=&B01111010 Data=&B01001001 Data=&B01001001 Data=&B01001001 Data=&B01001001 Data=&B01000110
NEGRILLA	 <pre> 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0 </pre>	Data=&B01111010 Data=&B01111001 Data=&B01001001 Data=&B01001001 Data=&B01001111 Data=&B01000110
PEQUEÑA	 <pre> 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 </pre>	Data=&B00111010 Data=&B00101010 Data=&B00101010 Data=&B00101010 Data=&B00101010 Data=&B00100100
DOBLE	 <pre> 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1 1 0 </pre>	Data=&B01111010 Data=&B01111001 Data=&B01001001 Data=&B01001001 Data=&B01001001 Data=&B01001001 Data=&B01001111 Data=&B01000110

Tabla 2.19: Tipos de letras y Códigos

En la tabla 2.20 se presenta la lista de los 88 caracteres que posee su propio código y las respectivas direcciones de la memoria donde inicia cada código de carácter guardado.

POSICION	CARÁCTER	NORMAL	NEGRILLA	PEQUEÑA	DOBLE
1	A	0	440	968	1496
2	B	5	446	974	1504
3	C	10	452	980	1512
4	D	15	458	986	1520
5	E	20	464	992	1528
6	F	25	470	998	1536
7	G	30	476	1004	1544
8	H	35	482	1010	1552
9	I	40	488	1016	1560
10	J	45	494	1022	1568
11	K	50	500	1028	1576
12	L	55	506	1034	1584
13	M	60	512	1040	1592
14	N	65	518	1046	1600
15	O	70	524	1052	1608
16	P	75	530	1058	1616
17	Q	80	536	1064	1624
18	R	85	542	1070	1632
19	S	90	548	1076	1640
20	T	95	554	1082	1648
21	U	100	560	1088	1656
22	V	105	566	1094	1664
23	W	110	572	1100	1672
24	X	115	578	1106	1680
25	Y	120	584	1112	1688

Tabla 2.20: Tipos de caracteres y direcciones

POSICION	CARÁCTER	NORMAL	NEGRILLA	PEQUEÑA	DOBLE
26	z	125	590	1118	1696
27		130	596	1124	1704
28	Ñ	135	602	1130	1712
29	a	140	608	1136	1720
30	b	145	614	1142	1728
31	c	150	620	1148	1736
32	d	155	626	1154	1744
33	e	160	632	1160	1752
34	f	165	638	1166	1760
35	g	170	644	1172	1768
36	h	175	650	1178	1776
37	i	180	656	1184	1784
38	j	185	662	1190	1792
39	k	190	668	1196	1800
40	l	195	674	1202	1808
41	m	200	680	1208	1816
42	n	205	686	1214	1824
43	o	210	692	1220	1832
44	p	215	698	1226	1840
45	q	220	704	1232	1848
46	r	225	710	1238	1856
47	s	230	716	1244	1864
48	t	235	722	1250	1872
49	u	240	728	1256	1880
50	v	245	734	1262	1888
51	w	250	740	1268	1896
52	x	255	746	1274	1904
53	y	260	752	1280	1912
54	z	265	758	1286	1920
55	ñ	270	764	1292	1928

Tabla 2.20: Tipos de caracteres y direcciones

POSICION	CARÁCTER	NORMAL	NEGRILLA	PEQUEÑA	DOBLE
56	0	275	770	1298	1936
57	1	280	776	1304	1944
58	2	285	782	1310	1952
59	3	290	788	1316	1960
60	4	295	794	1322	1968
61	5	300	800	1328	1976
62	6	305	806	1334	1984
63	7	310	812	1340	1992
64	8	315	818	1346	2000
65	9	320	824	1352	2008
66	+	325	830	1358	2016
67	-	330	836	1364	2024
68	*	335	842	1370	2032
69	/	340	848	1376	2040
70	=	345	854	1382	2048
71	.	350	860	1388	2056
72	:	355	866	1394	2064
73	,	360	872	1400	2072
74	;	365	878	1406	2080
75	@	370	884	1412	2088
76	#	375	890	1418	2096
77	\$	380	896	1424	2104
78	%	385	902	1430	2112
79	&	390	908	1436	2120
80	(395	914	1442	2128
81)	400	920	1448	2136
82	[405	926	1454	2144
83]	410	932	1460	2152
84	i	415	938	1466	2160
85	ß	420	944	1472	2168

Tabla 2.20: Tipos de caracteres y direcciones

POSICION	CARÁCTER	NORMAL	NEGRILLA	PEQUEÑA	DOBLE
86	>	425	950	1478	2176
87	<	430	956	1484	2184
88	'	435	962	1490	2192

Tabla 2.20: Tipos de caracteres y direcciones

2.2.3.2. Efectos de Presentación de Líneas

Un mensaje en el rótulo es presentado línea por línea y cada una puede ser visualizada de forma diferente según el tipo de efecto que posea, pudiendo tener cada línea uno de trece efectos.

2.2.3.2.1. Efecto 1 - Texto Corrido

Cuando una línea de un mensaje tiene este efecto, la línea es presentada en el rótulo, avanzando de derecha a izquierda, columna por columna, el efecto termina cuando el texto presentado a salido completamente fuera de la matriz de led's del rótulo.

En la figura 2.39 se puede observar el sentido de desplazamiento de la información de una línea al tener el efecto 1, en el ejemplo el desplazamiento es realizado en cinco pasos, pero el correcto desplazamiento se realiza columna por columna.

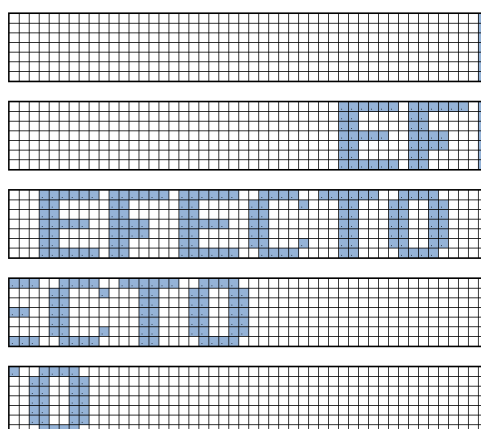


Figura 2.39: Efecto 1

2.2.3.2.2. Efecto 2 - Avanza Detiene

Este efecto es desarrollado utilizando el principio de funcionamiento del efecto 1. Cuando una línea de un mensaje es presentada en el rótulo, con este efecto, el texto avanza de derecha a izquierda, deteniéndose en el centro del rótulo un instante que depende de la velocidad que posea el mensaje, para luego continuar con su recorrido y salir completamente.

La figura 2.40 indica como es observado un texto con este efecto.

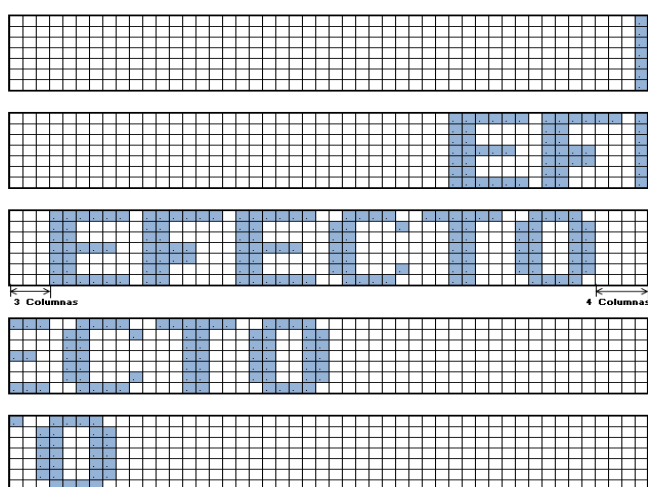


Figura 2.40: Efecto 2

El centrado de una frase en el rótulo no posee igual número de columnas de separación en sus extremos, cuando el número total de columnas de cada carácter de la línea, más los espacios es un número impar, en este caso siempre habrá entre las columnas de centrado la diferencia de una columna, como se indica en la figura 2.41.

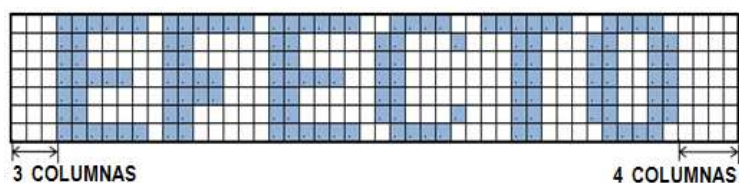


Figura 2.41: Espacios de centrado de texto

2.2.3.2.3. Efecto 3 - Avanza Flash

El efecto 3 es desarrollado utilizando el principio de funcionamiento del efecto 2. Cuando una línea de un mensaje tiene este efecto, el texto se desplaza de derecha a izquierda, se detiene en el centro, destella y luego sale completamente, la figura 2.42 indica como es observado un texto con este efecto.

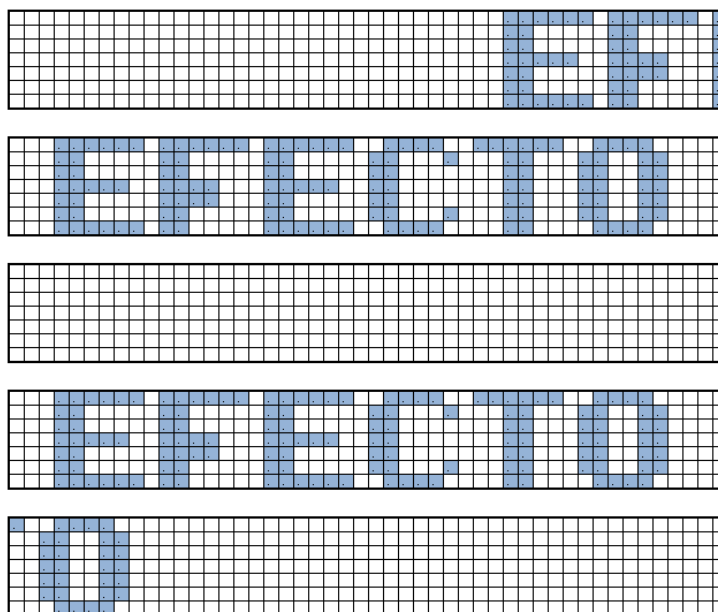


Figura 2.42: Efecto 3

La tabla 2.21 indica el número de destellos de la frase de una línea dependiendo de la velocidad que posea el mensaje.

Velocidad	Número de Destellos
1	3
2	4
3	5
4	6
5	7

Tabla 2.21: Número de destellos

2.2.3.2.4. Efecto 4 - Presenta Texto

El texto de una línea de un mensaje al tener este efecto es visualizado en el centro del rótulo, sin observarse el desplazamiento de la misma, en la figura 2.43 se visualiza la forma de presentación de una línea con este efecto.

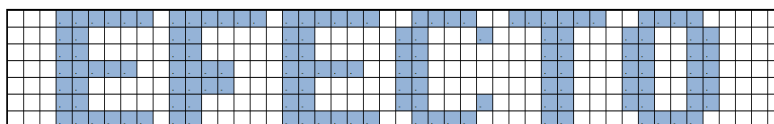


Figura 2.43: Efecto 4

El efecto 4 es desarrollado con el principio de funcionamiento del efecto 2, es decir el texto avanza de derecha a izquierda, se detiene en el centro y luego sale, con la diferencia de que los desplazamientos de la información se realizan dentro de los buffers (integrados 74LS595) a gran velocidad y su contenido es actualizado cuando la frase se encuentra centrada. El principio de desplazamiento de la información del efecto 4 es utilizado para la realización de los siguientes efectos³⁰:

- Efecto 5 – Destellante
- Efecto 6 - Abrir Texto 1
- Efecto 7 - Abrir Texto 2
- Efecto 8 - Abrir Texto 3
- Efecto 9 - Abrir Texto 4
- Efecto 10 - Ascender Texto
- Efecto 11 - Descender Texto
- Efecto 12 – Deletreando
- Efecto 13 - Abriendo Letra

Todos estos efectos son presentados en el centro del rótulo de diferentes maneras y su desarrollo tiene como dato principal, toda la string de salida de la frase fija que posee el efecto cuatro.

³⁰ Ver Anexo B

2.2.3.2.5. Efecto 5 – Destellante

El texto de una línea al poseer este efecto es presentado en el centro del rótulo de manera fija, para luego destellar un determinado número de veces que dependen de la velocidad del mensaje y luego continuar con la siguiente línea del mensaje, el número de destellos se puede observar en la tabla 2.21. La figura 2.44 indica como se presenta una línea con efecto 5.

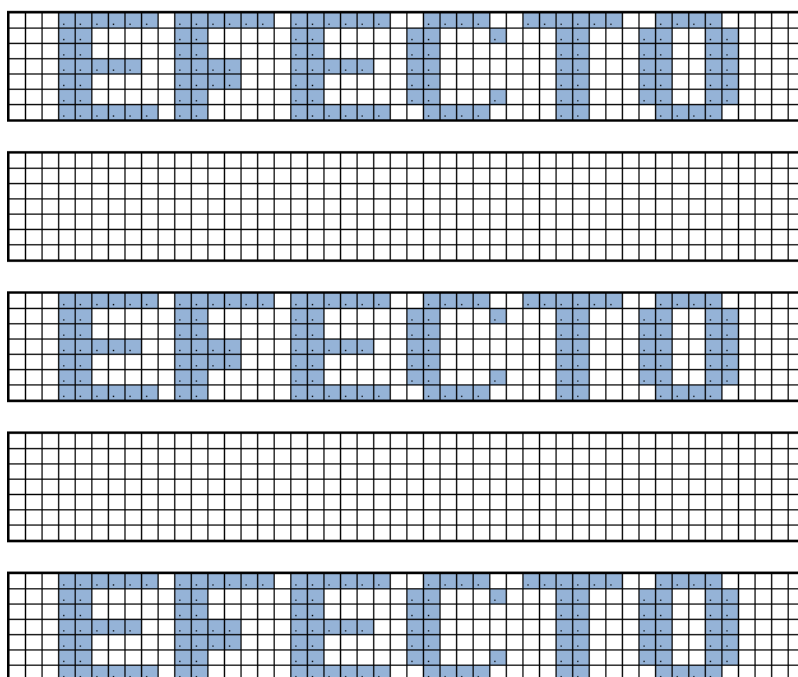


Figura 2.44: Efecto 5

2.2.3.2.6. Efecto 6 - Abrir Texto 1

La línea de un mensaje al poseer este efecto es presentado en el centro del rótulo, abriéndose verticalmente fila por fila, hacia sus lados superior e inferior, la figura 2.45 indica la forma de presentación de una línea con este efecto, en el ejemplo el desplazamiento se realiza en pocos pasos, pero el correcto desplazamiento se realiza fila por fila.

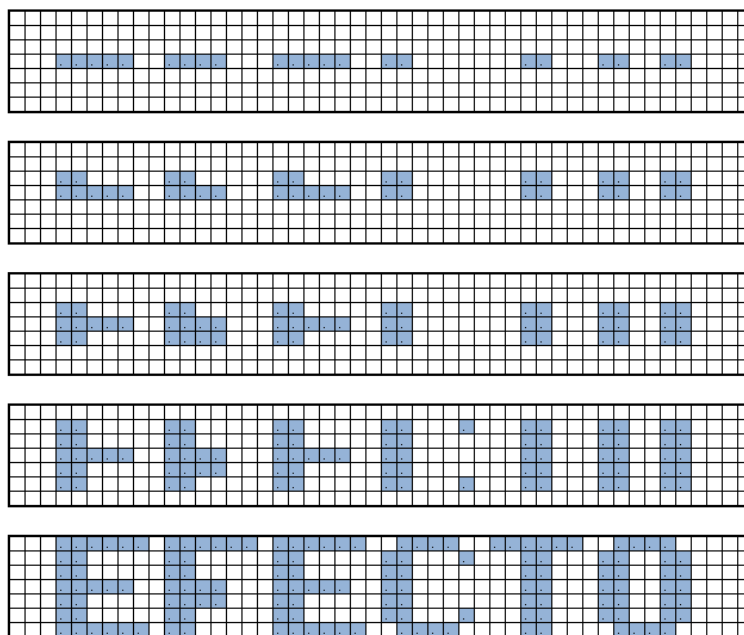


Figura 2.45: Efecto 6

2.2.3.2.7. *Efecto 7 - Abrir Texto 2*

La línea de un mensaje al poseer este efecto, es presentado en el centro del rótulo abriéndose fila por fila, desde la fila inferior hasta la superior, la figura 2.46 muestra la presentación de una línea con este efecto.

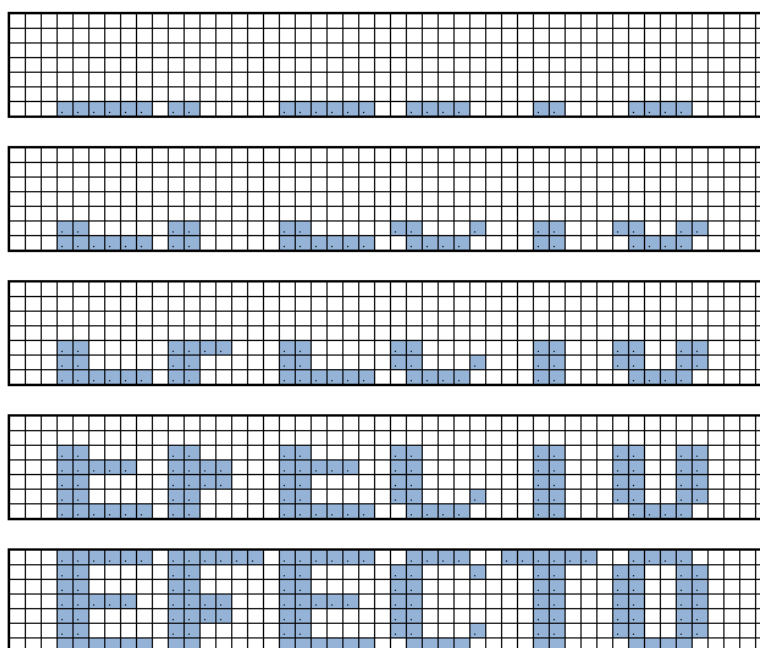


Figura 2.46: Efecto 7

2.2.3.2.8. Efecto 8 - Abrir Texto 3

El texto de una línea al poseer este efecto, es presentado en el centro del rótulo abriéndose fila por fila, desde la fila superior hasta la inferior, la figura 2.47 indica la presentación de una línea con este efecto.

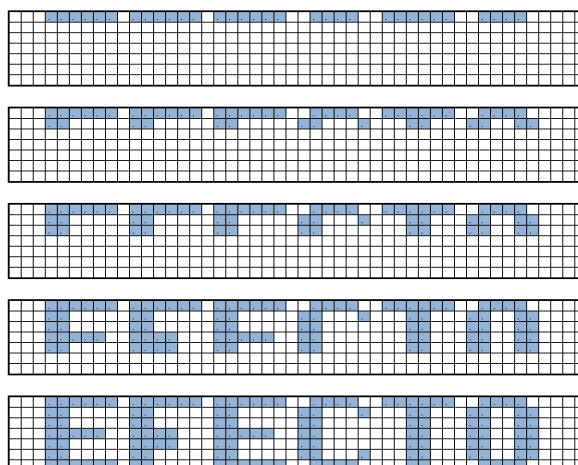


Figura 2.47: Efecto 8

2.2.3.2.9. Efecto 9 - Abrir Texto 4

Una línea de texto al poseer este efecto es presentado en el centro del rótulo, abriéndose horizontalmente columna por columna, hacia sus lados derecho e izquierdo, la figura 2.48 indica la presentación de una línea con este efecto.

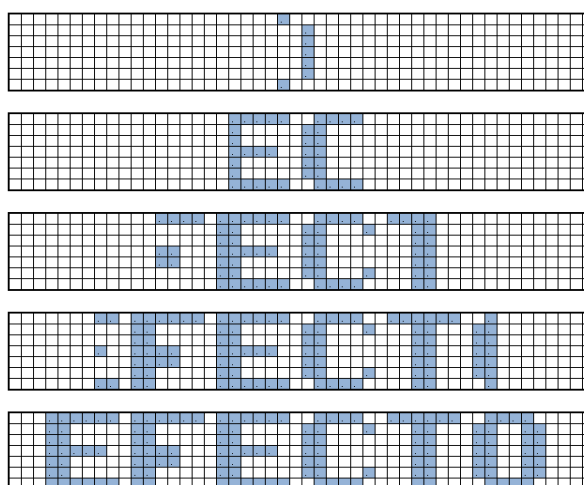


Figura 2.48: Efecto 9

2.2.3.2.10. Efecto 10 - Ascender Texto

La línea de un mensaje al poseer este efecto se presenta en el centro del rótulo desplazándose desde la fila inferior hacia la superior, en la figura 2.49 se visualiza una línea de texto con este efecto.

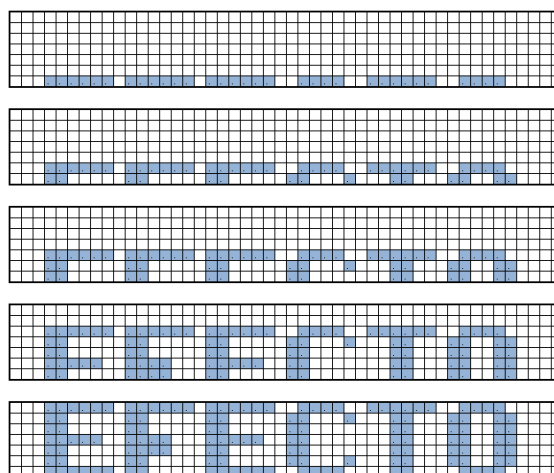


Figura 2.49: Efecto 10

2.2.3.2.11. Efecto 11 - Descender Texto

La línea de un mensaje al poseer este efecto es presentada en el centro del rótulo desplazándose desde la fila superior hacia la inferior. La figura 2.50 muestra como se presenta una línea de texto con este efecto.

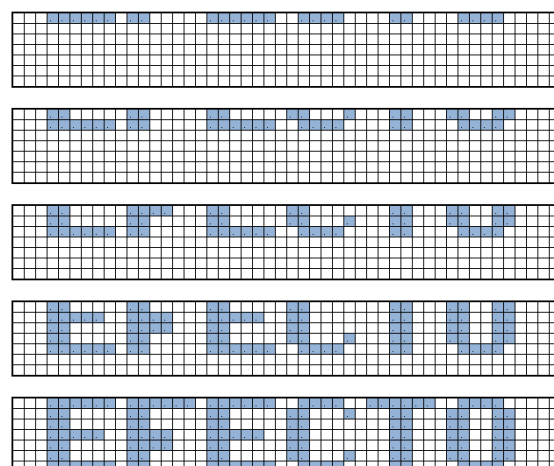


Figura 2.50: Efecto 11

2.2.3.2.12. Efecto 12 – Deletreando

Una línea de texto al poseer este efecto es mostrado de izquierda a derecha, carácter por carácter hasta que el texto completo es observado. La figura 2.51 indica la presentación de una línea de texto con este efecto.

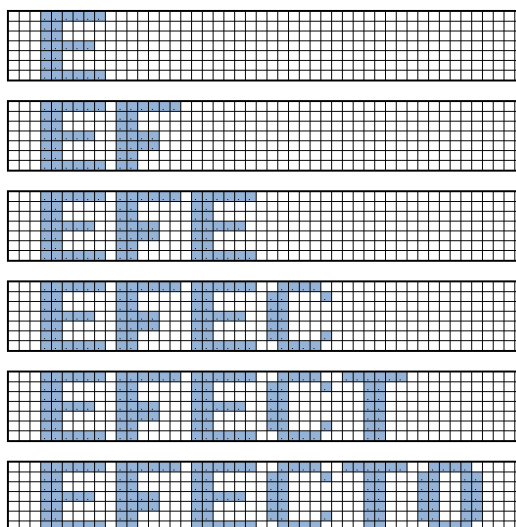


Figura 2.51: Efecto 12

2.2.3.2.13. Efecto 13 - Abriendo Letra

Este efecto presenta los caracteres de una línea, desplazándose de derecha a izquierda carácter por carácter. La figura 2.52 indica la presentación de una línea que posea este efecto.

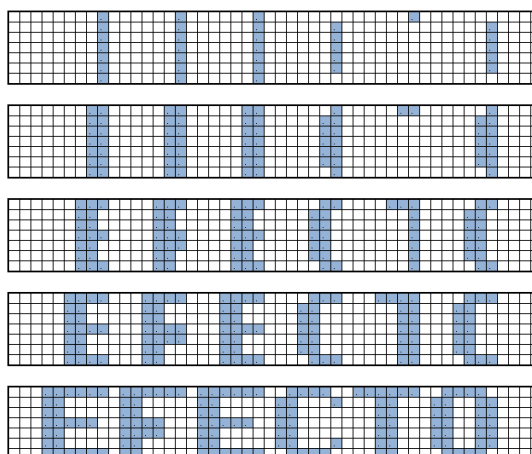


Figura 2.52: Efecto 13

2.2.3.3. Interacciones Entre Efectos

La interacción entre efectos es la manera que una línea de texto con un determinado efecto tiene que salir del rótulo para dar paso a la presentación de la siguiente línea de texto que posea un efecto similar u otro diferente.

2.2.3.3.1. Interacción Entre Efecto 1 y Efecto 1

Cuando se presenta una línea con el efecto 1 y la siguiente posee igual efecto, el texto es enviado a continuación sin esperar que la línea inicial mostrada, salga del rótulo, con esta acción es posible observar el desplazamiento de una frase completa en el rótulo.

2.2.3.3.2. Interacción Entre Efecto 1, 2 y Efecto 2, 3

Al pasar una línea de texto con efecto 1 o 2 a otra con el efecto 2 o 3, la siguiente línea es mostrada después de agregar un espacio equivalente a tres caracteres para diferenciar el cambio de una línea a otra. La figura 2.53 muestra como se presenta los espacios entre dos líneas de texto al poseer estos efectos.

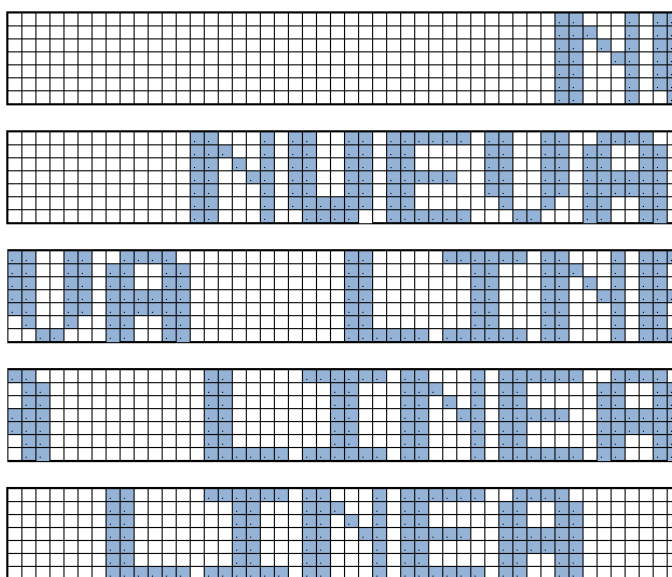


Figura 2.53: Cambio de Efecto 1 a Efecto 2

2.2.3.3.3. *Interacción Entre Efecto 2, 3 y Efecto 1*

Al pasar una línea de texto con efecto 2 o 3 a otra con el efecto 1, la siguiente línea se presenta después de agregar un espacio equivalente a tres caracteres.

2.2.3.3.4. *Interacción Entre Efecto 1, 2,3 y Efecto 4 Al 13*

Cuando el rótulo se encuentra mostrando una línea con efecto 1 y la siguiente posee un efecto fijo, la primera tiene que salir completamente de la matriz del rótulo para que el siguiente efecto pueda ser visualizado.

2.2.3.3.5. *Interacción Entre Efecto 4 Al 13 y Efecto 4 Al 13*

Al tener todos estos efectos una presentación fija, la primera línea es mostrada en el rótulo un tiempo que corresponde al valor de velocidad asignada en el mensaje, para posteriormente desaparecer del rótulo y continuar con la siguiente línea de texto.

2.2.3.3.6. *Interacción Entre Efecto 4 Al 13 y Efecto 1 Al 3*

Cuando se pasa de un efecto fijo a otro con desplazamiento, el primer efecto es mostrado el tiempo que corresponde al valor de velocidad asignada en el mensaje, para posteriormente desaparecer del rótulo y presentar el siguiente mensaje desplazándose desde la primera columna derecha del rótulo.

2.2.3.4. *Máximo Número de Caracteres por Tipo de Efecto*

Cuando la línea de un mensaje posee un efecto diferente al efecto 1, el programa verifica si el contenido puede ser detenido y centrado para ser visualizado en el rótulo, caso contrario la línea es presentada con el efecto 1, con esto se permite la presentación de todo el contenido de la línea debido a que un mensaje tiene que ser mostrado en su totalidad.

La cantidad de caracteres máximos que pueden ser visualizados en el rótulo varían dependiendo del tipo de letra que posea el mensaje y estos valores son presentados en la tabla 2.22.

Tipo de letra	Nº de columnas por carácter	Nº Máximo de Caracteres presentados
1 - Normal	5	8
2 - Negrilla	6	7
3 - Pequeña	6	7
4 - Doble	8	5

Tabla 2.22: Cantidad de caracteres

2.2.3.5. Tarjeta de Led's

En el rótulo los mensajes son visualizados en dos tarjetas de led's, en donde cada tarjeta está conformada por una matriz de led's de 7 filas x 24 columnas, conformada con los buffers 74LS595, transistores PNP, resistencias , un ULN y postes para la conexión de señales entre las tarjetas.

La información a ser mostrada es enviada por el microcontrolador mediante cadenas de datos utilizando los pines declarados como: Dat, Clk, Stb, Stb_fila, Clk_fila, Dat_fila. Los pines Dat, Clk y Stb envían la información del mensaje hacia las columnas del rótulo, mediante los buffers para luego pasar por una resistencia y posteriormente esta información se presenta en la base de un transistor PNP, que controla el encendido y apagado de los led's de una columna. Cada transistor tiene a la salida de su emisor una resistencia de protección para los led's, la misma que protege a cada led evitando que se queme.

Los pines Stb_fila, Clk_fila y Dat_fila controlan el orden de encendido de cada fila mediante un buffer que permite encender las mismas utilizando un ULN que soporta la corriente total presente en cada fila.

Cada tarjeta tiene distribuida sus pistas de manera ordenada, facilitando la conexión de elementos y manteniendo una estética de presentación, las pistas conservan igual orden de pines de conexión para el ingreso y salida de datos, permitiendo utilizar buses de datos en la conexión entre tarjetas de manera secuencial.

También cada tarjeta de led's posee la circuitería para la conexión de un regulador de voltaje 7805 el cual regula una entrada de voltaje de 12 voltios provenientes de una fuente.

En el gráfico presentado en la figura 2.54 se visualiza el diagrama esquemático de la tarjeta de led's.

En el gráfico 2.55 se indica el ruteado de las pistas de la tarjeta de led's y en la figura 2.56 la posición de los elementos presentes en la tarjeta.

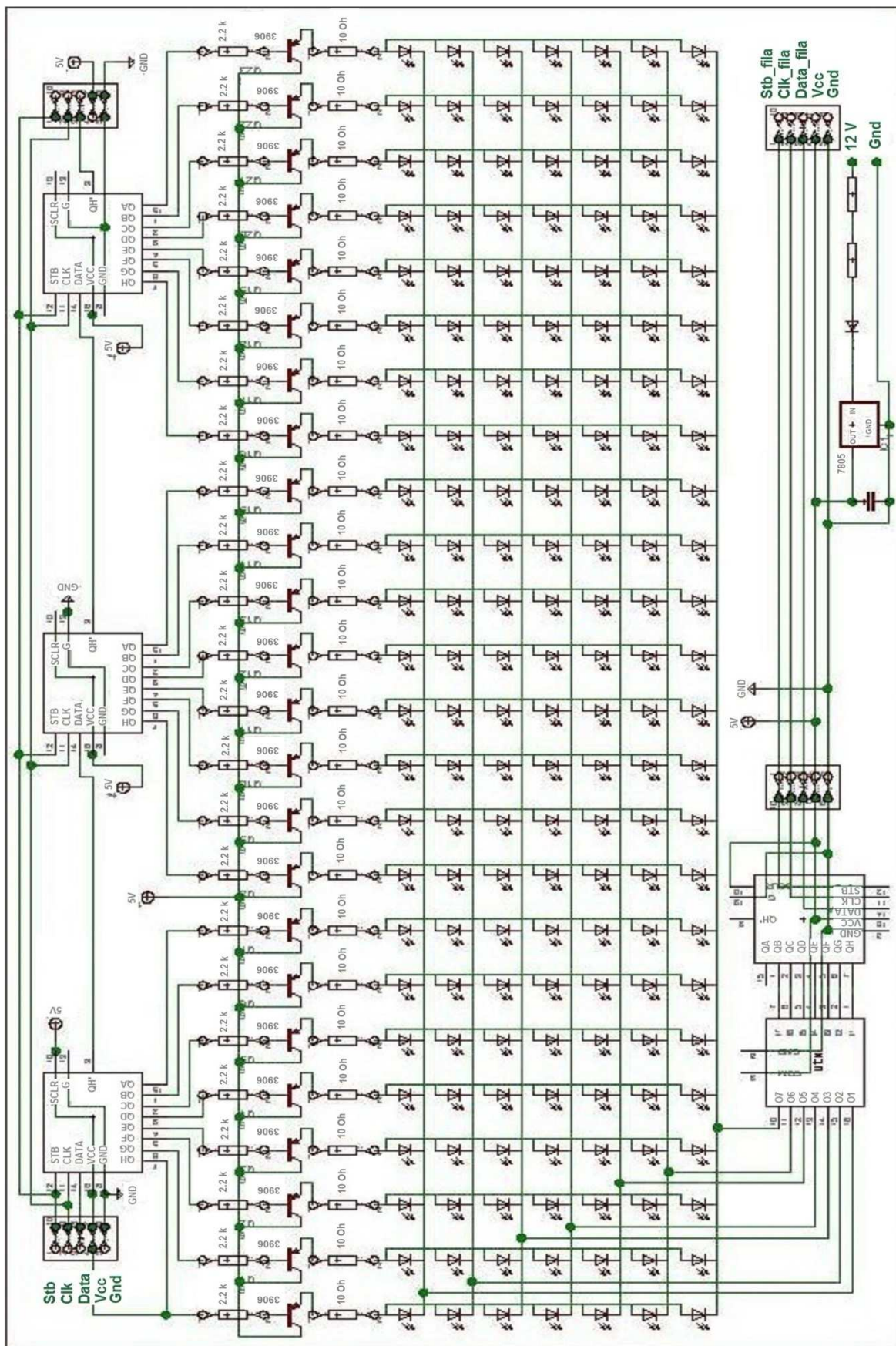


Figura 2.54: Diagrama Esquemático de la tarjeta de led's

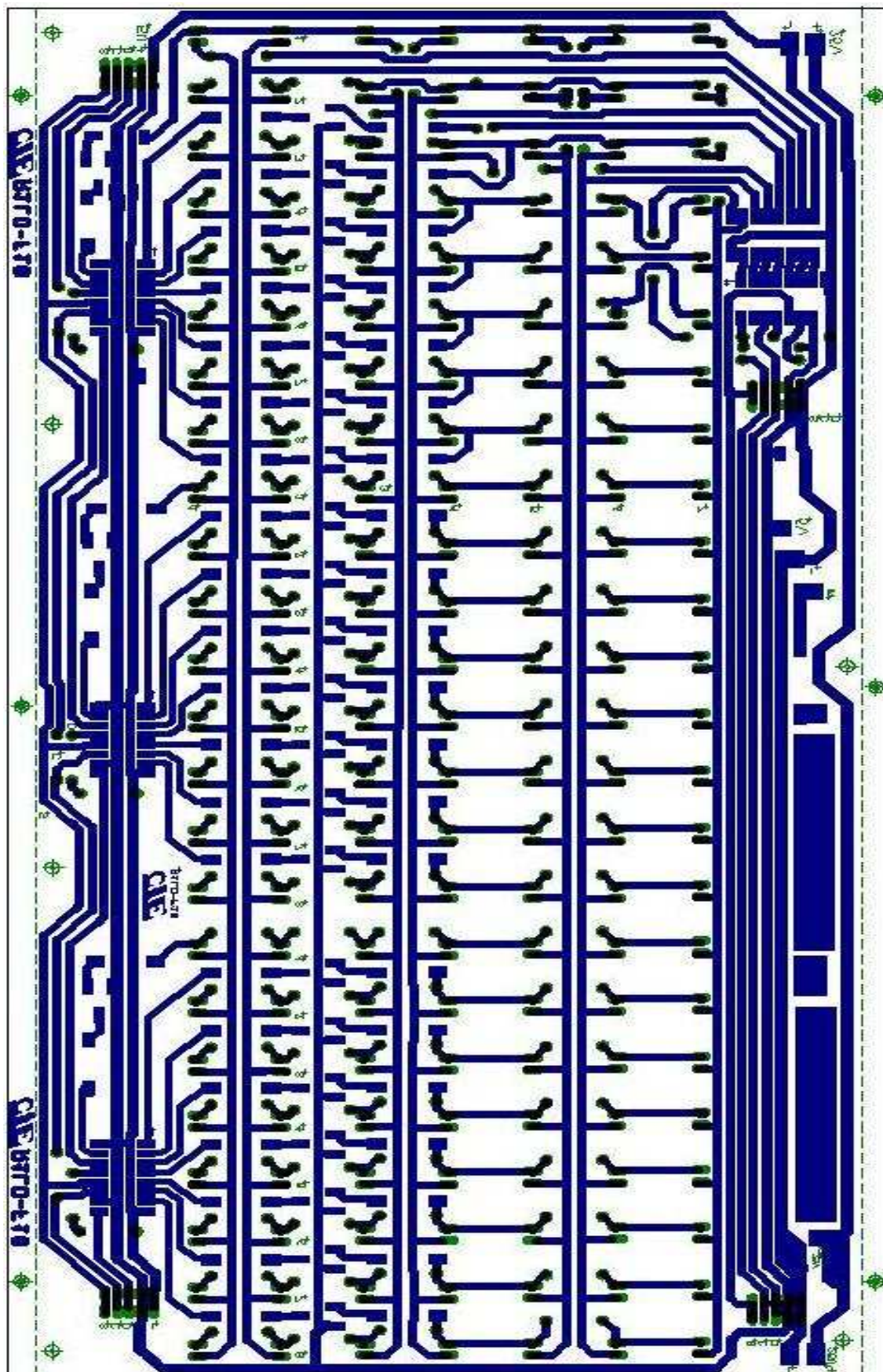


Figura 2.55: Líneas de conexión de la tarjeta de led's

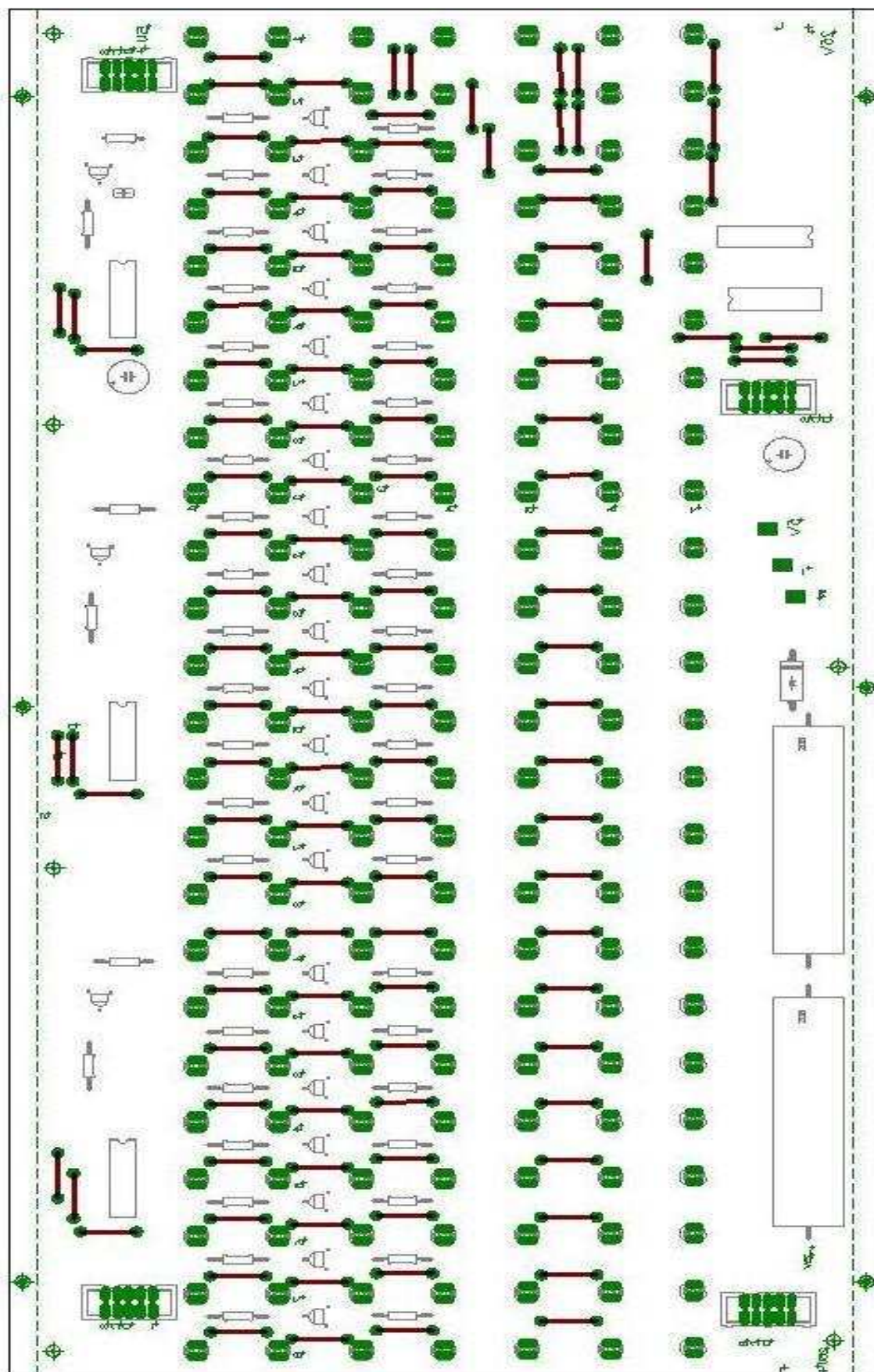


Figura 2.56: Disposición de elementos en tarjeta de led's

2.2.4. FUENTE

La alimentación para el cerebro de control y las tarjetas de led's, se establece con el regulador LM7805, cuya entrada de voltaje es la fuente de 12 voltios. El voltaje de la fuente después de ser regulado, provee voltaje al cerebro de control y a las dos tarjetas de led's que conforman el rótulo.

El consumo de corriente del cerebro de control, es máximo de 100 mA, mientras que cada tarjeta de led's consume una corriente máxima de 350 mA. Por lo tanto, la corriente que soporta la fuente del rótulo será alrededor de 800 mA.

Los 800 mA que soporta la fuente son proveídos por dos reguladores de voltaje LM7805, los cuales proporcionan el voltaje al sistema electrónico publicitario, como se presenta en la figura 2.57.

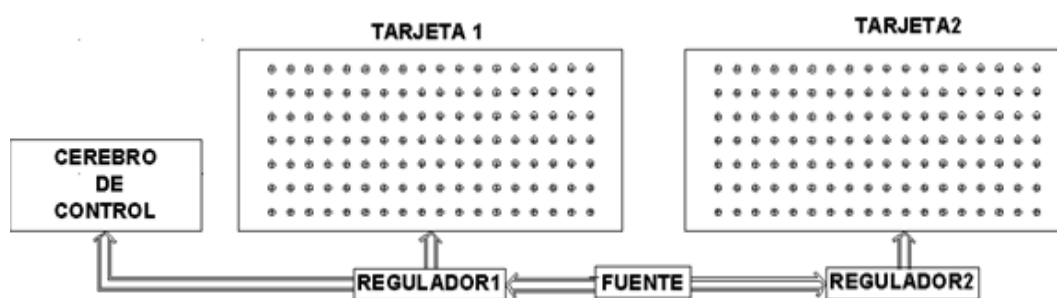


Figura 2.57: Distribución de Reguladores LM7805

La figura 2.58, presenta el diagrama de conexión de un regulador de voltaje, el mismo que es similar para las dos tarjetas de led's.

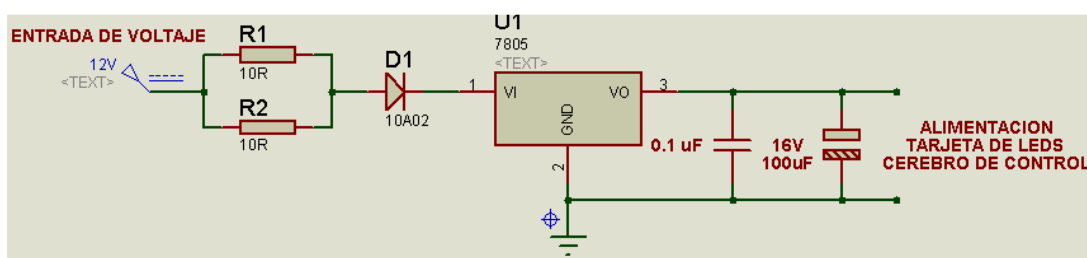


Figura 2.58: Diagrama de Regulación de Voltaje.

A la entrada del voltaje de la fuente se conectan dos resistencias de 10 ohmios en paralelo, que permiten que el voltaje que ingresa a los reguladores se presente en un valor menor a los 12 V, esto se realiza para que las resistencias ayuden a disipar el calor y la potencia al regulador, después posee un diodo de protección para evitar una mala polarización de voltaje y finalmente el voltaje ingresa al regulador LM 7805.

El condensador de 0.1uF permite remover cualquier ruido de alta frecuencia de circuitos vecinos, este valor es recomendado por el fabricante, el capacitor de 100µF estabiliza y mejora el rechazo de rizado a la salida.

Los valores de corriente soportados por cada regulador son:

$$\text{Regulador1} = \text{Cerebro de Control} + \text{Tarjeta1} = 100 \text{ mA} + 350 \text{ mA} = 450 \text{ mA}$$

$$\text{Regulador2} = \text{Tarjeta2} = 350 \text{ mA}$$

La potencia máxima disipada para cada regulador es:

$$\text{Potencia} = (\text{Voltaje_fuente} - \text{Voltaje_resistencia} - 5) \times I$$

$$\text{Potencia1} = [12 - (5 \times 450 \text{ mA}) - 5] \times 450 \text{ mA}$$

$$\text{Potencia1} = [12 - 2.25 - 5] \times 450 \text{ mA}$$

$$\text{Potencia1} = 2,13 \text{ w}$$

$$\text{Potencia2} = [12 - (5 \times 350 \text{ mA}) - 5] \times 350 \text{ mA}$$

$$\text{Potencia2} = [12 - 1,75 - 5] \times 350 \text{ mA}$$

$$\text{Potencia2} = 1.83 \text{ w}$$

2.3. IMPLANTACION DEL SISTEMA

2.3.1. TARJETA DE CEREBRO

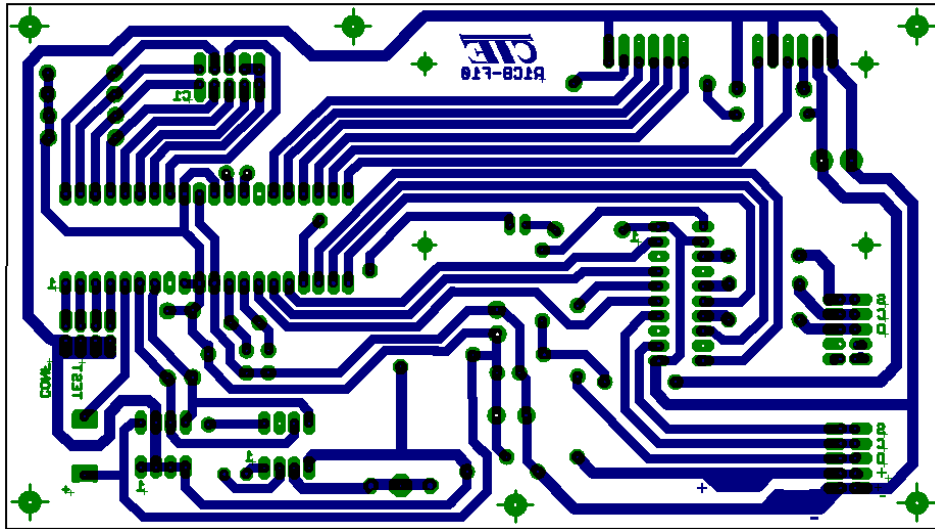


Figura 2.59: Diagrama en Eagle Cerebro de Control

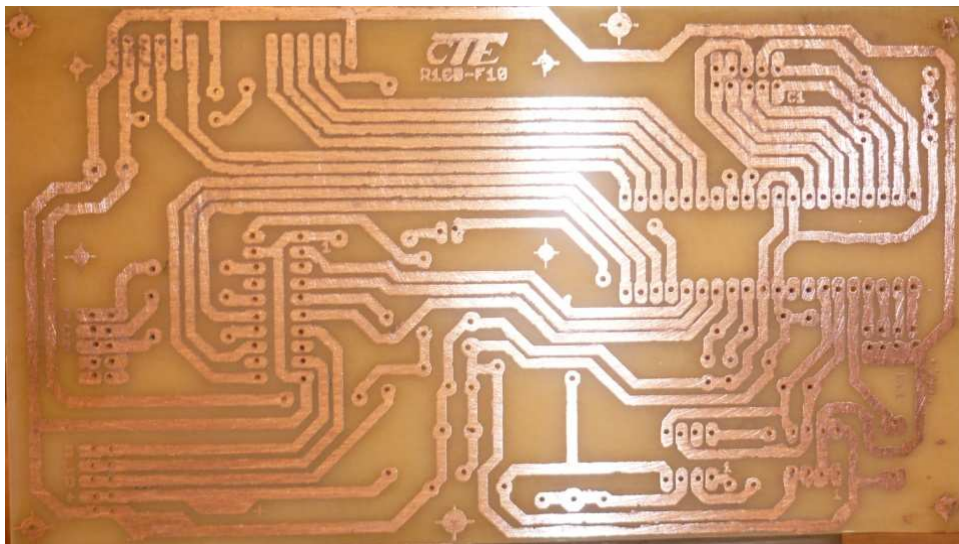


Figura 2.60: Tarjeta de Cerebro de Control

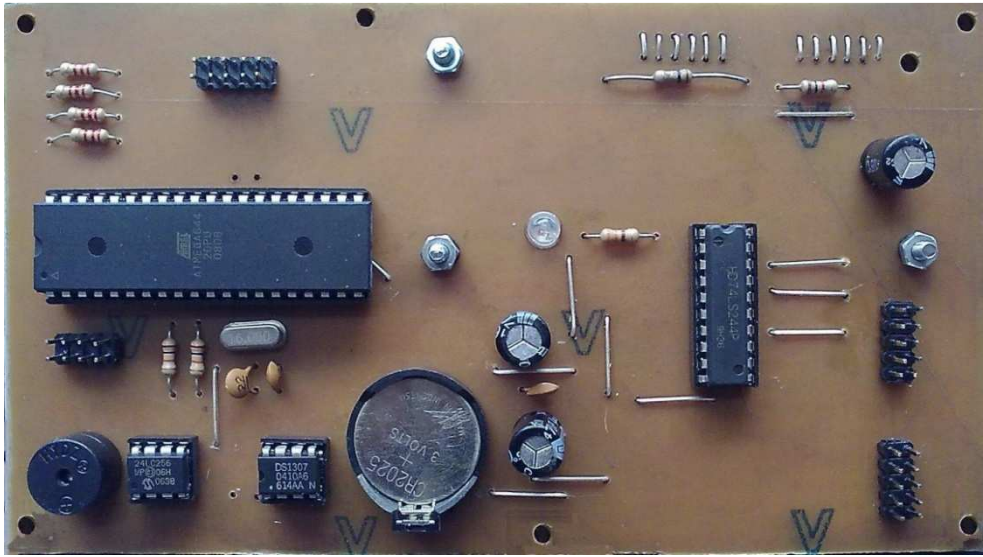


Figura 2.61: Cerebro de Control, lado de Elementos

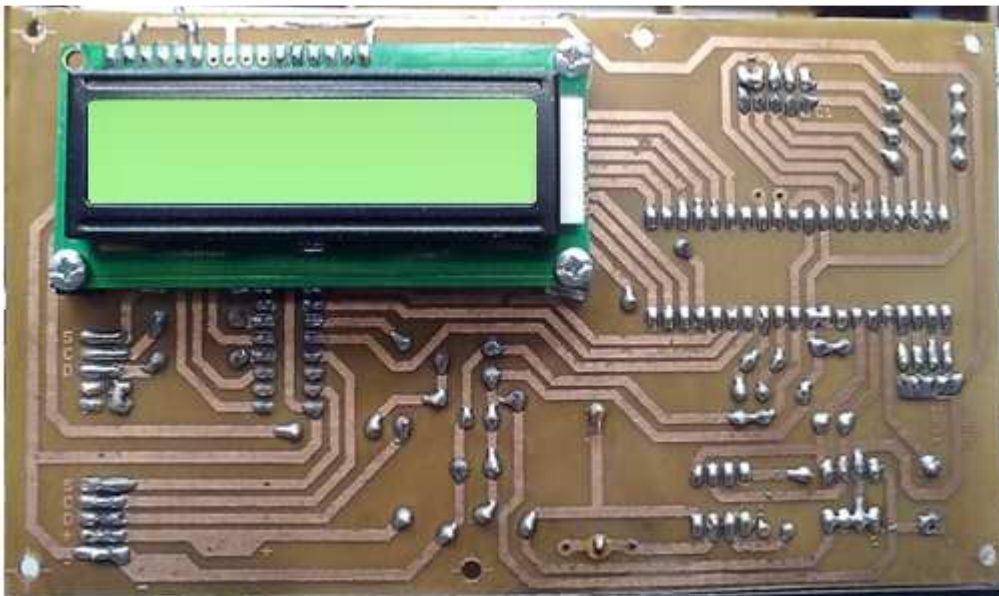


Figura 2.62: Cerebro de Control, lado de Pistas



Figura 2.63: Teclado del Cerebro de Control

2.3.2. TARJETA DE LED'S

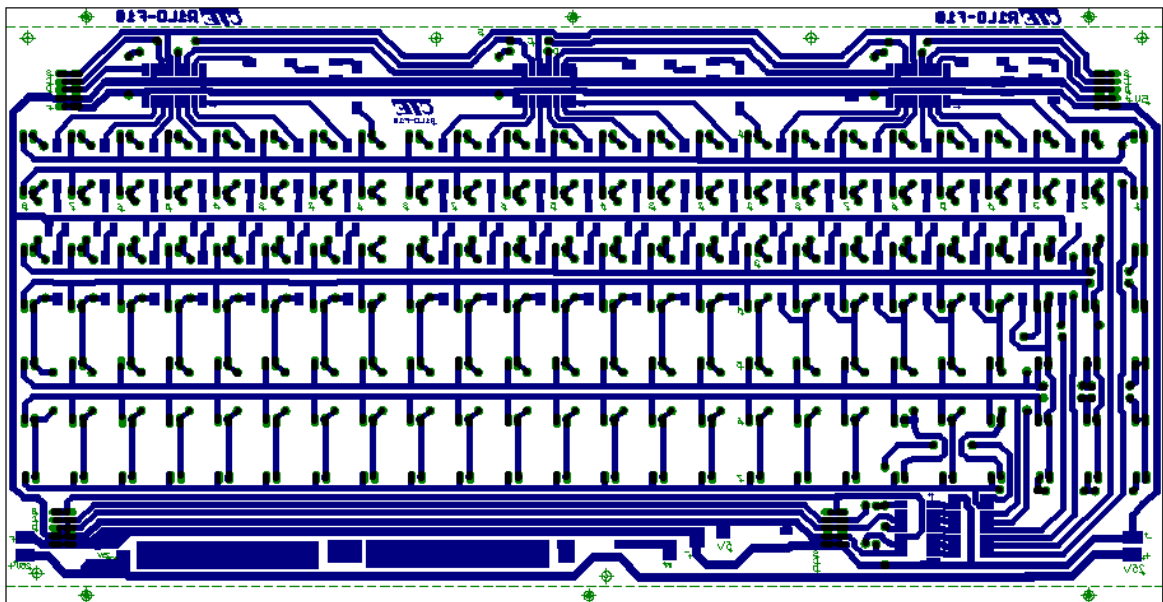


Figura 2.64: Diagrama en Eagle Tarjeta de Led's

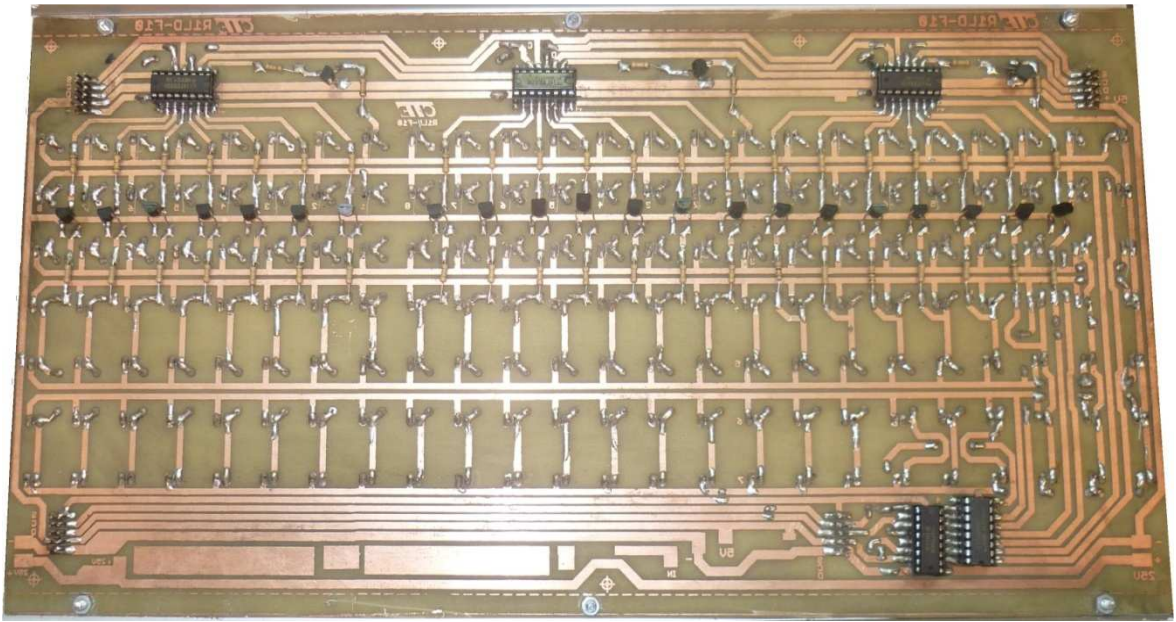


Figura 2.65: Tarjeta de Led's

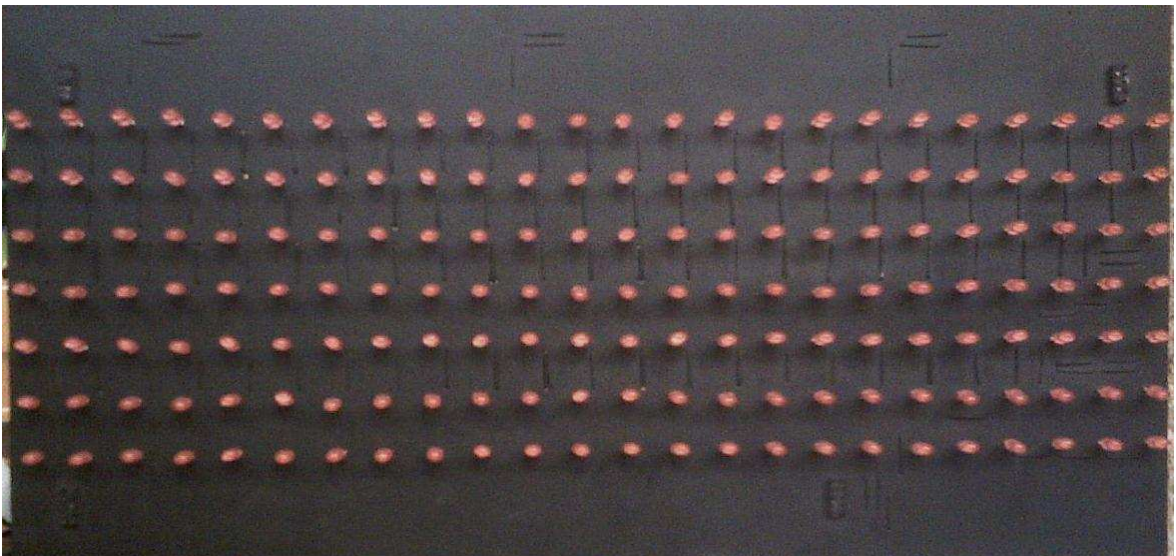


Figura 2.66: Tarjeta de Led's

2.3.3. ARMADO DEL ROTULO

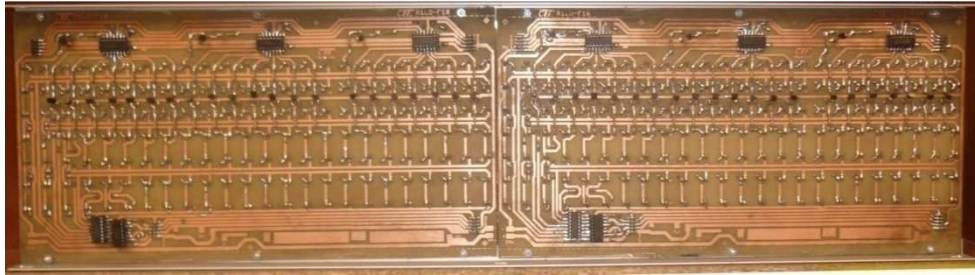


Figura 2.67: Ensamblaje del Rótulo lado de Pistas y Elementos

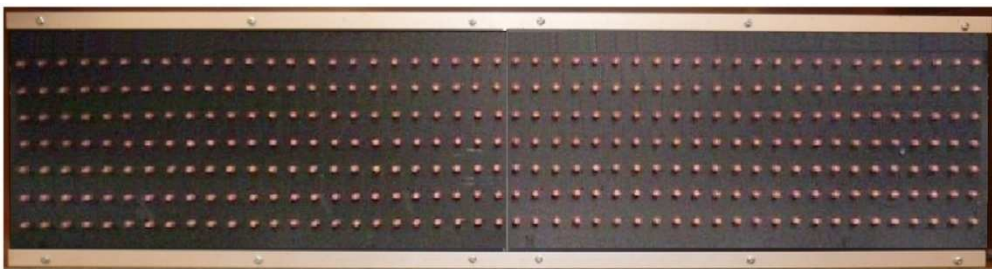


Figura 2.68: Ensamblaje del Rótulo lado de Led's



Figura 2.69: Tapas Laterales del Rótulo



Figura 2.70: Tapa Horizontal

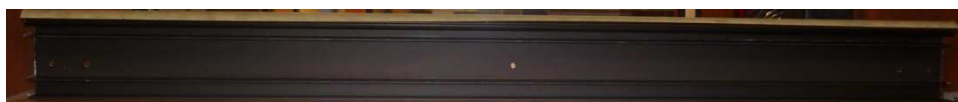


Figura 2.71: Tapa de Aluminio Superior



Figura 2.72: Tapa de Aluminio Inferior

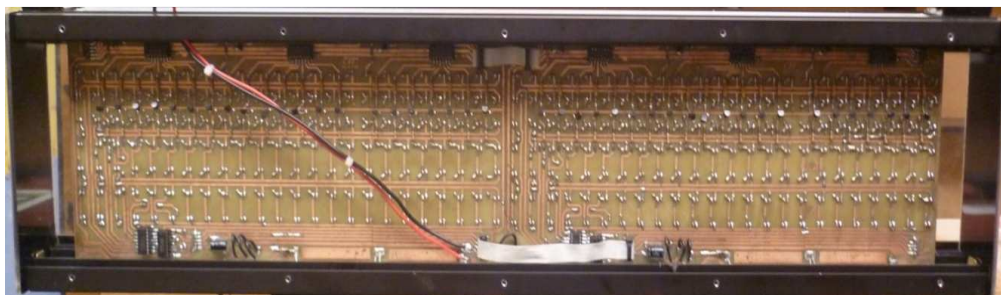


Figura 2.73: Rótulo Ensamblado

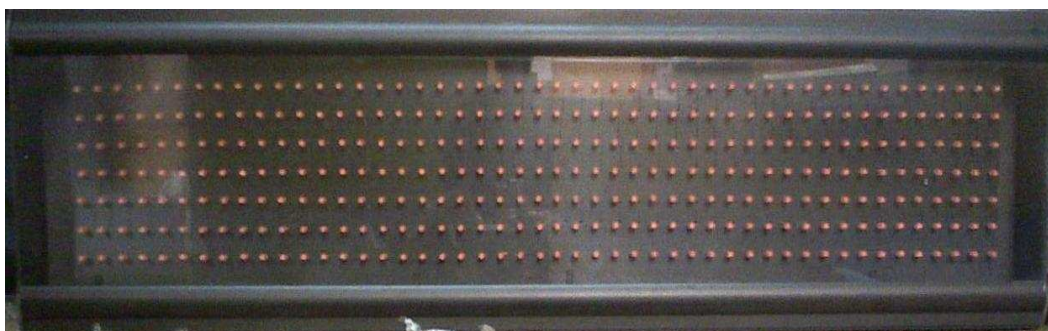


Figura 2.74: Rótulo Ensamblado Vista Frontal



Figura 2.75: Rótulo con Publicidad

2.4. DESCRIPCION DEL SOFTWARE

Los diagramas de flujo presentados a continuación indican el funcionamiento del programa principal. Los diagramas realizados solo presentan las banderas principales utilizadas para la habilitación de los menús, el camino que sigue el programa cuando una bandera cambia de estado en el menú principal, dando lugar al ingreso a un determinado menú, para la modificación de algún dato de configuración, parámetro de presentación del mensaje o modificación del mensaje y la condición que se debe cumplir para volver al menú principal.

En la explicación de cada diagrama de flujo se indica las funciones que realiza el programa y se menciona el funcionamiento de las partes que se han considerado necesarias para entender, como la información es procesada e interpretada.

2.4.1. DESCRIPCION DE LA FIGURA 2.76

Cuando se inicia el funcionamiento del rótulo en el programa se inicializan todas las variables a ser utilizadas, las librerías de control del integrado de reloj DS 1307 y las funciones que controlaran a la memoria 24LC256.

Luego de esto el programa ingresa a una sección de menús de configuraciones iniciales, en donde verifica si existe algún jumper conectado a tierra para permanecer en esta sección y direccionarse a un menú específico.

Dependiendo del tipo de jumper conectado el programa realiza:

- El jumper_1 permite ingresar al menú de configuraciones iniciales.
- El jumper_2 permite ingresar al menú de test de teclado en el que se verifica si alguna tecla fue presionada para enviar hacia el LCD el número de tecla presionada.
- El jumper_3 permite ingresar al menú de test de tarjetas para verificar el estado de los led's. En este menú los led's de una fila se encienden de manera progresiva de derecha a izquierda hasta que toda la fila se

encuentre encendida, para luego apagarse y realizar el mismo procedimiento en la siguiente fila.

Posteriormente al desconectar el jumper, el programa inicializa la bandera Band_publicidad = 1, para luego ingresar a la sección del programa que presenta la publicidad. La publicidad es leída carácter por carácter desde la memoria 24LC256. Cada carácter de la publicidad es leído, decodificado y presentado en el rótulo, hasta leer el carácter (ß), que indica el final de la publicidad. Encontrado el carácter (ß), la bandera Band_publicidad es llevado a cero, lo cual permite que el programa continúe con la siguiente sección de trabajo.

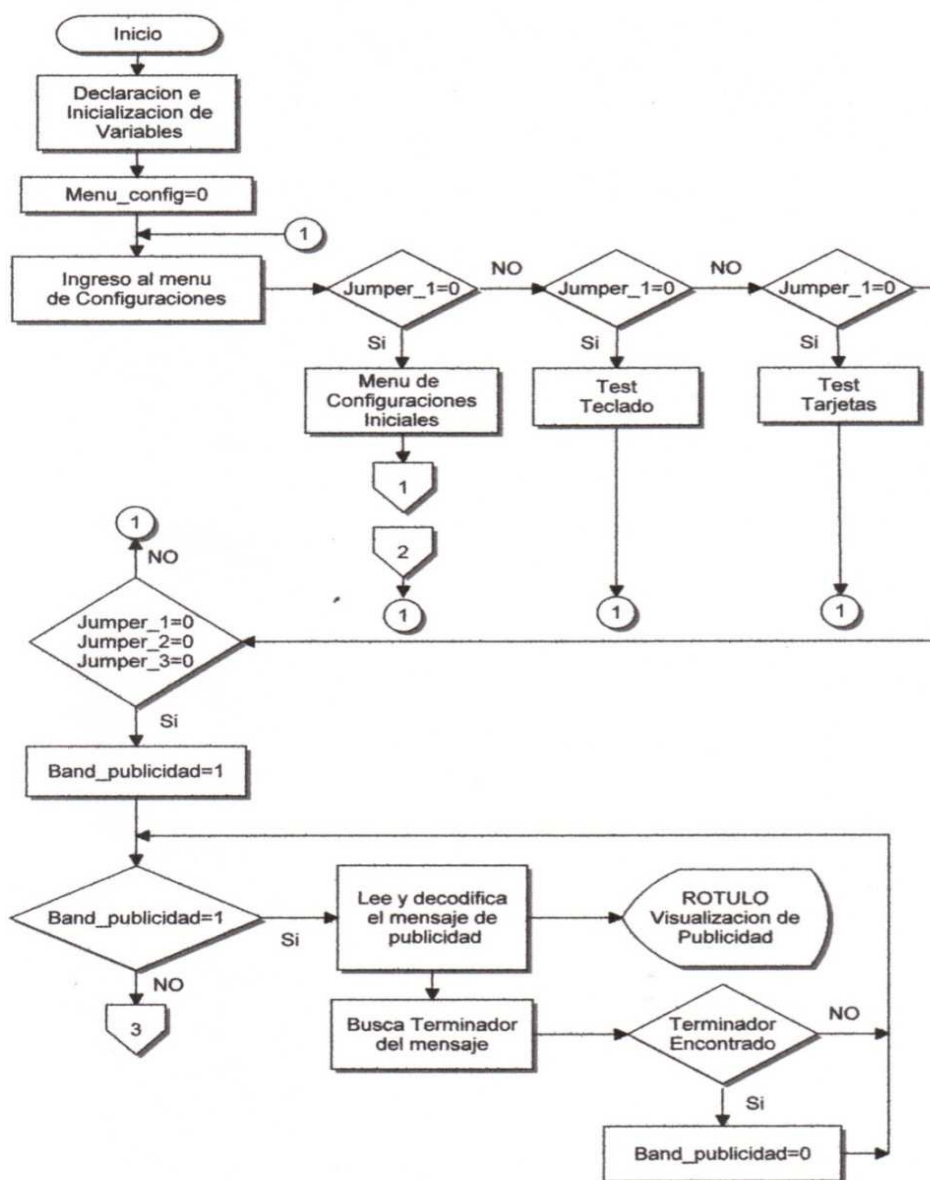


Figura 2.76: Diagrama del Menú de Verificación de Jumpers

2.4.2. DESCRIPCION DE LA FIGURA 2.77

El ingreso a este menú se realiza cuando en Jumper_1 es conectado a negativo, aquí el programa siempre chequea la tecla 10, 12 y 14. La tecla 12 cambia el estado de un byte entre cero y uno, para determinar la presencia o no del número telefónico en la publicidad. La tecla 10 determina el número de mensajes que presentará el rótulo. La tecla 14 incrementa la variable Menu_config, reiniciando este valor en cero cuando la variable es mayor a cuatro. Dependiendo del valor de la variable Menu_config se activa un menú diferente.

Si el valor de la variable Menu_config es uno se activa el menú que permite configurar la fecha de fabricación en el cual la tecla 1 permite incrementar y guardar el valor del día, la tecla 2 incrementa y guarda el valor del mes y la tecla 3 incrementa y guarda el valor del año.

Si el valor de la variable Menu_config es dos se activa el menú que permite configurar el número de serie en el cual la tecla 1 permite incrementar y guardar el valor de la centena, la tecla 2 incrementa y guarda el valor de la decena y la tecla 3 incrementa y guarda el valor de la unidad.

Si el valor de la variable Menu_config es tres se guarda el valor de la fecha de la revisión 1 o si es cuatro se guarda el valor de la fecha de la revisión 2, según estos menús sean seleccionados, en donde la tecla 1 incrementa y guarda el valor del día, la tecla 2 incrementa y guarda el valor del mes y la tecla 3 incrementa y guarda el valor del año.

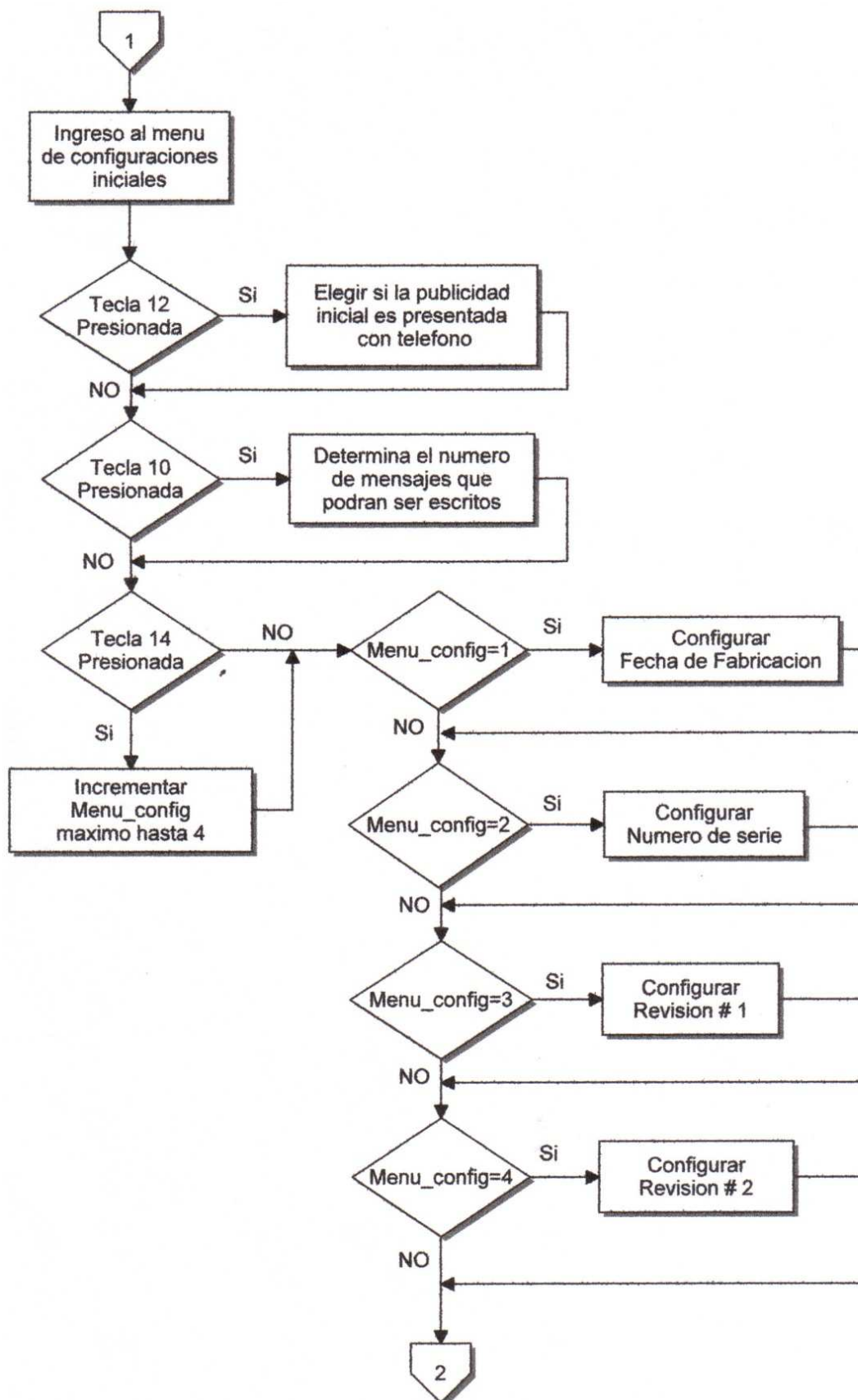


Figura 2.77: Diagrama del Menú de Configuraciones Iniciales.

2.4.3. DESCRIPCION DE LA FIGURA 2.78

En la figura se encuentran el diagrama del menú principal del programa en donde es posible modificar el contenido y las características de un mensaje. En este diagrama se presentan las rutas que debe seguir cada parte del programa cuando un diferente menú es activado.

El ingreso a esta sección del programa se realiza después de que la publicidad es presentada en el rótulo y previamente inicializando las banderas que permitirán controlar a toda esta sección.

El menú principal es controlado por la bandera Band_menu, Band_leer_msj y X_menu.

Al ingresar a esta sección del programa la bandera, Band_menu es igual a uno para habilitar el camino que permite mostrar el mensaje en el rótulo y chequear el teclado principal. Lo primero que realiza el programa es preguntar si la bandera Band_leer_msj es igual a uno para leer el mensaje a ser presentado en su totalidad carácter por carácter, hasta encontrar el terminador final del mensaje, con esto se determina la longitud total del mensaje, su número de caracteres, que permite conocer el número de líneas escritas. Si el terminador no es encontrado, un terminador es puesto automáticamente. Esto se determina conociendo que una línea tiene 16 caracteres y si los 16 últimos caracteres leídos poseen el valor de 32 en decimal, equivalente al espacio , significa que no se encuentra escrito nada a continuación y el terminador es escrito 16 espacios antes del último espacio leído, que es una posición después del último carácter escrito. Posteriormente la bandera Band_leer_msj es llevada a cero, permitiendo continuar con el programa.

Después el mensaje, es leído línea por línea si, X_menu es igual a cero, buscando en esta línea el terminador de línea o el terminador de mensaje, determinando si una línea puede ser mostrada con un determinado efecto, para saltar a la siguiente línea o si se encuentra el terminador final para presentar el

mensaje desde su inicio. Cada línea mostrada es leída carácter por carácter decodificada y presentada en el rótulo.

Si X_menu es igual a cero también se verifica el teclado principal que habilita los menús para cambiar el contenido o la presentación de un mensaje.

Si al iniciar la lectura del mensaje en su totalidad y el mismo solo obtenía espacios, se presenta en el rótulo la frase, “Corp. Electronica TELEVID”, con el efecto 1, para que el usuario visualice el funcionamiento del rótulo.

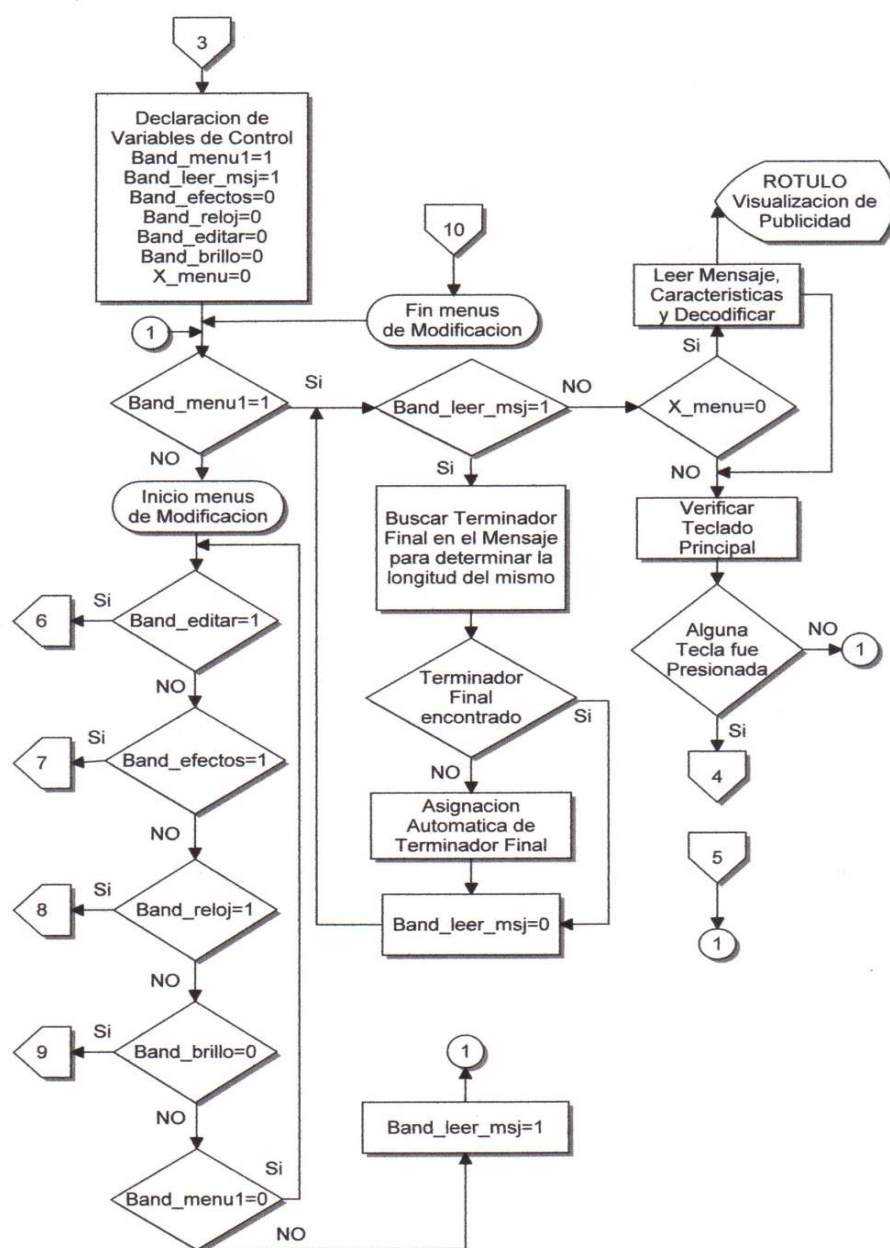


Figura 2.78: Diagrama del Menú Principal

2.4.4. DESCRIPCION DE LA FIGURA 2.79

En el teclado principal se verifica el número de veces que la tecla menú (Tecla 10) ha sido presionada para incrementar la variable X_menu que al ser diferente de cero no presenta el mensaje en el rótulo. Este menú muestra en el LCD la información de cada menú que puede ser seleccionado en el teclado y se elige uno presionando la tecla enter (Tecla15), el mismo que desactiva la bandera del menú principal Band_menu=0 y activa el menú seleccionado.

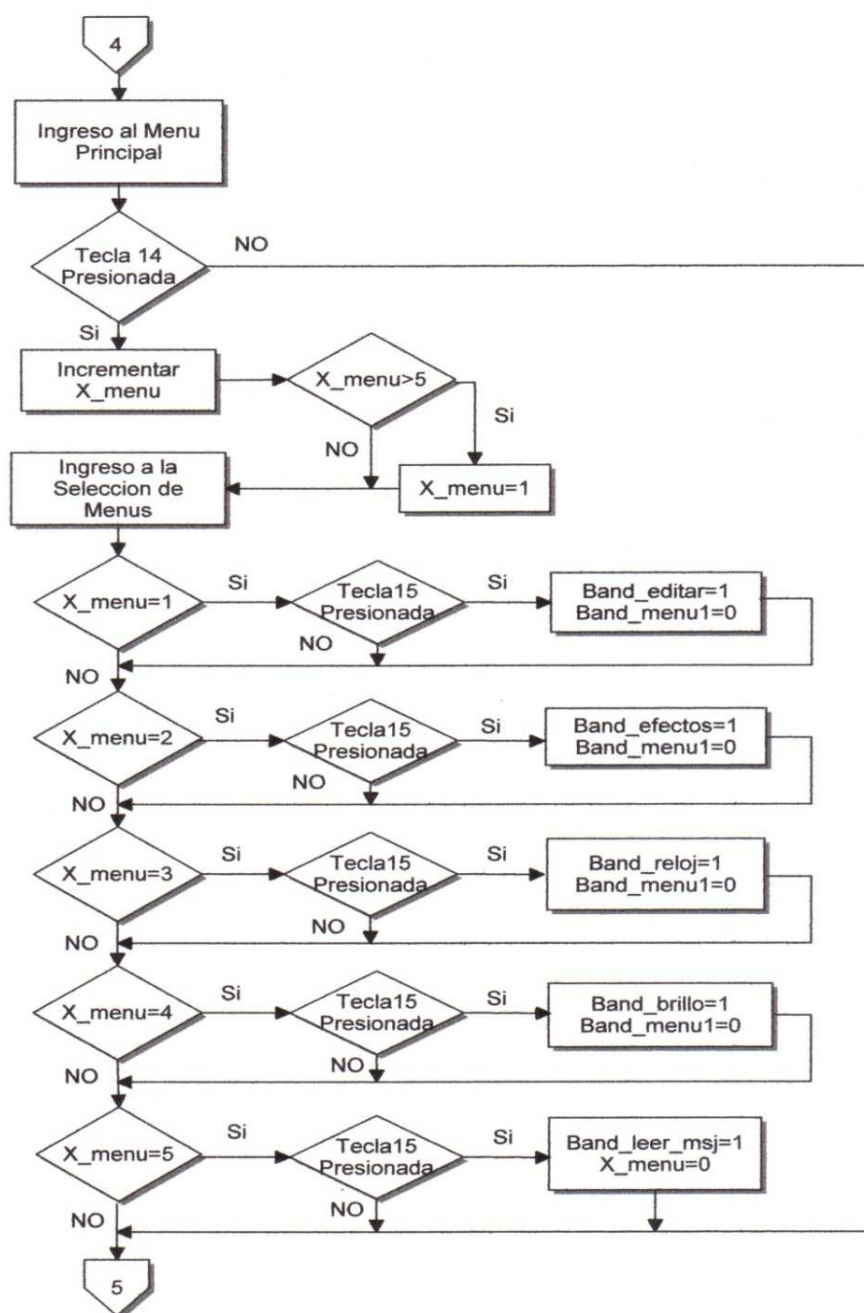


Figura 2.79: Diagrama del Teclado Principal

2.4.5. DESCRIPCION DE LA FIGURA 2.80

Al ingresar al menú edición en el teclado principal el programa inicia presentando un submenú que permite seleccionar la línea a ser editada o modificada, esta opción facilita el avance entre líneas rápidamente, para ingresar a la edición de la línea seleccionada se presiona la tecla enter. Realizado esto se ingresa a otro submenú que chequea todo el teclado, el mismo que permite escribir el mensaje y guardarlo en la memoria. Para salir de este submenú se presiona la tecla enter.

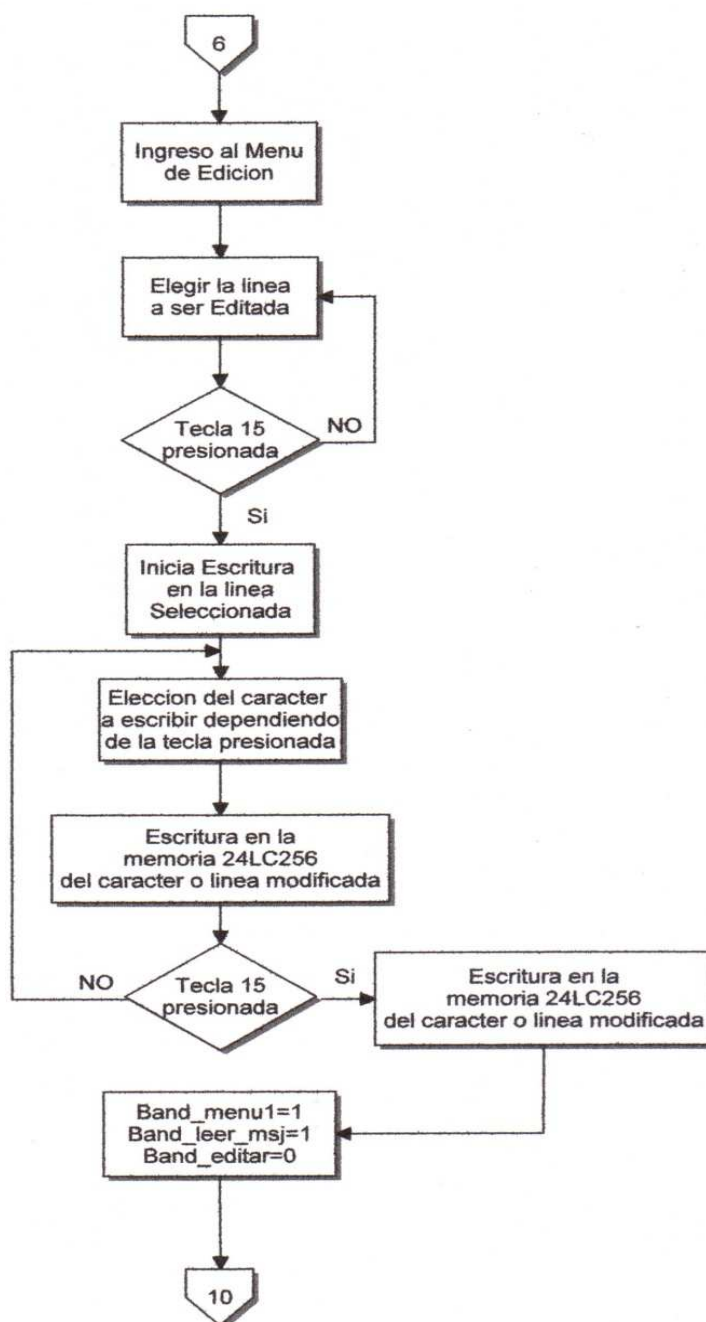


Figura 2.80: Diagrama del Menú de Edición

2.4.6. DESCRIPCION DE LA FIGURA 2.81

Al ingresar al menú de configuración de efectos se presenta en el rótulo la primera línea del mensaje con su respectivo efecto. El efecto es seleccionado con la tecla 15 y las teclas 13 y 16 permiten seleccionar la línea. Al instante de modificar el efecto de una línea, su nuevo valor es guardado en la memoria y seguidamente se presenta en el rótulo la línea con el nuevo efecto. Si la línea no puede ser mostrada con el efecto asignado debido a la cantidad de caracteres que posee la línea, se muestra con el efecto 1, permitiendo que el mensaje se presente completamente en el rótulo. Para salir de este menú se presiona la tecla 14.

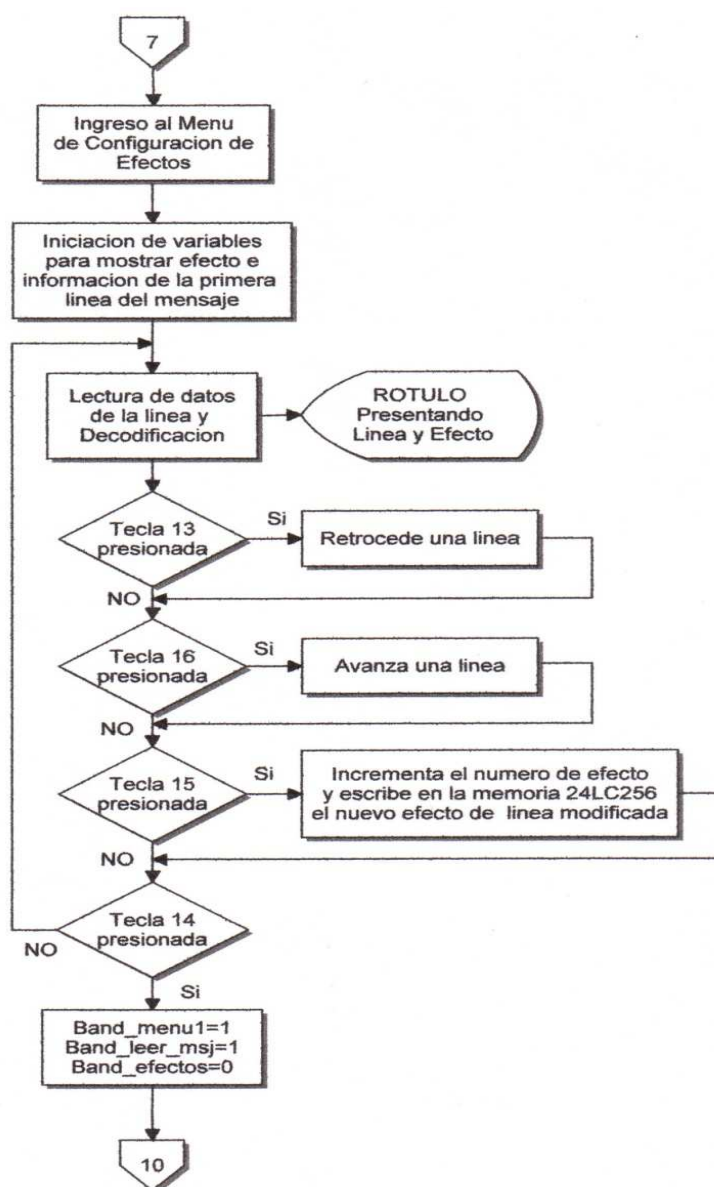


Figura 2.81: Diagrama del Menú Efectos

2.4.7. DESCRIPCION DE LA FIGURA 2.82

El menú de hora y fecha permite insertar la hora, la fecha o ambos datos al final de un mensaje. Este menú está conformado de cuatro submenús que son presentados de manera ordenada según los datos a ser mostrados. El momento de ingresar a este menú los datos de Band_hora y Band_fecha son leídos en la memoria 24LC256 para presentar en el LCD los valores de configuración que tiene el primer submenú (Figura 2.30).

El primer submenú permite elegir, si se desea observar la hora o la fecha, utilizando las teclas 13 y 16 que varían el valor de las banderas Band_hora y Band_fecha, para salir de este submenú se presiona la tecla enter. Realizado esto el programa verifica si alguna de las banderas es igual a uno, para continuar con el siguiente submenú, caso contrario el programa sale del menú porque esto significa que no se desea insertar ninguna información adicional que no sea parte del mensaje.

Si una bandera es igual a uno, en el segundo submenú se ingresa el número de veces que un mensaje tiene que ser visualizado antes de insertar la hora o la fecha, en pasos de dos en dos con las teclas 13 y 16, se confirma el cambio realizado con la tecla enter.

El tercer submenú se utiliza para igualar el valor de la hora, seleccionando horas y minutos.

2.4.8. DESCRIPCION DE LA FIGURA 2.83

El cuarto submenú iguala el valor de la fecha con las teclas 13,16 y 10 seleccionando día, mes y año respectivamente.

Dependiendo del valor de las banderas Band_hora y Band_fecha, el programa selecciona el menú a ser modificado, debido a que no se debe mostrar un menú que no fue habilitado.

Los cambios de cada valor de Band_hora, Band_fecha, las variables de número de ciclos y las que guardan los datos de hora y fecha son actualizados y guardados en la memoria 24LC256 el instante de presionar cada tecla para modificar su valor, en el submenú correspondiente.

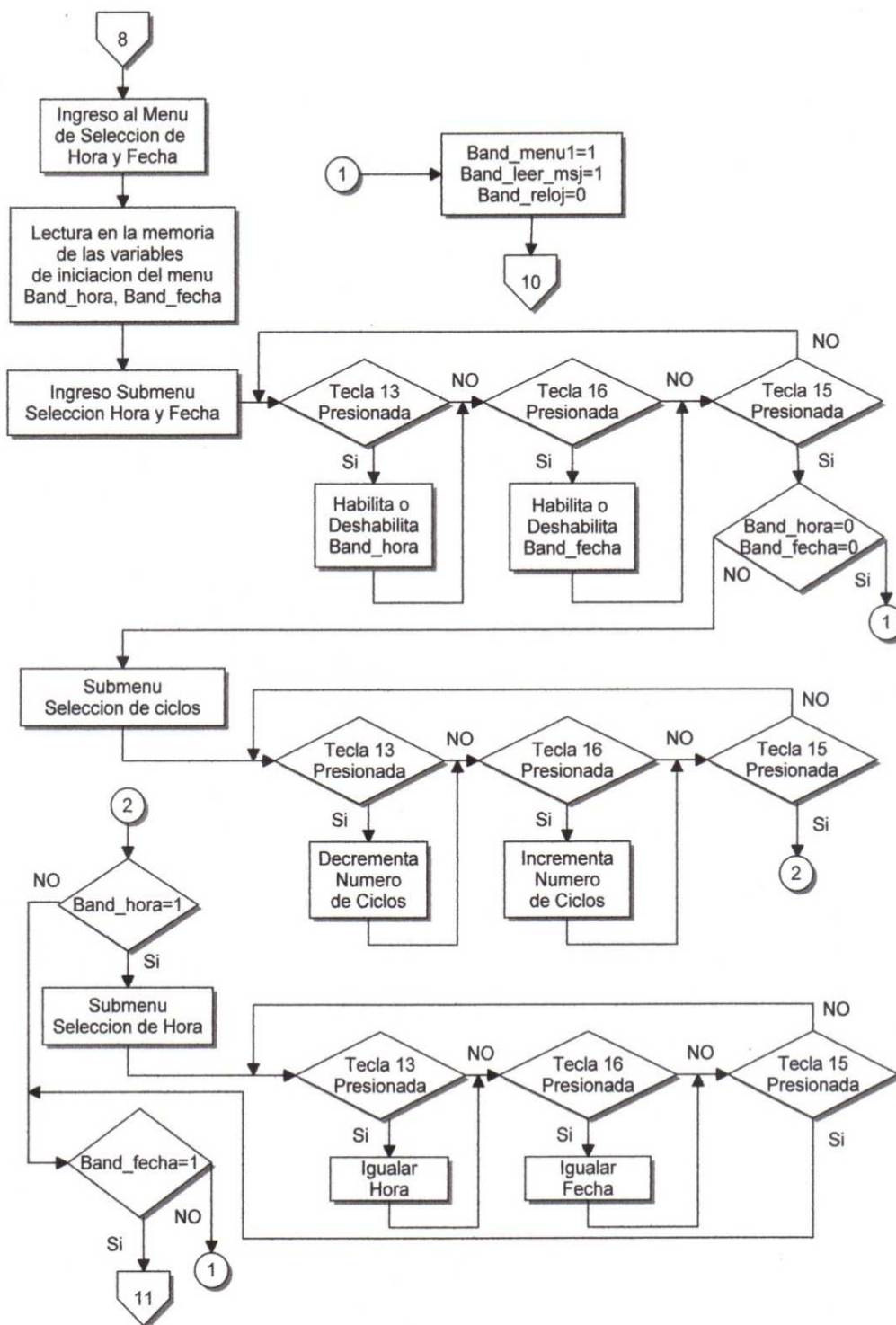


Figura 2.82: Diagrama del Submenú de Selección de Número de Ciclos y Hora

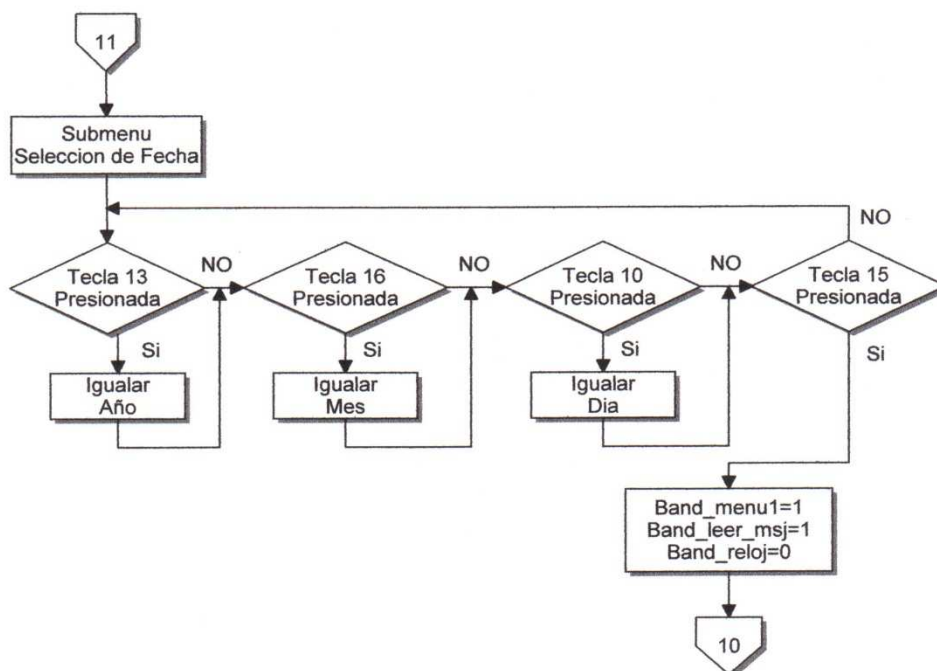


Figura 2.83: Diagrama del Submenú de Selección de Fecha

2.4.9. DESCRIPCIÓN DE LA FIGURA 2.84

Al ingresar al menú de brillo se selecciona la intensidad que tendrán los led's al momento de presentar un mensaje, la tecla 13 decrementa la intensidad de los led's y la tecla 16 incrementa la intensidad de los led's, la intensidad es seleccionable entre cinco valores diferentes.

Mientras el programa se encuentra en este menú, en el rótulo se presenta el mensaje escrito, permitiendo visualizar las variaciones de brillo con las que puede ser presentado el mensaje. Cada vez que se incrementa o decrementa el valor de brillo, este es actualizado y guardado en la memoria, para salir de este menú se presiona la tecla enter.

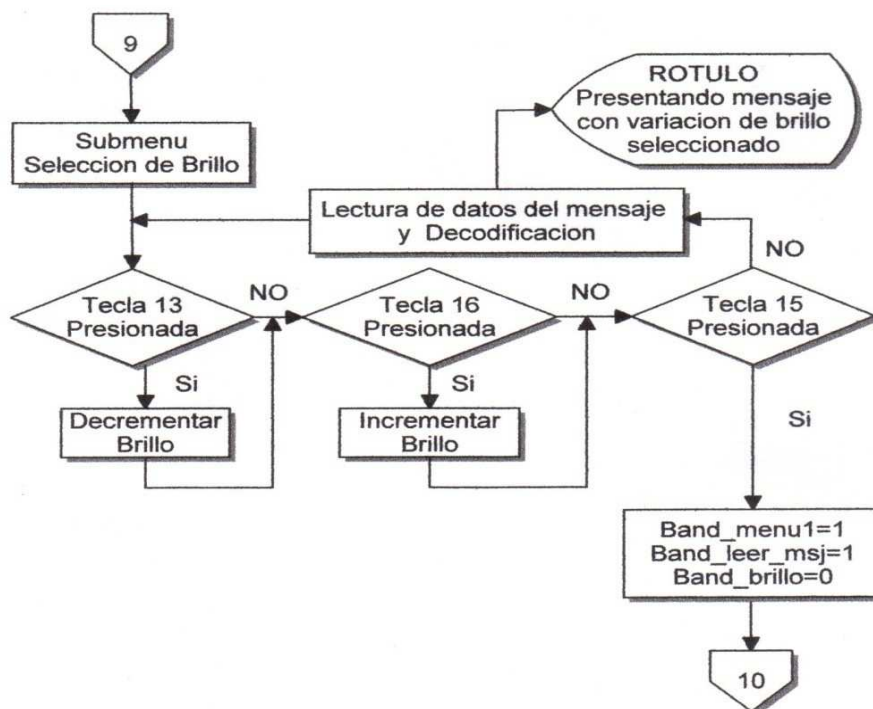


Figura 2.84: Diagrama del Menú de Selección de Brillo

2.4.10. SOFTWARE PARA EL SISTEMA

En el CAPITULO 1 se indicó los comandos básicos del software BASCOM AVR, a continuación estos son utilizados para la programación del microcontrolador ATmega 644 para así lograr el control del sistema electrónico publicitario.

-
- Lo primero que se tiene que indicar al iniciar el programa es especificar el microcontrolador a utilizar, la frecuencia de trabajo y habilitar las configuraciones iniciales.

```
$regfile = "m644def.dat"
$crystal = 16000000
Config Watchdog = 2048 : Start Watchdog
CIs : Cursor Off Noblink
```

-
- A continuación se declaran las variables y subrutinas a utilizar en la lectura y escritura de la memoria 24LC256.

```
'Definicion Para Manejo De Memorias Eeprom
Const Cod_esc = 160 : Const Cod_lec = 161
Config Scl = Portb.5 : Config Sda = Portb.6
Dim Direccion As Long At $100 , Direccion_h As Byte At $101 Overlay ,
Direccion_l As Byte At $100 Overlay
Dim Valor As Byte , String_lcd As String * 16 At $415 , String_lcd_x(16) As Byte At
$415 Overlay
Declare Sub Write_eeprom(byval ADR As Byte , Byval Value As Byte)
Declare Sub Write_eeprom16(byval ADR As Byte , Byval Value As Byte)
Declare Sub Read_eeprom(byval ADR As Byte , Value As Byte)
Declare Sub Read_eeprom16(byval ADR As Byte , Byval Value As Byte)
```

-
- Declaración de la librería y variables a utilizar en la lectura y escritura del Integrado de Reloj DS1307.
-

```
$lib "ds1307clock.lib"
```

```
Config Date = Mdy , Separator = / : Config Clock = User
```

```
'Configura Direcciones I2c Del Ds1307
```

```
Const Ds1307w = &HD0 : Const Ds1307r = &HD1: Dim _weekday As Byte , Hora  
As String * 8 , Fecha As String * 8 Band_hora_fecha As Bit
```

- Configuración de los pines del microcontrolador como entradas o salidas y asignación del nombre o alias a ser utilizado para cada pin.
-

```
'Configurar Teclado
```

```
Ddra.2 = 0 : Porta.2 = 1 : Columna1 Alias Pina.2
```

```
Ddra.1 = 0 : Porta.1 = 1 : Columna2 Alias Pina.1
```

```
Ddra.3 = 0 : Porta.3 = 1 : Columna3 Alias Pina.3
```

```
Ddra.0 = 0 : Porta.0 = 1 : Columna4 Alias Pina.0
```

```
Ddra.4 = 1 : Porta.4 = 1 : Fila1 Alias Porta.4
```

```
Ddra.7 = 1 : Porta.7 = 1 : Fila2 Alias Porta.7
```

```
Ddra.5 = 1 : Porta.5 = 1 : Fila3 Alias Porta.5
```

```
Ddra.6 = 1 : Porta.6 = 1 : Fila4 Alias Porta.6
```

```
'Definicion de Puertos de salida de senal para control de COLUMNAS
```

```
Ddrd.0 = 1 : Portd.0 = 0 : Dat Alias Portd.0
```

```
Ddrd.1 = 1 : Portd.1 = 0 : Clk Alias Portd.1
```

```
Ddrd.2 = 1 : Portd.2 = 0 : Stb Alias Portd.2
```

```
'Definicion de Puertos de salida de senal para control de FILAS
```

```
Ddrd.6 = 1 : Portd.6 = 0 : Dat_fila Alias Portd.6
```

```
Ddrd.5 = 1 : Portd.5 = 0 : Clk_fila Alias Portd.5
```

```
Ddrd.4 = 1 : Portd.4 = 0 : Stb_fila Alias Portd.4
```

```
'Controles Adicionales
```

```
Ddrd.3 = 1 : Portd.3 = 0 : Led Alias Portd.3
```

```
Ddrb.4 = 1 : Portb.4 = 1 : Beep Alias Portb.4
```

Ddrc.5 = 1 : Buffer Alias Portc.5 : Buffer = 0

'Jumpers para Configuracion

Ddrb.0 = 0 : Portb.0 = 1 : Jumper_1 Alias Pinb.0

Ddrb.1 = 0 : Portb.1 = 1 : Jumper_2 Alias Pinb.1

Ddrb.2 = 0 : Portb.2 = 1 : Jumper_3 Alias Pinb.2

-
- Declaración de constantes de control, utilizadas para calcular las diferentes direcciones de la memoria 24LC256 para leer y escribir los mensajes y sus respectivas características.

Const Letra_negrilla = 440 : Const Letra_pequena = 968

Const Letra_doble = 1496 : Const Publi_sintelf = 3000

Const Publi_contelf = 3048 : Const Letra_mensaje = 3500

Const Velocidad_mensaje = 3600 : Const Efecto_linea = 3700

Const Inicio_mensaje = 9200 : Const Lectura_hora = 5100

Const Lectura_fecha = 5200 : Const Lectura_num_ciclos = 5300

Const Lectura_brillo = 5400 : Const Const_publicidad = 2990

Const Const_cant_mensaje = 2991: Const Mensaje_memoria = 2992

Const Columnas_rotulo = 48 : Const Cant_string = 55 :

Const Ubicar_prox_columna = 51

-
- Códigos de caracteres especiales, presentados en el lcd.

Deflcdchar 0 , 32 , 1 , 7 , 31 , 7 , 1 , 32 , 32	' FIGURA IZQUIERDA
Deflcdchar 1 , 32 , 16 , 28 , 31 , 28 , 16 , 32 , 32	' FIGURA DERECHA
Deflcdchar 2 , 4 , 10 , 17 , 27 , 10 , 10 , 14 , 32	' TECLA SHIFT
Deflcdchar 3 , 4 , 10 , 21 , 4 , 21 , 10 , 4 , 32	' FLECHA ARRIBA ABAJO
Deflcdchar 4 , 4 , 14 , 31 , 14 , 14 , 14 , 14 , 32	' FLECHA ARRIBA
Deflcdchar 5 , 14 , 14 , 14 , 14 , 31 , 14 , 4 , 32	' FLECHA ABAJO
Deflcdchar 6 , 32 , 4 , 15 , 31 , 15 , 4 , 32 , 32	' FLECHA IZQUIERDA
Deflcdchar 7 , 32 , 4 , 30 , 31 , 30 , 4 , 32 , 32	' FLECHA DERECHA

-
- En esta parte se declara las variables a utilizar en la programación del microcontrolador.
-

'Banderas de Teclados

Dim Band_menu1 As Bit , Band_menu2 As Bit , Band_submenu1 As Bit

Dim Band_submenu2 As Bit, Band_efectos As Bit , Band_inst_hora_fecha As Bit

Dim Band_reloj As Bit , Band_fecha As Bit

Dim Band_ciclos_hora_fecha As Bit , Band_control_brillo As Bit

'Banderas de Lectura de textos

Dim Band_leer As Bit: Dim Mensaje As Byte , Linea As Byte , Linea1 As Byte ,
Linea_aux As Byte: Dim Velocidad As Byte , Tipo_letra As Byte , Tipo_efecto As
Byte , Efecto_aux As Byte: Dim Publicidad As Byte

Dim Cant_mensaje As Byte , Cant_lineas As Byte , Cant_caracteres_mensaje As
Word: Dim Cant_caracteres_letra As Byte , Cant_caracteres_letra_aux As Byte

Dim X_pausa As Word , Cant_pausa As Word: Dim X_indicativo As Byte ,
For_indicativo As Byte: Dim Direccion_aux As Word , Direccion_aux2 As Word ,
Direccion_aux3 As Word , Ubicacion_memoria As Word

Dim Prox_caracter As Word , Prox_caracter_aux1 As Word , Prox_caracter_aux2
As Word: Dim Posicion_tabla As Word , Columnas_tipo_letra As Byte ,
Columnas_tipo_letra_aux As Byte: Dim Ini_codigo_caracter As Word ,
Fin_codigo_caracter As Word

'Definir Variables para manejo de datos

Dim String_x As String * 1: Dim X_tecla As Byte , Tecla_string As String * 1 ,
Tecla_men As Byte , Tecla_men_aux As Byte: Dim X_pulso_tecla As Byte ,
For_pulso_tecla As Byte: Dim Avanza_caracter As Bit , Avanza_caracter_aux As
Word: Dim Band_sigue_linea As Bit , Band_sigue_linea_aux As Long

Dim Mayus_minus As Bit , X_cont_caracter_0 As Byte

Dim X_lcd_posicion As Byte , Band_lcd_posicion As Bit

Dim X_for As Word , Caracter_byte As Byte , Entries As Byte

Dim Fila_encendida As Byte , Fila_encendida_aux As Byte , For_fila As Byte ,
X_fila As Byte: Dim For_columna As Byte , Contar_columnas As Byte ,
Band_incr_columnas As Bit , Band_fin_mensaje As Bit, Band_linea_fija As Bit ,


```

Fin_reinicia As Byte. Dim Espacios_fin As Byte , Fin_mensaje As Word ,
Fin_mensaje_aux As Word:Dim Band_fin_linea As Bit , Band_fin_linea_aux As Bit
, Band_fin_linea_2 As Bit:Dim Band_centrado As Byte:Dim Band_leer_efecto As
Bit , Band_efecto2 As Bit , Band_efecto3 As Bit , Band_efecto4 As Bit , Fin_linea
As Byte:Dim Band_efec_1_4 As Bit,Dat_string As String * 1
Dim X_menu As Byte , Menu_configuracion As Byte
Dim Cant_columnas_centrado As Byte , Cant_columnas_centrado_ini As Byte ,
Espacio_centrado As Byte , Espacio_centrado_ini As Byte,Linea_aux2 As Word
'Variables para jumpers
Dim Band_jumper1 As Bit , Band_jumper2 As Bit , Band_jumper3 As Bit
Dim Dia As Byte , Mes As Byte , Anio As Byte:Dim Dia1 As Byte , Dia2 As Byte ,
Mes1 As Byte , Mes2 As Byte , Anio1 As Byte , Anio2 As Byte
Dim Centena As Byte , Decena As Byte , Unidad As Byte , Dir_men As Word ,
Pres_pub As Bit
'Variables para menus
Dim Dato_hora As Byte , Dato_fecha As Byte , Numero_ciclos As Byte ,
Dato_am_pm As String * 2:Dim Dato_day_year As String * 2 , Mes_lcd As String *
5 , Dato_brillo As Byte , Dato_brillo_aux As Byte:Dim Fin_linea_efectos As Word ,
Prox_caracter_inicial As Word , Prox_caracter_final As Word
Dim Band_mostrar_efec As Bit , Contador_terminador As Byte:
Band_no_incrementar As Bit
Dim Dat_string_aux As String * 1 , Ter_fin_linea As Byte , Ter_fin_mensaje As
Byte,Band_velocidad_fija As Bit , Band_velocidad_fija2 As Bit , Band_indicativo
As Bit,Band_fecha_hour As Bit , Band_restar_columnas As Bit , Sal_fecha As
Byte: Dim Cont_incr_columna_aux As Byte,Band_vacio As Bit,Men_vacia As Bit
Band_no_incrementar = 0 : Band_hora_fecha = 0 : Band_velocidad_fija2 = 0
'Variables para mostrar la hora y fecha
Dim Hora_aux As String * 8 , Fecha_aux As String * 9
Dim Dato_aux_1 As String * 1 , _hour_aux As Byte , Band_sigue_fecha As Bit
Dim Contador_caracter As Byte , Mostrar_pantalla As Bit , Conteo_lcd As Byte
Dim Dato_div As Word , Band_efecto As Bit , Band_efecto_aux As Bit ,
Conteo_efecto As Word: Dim Fin_publicidad As Bit , Fin_publicidad_aux As Byte
Dim Band_12 As Bit , Cant_esp As Byte , Efecto_aux1 As Byte

```

Dim Band_14 As Bit , Band_15 As Bit , Band_41 As Byte , Band_blanco As Bit
 Dim Dato_2 As Byte,X_pos_fin_linea As Byte , X_pos_fin_terminador As Byte ,
 X_pos_fin_linea_aux As Byte,Check_tecla As Bit,Apagando As Word
 Dim Dato_fila_matriz As Byte, Cont_filas As Byte , Direccion_matriz As Byte ,
 Direccion_matriz_aux As Byte,Posicion_matriz As Byte , Posicion_matriz_aux As
 Byte,Dim Band_terminador_efec2 As Bit
 Dim Rotulo_mitad As Byte , Cont_incr_columna As Byte , Cont_abrir_centro As
 Byte , Cont_decr_columna As Byte,Band_destello As Bit,Cont_efec12 As Byte
 Dim Cont_efec13 As Byte , Num_string As Byte
 Dim Num_posicion As Byte , Num_posicion1 As Byte , Fin_for_efec13 As Byte ,
 Cant_columnas_finales As Byte: Dim Cant_caracteres_iniciales As Byte ,
 Contador_efecto As Word , Cont_salida As Byte , Band_publicidad As Bit
 Dim Cont_terminador As Byte , Dato_espacios As Byte
 Dim String_columnas As String * 80 At \$1000 , String_columnas_x(7) As String * 9
 At \$1000 Overlay: Dim String_columnas_aux1 As String * 80
 Dim String_columnas_aux As String * 10 At \$400 , String_columnas_aux_x(10) As
 Byte At \$400 Overlay: Dim String_salidas_columnas As String * 80
 Dim String_salidas_columnas_aux As String * 80

-
- Asignación de valores iniciales de trabajo del microcontrolador.
-

X_menu = 1: Band_jumper1 = 0 : Band_jumper2 = 0 : Band_jumper3 = 0
 Cant_mensaje = 1 : Cant_lineas = 4 :Cant_caracteres_mensaje = Cant_lineas *
 16 : Incr Cant_caracteres_mensajeGosub Desactiva_teclados : Band_menu1 = 1
 Gosub Fila_blanqueo : Gosub Columna_blanqueo : Gosub
 Limpiar_string_columna: Band_hora_fecha = 1 : Gosub Getdatetime:
 If _month < 1 Or _month > 12 Then _month = 1 : If _year < 9 Or _year > 20 Then
 _year = 9
 If _day < 1 Or _day > 31 Then _day = 1 : If _sec =< 0 Or _sec > 60 Then _sec = 0
 If _min =< 0 Or _min > 60 Then _min = 0 : If _hour < 0 Or _hour > 23 Then _hour
 = 1
 Band_hora_fecha = 1 : Gosub Setdate: Band_hora_fecha = 1 : Gosub Settime
 Band_hora_fecha = 1 : Gosub Getdatetime: Gosub Actualiza_mes

```

Direccion = Const_publicidad : Call Read_eeeprom(1 , Valor) : Publicidad = Valor
If Publicidad > 1 Then : Publicidad = 1 : Direccion = Const_publicidad : Valor =
Publicidad : Call Write_eeeprom(1 , Valor) : End If
Gosub Beeps : Waitms 150 : Gosub Beeps : Waitms 150 : Gosub Beeps : Waitms
150

```

-
- Ingreso a los menús de Configuraciones Iniciales.
-

```

Do
  Reset Watchdog
  If Jumper_1 = 0 Then Gosub Teclado_configurar
  If Jumper_2 = 0 Then Gosub Test_teclado
  If Jumper_3 = 0 Then Gosub Test_targetas
  If Jumper_1 = 1 And Jumper_2 = 1 And Jumper_3 = 1 Then Goto Seguir
Loop
Seguir:
  Cant_mensaje = 1 : Cant_lineas = 3
  If Publicidad = 1 Then Ubicacion_memoria = Publi_contelf
  If Publicidad = 0 Then Ubicacion_memoria = Publi_sintelf
  Mensaje = 1 : Linea = 1: Band_mostrar_efec = 0 : Band_sigue_fecha = 0
  Band_leer = 1 : Contador_caracter = 1 : Band_efecto = 0 :Band_publicidad = 1

```

-
- Ingreso a la sección que permite presentar la publicidad inicial de la Corporación Electrónica Televid.
-

```

Do
  Reset Watchdog
  If Band_leer = 1 Then Gosub Leer_mensaje
  Band_velocidad_fija = 0
  If X_menu = 0 Then Gosub Mostrar_texto
  'CONDICION QUE SE DEVE CUMPLIR PARA SALIR DE ESTA SECCION
  If Band_publicidad = 0 Then Exit Do
Loop

```

-
- Asignación de valores iniciales para controlar los diferentes menús de usuario.
-

'Configuraciones Iniciales

Contador_terminador = 0 : Band_publicidad = 0

Direccion = Mensaje_memoria : Call Read_eeprom(1 , Valor) : Mensaje = Valor

Direccion = Const_cant_mensaje : Call Read_eeprom(1 , Valor) : Cant_mensaje = Valor

If Cant_mensaje <> 25 And Cant_mensaje <> 50 And Cant_mensaje <> 99

ThenCant_mensaje = 50 : Direccion = Const_cant_mensaje : Valor =

Cant_mensaje :Call Write_eeprom(1 , Valor)

End If

If Mensaje < 1 Or Mensaje > 99 Then Mensaje = 1 : Direccion =

Mensaje_memoria : Valor = Mensaje : Call Write_eeprom(1 , Valor)

If Cant_mensaje = 25 Then Cant_lineas = 56

If Cant_mensaje = 50 Then Cant_lineas = 28

If Cant_mensaje = 99 Then Cant_lineas = 14

Cant_caracteres_mensaje = Cant_lineas * 16 : Incr Cant_caracteres_mensaje

Band_mostrar_efec = 0 : Band_sigue_fecha = 0 : Band_velocidad_fija = 0

Band_leer = 1 : Contador_caracter = 1 : Band_efecto = 0 : Ubicacion_memoria =

Inicio_mensaje

- Ingreso a los Menús de Usuario.
-

Do

Reset Watchdog

If Band_menu1 = 1 Then Gosub Menu1

If Band_menu2 = 1 Then Gosub Menu2

If Band_submenu1 = 1 Then Gosub Submenu1

If Band_submenu2 = 1 Then Gosub Submenu2

If Band_efectos = 1 Then Gosub Menu_efectos

If Band_inst_hora_fecha = 1 Then Gosub Menu_inst_hora_fecha

```

If Band_ciclos_hora_fecha = 1 Then Gosub Menu_insertar_ciclos
If Band_reloj = 1 Then Gosub Menu_igualar_reloj
If Band_fecha = 1 Then Gosub Menu_igualar_fecha
If Band_control_brillo = 1 Then Gosub Menu_control_brillo
If Band_leer = 1 Then Gosub Leer_mensaje
If X_menu = 0 Then Gosub Mostrar_texto

```

Loop

-
- De aquí en adelante se presentan las subrutinas que son llamadas por las diferentes secciones del programa.
 - La siguiente subrutina lee el mensaje a ser presentado.
-

Leer_mensaje:

```

Gosub Limpiar_string_columna : Gosub Buscar_fin
Gosub Lcd_linea : Gosub Tipo_letra_aux
'Calculo de la Direccion Auxiliar del Inicio de cada Mensaje
Direccion_aux = Mensaje - 1 : Direccion_aux = Direccion_aux * 16 : Direccion_aux
= Direccion_aux * Cant_lineas
Direccion_aux = Direccion_aux + Ubicacion_memoria : Decr Direccion_aux :
If Men_vacia = 1 Then Direccion_aux = 32495
'Seteo de Banderas
Band_leer = 0 : X_menu = 0 : Band_efecto = 0 : Prox_caracter = 0 :
Band_efecto4 = 1 : Band_linea_fija = 0 : Contar_columnas = 0 :
Band_incr_columnas = 0 : Band_no_incrementar = 1
Return

```

-
- Subrutina que presenta el texto de un mensaje en el rótulo.
-

Mostrar_texto:

```

If Band_efecto = 0 Then
String_columnas = Mid(string_columnas , 2 , Cant_string)
If Band_leer_efecto = 1 Then
Linea_aux2 = Prox_caracter / 16 : Linea = Linea_aux2 : Incr Linea

```

```

Direccion = Mensaje - 1 : Direccion = Direccion * Cant_lineas : Direccion =
Direccion + Linea
Direccion = Direccion + Efecto_linea : Decr Direccion : Call Read_eeeprom(1 ,
Valor) : Tipo_efecto = Valor
If Tipo_efecto > 13 Then : Tipo_efecto = 1 : Valor = Tipo_efecto : Call
Write_eeeprom(1 , Valor) : End If
X_velocidad = 0 : Band_efecto2 = 1 : Band_leer_efecto = 0 : Band_efecto4 = 1 If
Tipo_efecto >= 6 And Velocidad >= 4 Then : Band_velocidad_fija2 = 1 : Else :
Band_velocidad_fija2 = 0 : End If
Gosub Calcula_salida_efec
If Band_publicidad = 1 Then : Tipo_efecto = 1 : Velocidad = 2 : Dato_brillo = 5 :
End
If Contador_terminador = Numero_ciclos And Dato_hora = 1 And Sal_fecha = 1
Then : Tipo_letra = 1 : Gosub Tipo_letra_aux : End If
If Contador_terminador = Numero_ciclos And Dato_hora = 1 And Sal_fecha = 2
Then : Tipo_letra = 1 : Gosub Tipo_letra_aux : End If
If Contador_terminador = Numero_ciclos And Dato_fecha = 1 And Sal_fecha = 3
Then : Tipo_letra = 2 : Gosub Tipo_letra_aux : End If
End If
If Band_incr_columnas = 1 Then Incr Contar_columnas: Band_incr_columnas = 1
If Contar_columnas => Columnas_tipo_letra_aux Then:Contar_columnas = 0 : Incr
Prox_caracter : Toggle Led: End If: Gosub Siguiente_columna
Else
Incr X_velocidad
If Tipo_efecto = 2 Then:Velocidad_aux = Velocidad * 40
If X_velocidad => Velocidad_aux Then : X_velocidad = 0 : Band_efecto = 0 : End
If
End If
If Tipo_efecto = 3 Then:Velocidad_aux = Velocidad * 20 : Velocidad_aux =
Velocidad_aux + 59 : Gosub Flashes
If X_velocidad => Velocidad_aux Then : X_velocidad = 0 : Band_efecto = 0 : End
If
End If

```

```

If Tipo_efecto = 4 Then
Velocidad_aux = Velocidad * 10 : Velocidad_aux = Velocidad_aux + 100: If
X_velocidad => Velocidad_aux Then : X_velocidad = 0 : Band_efecto = 0 : End If
End If
If Tipo_efecto = 5 Then
Velocidad_aux = Velocidad * 20 : Velocidad_aux = Velocidad_aux + 59 : Gosub
Flashes: If X_velocidad => Velocidad_aux Then : X_velocidad = 0 : Band_efecto =
0 : End If
End If
If Tipo_efecto >= 6 And Tipo_efecto <> 9 And Tipo_efecto < 12 Then Gosub
Salidas_efectos_filas
If Tipo_efecto = 9 Then Gosub Salidas_efectos_columnas
If Tipo_efecto = 12 Then Gosub Salidas_deletreando
If Tipo_efecto = 13 Then Gosub Salidas_abriendo_letras: End If
If Tipo_efecto >= 4 And Band_efecto = 0 And Band_efec_1_4 = 0 Then Return
If Tipo_efecto = 9 Or Tipo_efecto = 12 Or Tipo_efecto = 13 And Band_destello = 1
Then : Band_destello = 0 : Return : End If
If Velocidad = 1 Then Cant_pausa = 80
If Velocidad = 2 Then Cant_pausa = 180
If Velocidad = 3 Then Cant_pausa = 300
If Velocidad = 4 Then : Cant_pausa = 220 : Gosub Barrido_filas : End If
If Velocidad = 5 Then : Cant_pausa = 300 : Gosub Barrido_filas : End If
If Band_velocidad_fija2 = 0 Then : Gosub Barrido_filas : End If
Gosub Barrido_filas
Return

```

-
- Subrutinas que permiten que el texto destelle, encendiendo y apagando los led's del rótulo en el efecto 3 y en el efecto 5.
-

Flashes:

```

If X_velocidad = 20 Or X_velocidad = 40 Then : Gosub Pausa_flash : End If
If X_velocidad = 60 Or X_velocidad = 80 Then Gosub Pausa_flash
If X_velocidad = 100 Or X_velocidad = 120 Then Gosub Pausa_flash

```

```
If X_velocidad = 140 Or X_velocidad = 160 Then Gosub Pausa_flash
```

```
Return
```

```
Pausa_flash:
```

```
Gosub Fila_blanqueo : Gosub Columna_blanqueo : Waitms 150
```

```
Return
```

-
- Subrutina que limpia el contenido de las strings de datos que son utilizadas para presentar un mensaje en el rótulo, para no visualizar ningún carácter no deseado que se encuentre almacenado en la string de salida o se encienda alguna columna no deseada, antes de iniciar la lectura y presentación de una línea de mensaje.
-

```
Limpiar_valores_aux:
```

```
String_salidas_columnas_aux = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

```
String_salidas_columnas = String_salidas_columnas_aux
```

```
Return
```

-
- Subrutina que permite leer el código de las columnas de los caracteres que serán presentados en el rótulo.
-

```
Siguiente_columna
```

```
If Prox_caracter => Fin_mensaje_aux Or Prox_caracter < 1 Then
```

```
If Prox_caracter => Fin_mensaje_aux Then Decr Contador_terminador
```

```
Prox_caracter = 1
```

```
If Dato_fecha = 1 Or Dato_hora = 1 Then : Incr Contador_terminador : Else :
```

```
Contador_terminador = 0 : End If
```

```
If Band_no_incrementar = 1 Then : Decr Contador_terminador :
```

```
Band_no_incrementar = 0 : End If
```

```
If Contador_terminador <> Numero_ciclos Then Band_sigue_fecha = 0
```

```
Direccion = Mensaje - 1 : Direccion = Direccion + Letra_mensaje : Call
```

```
Read_eeprom(1 , Valor) : Tipo_letra = Valor
```



```

If Contador_terminador = Numero_ciclos And Dato_hora = 1 And
Band_sigue_fecha = 0 Then Tipo_letra = 1
If Contador_terminador = Numero_ciclos And Dato_fecha = 1 And
Band_sigue_fecha = 1 Then : Tipo_letra = 2 : Gosub Beeps : End If
If Band_publicidad = 1 Then Tipo_letra = 2: Gosub Tipo_letra_aux:
Band_velocidad_fija = 0 : Band_velocidad_fija2 = 0: End If
If Prox_caracter = 1 Then:Contador_caracter = 1 : Band_fin_mensaje = 0 :
Band_fin_linea = 0 : Band_fin_linea_2 = 0: End If: If Tipo_efecto = 1 And
Contador_terminador <> Numero_ciclos Then Gosub Efecto1
If Tipo_efecto = 2 Or Tipo_efecto = 3 And Contador_terminador <> Numero_ciclos
Then Gosub Efecto2: If Tipo_efecto >= 4 And Tipo_efecto < 14 And
Contador_terminador <> Numero_ciclos Then Gosub Efecto4: If
Contador_terminador = Numero_ciclos And Dato_hora = 1 And Dato_fecha = 0
Then Gosub Salidas_hora      'SOLO HORA
If Contador_terminador = Numero_ciclos And Dato_fecha = 1 And Dato_hora = 0
Then Gosub Salidas_fecha      'SOLO FECHA
If Contador_terminador = Numero_ciclos And Dato_hora = 1 And Dato_fecha = 1
And Band_sigue_fecha = 0 Then Gosub Salidas_
If Contador_terminador = Numero_ciclos And Dato_fecha = 1 And Dato_hora = 1
And Band_sigue_fecha = 1 Then Gosub Salidas_ Gosub
Decodificar_columna_proxima
Return

```

-
- Subrutina que permite leer de la memoria 24LC256 los códigos de caracteres de las líneas de un mensaje.
-

```

Decodificar_columna_proxima:
Posicion_tabla = Lookdown(caracter_byte ,Codigo , Entries) : Posicion_tabla =
Posicion_tabla - 1 : Posicion_tabla = Posicion_tabla / 2
If Tipo_letra = 1 Then : Ini_codigo_caracter = Posicion_tabla * 5 : End If
If Tipo_letra = 2 Then : Ini_codigo_caracter = Posicion_tabla * 6 :
Ini_codigo_caracter = Ini_codigo_caracter + Letra_negrilla : End If

```

```

If Tipo_letra = 3 Then : Ini_codigo_caracter = Posicion_tabla * 6 :
Ini_codigo_caracter = Ini_codigo_caracter + Letra_pequena : End If
If Tipo_letra => 4 Then : Ini_codigo_caracter = Posicion_tabla * 8 :
Ini_codigo_caracter = Ini_codigo_caracter + Letra_doble : End If
Fin_codigo_caracter = Ini_codigo_caracter + Columnas_tipo_letra
Direccion = Ini_codigo_caracter + Contar_columnas : Call Read_eeprom(1 , Valor)
If Valor = 0 Or Direccion = Fin_codigo_caracter Then Valor = 170
Mid(string_columnas , Ubicar_prox_columna , 1 ) = Valor
Return

```

-
- Subrutina que controla la presentación de una línea de mensaje que posea el efecto 1.
-

Efecto1:

```

Band_efec_1_4 = 1 : Direccion_matriz_aux = 0
Direccion = Direccion_aux + Prox_caracter : Call Read_eeprom(1 , Valor)
Caracter_byte = Valor: If Contar_columnas = 0 Then Incr Contador_caracter
If Prox_caracter = Fin_mensaje And Band_fin_mensaje = 0 Then :
Band_fin_mensaje = 1 : Fin_reinicia = 0 : End If
If Caracter_byte = "ß" And Band_fin_mensaje = 0 Then : Band_fin_mensaje = 1 :
Fin_reinicia = 0 : End If: If Band_fin_mensaje = 1 Then
Caracter_byte = " ":Espacios_fin = Columnas_tipo_letra_aux * 4 : Incr Fin_reinicia
If Fin_reinicia = Espacios_fin Then : Band_publicidad = 0
If Dato_hora = 0 And Dato_fecha = 0 Then Contador_terminador = 0
Sal_fecha = 0: Contar_columnas = 0:Band_incr_columnas = 0
If Dato_hora = 1 And Dato_fecha = 1 Then Sal_fecha = 1
If Dato_hora = 1 And Dato_fecha = 0 Then Sal_fecha = 2
If Dato_hora = 0 And Dato_fecha = 1 Then Sal_fecha = 3
Prox_caracter = 0 : Band_leer_efecto = 1 : Band_fin_mensaje = 0 : End If
End If
If Caracter_byte = "j" And Band_fin_linea = 1 Then : Band_fin_linea = 0 : End If
If Caracter_byte = "i" Then Caracter_byte = " "
If Band_fin_linea = 1 And Contar_columnas = Columnas_tipo_letra Then

```

```

If Dato_div = Prox_caracter Then : Dato_div = Prox_caracter / 16 : Else : Dato_div
= Prox_caracter / 16 : Incr Dato_div : End If
Dato_div = Dato_div * 16 : Prox_caracter = Dato_div : Incr Prox_caracter
Contar_columnas = 0 : Band_incr_columnas = 0 : Contador_caracter = 1 :
Band_fin_linea = 0 : Band_leer_efecto = 1: End If
If Contador_caracter = 16 And Contar_columnas = Columnas_tipo_letra And
Band_fin_mensaje = 0 Then: Incr Prox_caracter:Caracter_byte = " "
Contar_columnas = 0 : Band_incr_columnas = 0
Contador_caracter = 0 : Band_leer_efecto = 0
End If
Band_efecto = 0
Return

```

-
- Subrutina que controla la presentación de una línea de mensaje que posea el efecto 2 y el efecto 3.
-

Efecto2:

```

Band_efec_1_4 = 1 : Direccion_matriz_aux = 0
If Band_efecto2 = 1 Then
Band_efecto = 0 : Band_efecto2 = 0 : Band_efecto3 = 0
Direccion = Direccion_aux + Prox_caracter : Call Read_eeprom16(1 , Valor)
Dat_string = "¡" : X_pos_fin_linea = Instr(string_lcd , Dat_string ) : Decr
X_pos_fin_linea
Dat_string = "ß" : X_pos_fin_terminador = Instr(string_lcd , Dat_string ) : Decr
X_pos_fin_terminador: Band_terminador_efec2 = 0: Band_fin_mensaje = 0
If X_pos_fin_terminador < X_pos_fin_linea Then
X_pos_fin_linea = X_pos_fin_terminador: Band_terminador_efec2 = 0: End If
If X_pos_fin_terminador < 16 Then Band_fin_mensaje = 1
Cant_columnas_centrado = X_pos_fin_linea * Columnas_tipo_letra_aux
If Columnas_rotulo => Cant_columnas_centrado Then
Cant_columnas_centrado = Columnas_rotulo - Cant_columnas_centrado :
Cant_columnas_centrado = Cant_columnas_centrado / 2
Prox_caracter_aux2 = Prox_caracter + X_pos_fin_linea: Else

```

```

Cant_columnas_centrado = 0 : Prox_caracter_aux2 = Prox_caracter +
Cant_caracteres_letra: End If: Prox_caracter_aux1 = Prox_caracter
Cant_columnas_centrado_ini = Cant_columnas_centrado +
Columnas_tipo_letra_aux: Espacio_centrado = 0 : Band_centrado = 0
End If
If Band_centrado = 0 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado > Cant_columnas_centrado_ini Then
Prox_caracter = Prox_caracter_aux1: Contador_caracter = 0 : Contar_columnas =
0 : Band_centrado = 1: End If: End If
If Band_centrado = 1 Then
Direccion = Direccion_aux + Prox_caracter : Call Read_eeeprom(1 , Valor) :
Caracter_byte = Valor: If Contar_columnas = 0 Then Incr Contador_caracter
If Caracter_byte = "ß" Or Caracter_byte = "¡" Or Contador_caracter >
Cant_caracteres_letra Then : Espacio_centrado = 0 : Band_centrado = 2
End If: End If
If Band_centrado = 2 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado > Cant_columnas_centrado Then
If Band_fin_mensaje = 0 Then Band_centrado = 3
If Band_fin_mensaje = 1 Or Band_terminador_efec2 = 1 Then Band_centrado = 4
Prox_caracter = Prox_caracter_aux2: Contador_caracter = 0 : Contar_columnas =
0: Band_fin_linea = 0 : Fin_reinicia = 0 : Band_efecto = 1 : End If: End If
If Band_centrado = 3 Then: Caracter_byte = " "
If Contar_columnas => Columnas_tipo_letra Then
Prox_caracter = Prox_caracter_aux2: Dato_div = Prox_caracter / 16 : Incr Dato_div
: Dato_div = Dato_div * 16 : Prox_caracter = Dato_div : Incr Prox_caracter
Contar_columnas = 0 : Band_incr_columnas = 0: Band_leer_efecto = 1 :
Contador_caracter = 1: End If: End If
If Band_centrado = 4 Then : Caracter_byte = " ": Espacios_fin =
Columnas_tipo_letra_aux * 4 : Incr Fin_reinicia
If Fin_reinicia => Espacios_fin Then: Contar_columnas = 0 : Band_incr_columnas
= 0: Contador_caracter = 0 : Prox_caracter = 0 : Band_leer_efecto = 1: End If :
End If
Return

```

-
- Subrutina que controla la presentación de una línea de mensaje que posea el efecto 4 y la string de datos de este efecto es utilizado para el control de los efectos del 5 al 13.
-

Efecto4:

If Band_efecto4 = 1 Then

Band_efecto = 0 : Band_efecto4 = 0 : Band_efecto3 = 0

Direccion = Direccion_aux + Prox_caracter : Call Read_eeeprom16(1 , Valor)

Dat_string = "i" : X_pos_fin_linea = Instr(string_lcd , Dat_string) : Decr

X_pos_fin_linea:Dat_string = "B" : X_pos_fin_terminador = Instr(string_lcd ,

Dat_string) : Decr X_pos_fin_terminador

Band_fin_mensaje = 0 : Band_linea_fija = 0 : GOSUB Asigna_direccion

Band_restar_columnas = 0

If Tipo_letra = 2 Or Tipo_letra = 3 And X_pos_fin_linea = Cant_caracteres_letra

Then Band_restar_columnas = 1

If Tipo_letra = 4 And X_pos_fin_linea = Cant_caracteres_letra Then

Band_restar_columnas = 1

If X_pos_fin_terminador < X_pos_fin_linea Then

X_pos_fin_linea = X_pos_fin_terminador: End If

If X_pos_fin_terminador < 16 Then Band_fin_mensaje = 1

If X_pos_fin_terminador < 16 And Prox_caracter < 17 And Tipo_efecto = 4 Then

Band_linea_fija = 1

Cant_columnas_centrado = X_pos_fin_linea * Columnas_tipo_letra_aux

If Tipo_efecto = 12 Or Tipo_efecto = 13 Or Tipo_efecto = 9 Then

Cant_caracteres_iniciales = Cant_columnas_centrado

If Columnas_rotulo => Cant_columnas_centrado Then

Cant_columnas_centrado = Columnas_rotulo - Cant_columnas_centrado :

Cant_columnas_centrado = Cant_columnas_centrado / 2

Prox_caracter_aux2 = Prox_caracter + X_pos_fin_linea

Else: Cant_columnas_centrado = 0

Prox_caracter_aux2 = Prox_caracter + Cant_caracteres_letra: End If

```

If Tipo_efecto = 12 Or Tipo_efecto = 13 Then: Cant_columnas_finales =
Cant_columnas_centrado: Prox_caracter_aux1 = Prox_caracter
Cant_columnas_centrado_ini = Cant_columnas_centrado +
Columnas_tipo_letra_aux: Espacio_centrado = 0 : Band_centrado = 0
End If
If Band_efec_1_4 = 1 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado => Columnas_rotulo Then
Gosub Asigna_direccion:Prox_caracter = Prox_caracter_aux1 : Contador_caracter
= 1 : Contar_columnas = 0 : Band_centrado = 0: Band_efec_1_4 = 0:End If:Return
End If
If Band_centrado = 0 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado > Cant_columnas_centrado_ini Then
Prox_caracter = Prox_caracter_aux1 : Contador_caracter = 0 : Contar_columnas
= 0 : Band_centrado = 1: End If: End If
If Band_centrado = 1 Then
Direccion = Direccion_aux + Prox_caracter : Call Read_eeeprom(1 , Valor) :
Caracter_byte = Valor
If Contar_columnas = 0 Then Incr Contador_caracter
If Caracter_byte = "ß" Or Caracter_byte = "i" Or Contador_caracter >
Cant_caracteres_letra Then: Espacio_centrado = 0 : Band_centrado = 5
End If: End If
If Band_centrado = 2 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado > Cant_columnas_centrado Then
If Band_fin_mensaje = 0 And Band_linea_fija = 0 Then Band_centrado = 3
If Band_fin_mensaje = 1 And Band_linea_fija = 0 Then Band_centrado = 4
If Band_linea_fija = 1 And Tipo_efecto = 4 Then Band_centrado = 5
Prox_caracter = Prox_caracter_aux2: Contador_caracter = 0 : Contar_columnas
= 0: Band_fin_linea = 0 : Fin_reinicia = 0 : Band_efecto = 1
String_columnas_aux1 = String_columnas
If Tipo_efecto = 9 Or Tipo_efecto = 12 Or Tipo_efecto = 13 Then Band_destello =
1: End If: End If
If Band_centrado = 3 Then
Caracter_byte = " ": Incr Espacio_centrado : Caracter_byte = " "

```

```

If Espacio_centrado => Columnas_rotulo Then
Prox_caracter = Prox_caracter_aux2 : Gosub Fila_blanqueo : Gosub
Columna_blanqueo: Dato_div = Prox_caracter / 16 : Dato_div = Dato_div * 16 :
Decr Prox_caracter_aux2
If Dato_div = Prox_caracter_aux2 Then: Dato_div = Prox_caracter / 16 :Else
Dato_div = Prox_caracter / 16 : Incr Dato_div: End If
Incr Prox_caracter_aux2: Dato_div = Dato_div * 16 :
Prox_caracter = Dato_div : Incr Prox_caracter: Contar_columnas = 0 :
Band_incr_columnas = 0: Band_centrado = 0 : Band_leer_efecto = 1
Gosub Asigna_direccion: Band_efecto4 = 1
End If: End If
If Band_centrado = 4 Then: Caracter_byte = " ":Incr Fin_reinicia
If Fin_reinicia => Columnas_rotulo Then
Gosub Fila_blanqueo : Gosub Columna_blanqueo
Contar_columnas = 0 : Band_incr_columnas = 0
Contador_caracter = 1 : Prox_caracter = 0 : Band_leer_efecto = 1
Gosub Asigna_direccion:Band_efecto4 = 0:End If: End If
If Band_centrado = 5 And Tipo_efecto = 4 Then
String_columnas = String_columnas_aux1:Band_efecto = 1: End If
Return

```

-
- Subrutina que controla la presentación de una línea de mensaje que posea el efecto 9.
-

```

Salidas_efectos_columnas:
If X_velocidad = 1 Then
Velocidad_aux = Velocidad * 1 : Velocidad_aux = Velocidad_aux + 200
String_salidas_columnas_aux = String_columnas
Rotulo_mitad = 25 : Cont_incr_columna = 28 : Cont_abrir_centro = 0
Band_velocidad_fija = 0: End If
If Rotulo_mitad = 0 Then : String_columnas = String_salidas_columnas_aux :
Band_velocidad_fija = 0 : Goto No_decrementar : End If
If Cont_abrir_centro = 0 Then

```

```
String_columnas = String_salidas_columnas_aux Mid(string_columnas , 1 ,
Rotulo_mitad) = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

```
Mid(string_columnas , Cont_incr_columna , 25) =
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

```
End If: Incr Cont_abrir_centro
```

```
If Cont_abrir_centro = 2 Then: Cont_abrir_centro = 0 : Decr Rotulo_mitad : Incr
Cont_incr_columna:End If: No_decrementar:
```

```
If X_velocidad => Velocidad_aux Then
```

```
Band_efecto = 0 : Direccion_matriz_aux = 0 : Posicion_matriz = 0
```

```
Gosub Limpiar_string_columna:Gosub Limpiar_valores_aux : Return: End If
```

```
Return
```

-
- Subrutina que controla la presentación de una línea de mensaje que posea el efecto 12
-

```
Salidas_deletreando:
```

```
If X_velocidad = 1 Then
```

```
Velocidad_aux = Velocidad * 1 : Velocidad_aux = Velocidad_aux + 220
```

```
String_salidas_columnas = String_columnas: Gosub Limpiar_string_columna :
```

```
Gosub Fila_blanqueo : Gosub Columna_blanqueo
```

```
String_salidas_columnas_aux = String_salidas_columnas
```

```
Fin_for_efec13 = Columnas_rotulo - Cant_columnas_finales: Incr Fin_for_efec13
```

```
Num_posicion = Fin_for_efec13 - Cant_caracteres_iniciales
```

```
Cont_efect12 = Cant_caracteres_iniciales / Columnas_tipo_letra_aux : Incr
```

```
Cont_efect12: Num_string = Columnas_tipo_letra_aux : Cont_abrir_centro = 0
```

```
If Tipo_letra = 1 Or Tipo_letra = 2 Or Tipo_letra = 3 Then : Num_posicion =
```

```
Num_posicion + 2 : End If: If Tipo_letra = 4 Then : Num_posicion = Num_posicion
```

```
+ 2 : End If: Band_velocidad_fija = 1: End If
```

```
If Cont_efect12 = 0 Then
```

```
String_columnas = String_salidas_columnas : Band_velocidad_fija = 0
```

```
Band_velocidad_fija2 = 0 : Goto Frase_completa2
```

```
End If
```

```
If Cont_abrir_centro = 0 Then
```



```

String_salidas_columnas_aux = Mid(string_salidas_columnas , Num_posicion ,
Num_string ): Mid(string_columnas , Num_posicion , Num_string ) =
String_salidas_columnas_aux:Decr Cont_efect12
End If: Frase_completa2: Incr Cont_abrir_centro
If Cont_abrir_centro = 20 Then
Cont_abrir_centro = 0 : Num_string = Num_string + Columnas_tipo_letra_aux
End If
If X_velocidad => Velocidad_aux Then : Band_efecto = 0 : Direccion_matriz_aux
= 0 : Posicion_matriz = 0 : Gosub Limpiar_valores_aux : End If
Return

```

-
- Subrutina que controla la presentación de una línea de mensaje que posea el efecto 13.
-

```

Salidas_abriendo_letras:
If X_velocidad = 1 Then
Velocidad_aux = Velocidad * 10 : Velocidad_aux = Velocidad_aux + 140
String_salidas_columnas = String_columnas: Gosub Limpiar_string_columna :
Gosub Fila_blanqueo : Gosub Columna_blanqueo
String_salidas_columnas_aux = String_salidas_columnas
Fin_for_efec13 = Columnas_rotulo - Cant_columnas_finales: Incr Fin_for_efec13
Cont_incr_columna = Fin_for_efec13 - Cant_caracteres_iniciales
Cont_incr_columna = Cont_incr_columna + 2: Num_string = 1 : Cont_abrir_centro
= 0 : Band_velocidad_fija = 1
End If
If Num_string > Columnas_tipo_letra Then
String_columnas = String_salidas_columnas : Band_velocidad_fija = 0
Band_velocidad_fija2 = 0 : Goto Frase_completa
End If
If Cont_abrir_centro = 0 Then
For Cont_efec13 = Cont_incr_columna To Fin_for_efec13
If Cont_incr_columna > Fin_for_efec13 Then Exit For
Num_posicion = Cont_efec13

```

```

String_salidas_columnas_aux = Mid(string_salidas_columnas , Num_posicion ,
Num_string ): Num_posicion = Num_posicion + Columnas_tipo_letra :
Num_posicion = Num_posicion - Num_string: Mid(string_columnas ,
Num_posicion , Num_string ) = String_salidas_columnas_aux: Num_posicion =
Num_posicion - Columnas_tipo_letra : Num_posicion = Num_posicion +
Num_string:Cont_efec13 = Cont_efec13 + Columnas_tipo_letra: Next
Cont_efec13
End If: Frase_completa:
Incr Cont_abrir_centro: If Cont_abrir_centro = 10 Then
If X_velocidad => Velocidad_aux Then : Band_efecto = 0 : Direccion_matriz_aux =
0 : Posicion_matriz = 0 : Gosub Limpiar_valores_aux : Return : End If
Cont_abrir_centro = 0 : Incr Num_string: End If
Return

```

-
- Subrutina que controla la presentación de la hora.
-

```

Salida_hora:
If Band_efecto4 = 1 Then
Band_efecto = 0 : Band_efecto4 = 0 : Band_efecto3 = 0 : Band_sigue_fecha = 0:
Gosub Salida_fecha_hora:Tipo_efecto = 13:
If _hour = 10 Or _hour = 11 Or _hour = 12 Or _hour = 22 Or _hour = 23 Or _hour
= 0 Then: X_pos_fin_linea = 8 : Else : X_pos_fin_linea = 7
End If
X_pos_fin_linea_aux = X_pos_fin_linea + 1: Gosub Asigna_direccion
Cant_columnas_centrado = X_pos_fin_linea * Columnas_tipo_letra_aux
If Tipo_efecto = 13 Or Tipo_efecto = 12 Then Cant_caracteres_iniciales =
Cant_columnas_centrado: Cant_columnas_centrado = Columnas_rotulo -
Cant_columnas_centrado : Cant_columnas_centrado = Cant_columnas_centrado
/ 2: Prox_caracter_aux2 = Prox_caracter + X_pos_fin_linea
If Tipo_efecto = 13 Or Tipo_efecto = 12 Then Cant_columnas_finales =
Cant_columnas_centrado:Prox_caracter_aux1 = Prox_caracter:
Cant_columnas_centrado_ini = Cant_columnas_centrado +
Columnas_tipo_letra_aux: Espacio_centrado = 0 : Band_centrado = 0: End If

```

```

If Band_efec_1_4 = 0 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado => Columnas_rotulo Then: Gosub Asigna_direccion
Prox_caracter = Prox_caracter_aux1:Contador_caracter = 1 : Contar_columnas =
0 : Band_centrado = 0:Band_efec_1_4 = 0: End If: Return: End If
If Band_centrado = 0 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado > Cant_columnas_centrado_ini Then
Prox_caracter = Prox_caracter_aux1: Contador_caracter = 0 : Contar_columnas =
0 : Band_centrado = 1: End If: End If
If Band_centrado = 0 Then
Dato_aux_1 = Mid(hora_aux , Prox_caracter , 1) : Caracter_byte = Dato_aux_1
If Prox_caracter = X_pos_fin_linea_aux Then Caracter_byte = "ß"
If Contar_columnas = 0 Then Incr Contador_caracter
If Caracter_byte = "ß" Or Caracter_byte = "¡" Or Contador_caracter >
Cant_caracteres_letra Then: Espacio_centrado = 0 : Band_centrado = 2
End If: End If
If Band_centrado = 1 Then: Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado > Cant_columnas_centrado Then
If Band_fin_mensaje = 0 Then Band_centrado = 3
Prox_caracter = Prox_caracter_aux2: Contador_caracter = 0 : Contar_columnas =
0: Band_fin_linea = 0 : Fin_reinicia = 0 : Band_efecto = 1: String_columnas_aux1
= String_columnas: If Tipo_efecto = 9 Or Tipo_efecto = 12 Or Tipo_efecto = 13
Then Band_destello = 1: End If: End If
If Band_centrado = 2 Then
Caracter_byte = " ": Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado => Columnas_rotulo Then
Prox_caracter = Prox_caracter_aux2:Gosub Fila_blanqueo : Gosub
Columna_blanqueo: If Dato_fecha = 1 Then : Sal_fecha = 3 : Band_sigue_fecha =
1 : Else : Contador_terminador = 0 : End If: Prox_caracter = 0 : Contar_columnas
= 0 : Band_incr_columnas = 0 : Band_centrado = 0: Band_leer_efecto = 1 :
Band_no_incrementar = 1: Gosub Asigna_direccion: Gosub Limpiar_valores_aux
End If: End If
Return

```

-
- Subrutina que controla la presentación de la fecha.
-

Salida_fecha:

If Band_efecto4 = 1 Then

Band_efecto = 0 : Band_efecto4 = 0 : Band_efecto3 = 0:Gosub Salida_fecha_hora

Tipo_efecto = 7 : X_pos_fin_linea = 6 :Gosub Asigna_direccion

Cant_columnas_centrado = X_pos_fin_linea * Columnas_tipo_letra_aux

If Tipo_efecto = 13 Or Tipo_efecto = 12 Then Cant_caracteres_iniciales =

Cant_columnas_centrado: Cant_columnas_centrado = Columnas_rotulo -

Cant_columnas_centrado : Cant_columnas_centrado = Cant_columnas_centrado

/ 2: Prox_caracter_aux2 = Prox_caracter + X_pos_fin_linea

If Tipo_efecto = 13 Or Tipo_efecto = 12 Then Cant_columnas_finales =

Cant_columnas_centrado: Prox_caracter_aux1 = Prox_caracter

Cant_columnas_centrado_ini = Cant_columnas_centrado +

Columnas_tipo_letra_aux: Espacio_centrado = 0 : Band_centrado = 0:End If

If Band_efec_1_4 = 1 Then: Incr Espacio_centrado : Caracter_byte = " "

If Espacio_centrado => Columnas_rotulo Then

Gosub Asigna_direccion: Prox_caracter = Prox_caracter_aux1

Contador_caracter = 1 : Contar_columnas = 0 : Band_centrado = 0

Band_efec_1_4 = 0:End If: Return: End If

If Band_centrado = 0 Then: Incr Espacio_centrado : Caracter_byte = " "

If Espacio_centrado > Cant_columnas_centrado_ini Then

Prox_caracter = Prox_caracter_aux1: Contador_caracter = 0 : Contar_columnas =

0 : Band_centrado = 1: End If: End If

If Band_centrado = 1 Then

Dato_aux_1 = Mid(fecha_aux , Prox_caracter , 1) : Caracter_byte = Dato_aux_1

If Prox_caracter = 7 Then Caracter_byte = "ß": If Contar_columnas = 0 Then Incr

Contador_caracter: If Caracter_byte = "ß" Or Caracter_byte = "i" Or

Contador_caracter > Cant_caracteres_letra Then: Espacio_centrado = 0 :

Band_centrado = 2:

End If: End If

If Band_centrado = 2 Then: Incr Espacio_centrado : Caracter_byte = " "

```

If Espacio_centrado > Cant_columnas_centrado Then
If Band_fin_mensaje = 0 Then Band_centrado = 3
If Band_fin_mensaje = 0 And Band_linea_fija = 0 Then Band_centrado = 3
Prox_caracter = Prox_caracter_aux2: String_columnas_aux1 = String_columnas
If Tipo_efecto = 9 Or Tipo_efecto = 12 Or Tipo_efecto = 13 Then Band_destello =
1: End If: End If
If Band_centrado = 3 Then
Caracter_byte = " ": Incr Espacio_centrado : Caracter_byte = " "
If Espacio_centrado => Columnas_rotulo Then
Prox_caracter = Prox_caracter_aux2
Gosub Fila_blanqueo : Gosub Columna_blanqueo
Band_no_incrementar = 1:Gosub Asigna_direccion
Gosub Limpiar_valores_aux: End If: End If
Return

```

-
- Subrutina que permite leer la hora y la fecha en el integrado DS1307 para luego presentar esta información en el rótulo.
-

```

Salida_fecha_hora:
Band_hora_fecha = 1
Gosub Getdatetime:Hora_aux = Hora
If _hour > 9 And _hour < 13 Or _hour = 22 Or _hour = 23 Or _hour = 0 Then
Mid(hora_aux , 7 , 2) = Dato_am_pm : Mid(hora_aux , 6 , 1) = " "
Else:Hora_aux = Mid(hora_aux , 2 , 7)
Mid(hora_aux , 6 , 2) = Dato_am_pm : Mid(hora_aux , 5 , 1) = " "
End If
Gosub Actualiza_mes
Mes_lcd = Mid(mes_lcd , 2 , 4)
Mid(fecha_aux , 1 , 3) = Mes_lcd: Dato_day_year = ". "
Mid(fecha_aux , 4 , 2) = Dato_day_year
Dato_day_year = Mid(fecha , 4 , 2) : Mid(fecha_aux , 5 , 2) = Dato_day_year
Return

```

-
- Subrutina que controla el orden de encendido de las filas para los efectos 6, 7, 8, 10,11.
-

Salidas_efectos_filas:

Velocidad_aux = Velocidad * 10 : Velocidad_aux = Velocidad_aux + 120

If Contador_terminador = Numero_ciclos And Dato_hora = 1 Then

Velocidad_aux = Velocidad * 1 : Velocidad_aux = Velocidad_aux + 300: End If

If X_velocidad < 50 Then Band_velocidad_fija = 1

If X_velocidad > 50 Then : Band_velocidad_fija = 0 : Band_velocidad_fija2 = 0 :

End If

If X_velocidad => Velocidad_aux Then : Band_efecto = 0 : Direccion_matriz_aux =

0 : Posicion_matriz = 0 : Gosub Limpiar_string_columna : Gosub

Limpiar_valores_aux : Return : End If

If X_velocidad < 7 Then Posicion_matriz = 0

If X_velocidad > 14 Then Posicion_matriz = 1

If X_velocidad > 21 Then Posicion_matriz = 2

If X_velocidad > 28 Then Posicion_matriz = 3

If X_velocidad > 35 Then Posicion_matriz = 4

If X_velocidad > 42 Then Posicion_matriz = 5

If X_velocidad > 49 Then Posicion_matriz = 6

Return

- Subrutina que determina la posición inicial de los datos a leer dentro de una tabla, donde se encuentran los diferentes códigos de filas para encender las mismas de diferente manera.
-

Asigna_direccion:

If Tipo_efecto = 6 Then Direccion_matriz_aux = 105

If Tipo_efecto = 7 Then Direccion_matriz_aux = 7

If Tipo_efecto = 8 Then Direccion_matriz_aux = 154

If Tipo_efecto = 9 Or Tipo_efecto = 12 Then Direccion_matriz_aux = 0

If Tipo_efecto = 10 Then Direccion_matriz_aux = 56

If Tipo_efecto = 11 Then Direccion_matriz_aux = 203

Return

-
- Subrutinas que encienden las filas en el rótulo.
-

Barrido_filas:

Direccion_matriz = Direccion_matriz_aux : Posicion_matriz_aux = Posicion_matriz

* 7: Direccion_matriz = Direccion_matriz + Posicion_matriz_aux

If Tipo_efecto <= 5 Or Tipo_efecto = 9 Or Tipo_efecto => 12 Or Band_efecto = 0

Then Direccion_matriz = 0: Cont_filas = 0 :For Cont_filas = 6 To 0 Step -1

Dato_fila_matriz = Lookup(direccion_matriz , Matriz_efectos): Stb = 1 : Stb = 0 :

Fila_encendida = Dato_fila_matriz : Gosub Encender_fila: X_fila = Cont_filas :

Gosub Cod: If Band_velocidad_fija = 0 Then Gosub Pausa:Incr Direccion_matriz

Next Cont_filas: Direccion_matriz = 0

Return

Pausa:

For X_pausa = 0 To Cant_pausa: Nop: Nop: Nop: Nop: Nop: Next X_pausa

Return

-
- Subrutinas que envían el texto de los mensajes hacia los buffers que controlan las columnas del rótulo.
-

Cod:

Cont_salida = 10

String_columnas_aux = String_columnas_x(1) : Gosub Sal_serie

If Dato_brillo = 1 Then Gosub Fila_blanqueo

String_columnas_aux = String_columnas_x(2) : Gosub Sal_serie

If Dato_brillo = 2 Then Gosub Fila_blanqueo

String_columnas_aux = String_columnas_x(3) : Gosub Sal_serie

If Dato_brillo = 3 Then Gosub Fila_blanqueo

String_columnas_aux = String_columnas_x(4) : Gosub Sal_serie

If Dato_brillo = 4 Then Gosub Fila_blanqueo

```
String_columnas_aux = String_columnas_x(5) : If X_pos_fin_linea = 7 And
Tipo_letra = 2 Or Tipo_letra = 3 Then Cont_salida = 9 : Gosub Sal_serie
If Dato_brillo = 5 Then Gosub Fila_blanqueo
Return
```

Sal_serie:

```
Reset Watchdog
For For_columna = 1 To Cont_salida
Valor = String_columnas_aux_x(for_columna): If Valor = 170 Then Valor = 0
Toggle Valor.x_fila : Dat = Valor.x_fila : Clk = 0 : Nop : Clk = 1
Next For_columna
Return
```

-
- Subrutina que define la cantidad de caracteres que pueden ser presentados en el rótulo dependiendo del tipo de letra y también define la cantidad máxima de columnas que poseen los caracteres por tipo de letra.

Tipo_letra_aux:

```
If Tipo_letra = 1 Then:Cant_caracteres_letra = 8 : Columnas_tipo_letra = 5 : End If
If Tipo_letra = 2 Then:Cant_caracteres_letra = 7 : Columnas_tipo_letra = 6 : End If
If Tipo_letra = 3 Then:Cant_caracteres_letra = 7 : Columnas_tipo_letra = 6 : End If
If Tipo_letra = 4 Then:Cant_caracteres_letra = 5 : Columnas_tipo_letra = 8 : End If
If Tipo_letra > 4 Then:Cant_caracteres_letra = 7 : Columnas_tipo_letra = 6 : End If
Cant_caracteres_letra_aux = Cant_caracteres_letra + 1
Columnas_tipo_letra_aux = Columnas_tipo_letra + 1
Return
```

-
- Subrutina que verifica el número de tecla presionada, para determinar si todas las teclas se encuentran funcionando.

Test_teclado:

```
If Band_jumper2 = 0 Then : Band_jumper2 = 1 : X_indicativo = 10 : Gosub
Indicativo : Locate 2 , 1 : Lcd " MODO DE PRUEBA " : End If
```


X_tecla = 0 : Fila1 = 0 :

If Columna1 = 0 Then : Gosub Check_1 : If Check_tecla = 0 Then Return : X_tecla = 1 : End If

If Columna2 = 0 Then : Gosub Check_2 : If Check_tecla = 0 Then Return : X_tecla = 2 : End If

If Columna3 = 0 Then : Gosub Check_3 : If Check_tecla = 0 Then Return : X_tecla = 3 : End If

If Columna4 = 0 Then : Gosub Check_4 : If Check_tecla = 0 Then Return : X_tecla = 4 : End If

Fila1 = 1 : Fila2 = 0 :

If Columna1 = 0 Then : Gosub Check_5 : If Check_tecla = 0 Then Return : X_tecla = 5 : End If

If Columna2 = 0 Then : Gosub Check_6 : If Check_tecla = 0 Then Return : X_tecla = 6 : End If

If Columna3 = 0 Then : Gosub Check_7 : If Check_tecla = 0 Then Return : X_tecla = 7 : End If

If Columna4 = 0 Then : Gosub Check_8 : If Check_tecla = 0 Then Return : X_tecla = 8 : End If

Fila2 = 1 : Fila3 = 0 :

If Columna1 = 0 Then : Gosub Check_9 : If Check_tecla = 0 Then Return : X_tecla = 9 : End If

If Columna2 = 0 Then : Gosub Check_10 : If Check_tecla = 0 Then Return : X_tecla = 10 : End If

If Columna3 = 0 Then : Gosub Check_11 : If Check_tecla = 0 Then Return : X_tecla = 11 : End If

If Columna4 = 0 Then : Gosub Check_12 : If Check_tecla = 0 Then Return : X_tecla = 12 : End If

Fila3 = 1 : Fila4 = 0 :

If Columna1 = 0 Then : Gosub Check_13 : If Check_tecla = 0 Then Return : X_tecla = 13 : End If

If Columna2 = 0 Then : Gosub Check_14 : If Check_tecla = 0 Then Return : X_tecla = 14 : End If

```

If Column3 = 0 Then : Gosub Check_15 : If Check_tecla = 0 Then Return :
X_tecla = 15 : End If
If Column4 = 0 Then : Gosub Check_16 : If Check_tecla = 0 Then Return :
X_tecla = 16 : End If
Fila4 = 1
If X_tecla > 0 Then : Locate 1 , 1 : Lcd "  TECLA #    " : Locate 1 , 12 : Lcd
X_tecla : Gosub Beeps : X_tecla = 0 : Gosub Rebotos : End If: Reset Watchdog
Return

```

-
- Subrutina que posee el menú de configuraciones iniciales.
-

Teclado_configurar:

```

If Band_jumper1 = 0 Then : Band_jumper1 = 1 : X_indicativo = 9 : Gosub
Indicativo : End If:Fila3 = 0 :
If Column2 = 0 Then Gosub Trutas: If Column4 = 0 Then Gosub Tpublicidad
Fila3 = 1:Fila4 = 0 :If Column2 = 0 Then Gosub Tmenu_configurar: Fila4 = 1
If Menu_configuracion = 1 Or Menu_configuracion = 3 Or Menu_configuracion = 4
Then    'Fabricacion
Fila1 = 0 :If Column1 = 0 Then Gosub Dias: If Column2 = 0 Then Gosub Meses
If Column3 = 0 Then Gosub Anios: Fila1 = 1:End If
If Menu_configuracion = 2 Then: Fila1 = 0 :
If Column1 = 0 Then Gosub Centenas: If Column2 = 0 Then Gosub Decenas
If Column3 = 0 Then Gosub Unidades
Fila1 = 1: End If
Return

```

-
- Subrutina que permite configurar el número de mensajes que son posibles presentar en el rótulo.
-

Trutas:

```

Gosub Check_10 : If Check_tecla = 0 Then Return : Gosub Beeps
Direccion = Const_cant_mensaje : Call Read_eeprom(1 , Valor) : Cant_mensaje =
Valor:Cls : Lcd "CANTIDAD - RUTAS":If Cant_mensaje > 98 Then Cant_mensaje

```

```

= 0:Cant_mensaje = Cant_mensaje + 25 : If Cant_mensaje > 70 Then
Cant_mensaje = Cant_mensaje + 25
If Cant_mensaje = 100 Then Cant_mensaje = 99:Locate 2 , 1
If Cant_mensaje = 25 Then Lcd " SETEA 25 RUTAS "
If Cant_mensaje = 50 Then Lcd " SETEA 50 RUTAS "
If Cant_mensaje = 99 Then Lcd " SETEA 99 RUTAS "
Direccion = Const_cant_mensaje : Valor = Cant_mensaje : Call Write_eeprom(1 ,
Valor): Gosub Rebotes
Return

```

-
- Subrutina que permite configurar la presencia o no del número telefónico en la publicidad inicial.

```

Tpublicidad:
Gosub Check_12 : If Check_tecla = 0 Then Return : Gosub Beeps
Direccion = Const_publicidad : Call Read_eeprom(1 , Valor) : Publicidad = Valor
Incr Publicidad : If Publicidad > 1 Then Publicidad = 0:Cls : Lcd "TIPO
PUBLICIDAD" : Locate 2 , 1:If Publicidad = 0 Then : Lcd " SIN TELEFONO " :
Else : Lcd " CON TELEFONO " : End If: Direccion = Const_publicidad : Valor =
Publicidad : Call Write_eeprom(1 , Valor):Gosub Rebotes
Return

```

-
- Subrutinas que permiten presentar en el LCD, el menú de configuración que se desea seleccionar, con la respectiva actualización de datos, cuando un dato es modificado.

```

Tmenu_configurar:
Gosub Check_14 : If Check_tecla = 0 Then Return : Gosub Beeps
Gosub Desactiva_teclados : Cls:If Menu_configuracion >= 4 Then
Menu_configuracion = 0 : Incr Menu_configuracion
If Menu_configuracion = 1 Then : Lcd " FABRICACION " : Direccion = 3136 :
Gosub Conver_dia : Gosub Fecha_fab : End If:If Menu_configuracion = 2 Then :
Lcd "NUMERO DE SERIE " : Gosub Conver_serie : Gosub Series : End If

```

```

If Menu_configuracion = 3 Then : Lcd " REVISION # 1 " : Direccion = 3168 :
Gosub Conver_dia : Gosub Fecha_fab : End If:If Menu_configuracion = 4 Then :
Lcd " REVISION # 2 " : Direccion = 3184 : Gosub Conver_dia : Gosub Fecha_fab
: End If:Gosub Rebotos
Return

```

Conver_serie:

```

Direccion = 3153 : Call Read_eeprom(1 , Valor) : Centena = Valor : Centena =
Centena - 48
Direccion = 3154 : Call Read_eeprom(1 , Valor) : Decena = Valor : Decena =
Decena - 48
Direccion = 3155 : Call Read_eeprom(1 , Valor) : Unidad = Valor : Unidad =
Unidad - 48
Return

```

Conver_dia:

```

Call Read_eeprom(1 , Valor) : Dia2 = Valor : Dia2 = Dia2 - 48
Incr Direccion : Call Read_eeprom(1 , Valor) : Dia1 = Valor : Dia1 = Dia1 - 48
Incr Direccion : Incr Direccion : Call Read_eeprom(1 , Valor) : Mes2 = Valor :
Mes2 = Mes2 - 48
Incr Direccion : Call Read_eeprom(1 , Valor) : Mes1 = Valor : Mes1 = Mes1 - 48
Mes = Mes2 * 10 : Mes = Mes + Mes1 : _month = Mes : Gosub Actualiza_mes
Incr Direccion : Incr Direccion : Call Read_eeprom(1 , Valor) : Anio2 = Valor :
Anio2 = Anio2 - 48
Incr Direccion : Call Read_eeprom(1 , Valor) : Anio1 = Valor : Anio1 = Anio1 - 48
Return

```

Series:

```

Locate 2 , 1 : Lcd "          " : Locate 2 , 7 : Lcd Centena : Lcd Decena : Lcd
Unidad
Return

```

Fecha_fab:

```
Locate 2 , 1 : Lcd "          "
```

```
Locate 2 , 4 : Lcd Dia2 : Lcd Dia1 : Lcd Mes_lcd : Lcd Anio2 : Lcd Anio1
```

Return

-
- Las siguientes subrutinas permiten igualar y guardar en la memoria 24LC256, el valor de la fecha de fabricación cuando es modificada.
-

Dias:

```
Gosub Check_1 : If Check_tecla = 0 Then Return : Gosub Beeps
```

```
Incr Dia1 : If Dia1 => 10 Then : Dia1 = 0 : Incr Dia2 : End If
```

```
If Dia2 => 3 And Dia1 => 2 Then : Dia2 = 0 : Dia1 = 1 : End If
```

```
If Menu_configuracion = 1 Then:Direccion = 3136 : Valor = Dia2 + 48 : Call
```

```
Write_eeprom(1 , Valor):Incr Direccion : Valor = Dia1 + 48 : Call Write_eeprom(1 , Valor):End If
```

```
If Menu_configuracion = 3 Then:Direccion = 3168 : Valor = Dia2 + 48 : Call
```

```
Write_eeprom(1 , Valor):Incr Direccion : Valor = Dia1 + 48 : Call Write_eeprom(1 , Valor):End If
```

```
If Menu_configuracion = 4 Then
```

```
Direccion = 3184 : Valor = Dia2 + 48 : Call Write_eeprom(1 , Valor)
```

```
Incr Direccion : Valor = Dia1 + 48 : Call Write_eeprom(1 , Valor)
```

```
End If
```

```
Gosub Fecha_fab:Gosub Rebotes
```

Return

Anios:

```
Gosub Check_3 : If Check_tecla = 0 Then Return : Gosub Beeps
```

```
Incr Anio1 : If Anio1 => 10 Then : Anio1 = 0 : Incr Anio2 : End If
```

```
If Anio2 => 2 And Anio1 => 1 Then : Anio2 = 0 : Anio1 = 9 : End If
```

```
If Menu_configuracion = 1 Then:Direccion = 3142 : Valor = Anio2 + 48 : Call
```

```
Write_eeprom(1 , Valor):Incr Direccion : Valor = Anio1 + 48 : Call Write_eeprom(1 , Valor):End If
```

```
If Menu_configuracion = 3 Then
```

```

Direccion = 3174 : Valor = Anio2 + 48 : Call Write_eeeprom(1 , Valor)
Incr Direccion : Valor = Anio1 + 48 : Call Write_eeeprom(1 , Valor):End If
If Menu_configuracion = 4 Then
Direccion = 3190 : Valor = Anio2 + 48 : Call Write_eeeprom(1 , Valor)
Incr Direccion : Valor = Anio1 + 48 : Call Write_eeeprom(1 , Valor)
End If:Gosub Fecha_fab : Gosub Rebotes
Return

```

Meses:

```

Gosub Check_2 : If Check_tecla = 0 Then Return : Gosub Beeps
Incr Mes1 : If Mes1 => 10 Then : Mes1 = 0 : Mes2 = 1 : End If
If Mes2 => 1 And Mes1 => 3 Then : Mes2 = 0 : Mes1 = 1 : End If
If Menu_configuracion = 1 Then
Direccion = 3139 : Valor = Mes2 + 48 : Call Write_eeeprom(1 , Valor)
Incr Direccion : Valor = Mes1 + 48 : Call Write_eeeprom(1 , Valor):End If
If Menu_configuracion = 3 Then
Direccion = 3171 : Valor = Mes2 + 48 : Call Write_eeeprom(1 , Valor)
Incr Direccion : Valor = Mes1 + 48 : Call Write_eeeprom(1 , Valor):End If
If Menu_configuracion = 4 Then
Direccion = 3187 : Valor = Mes2 + 48 : Call Write_eeeprom(1 , Valor)
Incr Direccion : Valor = Mes1 + 48 : Call Write_eeeprom(1 , Valor)
End If
Incr Mes : If Mes > 12 Then Mes = 1 : _month = Mes : Gosub Actualiza_mes
Gosub Fecha_fab: Gosub Rebotes
Return

```

-
- Las siguientes subrutinas permiten igualar y guardar en la memoria 24LC256, el valor del número de serie cuando es modificado.
-

Centenas:

```

Gosub Check_1 : If Check_tecla = 0 Then Return : Gosub Beeps
Incr Centena : If Centena => 10 Then Centena = 0:Direccion = 3153 : Valor =
Centena + 48 : Call Write_eeeprom(1 , Valor):Gosub Series :

```

Gosub Rebotes

Return

Decenas:

Gosub Check_2 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr Decena : If Decena => 10 Then Decena = 0

Direccion = 3154 : Valor = Decena + 48 : Call Write_eeeprom(1 , Valor)

Gosub Series : Gosub Rebotes

Return

Unidades:

Gosub Check_3 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr Unidad : If Unidad => 10 Then Unidad = 0

Direccion = 3155 : Valor = Unidad + 48 : Call Write_eeeprom(1 , Valor)

Gosub Series : Gosub Rebotes

Return

-
- Subrutina que realiza el Test de Tarjetas.
-

Test_targetas:

If Band_jumper3 = 0 Then

Band_jumper3 = 1 : X_indicativo = 11 : Gosub Indicativo

Cls : Lcd " TARGETAS " : Locate 2 , 1 : Lcd " MODO DE PRUEBA "

Fila_encendida_aux = 0 : Fila_encendida = 2

End If: Dat = 0 : Clk = 0 : Clk = 1 : Stb = 1 : Stb = 0: Stb_fila = 0

For For_fila = 1 To 8

Dat_fila = Fila_encendida.for_fila : Clk_fila = 1 : Clk_fila = 0 :Next For_fila

Stb_fila = 1 : Stb_fila = 0: Incr Fila_encendida_aux

If Fila_encendida_aux = Cant_string Then

If Fila_encendida = 128 Then Fila_encendida = 1: Fila_encendida =

Fila_encendida * 2 : Gosub Columna_blanqueo : Fila_encendida_aux = 0

End If: Toggle Led

Return

-
- Subrutina que permite que la chicharra emita una señal auditiva.
-

```
Beeps: Beep = 0 : Waitms 20 : Beep = 1:Return
```

- Subrutina que chequea el menú principal.
-

```
Menu1:
```

```
Fila3 = 0
```

```
If Columna2 = 0 Then Gosub Tvelocidad
```

```
If Columna4 = 0 Then Gosub Tipo_texto
```

```
Fila3 = 1
```

```
Fila4 = 0
```

```
If Columna1 = 0 Then Gosub Izq_texto
```

```
If Columna2 = 0 Then Gosub Tmenu
```

```
If Columna4 = 0 Then Gosub Der_texto
```

```
Fila4 = 1
```

```
Return
```

- Subrutina que permite cambiar la velocidad de un mensaje.
-

```
Tvelocidad:
```

```
Gosub Check_10 : If Check_tecla = 0 Then Return 'Cambio del menu 1
```

```
Gosub Fila_blanqueo : Gosub Columna_blanqueo : Gosub Beeps
```

```
Gosub Desactiva_teclados : Band_menu1 = 1 : Band_leer = 1
```

```
If Velocidad >= 5 Then Velocidad = 0 : Incr Velocidad:Locate 1 , 1 : Lcd "
```

```
" : Locate 1 , 1 : Waitms 40:Locate 1 , 1 : Lcd " VELOCIDAD ^ " : Locate 1 , 15 :
```

```
Lcd Velocidad:Reset Watchdog : Waitms 960 : Reset Watchdog:Locate 1 , 1 : Lcd
```

```
" : Waitms 40:Direccion = Mensaje - 1 : Direccion = Direccion +
```

```
Velocidad_mensaje : Valor = Velocidad : Call Write_eeprom(1 , Valor)
```

```
Contador_terminador = 0:Gosub Limpiar_valores_salidas:Gosub
```

```
Limpiar_valores_aux:Gosub Rebotes
```

```
Return
```

-
- Subrutina que permite seleccionar el tipo de texto de un mensaje.
-

Tipo_texto:

```
Gosub Check_12 : If Check_tecla = 0 Then Return
Gosub Fila_blanqueo : Gosub Columna_blanqueo : Gosub Beeps
Gosub Desactiva_teclados : Band_menu1 = 1 : Band_leer = 1
If Tipo_letra >= 4 Then Tipo_letra = 0 : Incr Tipo_letra
Locate 1 , 1 : Lcd "          " : Locate 1 , 1 : Waitms 40
If Tipo_letra = 1 Then Lcd "T1 TEXTO NORMAL "
If Tipo_letra = 2 Then Lcd "T2 TEXT NEGRILLA"
If Tipo_letra = 3 Then Lcd "T3 TEXTO PEQUENO"
If Tipo_letra = 4 Then Lcd "T4 TEXTO DOBLE "
Direccion = Mensaje - 1 : Direccion = Direccion + Letra_mensaje : Valor =
Tipo_letra : Call Write_eeprom(1 , Valor)
Reset Watchdog : Waitms 960 : Reset Watchdog
Locate 1 , 1 : Lcd "          " : Waitms 40:Gosub Limpiar_valores_salidas
Gosub Tipo_letra_aux:Gosub Limpiar_valores_aux:Gosub Rebotes
Return
```

- Subrutinas que permiten seleccionar el mensaje a ser presentado.
-

lzcq_texto:

```
Gosub Check_13 : If Check_tecla = 0 Then Return
Gosub Fila_blanqueo : Gosub Columna_blanqueo : Gosub Gosub
Desactiva_teclados : Band_menu1 = 1 : Band_leer = 1
Direccion = Mensaje_memoria : Call Read_eeprom(1 , Valor) : Mensaje = Valor
If Mensaje > 1 Then : Mensaje = Mensaje - 2 : Else : Mensaje = Cant_mensaje - 1
: End If :Mostrar_pantalla = 0 : Conteo_lcd = 0 : Contador_terminador = 0
Gosub Limpiar_valores_aux:Gosub Textos : Gosub Rebotes
Return
```

Der_texto:

```
Gosub Check_16 : If Check_tecla = 0 Then Return
Gosub Fila_blanqueo : Gosub Columna_blanqueo : Gosub Beeps
Gosub Desactiva_teclados : Band_menu1 = 1 : Band_leer = 1
Direccion = Mensaje_memoria : Call Read_eeprom(1 , Valor) : Mensaje = Valor
Gosub Textos : Gosub Limpiar_valores_aux:Mostrar_pantalla = 0 : Conteo_lcd = 0
: Contador_terminador = 0:Gosub Rebotes
Return
```

Textos:

```
Incr Mensaje : If Mensaje > Cant_mensaje Then Mensaje = 1
Direccion = Mensaje_memoria : Valor = Mensaje : Call Write_eeprom(1 , Valor)
Return
```

Sub_textos:Incr Linea : If Linea > Cant_lineas Then : Linea = 1 : End If: Return

-
- Subrutinas que muestran en el LCD la información de las líneas y el número de mensaje.
-

Pos_caracter:

```
Locate 1 , 11 : Lcd "~ ~ "
Locate 1 , 15 : If X_lcd_posicion < 10 Then Lcd "0" : Lcd X_lcd_posicion
Return
```

Visual:

```
Locate 1 , 1 : Lcd "Msj : " : Locate 1 , 5 : If Mensaje < 10 Then Lcd "0" : Lcd
Mensaje:Locate 1 , 8 : If Linea < 10 Then Lcd "0" : Lcd Linea
Return
```

Leer1:

```
If X_menu = 0 Then : Locate 1 , 1 : Lcd " TEXTO ~# : " : End If
If X_menu = 1 Then : Locate 1 , 1 : Lcd "Msj Linea " : End If
Locate 1 , 5 : If Mensaje < 10 Then Lcd "0" : Lcd Mensaje
```

Locate 1 , 15 : If Linea < 10 Then Lcd "0" : Lcd Linea

Leer:

Direccion_aux3 = Mensaje - 1 : Direccion_aux2 = Cant_lineas * 16

Direccion_aux3 = Direccion_aux3 * Direccion_aux2 : Direccion_aux3 =

Direccion_aux3 + Ubicacion_memoria: Direccion = Linea - 1 : Direccion =

Direccion * 16 : Direccion = Direccion + Direccion_aux3

Call Read_eeeprom16(1 , Valor)

Return

-
- Subrutina que guarda en la memoria 24LC256 la información de una línea escrita.
-

Guardar:

Direccion_aux3 = Mensaje - 1 : Direccion_aux2 = Cant_lineas * 16

Direccion_aux3 = Direccion_aux3 * Direccion_aux2 : Direccion_aux3 =

Direccion_aux3 + Ubicacion_memoria: Direccion = Linea - 1 : Direccion =

Direccion * 16 : Direccion = Direccion + Direccion_aux3

Call Write_eeeprom16(1 , String_lcd_x(1)): Gosub Asigna_terminador

Return

-
- Subrutina que verifica si la línea escrita posee el terminador de línea.
-

Asigna_terminador:

Dato_espacios = 0

For Cont_terminador = 1 To 16

Dat_string = Mid(string_lcd , Cont_terminador , 1)

If Dat_string <> " " Then : Dato_espacios = Cont_terminador : End If

If Dat_string = "¡" Or Dat_string = "ß" Then Exit For:

Next Cont_terminador

If Dato_espacios = 0 Or Dat_string = "¡" Or Dat_string = "ß" Then Goto

No_ter_linea

If Dato_espacios < 14 Then: Direccion = Direccion + Dato_espacios

Valor = 161 : Call Write_eeeprom(1 , Valor): Direccion = Direccion - Dato_espacios

End If:No_ter_linea:

Return

-
- Subrutina que contiene los mensajes que son mostrados en el LCD, cuando un menú es seleccionado.
-

Indicativo:

Cursor Off Noblink

Reset Watchdog

For For_indicativo = 1 To 3

Select Case X_indicativo

Case 1 :Cls : Lcd " SALIR DE MENU "

Case 2 :Locate 1 , 1 : Lcd " DIGITE AHORA "

Case 3 :Locate 2 , 1 : Lcd "GUARDA - MEMORIA"

Case 4 :Locate 1 , 1 : Lcd " FIN DE MENSAJE " : Locate 2 , 1 : Lcd "REINICIA SISTEMA"

Case 5 :Locate 1 , 1 : Lcd "INICIO - RUTA " : Locate 1 , 15 : Lcd Mensaje : Locate 2 , 1 : Lcd " DIGITE AHORA "

Case 6 :Locate 1 , 1 :

Lcd "SALIR CONFIGURAR"

Case 7 :Gosub Fila_blanqueo : Locate 1 , 1 : Lcd "SOLO EFECTO 1 "

Case 8 :If For_indicativo = 1 Or For_indicativo = 2 And Band_indicativo = 0 Then
Cls : Lcd " ACTUALIZANDO " : Locate 2 , 1 : Lcd " BRILLO "

If For_indicativo = 2 Then : Band_indicativo = 1 : Decr For_indicativo : End If

End If: If For_indicativo = 2 Or For_indicativo = 3 And Band_indicativo = 1 Then
Cls : Lcd " REINICIANDO " : Locate 2 , 1 : Lcd " SISTEMA " :End If

Case 9 :Cls : Lcd "MODO CONFIGURAR"

Case 10 :Cls : Lcd " TECLADO " : Locate 2 , 1 : Lcd " MODO DE PRUEBA "

Case 11 :Cls : Lcd " TARGETAS " : Locate 2 , 1 : Lcd " MODO DE PRUEBA "

Case 12 :Cls : Lcd " SELECCIONAR " : Locate 2 , 1 : Lcd " LINEA A EDITAR "

Case 13 :If For_indicativo = 1 Or For_indicativo = 2 And Band_indicativo = 0 Then
Cls : Lcd " ACTUALIZANDO " : Locate 2 , 1 : Lcd " RELOJ "

If For_indicativo = 2 Then : Band_indicativo = 1 : Decr For_indicativo : End If

```

End If: If For_indicativo = 2 Or For_indicativo = 3 And Band_indicativo = 1 Then
Cls : Lcd " REINICIANDO " : Locate 2 , 1 : Lcd " SISTEMA " : End If
Case 14 : If For_indicativo = 1 Or For_indicativo = 2 And Band_indicativo = 0 Then
Cls : Lcd " ACTUALIZANDO " : Locate 2 , 1 : Lcd " HORA "
If For_indicativo = 2 Then : Band_indicativo = 1 : Decr For_indicativo : End If: End If
If: If For_indicativo = 2 Or For_indicativo = 3 And Band_indicativo = 1 Then
Cls : Lcd " REINICIANDO " : Locate 2 , 1 : Lcd " SISTEMA " : End If
Case 15 : If For_indicativo = 1 Or For_indicativo = 2 And Band_indicativo = 0 Then
Cls : Lcd " ACTUALIZANDO " : Locate 2 , 1 : Lcd " FECHA " : If
For_indicativo = 2 Then : Band_indicativo = 1 : Decr For_indicativo : End If: End If
If For_indicativo = 2 Or For_indicativo = 3 And Band_indicativo = 1 Then
Cls : Lcd " REINICIANDO " : Locate 2 , 1 : Lcd " SISTEMA " : End If
Case 17: Cls : Lcd " SELECCIONAR " : Locate 2 , 1 : Lcd " LINEA Y EFECTOS"
Case 18: If For_indicativo = 1 Or For_indicativo = 2 And Band_indicativo = 0 Then
Cls : Lcd " ACTUALIZANDO " : Locate 2 , 1 : Lcd " EFECTOS "
If For_indicativo = 2 Then : Band_indicativo = 1 : Decr For_indicativo : End If
End If
If For_indicativo = 2 Or For_indicativo = 3 And Band_indicativo = 1 Then: Cls : Lcd
" REINICIANDO " : Locate 2 , 1 : Lcd " SISTEMA " : End If
End Select
Reset Watchdog : Waitms 500
If X_indicativo = 1 Then : Cls : End If
If X_indicativo = 2 Then : Locate 1 , 1 : Lcd " " : End If
If X_indicativo = 3 Then : Locate 2 , 1 : Lcd " " : End If
If X_indicativo => 4 And X_indicativo <> 7 And X_indicativo <> 9 Then : Cls : End
If: If X_indicativo = 7 Then : Locate 1 , 1 : Lcd " " : End If
Reset Watchdog : Gosub Beeps :
Waitms 100
Next For_indicativo
X_indicativo = 0
Return

```

-
- Subrutina que verifica el número de tecla presionada y el número de veces que se presionó la misma para determinar el carácter a escribir.
-

Decodifica_teclado:

Gosub Beeps

Tecla_men_aux = X_tecla : Avanza_caracter = 0 : Avanza_caracter_aux = 0

If Tecla_men = X_tecla Then: Incr X_pulso_tecla : If X_pulso_tecla > 5 Then

X_pulso_tecla = 0

Else: X_pulso_tecla = 0

If Band_lcd_posicion = 0 Then: If X_lcd_posicion < 16 Then

Incr X_lcd_posicion : End If: End If

End If

Band_lcd_posicion = 0: Cls : Gosub Visual : Locate 1 , 10

For For_pulso_tecla = 0 To 5

X_tecla = Tecla_men_aux - 1 : X_tecla = X_tecla * 6 : X_tecla = X_tecla +

For_pulso_tecla: If Mayus_minus = 1

Then: Tecla_string = Lookupstr(x_tecla , Decodifica_tecla_may): Else:

Tecla_string = Lookupstr(x_tecla , Decodifica_tecla_min)

End If: Lcd Tecla_string

Next For_pulso_tecla

X_tecla = Tecla_men_aux - 1 : X_tecla = X_tecla * 6 : X_tecla = X_tecla +

X_pulso_tecla

If Mayus_minus = 1 Then

Tecla_string = Lookupstr(x_tecla , Decodifica_tecla_may)

Else

Tecla_string = Lookupstr(x_tecla , Decodifica_tecla_min)

End If

Tecla_men = Tecla_men_aux: If Tecla_string = "_" Then Tecla_string = " "

Mid(string_lcd , X_lcd_posicion , 1) = Tecla_string

Signe: Locate 4 , 1 : Lcd String_lcd : Locate 2 , X_lcd_posicion

Return

-
- Subrutina que verifica si las teclas menú y enter han sido presionadas.
-

Menu2:

Fila4 = 0 :

 If Columna2 = 0 Then Gosub Tmenu

 If Columna3 = 0 Then Gosub Tenter

Fila4 = 1

Return

- Subrutina que determina el menú de usuario que se desea seleccionar.
-

Tmenu:

Gosub Check_14 : If Check_tecla = 0 Then Return

Gosub Fila_blanqueo : Gosub Columna_blanqueo : Gosub Beeps

Gosub Desactiva_teclados : Band_menu2 = 1 :

Initlcd : Cls : Lcd "MENU ~ "

If X_menu >= 5 Then X_menu = 0 : Incr X_menu

Locate 1 , 8 : Lcd X_menu : Locate 2 , 1

If X_menu = 1 Then Lcd " EDITAR MENSAJE "

If X_menu = 2 Then Lcd " CONFIG. EFECTOS"

If X_menu = 3 Then Lcd " INSERTAR RELOJ "

If X_menu = 4 Then Lcd " NIVEL DE BRILLO"

If X_menu = 5 Then Lcd " SALIR DE MENU "

Gosub Limpiar_valores_aux: Gosub Rebotes :

Mostrar_pantalla = 0 : Conteo_lcd = 0

Return

- Subrutina que permite ingresar al menú seleccionado.
-

Tenter:

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps

Gosub Desactiva_teclados : Band_velocidad_fija = 0 : Band_velocidad_fija2 = 0


```

If X_menu = 1 Then: Band_submenu1 = 1 : Contador_terminador = 0
X_indicativo = 12 : Gosub Indicativo : Linea = 1 : Gosub Leer1 : Locate 2 , 1 : Lcd
String_lcd: End If
If X_menu = 2 Then
Fin_linea_efectos = Fin_linea_efectos + 15 : Fin_linea_efectos =
Fin_linea_efectos / 16 : Decr Fin_linea_efectos: Band_efectos = 1 : X_indicativo =
17 : Gosub Indicativo : Linea = 0: Gosub Actualizar_linea : Band_mostrar_efec = 1
: Contador_efecto = 0
End If
If X_menu = 3 Then: Band_inst_hora_fecha = 1 : Gosub Ins_hora_fecha: End If
If X_menu = 4 Then:Band_control_brillo = 1 : Gosub Presenta_brillo_lcd: End I
If X_menu = 5 Then: X_menu = 0 : Linea = Cant_lineas
X_indicativo = 1 : Gosub Indicativo : Band_leer = 1 : Band_menu1 = 1
End If
Gosub Limpiar_valores_aux: Gosub Rebotes
Return

```

-
- Subrutinas que verifican el menú de configuraciones iniciales.
-

Submenu1:

```

Fila4 = 0 :
If Columna1 = 0 Then Gosub Izq_subtexto
If Columna2 = 0 Then Gosub Tmenu
If Columna3 = 0 Then Gosub Tenter2
If Columna4 = 0 Then Gosub Der_subtexto
Fila4 = 1
Return

```

Izq_subtexto:

```

Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps
If Linea > 1 Then : Linea = Linea - 2 : Else : Linea = Cant_lineas : End If
Gosub Sub_textos : Gosub Leer1 : Locate 2 , 1 : Lcd String_lcd : Gosub Rebotes
Return

```

Tenter2:

```
Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps
Gosub Desactiva_teclados : Band_submenu2 = 1: X_indicativo = 2 : Gosub
Indicativo: Mayus_minus = 1 : X_lcd_posicion = 0 : Gosub Der_caracter1 :
Gosub Limpiar_valores_aux: Gosub Rebotes
Return
```

Der_subtexto:

```
Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps
Gosub Sub_textos : Gosub Leer1 : Locate 2 , 1 : Lcd String_lcd : Gosub Rebotes
Return
```

-
- Subrutina que verifica la tecla presionada mientras se edita un mensaje.
-

Submenu2:

```
X_tecla = 0
Fila1 = 0 :
If Columna1 = 0 Then Gosub Tecla_1
If Columna2 = 0 Then Gosub Tecla_2
If Columna3 = 0 Then Gosub Tecla_3
If Columna4 = 0 Then Gosub Tecla_4
Fila1 = 1
Fila2 = 0 :
If Columna1 = 0 Then Gosub Tecla_5
If Columna2 = 0 Then Gosub Tecla_6
If Columna3 = 0 Then Gosub Tecla_7
If Columna4 = 0 Then Gosub Tecla_8
Fila2 = 1
Fila3 = 0 :
If Columna1 = 0 Then Gosub Tecla_9
If Columna2 = 0 Then Gosub May_min
If Columna3 = 0 Then Gosub Tecla_11
If Columna4 = 0 Then Gosub Tecla_12
```

```

Fila3 = 1
Fila4 = 0 :
If Columna1 = 0 Then Gosub Izq_caracter
If Columna4 = 0 Then Gosub Der_caracter
If Columna2 = 0 Then Gosub Tmenu
If Columna3 = 0 Then Gosub Tenter3
Fila4 = 1
If X_tecla > 0 Then : X_cont_caracter_0 = 0 : Band_sigue_linea_aux = 0 :
Band_sigue_linea = 0. Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes
End If
If Band_sigue_linea = 1 Then
Incr Band_sigue_linea_aux:If Band_sigue_linea_aux = 120000 Then: Gosub
Beeps: Gosub Visual : Gosub Pos_caracter : Gosub Sigue : Cursor Blink :
Band_sigue_linea = 0 : Band_sigue_linea_aux = 0: End If: End If
If Avanza_caracter = 1 Then : Incr Avanza_caracter_aux
If Avanza_caracter_aux = 40000 Then
If X_lcd_posicion < 16 Then Gosub Der_caracter1
If X_lcd_posicion = 16 Then : Decr X_lcd_posicion : Gosub Der_caracter1 : End If
Avanza_caracter_aux = 0 : Avanza_caracter = 0 : X_pulso_tecla = 6 :
Band_lcd_posicion = 1: End If: End If
Return

```

-
- Subrutinas que verifican si una tecla ha sido presionada al momento de escribir o modificar un mensaje.
-

```

Tecla_1: Gosub Check_1 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If
X_tecla = 1 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes
Return
Tecla_2: Gosub Check_2 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If
X_tecla = 2 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes
Return
Tecla_3: Gosub Check_3 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If
X_tecla = 3 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

```

Return

Tecla_4: Gosub Check_4 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If

X_tecla = 4 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_5: Gosub Check_5 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If

X_tecla = 5 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_6: Gosub Check_6 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If

X_tecla = 6 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_7: Gosub Check_7 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If

X_tecla = 7 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_8: Gosub Check_8 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If

X_tecla = 8 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_9: Gosub Check_9 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End If

X_tecla = 9 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_11: Gosub Check_11 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End

If: X_tecla = 10 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

Tecla_12: Gosub Check_12 : If Check_tecla = 0 Then : X_tecla = 0 : Return : End

If: X_tecla = 11 : Gosub Decodifica_teclado : X_tecla = 0 : Gosub Rebotes

Return

May_min:

Gosub Check_10 : If Check_tecla = 0 Then Return : Gosub Beeps : X_tecla = 0 :

Locate 1 , 11 : If Mayus_minus = 0 Then : Mayus_minus = 1 : Lcd "MAYUS." :

Else : Mayus_minus = 0 : Lcd "minus." : End If: Gosub Rebotes

Return

Izq_caracter:

Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps : X_tecla = 0

```

Cursor Off Noblink: If X_lcd_posicion > 1 Then : Decr X_lcd_posicion :
X_cont_caracter_0 = 0 : End If: If X_lcd_posicion = 1 Then Incr X_cont_caracter_0
If X_cont_caracter_0 = 3 Then: X_cont_caracter_0 = 0: If Linea = 1 Then:
X_indicativo = 5 : Gosub Indicativo : X_lcd_posicion = 1: Else
Decr Linea : Gosub Leer : Lcd String_lcd : X_lcd_posicion = 16: End If: End If
Gosub Visual : Gosub Pos_caracter: Gosub Sigue : Cursor Blink : Gosub Rebotes
Avanza_caracter = 0 : Avanza_caracter_aux = 0 : X_pulso_tecla = 6
Band_sigue_linea = 0 : Band_sigue_linea_aux = 0
Return

```

Der_caracter:

```

Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps : X_tecla = 0
Cursor Off Noblink

```

Der_caracter1:

```

Incr X_lcd_posicion : Band_lcd_posicion = 1:Band_sigue_linea = 0 :
Band_sigue_linea_aux = 0
If X_lcd_posicion < 17 Then: Gosub Visual : Gosub Pos_caracter: Else
Locate 1 , 1 : Lcd String_lcd : Gosub Guardar
X_lcd_posicion = 1 : Gosub Sub_textos
If Linea = 1 Then
X_indicativo = 4 : Gosub Indicativo : X_menu = 0 : Linea = Cant_lineas
Band_leer = 1 : Gosub Desactiva_teclados : Band_menu1 = 1
Gosub Limpiar_valores_aux: Return: End If
Gosub Leer : Band_lcd_posicion = 1: Band_sigue_linea = 1 :
Band_sigue_linea_aux = 0: End If: Gosub Sigue : Cursor Blink : Gosub Rebotes
Avanza_caracter = 0 : Avanza_caracter_aux = 0 : X_pulso_tecla = 6
Return

```

-
- Subrutina que permite salir del menú de edición de un mensaje.
-

Tenter3:

```

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps
Gosub Desactiva_teclados : Band_menu1 = 1 : Band_leer = 1

```

```
X_indicativo = 3 : Gosub Indicativo : Gosub Guardar
Gosub Limpiar_valores_aux:Gosub Rebotes
Return
```

-
- Subrutina que verifica el menú que permite modificar el tipo de efecto de una línea.
-

Menu_efectos:

```
Fila4 = 0 :
If Columna1 = 0 Then Gosub Izq_retrocede_linea_efecto
If Columna2 = 0 Then Gosub Actualizar_efec
If Columna3 = 0 Then Gosub Seleccion_efectos
If Columna4 = 0 Then Gosub Der_avanza_linea_efecto
Fila4 = 1
Prox_caracter_inicial = Linea - 1 : Prox_caracter_inicial = Prox_caracter_inicial *
16 : Incr Prox_caracter_inicial: Prox_caracter_final = Prox_caracter_inicial + 16
Contador_terminador = 0: If Prox_caracter => Prox_caracter_final Then :
Prox_caracter = Prox_caracter_inicial : Band_leer_efecto = 1 :End If
Incr Contador_efecto
If Contador_efecto = 100 Then Gosub Seleccionar_efectos
If Contador_efecto = 200 Then : Contador_efecto = 0 : Gosub
Seleccionar_efectos_aux : End If: Gosub Mostrar_texto
If Prox_caracter = 0 And Tipo_efecto = 1 Then Prox_caracter =
Prox_caracter_inicial - 1: If Prox_caracter = 0 And Tipo_efecto <> 1 Then :
Prox_caracter = Prox_caracter_inicial : Band_leer_efecto = 1 : End If
Return
```

-
- Subrutinas que permiten elegir una línea, para modificar el tipo de efecto de la misma y guardarlo en la memoria 24LC256.
-

Izq_retrocede_linea_efecto:

```
Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps
If Linea > 1 Then : Linea = Linea - 2 : Else : Linea = Fin_linea_efectos : End If
```

```
Gosub Actualizar_linea: Gosub Limpiar_valores_aux: Gosub Rebotes
Return
```

```
Der_avanza_linea_efecto:
```

```
Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps
If X_pos_fin_terminador < 16 Then : Linea = 0 : Prox_caracter = 0 : End If
Gosub Actualizar_linea: Gosub Limpiar_valores_aux: Gosub Rebotes
Return
```

```
Actualizar_linea:
```

```
Cls : Gosub Sub_textos : Gosub Leer
Direccion = Mensaje - 1 : Direccion = Direccion * Cant_lineas : Direccion =
Direccion + Linea : Direccion = Direccion + Efecto_linea
Decr Direccion : Call Read_eeprom(1 , Valor) : Tipo_efecto = Valor
Dat_string = "ß" : X_pos_fin_terminador = Instr(string_lcd , Dat_string ) : Decr
X_pos_fin_terminador: If X_pos_fin_terminador > 100 Then X_pos_fin_terminador
= 20: Gosub Calcula_salida_efec
Contador_efecto = 0 : Gosub Seleccionar_efectos_aux
Contador_caracter = 0 : Contar_columnas = 0 : Band_incr_columnas = 0 :
Band_fin_linea = 0 : Fin_reinicia = 0: Band_efecto = 1 : Band_efecto = 0 :
Direccion_matriz_aux = 0 : Posicion_matriz = 0
Prox_caracter_inicial = Linea - 1 : Prox_caracter_inicial = Prox_caracter_inicial *
16 : Incr Prox_caracter_inicial
Prox_caracter = Prox_caracter_inicial: Prox_caracter_aux2 = Prox_caracter_inicial
Gosub Limpiar_string_columna : Gosub Fila_blanqueo : Gosub
Columna_blanqueo: String_salidas_columnas_aux = String_salidas_columnas
Gosub Asigna_direccion
Return
```

```
Calcula_salida_efec:
```

```
Direccion = Linea - 1 : Direccion = Direccion * 16 : Direccion = Direccion_aux +
Direccion: Incr Direccion : Call Read_eeprom16(1 , Valor)
```

```

Dat_string_aux = "i" : Ter_fin_linea = Instr(string_lcd , Dat_string_aux ) : If
Ter_fin_linea > 0 Then Decr Ter_fin_linea
Dat_string_aux = "ß" : Ter_fin_mensaje = Instr(string_lcd , Dat_string_aux ) : If
Ter_fin_mensaje > 0 Then Decr Ter_fin_mensaje
If Ter_fin_linea > Cant_caracteres_letra Then
Tipo_efecto = 1 : Contador_caracter = 0 : Contar_columnas = 0: End If
If Ter_fin_mensaje > Cant_caracteres_letra Then: Tipo_efecto = 1 :
Contador_caracter = 0 : Contar_columnas = 0: End If
If Ter_fin_mensaje = 0 And Ter_fin_linea = 0 Then
Tipo_efecto = 1 : Contador_caracter = 0 : Contar_columnas = 0: End If
Return

```

-
- Subrutinas que presentan en el LCD los efectos que pueden ser elegidos.
-

Seleccionar_efectos:

Cls

Select Case Tipo_efecto

Case 1 : Lcd " TEXTO CORRIDO "

Case 2 : Lcd " AVANZA DETIENE "

Case 3 : Lcd " AVANZA FLASH "

Case 4 : Lcd " PRESENTA TEXTO "

Case 5 : Lcd " DESTELLANTE "

Case 6 :Lcd Chr(3) : Lcd " ABRIR TEXTO " : Lcd Chr(3)

Case 7 :Lcd Chr(4) : Lcd " ABRIR TEXTO " : Lcd Chr(4)

Case 8 :Lcd Chr(5) : Lcd " ABRIR TEXTO " : Lcd Chr(5)

Case 9 :Lcd Chr(6) : Lcd Chr(7) : Lcd "ABRIR TEXTO" : Lcd Chr(6) : Lcd Chr(7)

Case 10 :Lcd Chr(4) : Lcd "ASCENDER TEXTO" : Lcd Chr(4)

Case 11 :Lcd Chr(5) : Lcd "DESCENDER TEXT" : Lcd Chr(5)

Case 12 :Lcd Chr(7) : Lcd " DELETREANDO " : Lcd Chr(7) : Lcd Chr(7)

Case 13 : Lcd "ABRIENDO LETRAS "

End Select

Gosub Asigna_direccion : Gosub Leer: Locate 2 , 1 : Lcd String_lcd

Return

Seleccionar_efectos_aux:

Gosub Leer

Cls : Lcd "Efecto " : Locate 1 , 8 : Lcd Tipo_efecto

Locate 1 , 12 : If Mensaje < 10 Then Lcd "0" : Lcd Mensaje : Lcd ":",

Locate 1 , 15 : If Linea < 10 Then Lcd "0" : Lcd Linea

Locate 2 , 1 : Lcd String_lcd

Return

-
- Subrutina que permite salir del menú de selección de efectos.
-

Actualizar_efec:

Gosub Check_14 : If Check_tecla = 0 Then Return : Gosub Beeps

Gosub Desactiva_teclados: X_menu = 0 : Linea = Cant_lineas :

Band_mostrar_efec = 0: Band_indicativo = 0 : X_indicativo = 18 : Gosub Indicativo

: Band_leer = 1 : Band_menu1 = 1: Gosub Limpiar_valores_salidas:Gosub

Limpiar_valores_aux

Return

-
- Subrutina que permite seleccionar los efectos.
-

Seleccion_efectos:

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps :

If Ter_fin_linea > Cant_caracteres_letra Then : Tipo_efecto = 1 : X_indicativo = 7 :

Gosub Indicativo : Goto Solo_efecto_1 : End If

If Ter_fin_mensaje > Cant_caracteres_letra Then : Tipo_efecto = 1 : X_indicativo

= 7 : Gosub Indicativo : Goto Solo_efecto_1 : End If

If Ter_fin_mensaje = 0 And Ter_fin_linea = 0 Then : Tipo_efecto = 1 : X_indicativo

= 7 : Gosub Indicativo : Goto Solo_efecto_1 : End If

If Tipo_efecto >= 13 Then Tipo_efecto = 0 : Incr Tipo_efecto : Locate 1 , 1

Contador_efecto = 0 : Gosub Seleccionar_efectos_aux

Direccion = Mensaje - 1 : Direccion = Direccion * Cant_lineas

Direccion = Direccion + Linea : Direccion = Direccion + Efecto_linea: Decr

Direccion: Valor = Tipo_efecto : Call Write_eeprom(1 , Valor)

```

Contador_caracter = 0 : Contar_columnas = 0 : Band_fin_linea = 0 : Fin_reinicia =
0 : Band_efecto = 1: Band_efecto = 0 : Direccion_matriz_aux = 0 : Posicion_matriz
= 0

```

```

Solo_efecto_1:

```

```

Gosub Asigna_direccion : Gosub Limpiar_string_columna : Gosub Fila_blanqueo :

```

```

Gosub Columna_blanqueo: Prox_caracter_inicial = Linea - 1 :

```

```

Prox_caracter_inicial = Prox_caracter_inicial * 16 : Incr Prox_caracter_inicial

```

```

Prox_caracter = Prox_caracter_inicial: Contador_caracter = 0 : Contar_columnas

```

```

= 0 : Band_fin_linea = 0 : Fin_reinicia = 0 : Band_efecto = 1: Band_efecto = 0 :

```

```

Direccion_matriz_aux = 0 : Posicion_matriz = 0 :Gosub Rebotes

```

```

Return

```

-
- Subrutinas del Menú Insertar Reloj.
-

```

Menu_inst_hora_fecha:

```

```

Fila4 = 0 :

```

```

If Columna1 = 0 Then Gosub Izq_ins_hora

```

```

If Columna3 = 0 Then Gosub Acepta_hora_fecha

```

```

If Columna4 = 0 Then Gosub Der_ins_fecha

```

```

Fila4 = 1

```

```

Return

```

```

Izq_ins_hora:

```

```

Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps

```

```

Incr Dato_hora: If Dato_hora > 1 Then Dato_hora = 0: Gosub Ins_hora_fecha

```

```

Gosub Rebotes

```

```

Return

```

```

Acepta_hora_fecha:

```

```

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps

```

```

Gosub Desactiva_teclados:Gosub Limpiar_valores_aux

```

```

If Dato_hora = 1 Or Dato_fecha = 1 Then: If Dato_hora = 1 And Dato_fecha = 1

```

```

Then : Band_fecha_hour = 1 : Else : Band_fecha_hour = 0 : End If

```

```

Band_ciclos_hora_fecha = 1 : Gosub Ciclos: Else. X_menu = 0 : Linea =
Cant_lineas. Band_indicativo = 0 : X_indicativo = 13 : Gosub Indicativo :
Band_leer = 1 : Band_menu1 = 1: Gosub Limpiar_valores_salidas: End If
Gosub Rebotes
Return

```

```

Der_ins_fecha:
Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps
Incr Dato_fecha: If Dato_fecha > 1 Then Dato_fecha = 0
Gosub Ins_hora_fecha: Gosub Rebotes
Return

```

```

Ins_hora_fecha:
Cls : Home U: Lcd Chr(0) : Lcd "Insert Hora  "
Locate 1 , 15 : If Dato_hora = 1 Then : Lcd "SI" : Else : Lcd "NO" : End If
Home L: Lcd Chr(1) : Lcd "Insert Fecha  "
Locate 2 , 15 : If Dato_fecha = 1 Then : Lcd "SI" : Else : Lcd "NO" : End If
Direccion = Mensaje - 1 : Direccion = Lectura_hora + Direccion : Valor =
Dato_hora : Call Write_eeprom(1 , Valor):
Direccion = Mensaje - 1 : Direccion = Lectura_fecha + Direccion : Valor =
Dato_fecha : Call Write_eeprom(1 , Valor)
Return

```

-
- Subrutinas que permiten elegir el número de ciclos, para que un mensaje sea presentado antes de que la hora o la fecha sea visualizada.
-

```

Menu_insertar_ciclos:
Fila4 = 0 :
If Columna4 = 0 Then Gosub Der_aumenta_num_ciclos
If Columna3 = 0 Then Gosub Acepta_num_ciclos
If Columna1 = 0 Then Gosub Izq_disminuye_num_ciclos
Fila4 = 1
Return

```

Izq_disminuye_num_ciclos:

```
Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps
Numero_ciclos = Numero_ciclos - 2 : Gosub Ciclos: Gosub Rebotes
Return
```

Acepta_num_ciclos:

```
Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps
Gosub Desactiva_teclados: Gosub Limpiar_valores_aux
If Dato_hora = 1 Then: Band_reloj = 1 : Gosub Actualiza_hora: Else
Band_fecha = 1 : Gosub Actualiza_fecha: End If: Gosub Rebotes
Return
```

Der_aumenta_num_ciclos:

```
Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps
Numero_ciclos = Numero_ciclos + 2 : Gosub Ciclos: Gosub Rebotes
Return
```

Ciclos:

```
If Numero_ciclos < 2 Then Numero_ciclos = 20
If Numero_ciclos > 20 Then Numero_ciclos = 2
Direccion = Mensaje - 1 : Direccion = Lectura_num_ciclos + Direccion : Valor =
Numero_ciclos : Call Write_eeprom(1 , Valor): Lcd " INSERTAR RELOJ "
Locate 2 , 1 : Lcd "Cada " : Lcd Chr(0) : Locate 2 , 8 : If Numero_ciclos < 10 Then
Lcd "0": Lcd Numero_ciclos : Locate 2 , 11 : Lcd Chr(1) : Locate 2 , 13 : Lcd "Msj."
Return
```

-
- Subrutinas que permiten igualar la hora.
-

Menu_igualar_reloj:

```
Fila4 = 0 :
If Columna1 = 0 Then Gosub Izq_igualar_hora
If Columna3 = 0 Then Gosub Salir_igualar_hora
If Columna4 = 0 Then Gosub Der_igualar_minuto
```

Fila4 = 1

Return

Izq_igualar_hora:

Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr_hour : If _hour > 23 Then _hour = 0

Gosub Actualiza_hora: Gosub Rebotes

Return

Salir_igualar_hora:

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps

Gosub Desactiva_teclados: Gosub Limpiar_valores_aux

If Dato_fecha = 1 Then: Band_fecha = 1 :

Gosub Actualiza_fecha: Else

X_menu = 0 : Linea = Cant_lineas: Band_indicativo = 0 : X_indicativo = 14 :

Gosub Indicativo : Band_leer = 1 : Band_menu1 = 1:

Gosub Limpiar_valores_salidas

End If: Gosub Rebotes

Return

Der_igualar_minuto:

Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr_min : If _min = 60 Then _min = 0 : Gosub Actualiza_hora: Gosub Rebotes

Return

Actualiza_hora:

Cls : _sec = 0

Band_hora_fecha = 1 : Gosub Settime: Band_hora_fecha = 1 : Gosub

Getdatetime

Lcd " HORA ACTUAL ": Locate 2 , 3 : Lcd Chr(0) : Locate 2 , 5 : Lcd Hora :

Locate 2 , 10 : Lcd " " : Lcd Chr(1) : Lcd " " : Lcd Dato_am_pm: Cursor Off Noblink

Return

-
- Subrutinas que permiten igualar la fecha.
-

Menu_igualar_fecha:

Fila3 = 0 : If Columna2 = 0 Then Gosub Igualar_year: Fila3 = 1

Fila4 = 0 :

If Columna1 = 0 Then Gosub Izq_igualar_dia

If Columna3 = 0 Then Gosub Salir_igualar_fecha

If Columna4 = 0 Then Gosub Der_igualar_mes

Fila4 = 1

Return

Igualar_year:

Gosub Check_10 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr_year : If _year > 20 Then _year = 9 : Gosub Actualiza_fecha: Gosub Rebotes

Return

Izq_igualar_dia:

Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr_day : If _day > 31 Then _day = 1 : Gosub Actualiza_fecha : Gosub Rebotes

Return

Salir_igualar_fecha:

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps

X_menu = 0 : Linea = Cant_lineas

If Band_fecha_hour = 0 Then : Band_indicativo = 0 : X_indicativo = 15 : End If

If Band_fecha_hour = 1 Then : Band_indicativo = 0 : X_indicativo = 13 : End If

Gosub Indicativo : Band_leer = 1 : Band_menu1 = 1

Gosub Limpiar_valores_salidas : Gosub Rebotes

Return

Der_igualar_mes:

Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr_month : If _month > 12 Then _month = 1 : Gosub Actualiza_fecha

Gosub Rebotes

Return

- Subrutina que guarda el valor de la fecha cuando esta es igualada.
-

Actualiza_fecha:

Cls: Band_hora_fecha = 1 : Gosub Setdate

Band_hora_fecha = 1 : Gosub Getdatetime

Lcd " FECHA ACTUAL ": Gosub Actualiza_mes

Locate 2 , 1 : If _day < 10 Then Lcd "0" : Lcd _day : Lcd Mes_lcd : If _year < 10

Then Lcd "0" : Lcd _year: Locate 2 , 12 : Lcd Chr(0) : Lcd " " : Lcd Chr(1) : Lcd " " :

Lcd Chr(2): Cursor Off Noblink

Return

- Subrutina que permite seleccionar el tipo de dato correspondiente al mes que será visualizado en el LCD cuando la fecha es igualada.
-

Actualiza_mes:

Select Case _month

Case 1 : Mes_lcd = "-Ene-"

Case 2 : Mes_lcd = "-Feb-"

Case 3 : Mes_lcd = "-Mar-"

Case 4 : Mes_lcd = "-Abr-"

Case 5 : Mes_lcd = "-May-"

Case 6 : Mes_lcd = "-Jun-"

Case 7 : Mes_lcd = "-Jul-"

Case 8 : Mes_lcd = "-Ago-"

Case 9 : Mes_lcd = "-Sep-"

Case 10 : Mes_lcd = "-Oct-"

Case 11 : Mes_lcd = "-Nov-"

Case 12 : Mes_lcd = "-Dic-"

End Select

Return

-
- Subrutina que limpia los buffers 74LS595 del rótulo.
-

Limpiar_valores_salidas:

Gosub Limpiar_string_columna

Gosub Fila_blanqueo : Gosub Columna_blanqueo: Contador_terminador = 0

Return

- Subrutinas del Menú Nivel de Brillo.
-

Menu_control_brillo:

Fila4 = 0 :

If Columna1 = 0 Then Gosub Izq_disminuir_brillo

If Columna3 = 0 Then Gosub Salir_control_brillo

If Columna4 = 0 Then Gosub Der_aumentar_brillo

Fila4 = 1: Gosub Mostrar_texto

Return

Izq_disminuir_brillo:

Gosub Check_13 : If Check_tecla = 0 Then Return : Gosub Beeps

Decr Dato_brillo : If Dato_brillo < 1 Then Dato_brillo = 5

Gosub Presenta_brillo_lcd : Gosub Rebotes

Return

Salir_control_brillo:

Gosub Check_15 : If Check_tecla = 0 Then Return : Gosub Beeps

X_menu = 0 : Linea = Cant_lineas: Band_indicativo = 0 : X_indicativo = 8 : Gosub

Indicativo : Band_leer = 1 : Band_menu1 = 1: Gosub Limpiar_string_columna :

Gosub Fila_blanqueo : Gosub Columna_blanqueo

String_salidas_columnas_aux = String_salidas_columnas : Contador_terminador

= 0 : Prox_caracter = 0 : Contar_columnas = 0 : Gosub Rebotes

Return

Der_aumentar_brillo:

Gosub Check_16 : If Check_tecla = 0 Then Return : Gosub Beeps

Incr Dato_brillo : If Dato_brillo > 5 Then Dato_brillo = 1

Gosub Presenta_brillo_lcd: Gosub Rebotos

Return

Presenta_brillo_lcd:

Dato_brillo_aux = Dato_brillo * 20 : Cls : Lcd "NIVEL DE BRILLO "

Locate 2 , 5 : Lcd Chr(0) : Lcd " % " : Lcd Chr(1) :

Locate 2 , 7 : Lcd Dato_brillo_aux : Lcd " %" : Direccion = Mensaje -

1 : Direccion = Lectura_brillo + Direccion : Valor = Dato_brillo : Call

Write_eeeprom(1 , Valor) : Cursor Off Noblink

Return

-
- Subrutinas que verifican si una tecla ha sido realmente presionada.
-

Check_1: Gosub Verifica_tecla : Fila1 = 0 : Waitms 5 : If Columna1 = 0 Then

Check_tecla = 1 : Fila1 = 1

Return

Check_2: Gosub Verifica_tecla : Fila1 = 0 : Waitms 5 : If Columna2 = 0 Then

Check_tecla = 1 : Fila1 = 1

Return

Check_3: Gosub Verifica_tecla : Fila1 = 0 : Waitms 5 : If Columna3 = 0 Then

Check_tecla = 1 : Fila1 = 1

Return

Check_4: Gosub Verifica_tecla : Fila1 = 0 : Waitms 5 : If Columna4 = 0 Then

Check_tecla = 1 : Fila1 = 1

Return

Check_5: Gosub Verifica_tecla : Fila2 = 0 : Waitms 5 : If Columna1 = 0 Then

Check_tecla = 1 : Fila2 = 1

Return

Check_6: Gosub Verifica_tecla : Fila2 = 0 : Waitms 5 : If Columna2 = 0 Then

Check_tecla = 1 : Fila2 = 1

Return

Check_7: Gosub Verifica_tecla : Fila2 = 0 : Waitms 5 : If Columna3 = 0 Then
Check_tecla = 1 : Fila2 = 1

Return

Check_8: Gosub Verifica_tecla : Fila2 = 0 : Waitms 5 : If Columna4 = 0 Then
Check_tecla = 1 : Fila2 = 1

Return

Check_9: Gosub Verifica_tecla : Fila3 = 0 : Waitms 5 : If Columna1 = 0 Then
Check_tecla = 1 : Fila3 = 1

Return

Check_10: Gosub Verifica_tecla : Fila3 = 0 : Waitms 5 : If Columna2 = 0 Then
Check_tecla = 1 : Fila3 = 1

Return

Check_11: Gosub Verifica_tecla : Fila3 = 0 : Waitms 5 : If Columna3 = 0 Then
Check_tecla = 1 : Fila4 = 1

Return

Check_12: Gosub Verifica_tecla : Fila3 = 0 : Waitms 5 : If Columna4 = 0 Then
Check_tecla = 1 : Fila3 = 1

Return

Check_13: Gosub Verifica_tecla : Fila4 = 0 : Waitms 5 : If Columna1 = 0 Then
Check_tecla = 1 : Fila4 = 1

Return

Check_14: Gosub Verifica_tecla : Fila4 = 0 : Waitms 5 : If Columna2 = 0 Then
Check_tecla = 1 : Fila4 = 1

Return

Check_15: Gosub Verifica_tecla : Fila4 = 0 : Waitms 5 : If Columna3 = 0 Then
Check_tecla = 1 : Fila4 = 1

Return

Check_16: Gosub Verifica_tecla : Fila4 = 0 : Waitms 5 : If Columna4 = 0 Then
Check_tecla = 1 : Fila4 = 1

Return

Verifica_tecla:

Check_tecla = 0 : Fila1 = 1 : Fila2 = 1 : Fila3 = 1 : Fila4 = 1 : Waitms 1

Return

Rebotes:

Fila1 = 1 : Fila2 = 1 : Fila3 = 1 : Fila4 = 1 : Waitms 250

Return

-
- Subrutina que busca el terminador final en un mensaje para determinar el número de caracteres que posee el mensaje.
-

Buscar_fin:

Direccion_aux3 = Mensaje - 1

Direccion_aux2 = Cant_lineas * 16

Direccion_aux3 = Direccion_aux3 * Direccion_aux2 : Direccion = Direccion_aux3
+ Ubicacion_memoria : Fin_mensaje = Cant_caracteres_mensaje: Band_vacio = 1
: Men_vacia = 1 : Cont_terminador = 0

For X_for = 0 To Cant_caracteres_mensaje

Call Read_eeprom(1 , Valor) : Incr Direccion

If Valor <> 32 Then

If Band_vacio = 1 Then : Band_vacio = 0 : Men_vacia = 0 : End If: End If

If Band_publicidad = 0 Then:

If Men_vacia = 1 Then Goto Solo_publicidad

If Valor = 32 Then : Else : Cont_terminador = 0 : If Valor = 161 Then

Cont_terminador = 1 : End If

If Cont_terminador => 16 Then : Direccion = Direccion - 16

Valor = 223 : Call Write_eeprom(1 , Valor):X_for = X_for - 15

End If

Incr Cont_terminador:Solo_publicidad: End If

If Valor = 223 Then : Fin_mensaje = X_for + 15 : Fin_linea_efectos = X_for : Exit

For : End If

Next X_for

Dato_div = Fin_mensaje / 16

Fin_mensaje = Dato_div * 16 : Incr Fin_mensaje

Fin_mensaje_aux = Fin_mensaje + 4

Return

-
- Subrutina que escribe en el LCD, el tipo de letra, la velocidad, el dato de brillo cuando un mensaje es mostrado en el rótulo. También lee las características que posee el mensaje.
-

Lcd_linea:

```

Direccion = Mensaje - 1 : Direccion = Direccion * 16 : Direccion = Direccion *
Cant_lineas : Direccion = Direccion + Ubicacion_memoria
Call Read_eeprom(1 , Valor)
Direccion = Mensaje - 1 : Direccion = Direccion + Letra_mensaje : Call
Read_eeprom(1 , Valor) : Tipo_letra = Valor : If Tipo_letra > 4 Then : Tipo_letra =
2 : Valor = Tipo_letra : Call Write_eeprom(1 , Valor) : End If
If Band_publicidad = 1 Then Tipo_letra = 2
Direccion = Mensaje - 1 : Direccion = Direccion + Velocidad_mensaje : Call
Read_eeprom(1 , Valor) : Velocidad = Valor : If Velocidad > 5 Then : Velocidad =
2 : Valor = Velocidad : Call Write_eeprom(1 , Valor) : End If
Direccion = Mensaje - 1 : Direccion = Direccion * Cant_lineas : Direccion =
Direccion + Linea : Direccion = Direccion + Efecto_linea : Decr Direccion
Call Read_eeprom(1 , Valor) : Tipo_efecto = Valor
If Tipo_efecto > 13 Then : Tipo_efecto = 1 : Valor = Tipo_efecto : Call
Write_eeprom(1 , Valor) : End If
Direccion = Mensaje - 1 : Direccion = Lectura_hora + Direccion : Call
Read_eeprom(1 , Valor) : Dato_hora = Valor
Direccion = Mensaje - 1 : Direccion = Lectura_fecha + Direccion : Call
Read_eeprom(1 , Valor) : Dato_fecha = Valor
If Dato_hora > 1 Then Dato_hora = 1 : If Dato_fecha > 1 Then Dato_fecha = 1
Direccion = Mensaje - 1 : Direccion = Lectura_num_ciclos + Direccion : Call
Read_eeprom(1 , Valor) : Numero_ciclos = Valor
If Numero_ciclos < 2 Or Numero_ciclos > 20 Then Numero_ciclos = 2
Direccion = Mensaje - 1 : Direccion = Lectura_brillo + Direccion : Call
Read_eeprom(1 , Valor) : Dato_brillo = Valor : If Dato_brillo < 1 Or Dato_brillo > 5
Then Dato_brillo = 2 : Cls : Lcd "Msj      " : Locate 1 , 5 : Lcd Mensaje
If Band_publicidad = 1 Then : Cls : Lcd " CTE  "

```

```
Tipo_letra = 2 : Velocidad = 2 : Dato_brillo = 5 : End If
Locate 1 , 9 : Lcd "T" : Lcd Tipo_letra : Locate 1 , 12 : Lcd "V" : Lcd Velocidad
Locate 1 , 15 : Lcd "B" : Lcd Dato_brillo : Locate 2 , 1 : Lcd String_lcd
Return
```

-
- Subrutina que lee la hora y la fecha del integrado DS1307.
-

Getdatetime:

```
If Band_hora_fecha = 1 Then
```

```
I2cstart
```

```
I2cwbyte Ds1307w
```

```
I2cwbyte 0
```

```
I2cstart
```

```
I2cwbyte Ds1307r
```

```
I2crbyte _sec , Ack
```

```
I2crbyte _min , Ack
```

```
I2crbyte _hour , Ack
```

```
I2crbyte _weekday , Ack
```

```
I2crbyte _day , Ack
```

```
I2crbyte _month , Ack
```

```
I2crbyte _year , Nack
```

```
I2cstop
```

```
_sec = Makedec(_sec) : _min = Makedec(_min) : _hour = Makedec(_hour) : _day
= Makedec(_day) : _month = Makedec(_month) : _year = Makedec(_year)
```

```
_hour_aux = _hour: Dato_am_pm = "AM"
```

```
If _hour > 11 Then Dato_am_pm = "PM"
```

```
If _hour = 0 Then _hour = 12
```

```
If _hour > 12 Then _hour = _hour - 12
```

```
Band_hora_fecha = 0 : Hora = Time$ : Fecha = Date$ : _hour = _hour_aux
```

```
End If
```

```
Return
```

-
- Subrutina que guarda la fecha en el integrado DS1307.
-

Setdate:

If Band_hora_fecha = 1 Then

_day = Makebcd(_day) : _month = Makebcd(_month) : _year = Makebcd(_year)

I2cstart

I2cwbyte Ds1307w

I2cwbyte 4

I2cwbyte _day

I2cwbyte _month

I2cwbyte _year

I2cstop

Band_hora_fecha = 0

End If

Return

- Subrutina que guarda la hora en el integrado DS1307.
-

Settime:

If Band_hora_fecha = 1 Then

_sec = Makebcd(_sec) : _min = Makebcd(_min) : _hour = Makebcd(_hour)

I2cstart

I2cwbyte Ds1307w

I2cwbyte 0

I2cwbyte _sec

I2cwbyte _min

I2cwbyte _hour

I2cstop

Band_hora_fecha = 0

End If

Return

-
- Subrutina que escribe los caracteres en la memoria 24LC256 de uno en uno.
-

```
Sub Write_eeprom(byval Adr As Byte , Byval Value As Byte)
```

```
I2cstart
```

```
I2cwbyte Cod_esc
```

```
I2cwbyte Direccion_h
```

```
I2cwbyte Direccion_l
```

```
I2cwbyte Valor
```

```
I2cstop
```

```
End Sub
```

- Subrutina que lee los caracteres de la memoria 24LC256 de uno en uno.
-

```
Sub Read_eeprom(byval Adr As Byte , Value As Byte)
```

```
I2cstart
```

```
I2cwbyte Cod_esc
```

```
I2cwbyte Direccion_h
```

```
I2cwbyte Direccion_l
```

```
I2cstart
```

```
I2cwbyte Cod_lect
```

```
I2crbyte Valor , Nack
```

```
I2cstop
```

```
End Sub
```

- Subrutina que lee 16 caracteres de la memoria 24LC256 correspondientes a una línea de mensaje para ser mostrados en el lcd.
-

```
Sub Read_eeprom16(byval Adr As Byte , Value As Byte)
```

```
I2cstart
```

```
I2cwbyte Cod_esc
```

```
I2cwbyte Direccion_h
```

```

I2cwbyte Direccion_I
I2cstart
I2cwbyte Cod_lec
I2crbyte String_lcd_x(1) , Ack
I2crbyte String_lcd_x(2) , Ack
I2crbyte String_lcd_x(3) , Ack
I2crbyte String_lcd_x(4) , Ack
I2crbyte String_lcd_x(5) , Ack
I2crbyte String_lcd_x(6) , Ack
I2crbyte String_lcd_x(7) , Ack
I2crbyte String_lcd_x(8) , Ack
I2crbyte String_lcd_x(9) , Ack
I2crbyte String_lcd_x(10) , Ack
I2crbyte String_lcd_x(11) , Ack
I2crbyte String_lcd_x(12) , Ack
I2crbyte String_lcd_x(13) , Ack
I2crbyte String_lcd_x(14) , Ack
I2crbyte String_lcd_x(15) , Ack
I2crbyte String_lcd_x(16) , Nack
I2cstop
End Sub

```

-
- Subrutina que escribe 16 caracteres en la memoria 24LC256 cuando se escribe o modifica el contenido de la línea de un mensaje.
-

```

Sub Write_eeeprom16(byval Adr As Byte , Byval Value As Byte)
I2cstart
I2cwbyte Cod_esc
I2cwbyte Direccion_h
I2cwbyte Direccion_I
I2cwbyte String_lcd_x(1)
I2cwbyte String_lcd_x(2)
I2cwbyte String_lcd_x(3)

```



```

I2cwbyte String_lcd_x(4)
I2cwbyte String_lcd_x(5)
I2cwbyte String_lcd_x(6)
I2cwbyte String_lcd_x(7)
I2cwbyte String_lcd_x(8)
I2cwbyte String_lcd_x(9)
I2cwbyte String_lcd_x(10)
I2cwbyte String_lcd_x(11)
I2cwbyte String_lcd_x(12)
I2cwbyte String_lcd_x(13)
I2cwbyte String_lcd_x(14)
I2cwbyte String_lcd_x(15)
I2cwbyte String_lcd_x(16)
I2cstop
Waitms 10
End Sub

```

-
- Tabla de datos que contiene los códigos de las filas a encender, según el tipo de efecto que posea la línea que es presentada en el rótulo.
-

Matriz_efectos:

Data 2 , 128 , 64 , 32 , 16 , 8 , 4

Data 0 , 128 , 0 , 0 , 0 , 0 , 0 , 0 , 128 , 64 , 0 , 0 , 0 , 0 , 0 , 128 , 64 , 32 , 0 , 0 , 0 ,
0 , 128 , 64 , 32 , 16 , 0 , 0 , 0 , 128 , 64 , 32 , 16 , 8 , 0 , 0 , 128 , 64 , 32 , 16 , 8 ,
4 , 2 , 128 , 64 , 32 , 16 , 8 , 4

Data 128 , 0 , 0 , 0 , 0 , 0 , 0 , 64 , 0 , 0 , 0 , 0 , 0 , 128 , 32 , 0 , 0 , 0 , 0 , 128 , 64 ,
16 , 0 , 0 , 0 , 128 , 64 , 32 , 8 , 0 , 0 , 128 , 64 , 32 , 16 , 4 , 0 , 128 , 64 , 32 , 16 ,
8 , 2 , 128 , 64 , 32 , 16 , 8 , 4

Data 0 , 0 , 0 , 0 , 16 , 0 , 0 , 0 , 0 , 0 , 0 , 16 , 8 , 0 , 0 , 0 , 0 , 32 , 16 , 8 , 0 , 0 , 0 ,
0 , 32 , 16 , 8 , 4 , 0 , 0 , 64 , 32 , 16 , 8 , 4 , 2 , 0 , 64 , 32 , 16 , 8 , 4 , 2 , 128 , 64 ,
32 , 16 , 8 , 4

Data 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 4, 2, 0, 0, 0, 0, 8, 4, 2, 0, 0, 0, 16, 8, 4, 2, 0, 0, 32, 16, 8, 4, 2, 0, 64, 32, 16, 8, 4, 2, 128, 64, 32, 16, 8, 4

Data 0, 2, 0, 0, 0, 0, 0, 0, 4, 2, 0, 0, 0, 0, 0, 8, 4, 2, 0, 0, 0, 0, 16, 8, 4, 2, 0, 0, 0, 32, 16, 8, 4, 2, 0, 0, 64, 32, 16, 8, 4, 2, 2, 128, 64, 32, 16, 8, 4

-
- Tabla de datos que es utilizada para calcular la dirección donde inicia el código de un carácter que será presentado en el rótulo.
-

Codigo:

Data "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", " ", "ø", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "î", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "+", "-", "*", "/", "=", ".", ":", ";", ",", "@", "#", "\$", "%", "&", "(", ")", "[", "]", "ï", "ß", ">", "<", ""

- Las siguientes tablas permiten determinar el carácter que se ha digitado cuando se modifica el contenido de la línea de un mensaje, previo la determinación de la posición del carácter.
-

Decodifica_tecla_may:

Data "+", "-", "*", "/", "=", "1", "A", "B", "C", "2", ".", ":", "D", "E", "F", "3", ",", ";", "G", "H", "I", "4", "(", ")", "J", "K", "L", "5", "[", "]", "M", "N", "O", "ø", "6", "<", "P", "Q", "R", "S", "7", ">", "T", "U", "V", "8", "?", "!", "W", "X", "Y", "Z", "9", "", "_", "0", "@", "#", "\$", "%", "ï", "ß", "ï", "ß", "ï", "ß"

Decodifica_tecla_min:

Data "+", "-", "*", "/", "=", "1", "a", "b", "c", "2", ".", ":", "d", "e", "f", "3", ",", ";", "g", "h", "i", "4", "(", ")", "j", "k", "l", "5", "[", "]", "m", "n", "o", "î", "6", "<", "p", "q", "r", "s", "7", ">", "t", "u", "v", "8", "?", "!", "w", "x", "y", "z", "9", "", "_", "0", "@", "#", "\$", "%", "ï", "ß", "ï", "ß", "ï", "ß"

2.5. PRUEBAS Y RESULTADOS

2.5.1. CEREBRO DE CONTROL

- El cerebro de control tiene un led que permite verificar si la información, se encuentra siendo enviada desde el cerebro de control hacia el rótulo ya que la instrucción que le permite encenderse y apagarse, se encuentra en la subrutina que envía el texto hacia el rótulo.
- Cuando no se visualiza información en el rótulo, el led de control es de gran ayuda ya que nos indica que la información esta siendo enviada hacia el rótulo y por medio de esto descartar una falla de programación del código en el microcontrolador.
- También en el cerebro de control se encuentra una chicharra que emite una señal auditiva cada instante que se presiona una tecla dentro de un menú definido.
- La señal auditiva es de gran ayuda, debido a que permite conocer si una función fue o no seleccionada.
- El integrado de hora y fecha posee pines para la conexión a una batería además tiene un circuito interno que detecta las caídas de voltaje, el cual al detectar una caída de voltaje permite que la batería entre en funcionamiento suministrando el voltaje al integrado, esto permite que el reloj no se desigale, el funcionamiento de esta opción se verificó igualando la hora y la fecha para posteriormente monitorear sus valores, comprobando que la hora y la fecha no se desigualaban.
- En la memoria 24LC256 es almacenada toda la información de los mensajes escritos de manera ordenada y en direcciones específicas, que son previamente calculadas mediante el programa, la verificación de la escritura de los datos en sus direcciones correspondientes, se realizó

escribiendo varios mensajes con diferentes características en la memoria y verificando la posición de sus datos, utilizando el lector de memorias (WINPIC 800).

- El programa posee diversos menús y cada uno activa diferentes teclas, según el tipo de menú seleccionado, se comprobó que esto funcione correctamente mediante la activación de cada menú y verificando que cada tecla realice su función correspondiente, además se presiona las teclas que no corresponden al menú para verificar que no se encuentren activadas o realicen alguna función que no corresponde al menú activado.
- Los mensajes son presentados en el LCD dependiendo del tipo de menú seleccionado indicando la función que se encuentra realizando el usuario, la comprobación de esto se realizó mediante la selección de cada menú por el teclado y verificando que se muestre el mensaje que corresponde a la función realizada.

2.5.2. TARJETA DE LED'S

- Las líneas de texto de un mensaje son visualizadas en el rótulo en orden y según el tipo de efecto que posean, interactúan de diferente manera siguiendo un procedimiento determinado. Para comprobar la interacción entre efectos, se asignó diversos efectos a varias líneas, con las combinaciones de efectos que pueden ser presentados y luego se verificó su funcionamiento observando en el rótulo las líneas de texto del mensaje.
- La intensidad de los led's al presentar un mensaje cambia al seleccionar un porcentaje de brillo. Este cambio se observó al seleccionar un valor de brillo y presentar el mensaje en el rótulo.
- Insertando la hora y la fecha mediante el menú se puede visualizar que esta información aparece después del número de ciclos definidos.

2.6. ANALISIS TECNICO Y ECONOMICO

2.6.1. ANALISIS TECNICO

El eje central del sistema electrónico publicitario es el ATmega 644, que es el encargado de controlar el teclado detectando el tipo de dato ingresado, para grabarlo en la memoria y posteriormente mostrar en el rótulo diferentes mensajes escritos.

El ATmega 644 ha sido seleccionado debido a la cantidad de código de programa que puede almacenar ya que las funciones especiales utilizadas del microcontrolador son pocas como la comunicación I2C y el control del LCD.

Por programa se declaró para el ATmega 644 una frecuencia de trabajo de 16 Mhz pero también puede trabajar a 20 Mhz lo que significa que la transferencia de datos hacia el rótulo puede realizarse más rápidamente si se observara el salto de información entre filas, cabe mencionar que los integrados que llevan la información también trabajan a una máxima frecuencia de 20 Mhz.

La información a ser mostrada en el rótulo es enviada por el microcontrolador mediante la utilización de 6 pines de control denominados : Stb,Clk,Data, para el control de las columnas y Stb_fila, Clk_fila , Data_fila para el control de las filas, los datos son enviados serialmente y actualizados paralelamente. Esta forma de envío de datos permite disminuir el número pines del microcontrolador para el control del rótulo, los cuales tampoco se poseen porque serían necesarios 48 pines para el control de las columnas y 7 para el control de las filas.

La protección de los pines del ATmega 644 que envían los datos hacia el rótulo y la amplificación de los datos se realiza con el buffer 74LS244 y el ULN 2803. Estos integrados proveen la corriente necesaria para controlar los registros de desplazamiento de las columnas y para encender toda una fila de led's, debido a que la corriente que el ATmega suministra por pin es limitada, no puede controlar directamente las tarjetas de led's.

2.6.2. ANALISIS ECONOMICO

El analisis económico presenta el costo total requerido para construir el rótulo electrónico.

El costo total es la suma del costo de los materiales necesarios para construir el rótulo, más el costo de la mano de obra que es el costo por la construcción del rótulo, incluyendo además el costo de abastecimiento que determina la cantidad de dinero que se requirió durante la movilización, para adquirir los materiales del rótulo y consultar la información necesaria para realizar el software del sistema electrónico publicitario.

Costo de Materiales	\$ 223.49
Costo de Abastecimiento	\$ 100.00
Costo de mano de Obra	\$ 100.00
COSTO TOTAL	\$ 423.49

CEREBRO ELECTRONICO DE ROTULO					\$ 40.08
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Resistencia	2,2k a 1/2w	4	0.03	0.12
2	Resistencia	10k a 1/2w	2	0.03	0.06
3	Resistencia	330 a 1/2w	1	0.03	0.03
4	Resistencia	10 a 1/2w	1	0.03	0.03
5	Resistencia	1k a1/2w	1	0.03	0.03
6	Condensador Fijo	22 pf	2	0.04	0.08
7	Condensador Fijo	103	1	0.04	0.04
8	Condensador Electrolitico	470 uf /16v	3	0.39	1.17
9	Led de baja		1	0.05	0.05
10	Cristal Relej	32768 Khz	1	0.45	0.45
11	Cristal	16 Mhz	1	0.50	0.50
12	Chicharra	Zumbador	1	0.50	0.50
13	Socalo	4x2	2	0.05	0.10
14	Socalo	10x2	1	0.10	0.10
15	Socalo	20x2	1	0.15	0.15
16	Conector macho doble	5x2	4	0.10	0.40
17	Pantalla LCD	16x2	1	7.50	7.50
18	Memoria EEPROM	24LC256	1	5.00	5.00
19	Integrado DS1307	Fecha - Hora	1	4.50	4.50
20	Zocalo Bateria	Zocalo	1	0.15	0.15
21	Bateria	3V	1	3.00	3.00
22	Buffer	74LS244	1	0.50	0.50

CEREBRO ELECTRONICO DE ROTULO					\$ 40.08
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
23	Microtrolador	ATMEGA644	1	12.54	12.54
24	Tornillos Milimetricos	m3x20	3	0.10	0.30
25	Tuercas	m3	3	0.05	0.15
26	Baquelita	159mmx90mm	1	1.60	1.60
27	Impresion de Cerebro Rótulo		1	0.73	0.73
28	Estaño	60cm	0.6	0.50	0.30

TECLADO DE ROTULO					\$ 4.35
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Bus de Datos	10cm x 8 hilos	0.1	0.40	0.04
2	Conector de cerebro	Hembra 5X2	1	0.32	0.32
3	Silicon en Barra		1	0.15	0.15
4	Papel adhesivo		1	0.17	0.17
5	Cartulina		1	0.17	0.17
6	Papel aluminio	16x9 cm	1	0.017	0.017
7	Cemento de contacto		1	1.00	1.00
8	Baquelita	159mmx90mm	1	1.60	1.60
9	Impresion de Teclado Rótulo		1	0.73	0.73
10	Estaño	30cm	0.3	0.50	0.15

CERCO DE ROTULO					\$ 4.65
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Aluminio en T 1'x3/4'	60cm	1	3.00	3.00
2	Aluminio en Angulo 1'x1'	1,5cm x 4	4	0.067	0.271
3	Tornillos Autoroscantes	3/8' x 4mm negro	6	0.004	0.024
4	Alza de Acrilico	5mm	3	0.0021	0.0063
5	Alza de Baquelita	1,5mm	6	0.0021	0.0126
6	Aluminio en U	2 cm	6	0.20	1.20
7	Tornillos cabeza avellanada	m3x15	6	0.0125	0.075
8	Rodelas de Presión	m3	6	0.03	0.18
9	Tuercas	m3	6	0.01	0.06

TARGETAS DE LED'S DE ROTULO					\$ 90.58
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Resistencias	10 a 1/2w	48	0.03	1.44
2	Resistencias	2,2k a 1/2w	48	0.03	1.44
3	Leds	de alta	336	0.15	50.4
4	Conector Macho doble	5x2	8	0.05	0.40
5	Socalos	8x2	8	0.05	0.40
6	Socalos	9x2	2	0.05	0.10
7	Integrados	74595	8	0.55	4.40

TARGETAS DE LED'S DE ROTULO					\$ 90.58
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
8	Integrados	ULN2803	2	0.65	1.30
9	Transistores	3906	48	0.15	7.20
10	Regulador de voltaje	7805	2	0.45	0.90
11	Conector de cerebro	Hembra 5X2	8	0.32	2.56
12	Bus de datos	85cm x 10 hilos	0.85	0.40	0.34
13	Resistencias	10 a 2w	4	0.32	1.28
14	Baquelita	363mmx187mm	2	7.73	15.46
15	Impresion de Tarjetas Rótulo		2	0.73	1.46
16	Estaño	3m	3	0.50	1.50

PERFIL DE ALUMINIO					\$ 35.26
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Perfil Publicitario	78 cm	2	2.32	4.65
2	Angulo 1' x 1'	72 cm	2	0.776	1.552
3	Junquillo	78 cm	2	0.625	1.25
4	Tornillos cabeza Avellanados	m3 x 15	11	0.0125	0.1375
5	Tornillos	m3 x 20	12	0.01205	0.1446
6	Rodelas de presion	m4	23	0.01	0.23
7	Tuercas	m3	23	0.01	0.23
8	Tornillos Autoroscantes	1' x 5mm Negro	10	0.01	0.10
9	Tapas Laterales	8 x 21 x 0,5 cm	2	0.94	1.88
10	Tornillos Autoroscantes	1' x 5mm	4	0.01	0.04
11	Papel Adesivo negro	10 x 22 cm	2	0.25	0.50
12	Angulos Dorados		4	0.01	0.04
13	Tapa Tracera	19,7 x 78 cm	1	1.50	1.50
14	Tapa Tracera	Pintura	1	20.00	20.00
15	Acrilico Transparente	16 x 78	1	3.00	3.00

Caja para Cerebro Externo					\$ 5.37
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Caja de Acrilico	414cm2	1	1.50	1.50
2	Cable de Red	2mm	1	0.91	0.91
3	Conector	RJ45	2	1.20	2.40
4	Tornillos cabeza avelladada	m3 x 15	4	0.0125	0.05
5	Tuercas	m3	4	0.01	0.04
6	Tornillos Avellanados	1/2' x 3m	4	0.004	0.016
7	Silicon en Barra		3	0.15	0.45

Materiales					\$ 43.20
Items	Materiales	Descripcion	Cantidad	Costo Unitario	Costo por Items
1	Cable RJ45	2m	1	10.00	10.00
2	Fuente	12V 3A	1	15.00	15.00
3	Acido Ferrico	Bolsitas	14	0.50	7.00
4	Hojas	Impresión	4	0.25	1.00
5	Masquin		1	1.20	1.20
6	Sicaflex		1	4.50	4.50
7	Silicon Abro		1	4.00	4.00
8	Grampas	Caja	1	0.50	0.50

CONCLUSIONES

1. El proyecto desarrollado e implementado es una herramienta que permite promocionar un producto o un servicio, mediante la presentación de varios mensajes y cada uno con diferentes efectos.
2. La utilización de una fuente de tensión existente en el mercado, evita la construcción de otra de menor calidad y ahorra el tiempo que sería necesario para la construcción de la misma y las pruebas para comprobar si su funcionamiento es óptimo para ser parte de un producto terminado, también por la forma de presentación que posee la fuente, está asociada a la seguridad y a la comodidad.
3. El lenguaje para programar el microcontrolador Bascom Avr, es una herramienta de gran utilidad para la creación del código del programa, ya que su lenguaje de programación es Basic.
4. El ATmega 644 es utilizado por poseer grandes ventajas, en su estructura interna (mayor capacidad de memoria flash, memoria eeprom, memoria sram, contadores, líneas de entrada/salida, etc.), también porque actualmente puede ser encontrada en el mercado con facilidad a pesar de que las ventajas de su tecnología son poco conocidas.
5. El ATmega 644 al poseer una gran memoria de programa , me permitió realizar un programa con varias opciones y realizar varias subrutinas para cada función específica, teniendo un completo orden en el camino que debe seguir el programa según la función a ser realizada.
6. El diseño de la tarjeta de led's y el cerebro de control se planificó de tal manera que facilite la correcta ubicación y la fácil instalación de cada uno de los dispositivos.

RECOMENDACIONES

1. Antes de conectar la fuente a la red se recomienda verificar primeramente el voltaje de la red y que la conexión este correctamente realizada.
2. Si el equipo va ha ser instalado en lugares donde exista alta temperatura o exposición a los rayos del sol instale dispositivos de enfriamiento .
3. La adquisición de una fuente, brinda una tensión estable al ingreso del rótulo y ahorra tiempo de construcción en el sistema electrónico publicitario.

BIBLIOGRAFÍA

BIBLIOGRAFIA

Ramiro Valencia B., Aplicaciones Electronicas con Microcontroladores
MCS Electronics, Manual del usuario BASCOM-AVR

REFERENCIAS

http://www.electrofisei.com/Tutorial_AVR.pdf

http://www.mcselec.com/index.php?option=com_docman&task=cat_view&gid=99&Itemid=54

<http://www.atmel.com>

<http://www.todopic.com.ar/foros/index.php?action=printpage;topic=8108.0>

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

<http://perso.wanadoo.es/pictob/micropic.htm>

<http://www.webelectronica.com.ar/>

<http://perso.wanadoo.es/pictob/microcr.htm>

<http://ayudaelectronica.com/tipos-de-encapsulados/>

<http://www.avrfreaks.net/index.php?module=photoshare&func=viewimage&thumbnail=&iid=414>

<http://www.ermicro.com/blog/?p=744>

http://axxon.com.ar/rob/Comunicacion_busI2C.htm

<http://www.ermicro.com/blog/?p=1239>

<http://www.clubse.com.ar/DIEGO/NOTAS/notas19/nota31.htm>

http://www.disca.upv.es/aperles/web51/pdf/placas/sch/modulo_07.pdf

http://usuarios.multimania.es/carlosyaco/microcontroaldor/matriz_leds.htm

<http://www.olimex.cl/index.php?cPath=51>

http://www.caveo.com.ar/lcd_16x2_sinc_big.htm

http://picaxe.electronicasimple.com/2008/11/uln2803-qu-es-cual-es-su-funcin_11.html

<http://www.geocities.com/consultacecat/proyectos.html>

http://www.iearobotics.com/personal/ricardo/articulos/diodos_led/index.html

<http://www.alu.us.es/a/amaluqsen/Transformadores.doc>

<http://www.juntadeandalucia.es/averroes/iesbahiadecadiz/pdf/electr/cesarsanchez/modulo8/tema8.pdf>

http://picmania.garcia-cuervo.net/eagle_tutlbr_i_library.php

http://www.mcselec.com/index.php?option=com_docman&Itemid=54

www.atmel.com

ANEXOS

ANEXO A: MEDIDAS DE TECLADO Y TARJETA DE LED'S

ANEXO B: MENSAJES PRESENTANDO LOS EFECTOS EN EL LCD

ANEXO C: DATASHEET 24LC 256

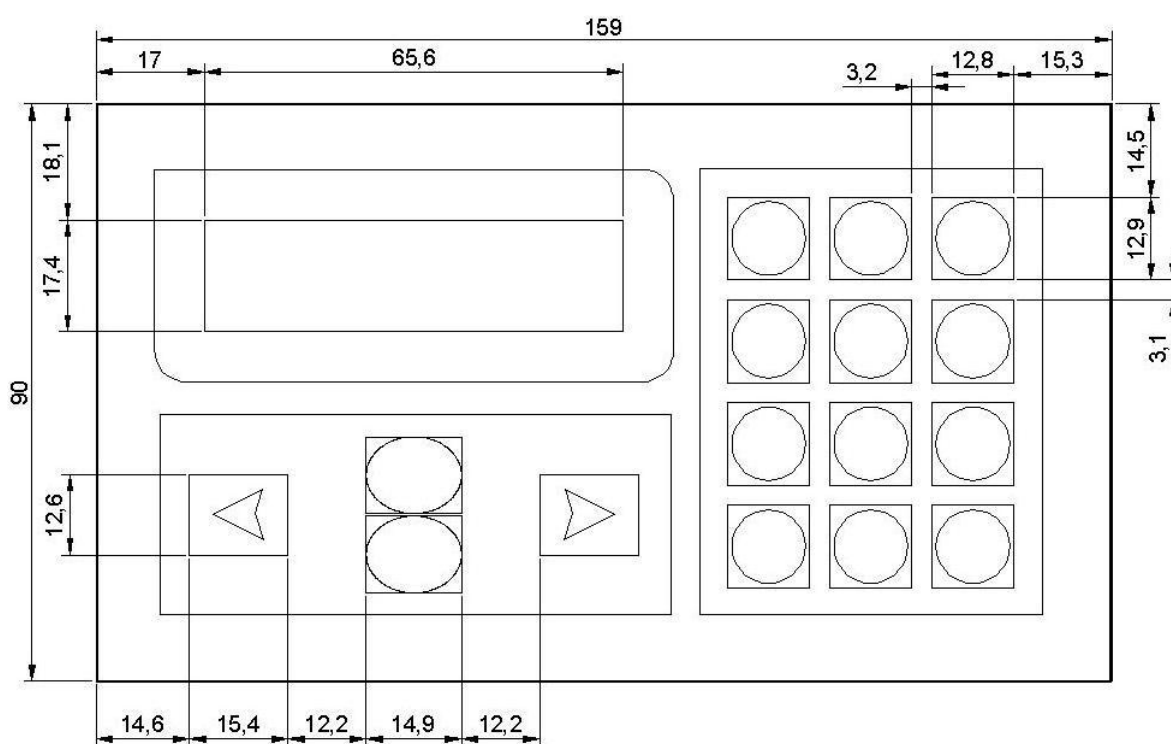
ANEXO D: DATASHEET IDS 1307

ANEXO E: DATASHEET ATMEGA-644

ANEXO A

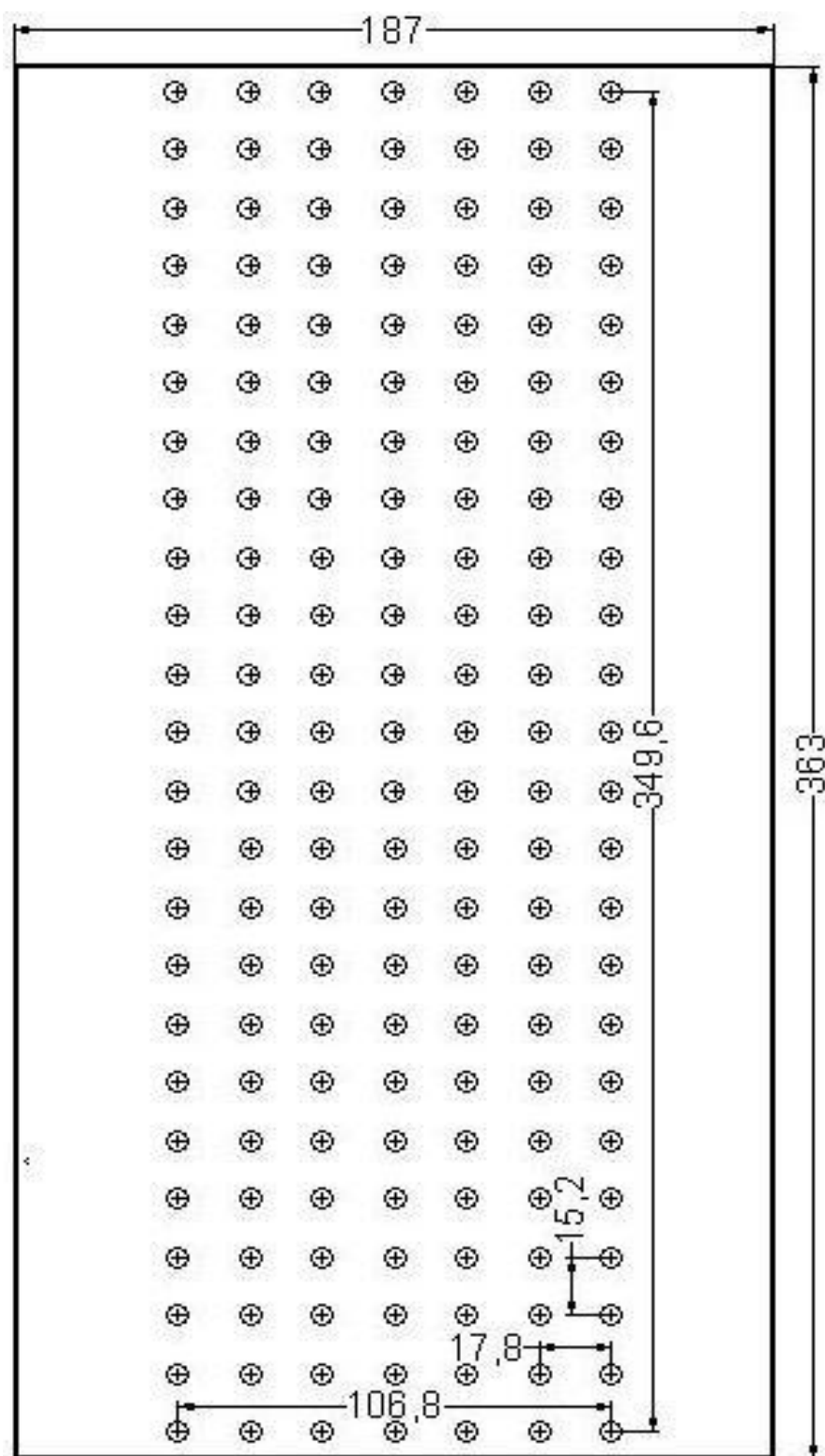
MEDIDAS DE TECLADO Y TARJETA DE LED'S

TECLADO DE CEREBRO DE CONTROL



Medidas expresadas en milímetros

TARJETA DE LED'S

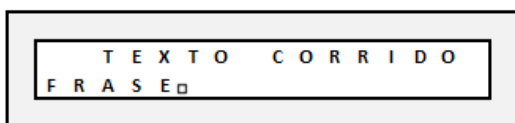


Medidas expresadas en milímetros

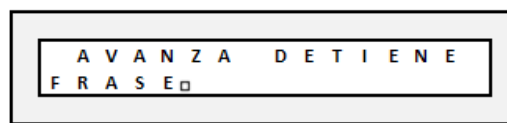
ANEXO B

MENSAJES PRESENTANDO LOS EFECTOS EN EL LCD

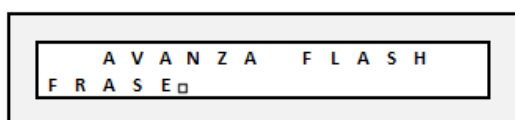
EFECTO 1



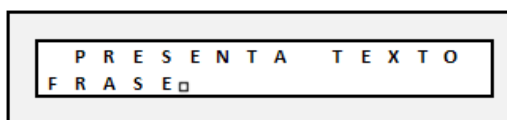
EFECTO 2



EFECTO 3



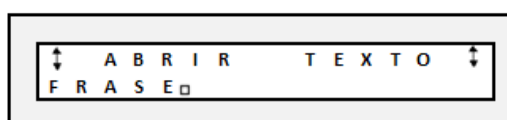
EFECTO 4



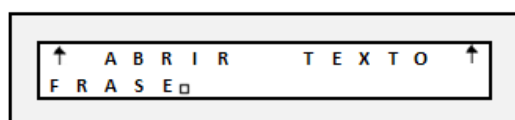
EFECTO 5



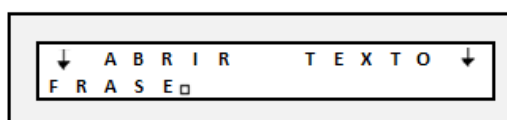
EFECTO 6



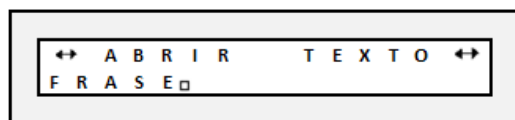
EFECTO 7



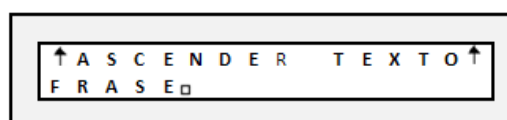
EFECTO 8



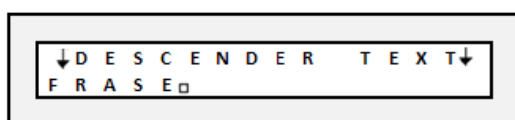
EFECTO 9



EFECTO 10



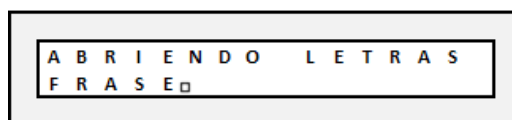
EFECTO 11




EFECTO 12



EFECTO 13



ANEXO C**DATASHEET 24LC 256**


MICROCHIP 24AA256/24LC256/24FC256
256K I²C™ CMOS Serial EEPROM

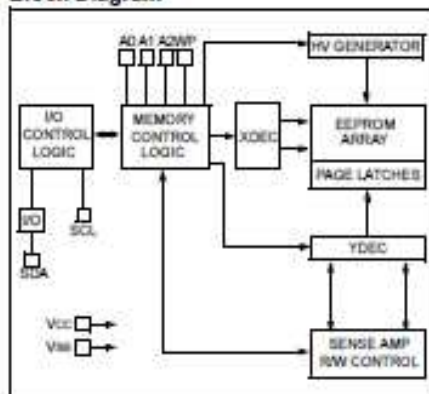
Features

- Low power CMOS technology
 - Maximum write current 3 mA at 5.5 V
 - Maximum read current 400 μ A at 5.5 V
 - Standby current 100 nA typical at 5.5 V
- 2-wire serial interface bus, I²C compatible
- Cascadable for up to eight devices
- Self-timed ERASE/WRITE cycle
- 64-byte page-write mode available
- 5 ms max write-cycle time
- Hardware write protect for entire array
- Output slope control to eliminate ground bounce
- Schmitt trigger inputs for noise suppression
- 1,000,000 erase/write cycles
- Electrostatic discharge protection > 4000 V
- Data retention > 200 years
- 8-pin PDIP, SOIC, TSSOP, MSOP, and DFN packages
- 14-lead TSSOP package
- Temperature ranges:
 - Industrial (I): -40°C to +85°C
 - Automotive (E): -40°C to +125°C

Description

The Microchip Technology Inc. 24AA256/24LC256/24FC256 (24XX256*) is a 32K x 8 (256 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8 V to 5.5 V). It has been developed for advanced, low power applications such as personal communications or data acquisition. This device also has a page-write capability of up to 64 bytes of data. This device is capable of both random and sequential reads up to the 256K boundary. Functional address lines allow up to eight devices on the same bus, for up to 2 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIC, TSSOP, MSOP, DFN and 14-lead TSSOP packages.

Block Diagram

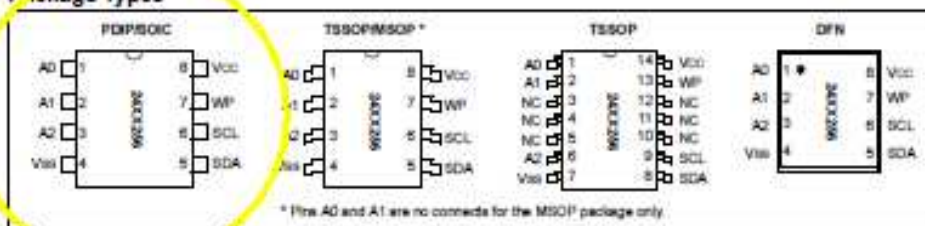


Device Selection Table

Part Number	Vcc Range	Max. Clock Frequency	Temp. Ranges
24AA256	1.8-5.5 V	400 kHz ⁽¹⁾	I
24LC256	2.5-5.5 V	400 kHz	I, E
24FC256	2.5-5.5 V	1 MHz	I

Note 1: 100 kHz for Vcc < 2.5 V.

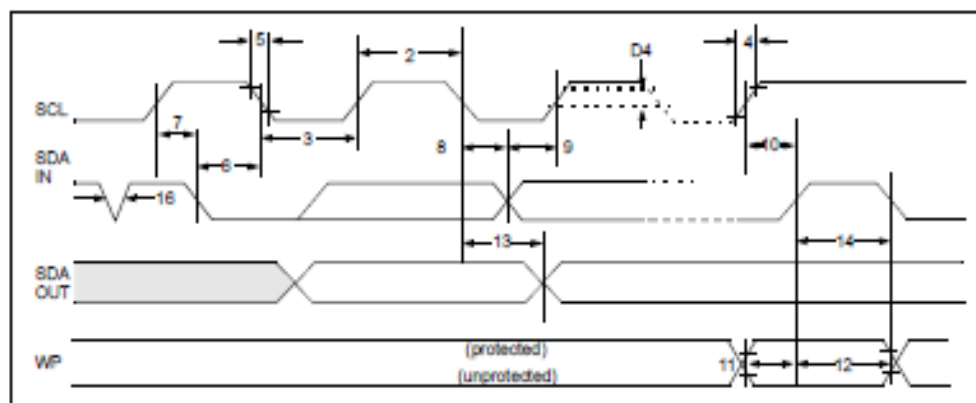
Package Types



*24XX256 is used in this document as a generic part number for the 24AA256/24LC256/24FC256 devices.

24AA256/24LC256/24FC256

FIGURE 1-1: BUS TIMING DATA



24AA256/24LC256/24FC256

5.0 DEVICE ADDRESSING

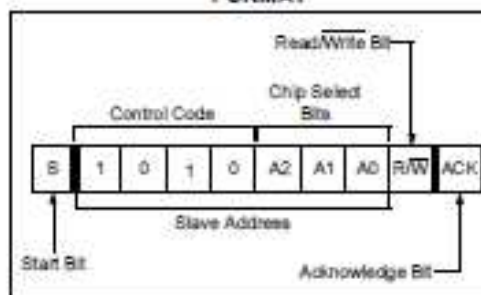
A control byte is the first byte received following the start condition from the master device (Figure 5-1). The control byte consists of a 4-bit control code. For the 24XX256, this is set as 1010 binary for read and write operations. The next three bits of the control byte are the chip select bits (A2, A1, A0). The chip select bits allow the use of up to eight 24XX256 devices on the same bus and are used to select which device is accessed. The chip select bits in the control byte must correspond to the logic levels on the corresponding A2, A1 and A0 pins for the device to respond. These bits are, in effect, the three most significant bits of the word address.

For the MDOP package, the A0 and A1 pins are not connected. During device addressing, the A0 and A1 chip select bits (Figures 5-1 and 5-2) should be set to '0'. Only two 24XX256 MDOP packages can be connected to the same bus.

The last bit of the control byte defines the operation to be performed. When set to a one, a read operation is selected. When set to a zero, a write operation is selected. The next two bytes received define the address of the first data byte (Figure 5-2). Because only A14...A0 are used, the upper address bits are a don't care. The upper address bits are transferred first, followed by the less significant bits.

Following the start condition, the 24XX256 monitors the SDA bus checking the device type identifier being transmitted. Upon receiving a 1010 code and appropriate device select bits, the slave device outputs an acknowledge signal on the SDA line. Depending on the state of the R/W bit, the 24XX256 will select a read or write operation.

FIGURE 5-1: CONTROL BYTE FORMAT

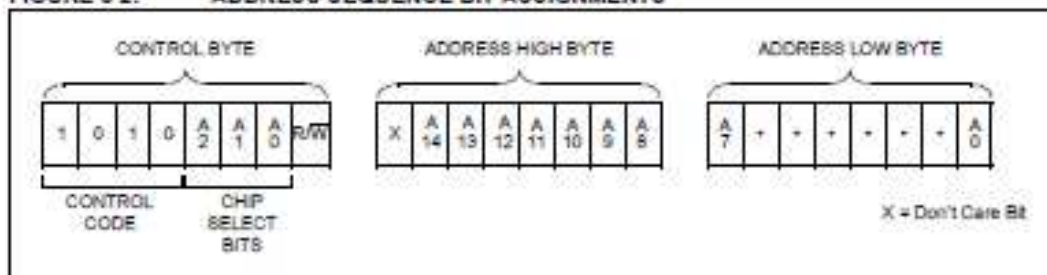


5.1 Contiguous Addressing Across Multiple Devices

The chip select bits A2, A1, A0 can be used to expand the contiguous address space for up to 2 Mbit by adding up to eight 24XX256s on the same bus. In this case, software can use A0 of the control byte as address bit A15; A1 as address bit A16; and A2 as address bit A17. It is not possible to sequentially read across device boundaries.

For the MDOP package, up to two 24XX256 devices can be added for up to 512 Kbit of address space. In this case, software can use A2 of the control byte as address bit A17. Bits A0 (A15) and A1 (A16) of the control byte must always be set to a logic '0' for the MDOP.

FIGURE 5-2: ADDRESS SEQUENCE BIT ASSIGNMENTS



24AA256/24LC256/24FC256

6.0 WRITE OPERATIONS

6.1 Byte Write

Following the start condition from the master, the control code (four bits), the chip select (three bits) and the R/W bit (which is a logic low) are clocked onto the bus by the master transmitter. This indicates to the addressed slave receiver that the address high byte will follow after it has generated an acknowledge bit during the ninth clock cycle. Therefore, the next byte transmitted by the master is the high-order byte of the word address and will be written into the address pointer of the 24XX256. The next byte is the least significant address byte. After receiving another acknowledge signal from the 24XX256, the master device will transmit the data word to be written into the addressed memory location. The 24XX256 acknowledges again and the master generates a stop condition. This initiates the internal write cycle and during this time, the 24XX256 will not generate acknowledge signals (Figure 6-1). If an attempt is made to write to the array with the WP pin held high, the device will acknowledge the command but no write cycle will occur, no data will be written, and the device will immediately accept a new command. After a byte write command, the internal address counter will point to the address location following the one that was just written.

6.2 Page Write

The write control byte, word address and the first data byte are transmitted to the 24XX256 in much the same way as in a byte write. The exception is that instead of generating a stop condition, the master transmits up to 63 additional bytes, which are temporarily stored in the on-chip page buffer and will be written into memory once the master has transmitted a stop condition. Upon receipt of each word, the six lower address pointer bits are internally incremented by one. If the master should

transmit more than 64 bytes prior to generating the stop condition, the address counter will roll over and the previously received data will be overwritten. As with the byte write operation, once the stop condition is received, an internal write cycle will begin (Figure 6-2). If an attempt is made to write to the array with the WP pin held high, the device will acknowledge the command but no write cycle will occur, no data will be written and the device will immediately accept a new command.

6.3 Write Protection

The WP pin allows the user to write-protect the entire array (0000-7FFF) when the pin is tied to V_{CC}. If tied to V_{SS} or left floating, the write protection is disabled. The WP pin is sampled at the STOP bit for every write command (Figure 1-1). Toggling the WP pin after the STOP bit will have no effect on the execution of the write cycle.

Note: Page write operations are limited to writing bytes within a single physical page, regardless of the number of bytes actually being written. Physical page boundaries start at addresses that are integer multiples of the page buffer size (or 'page size') and end at addresses that are integer multiples of (page size - 1). If a page write command attempts to write across a physical page boundary, the result is that the data wraps around to the beginning of the current page (overwriting data previously stored there), instead of being written to the next page, as might be expected. It is, therefore, necessary for the application software to prevent page write operations that would attempt to cross a page boundary.

FIGURE 6-1: BYTE WRITE

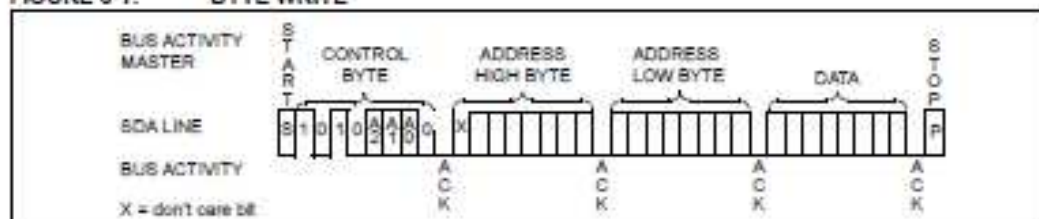
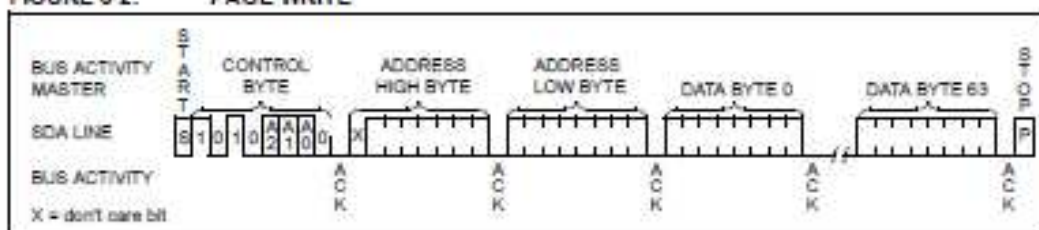


FIGURE 6-2: PAGE WRITE



ANEXO D

DATASHEET IDS 1307



DS1307 64 x 8 Serial Real-Time Clock

www.maxim-ic.com

FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

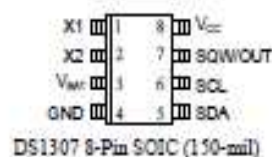
ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

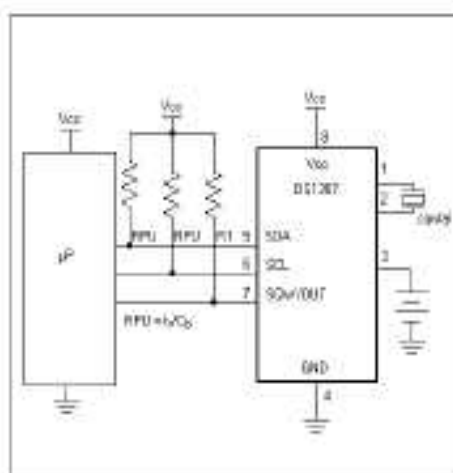
PIN ASSIGNMENT



PIN DESCRIPTION

V _{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V _{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

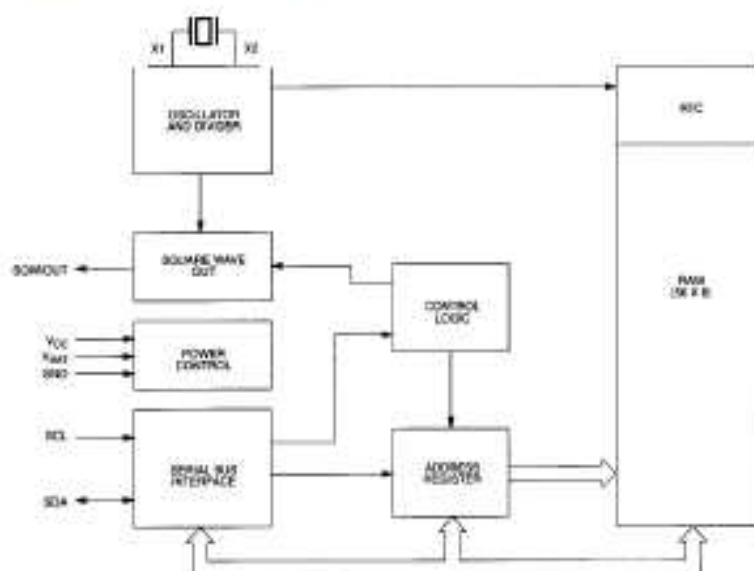
TYPICAL OPERATING CIRCUIT



OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below $1.25 \times V_{BAT}$ the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low-current battery backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than $V_{BAT} + 0.2V$ and recognizes inputs when V_{CC} is greater than $1.25 \times V_{BAT}$. The block diagram in Figure 1 shows the main elements of the serial RTC.

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5V input. When 5V is applied within normal limits, the device is fully accessible and data can be written and read. When a 3V battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} – Battery input for any standard 3V lithium cell or other energy source. Battery voltage must be held between 2.0V and 3.5V for proper operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See “Conditions of Acceptability” at <http://www.maxim-ic.com/TechSupport/QA/ntrl.htm>.

SCL (Serial Clock Input) – SCL is used to synchronize data movement on the serial interface.

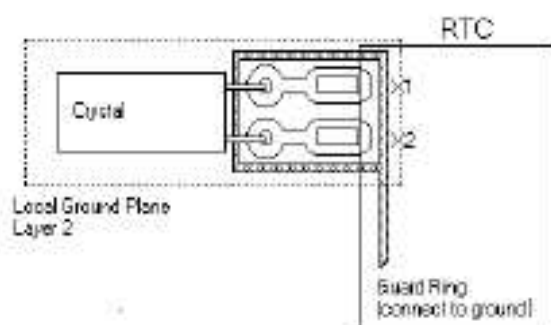
SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

SQW/OUT (Square Wave/Output Driver) – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pull-up resistor. SQW/OUT will operate with either V_{CC} or V_{BAT} applied.

X1, X2 – Connections for a standard 32.768kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5pF.

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, “Crystal Considerations with Dallas Real-Time Clocks.” The DS1307 can also be driven by an external 32.768kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

RECOMMENDED LAYOUT FOR CRYSTAL



CLOCK ACCURACY

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. See Application Note 58, "Crystal Considerations with Dallas Real-Time Clocks" for detailed information.

Please review Application Note 95, "Interfacing the DS1307 with a 8051-Compatible Microcontroller" for additional information.

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

DS1307 ADDRESS MAP Figure 2

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The RTC registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. Bit 7 of register 0 is the clock halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

Please note that the initial power-on state of all registers is not defined. Therefore, it is important to enable the oscillator (CH bit = 0) during initial configuration.

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.

DS1307 TIMEKEEPER REGISTERS Figure 3

		BIT 7								BIT 0		
00H	0H	16 SECONDS				SECONDS				00-0F		
	1H	16 MINUTES				MINUTES				00-0F		
	2H	10	24	30 MIN AP	16 HR	HOURS				01-12 0A-0D		
	3H	0	0	0	0	0	DAY				0-7	
	4H	0	0	10 DATE		DATE				01-09/09 01-31 01-31		
	5H	0	0	0	0	12 MONTH		MONTH		01-12		
01H	10 YEAR				YEAR				00-9F			
	OUT	0	0	SQWE	0	0	RS1	RS0				

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

OUT (Output control): This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0.

SQWE (Square Wave Enable): This bit, when set to a logic 1, will enable the oscillator output. The frequency of the square wave output depends upon the value of the RS0 and RS1 bits. With the square wave output set to 1Hz, the clock registers update on the falling edge of the square wave.

RS (Rate Select): These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

SQUAREWAVE OUTPUT FREQUENCY Table 1

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz