



Objetivo

- Encender-Apagar lámparas mediante control remoto y monitorear el estado de dichas lámparas.
- Utilizar el microcontrolador QY4A para monitorear el estado de las lámparas, y estos estados mostrarlos en un display LCD (20x4, bus de datos 8bits).

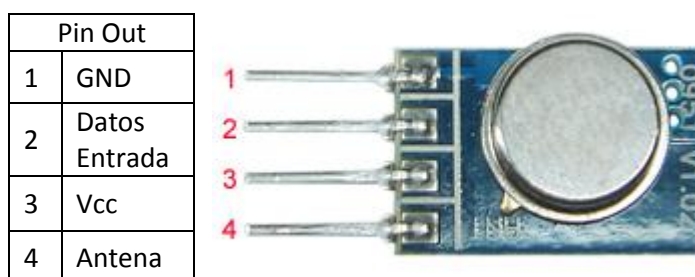
Introducción

El trabajo tiene como fin controlar a distancia el encendido-apagado de las luces de un hogar, ya que en una casa se encuentran obstáculos para la señal (paredes, puertas, muebles, etc.) se utiliza módulos de radiofrecuencia (RF) tanto para el transmisor como para el receptor. Con este proyecto se familiariza uno con la transmisión inalámbrica mediante RF, con el armado de antenas para dichos módulos, con la parte de potencia para controlar las lámparas (conectadas a la red 220VAC) y con la toma de los datos que entrega los integrados para llevarlos a la entrada del microcontrolador (QY4A) y así mostrar dichos datos en un display LCD.

Módulos RF y los integrados utilizados con dichos módulos

Para la transmisión-recepción de RF se utiliza los módulos TWS-418 y RWS-418 (418Mhz) muy pequeños y de bajo costo, y los IC HT12E-HT12D para la codificación-decodificación de la señal RF.

Transmisor y Receptor TWS-418 y RWS-418: El módulo transmisor viene ya ajustado en una frecuencia, que es 418Mhz. Está listo para su uso. Sólo se debe colocar una antena (de la longitud indicada), conectarle la alimentación y comenzar a enviarle datos. Para facilitar la transmisión de datos codificados, existe un codificador que es el Holtek HT12E, que también es de un precio relativamente bajo, comparado con un microcontrolador, por eso se lo utilizo.



El motivo por el cual usar estos modulos de RF es por su mayor distancia entre Tx-Rx comparados con la transmision por Infrarrojo, en este caso, según la hoja de dato del fabricante se logra un alcance de 20m, pero esto depende mucho de las condiciones del ambiente y la antena.

El transmisor listado se comunica con receptor de la misma frecuencia, que también vienen en un valor predeterminado 418 MHz (RWS-418). No requiere componentes externos para recibir la señal, simplemente una antena adecuada a dicha frecuencia. Para decodificar la señal que llega a este receptor se puede utilizar el decodificador Holtek HT12D.



Técnicas Digitales II (FINAL)

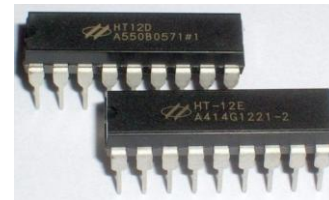
Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

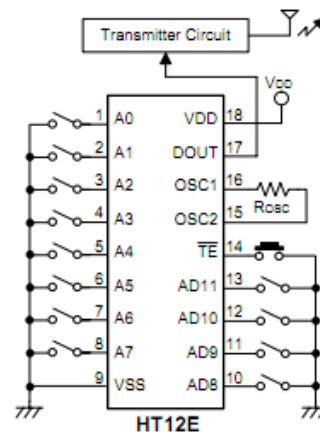
Pin Out	
1	GND
2	Datos Salida
3	Salida
4	Vcc
5	Vcc
6	GND
7	GND
8	Antena



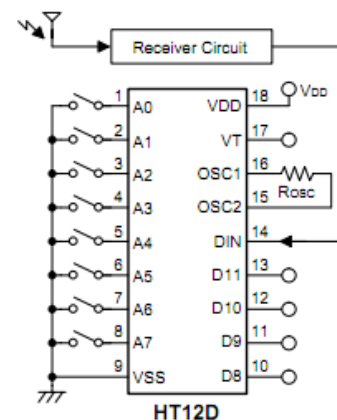
Integrados HT12E - HT12D: Este juego de integrados codifica y decodifica una palabra de 12 bits, compuesta por una dirección de 8 bits y una sección de datos de 4 bits. Con esta cantidad de bits se pueden comandar 256 dispositivos diferentes, enviándoles hasta 16 comandos distintos a cada uno.



Transmision: Los pines 1 a 8 eligen la dirección (que tendría que ser igual en el transmisor y receptor para lograr comunicación), en este caso conectadas todas a masa, los pines 10 a 13 son los datos que se desean enviar, serian las llaves de encendido-apagado de cada lampara, el pin 14 controla el envío (al dejarla en estado bajo transmite, esta a 0 para poder transmitir constantemente), los pines 15 y 16 necesita ser conectada a una resistencia para generar la señal de clock interna, el pin 17 es la salida de datos, que debe ser conectada al módulo de transmisión.



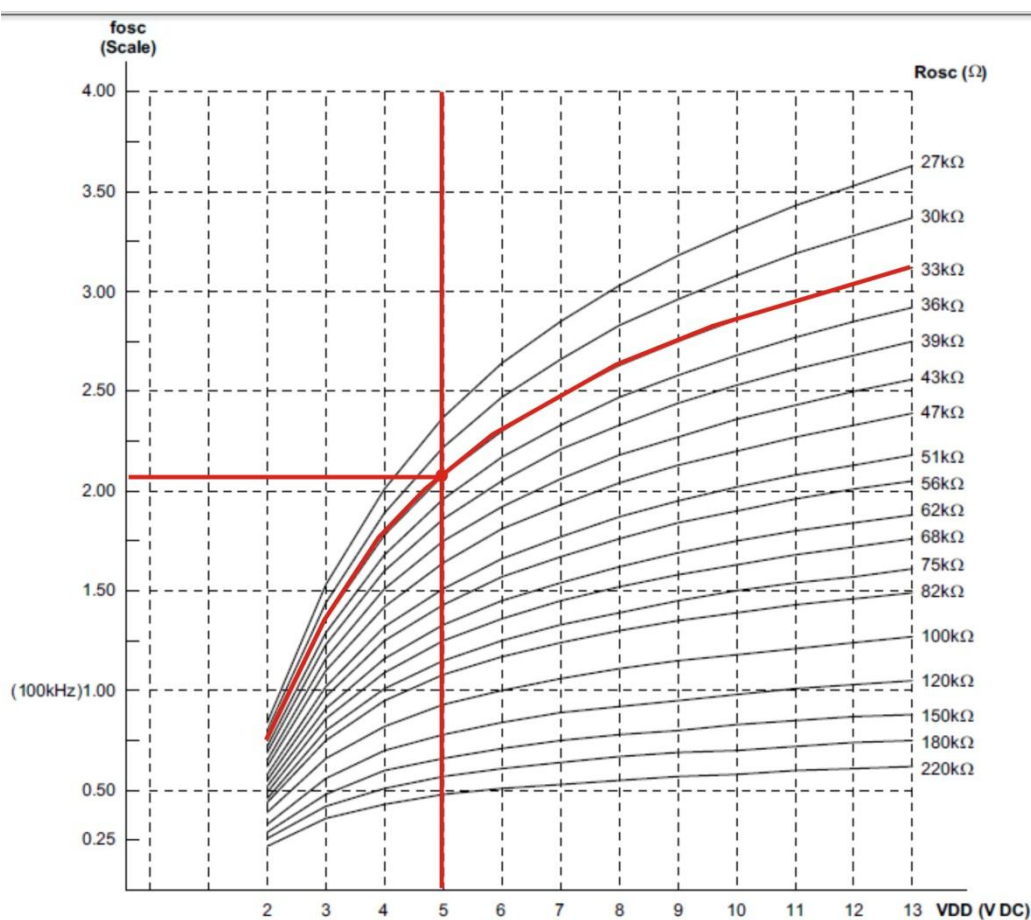
Recepcion: Nuevamente, los pines 1 a 8 eligen la dirección (que tendría que ser igual en el transmisor y receptor para lograr comunicación), los pines 10 a 13 son los datos recibidos, el pin 14 deberá ser conectada a la salida del módulo de recepción de RF, los pines 15 y 16 necesitan ser conectadas a una resistencia para generar la señal de clock interna, el pin 17 dice si la recepción fue correcta o no, en este caso siempre va a estar encendido el LED indicando transmision, se apaga momentaneamente cuando recibe el dato.





En ambos integrados entre los pines 15 y 16 deben tener una resistencia de un determinado valor para poder generar la señal de clock, según la hoja de datos, realizo los siguientes procedimientos para calcularlas:

1. Tomo la grafica del HT12D (decodificador) porque al estar conectado al display y al QY4A todos trabajan a 5V, entonces con este dato me voy a la grafica y eligo una resistencia comercial de 33KΩ.

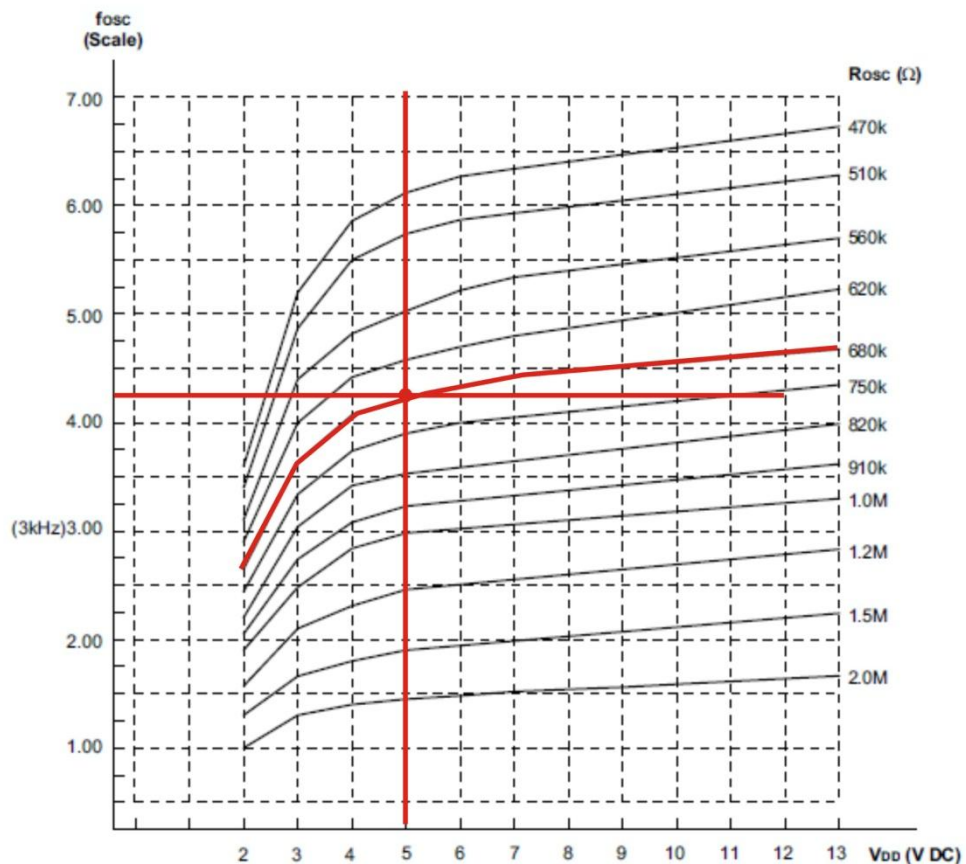


Lo que me da una frecuencia aproximadamente de 212,5Khz.

2. Con la ecuacion que esta en la hoja de dato obtengo la otra frecuencia, del HT12E

$$f_{OSCD}(HT12D) \cong 50 \cdot f_{OSCE}(HT12E)$$
$$f_{OSCE}(HT12E) \cong \frac{212,5Khz}{50} \cong 4,25Khz$$

3. Luego con la grafica del HT12E y con la frecuencia de 4,25Khz y con una tension de 5V se obtiene una resistencia muy proxima a 680KΩ. Aclaracion: Con este valor calculado de frecuencia y tension, yo probe ultizar una pila de 12V y se obtuvieron los mismos resultados satisfactorios que con 5V.



Antenas

Ambos modulos trabajan a una frecuencia de 418Mhz, entonces se tiene que calcular la longitud de onda (λ) a partir de esta frecuencia:

$$\lambda = \frac{c}{F} = \frac{300}{F(\text{Mhz})} = \frac{300}{418} \cong 0,72\text{m}$$

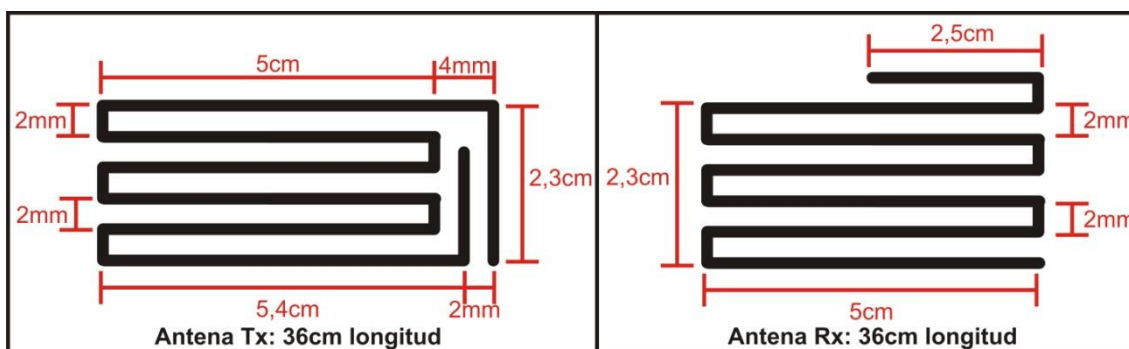
Teniendo la longitud de la onda, calculo la antena para $\frac{1}{2}$ y un $\frac{1}{4}$ de onda:

- Para $\lambda/2$: 36cm el tamaño de la antena
- Para $\lambda/4$: 18cm el tamaño de la antena

Primeramente hice 2 pruebas, que consistia en usar como antenas cables de esos tamaños calculados, ambos valores daban una buena transmision, el inconveniente de usas cables de estas dimensiones eran que se caian y quedaban “anti-estetico” en la placa.

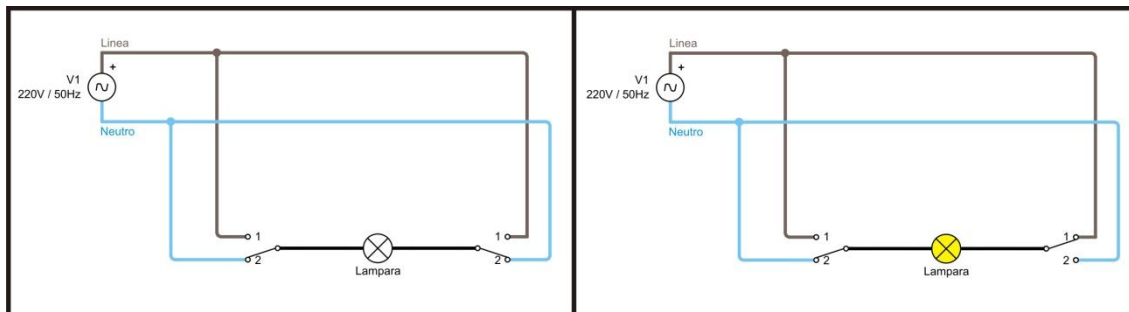
Luego de estas 2 antenas (cable) busque informacion para realizar las antenas en PCB, como las que tienen los controles remotos de alarmas de automoviles.

Se recomienda hacer antenas en pcb con distancias entre pistas de ancho de 1mm (o un poco superior) y separacion entre pistas de 2mm, tambien realice dos pruebas con antenas en PCB, para $\lambda/2$ y para $\lambda/4$, para la primera antena no tuve inconvenientes, para la segunda la transmision no se realizaba bien, o a veces dejaba de transmitir, entonces las antenas realizadas en PCB fueron de $\lambda/2$ para el proyecto.



Llave conmutada y relees

Las salidas del HT12D se conectan a una compuerta XOR para simular una llave conmutada. Se utilizan las llaves conmutadas en un hogar cuando se quiere apagar-encender una lámpara desde lugares distintos, es decir mediante distintos interruptores. En este caso decidí colocar estas llaves (XOR) pensando en una posible falla en los circuitos de RF, en caso de ser así, las lámparas se las podrían controlar manualmente. No se utilizó el microcontrolador debido a que el QY4A tiene todos sus pines ocupados. Pero en el caso de tener un micro con más pines, es factible hacerlo. En mi caso me salía más económico comprar la compuerta XOR que un microcontrolador con más pines, por eso tome la decisión de usar la compuerta.



Del esquema anterior se puede observar que si las llaves están ambas en la misma posición (1-1 o 2-2) la lámpara permanece apagada, en cambio si están en posiciones opuestas la lámpara se enciende, con estos datos se puede armar una tabla de estados:

Posición		Lámpara
Llave 1	Llave 2	
1	1	Apagada
1	2	Encendida
2	1	Encendida
2	2	Apagada

Comparando esta tabla de estado con la tabla de la compuerta XOR, son idénticas, por eso se utilizó esta compuerta para simular la llave conmutada.



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

Para la parte del control de las lámparas, se utiliza el integrado ULN2803, que es un arreglo de 8 Darlington, conectados a las salidas de las compuertas XOR, se utiliza este integrado para aumentar la corriente que hay en la salida de la compuerta. 4 de estos Darlington se conectan a relés, los cuales se conectan a las lámparas y a 220VAC. De esta forma las lámparas trabajan con VAC, mientras que el resto del circuito con VDC.

Se utiliza el ULN2803 en vez del ULN2003 (7 Darlington), ya que los otros 4 Darlington sobrantes, se los conecta a la entrada del microcontrolador y así asegurar la corriente necesaria a la entrada del micro.

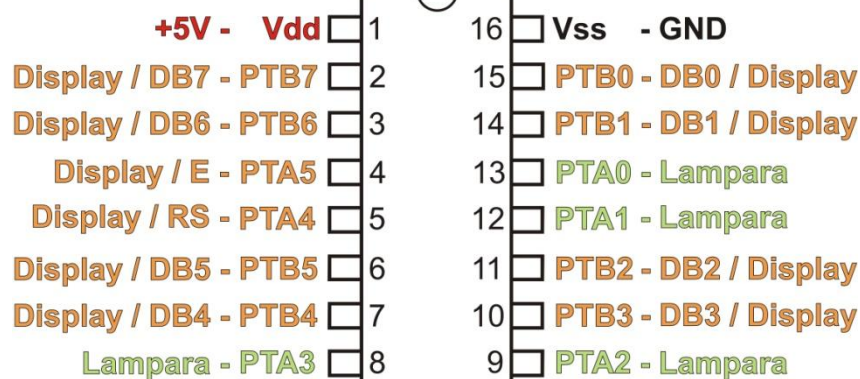
Las salidas del HT12D, entradas de la compuerta XOR, y salidas del ULN2803 se encuentran conectadas mediante pull-up para asegurar una buena conmutación entre el estado ALTO y BAJO.

Funciones de los pines del micro

Cuatro pines de este están destinadas como entradas de información que proviene del ULN2803, estas indican los estados de cada lámpara, dependiendo de cuáles sean, se mostrara en un display LCD los estados.

Como la comunicación con el display es a través de un bus de 8bits, se tiene 8 pines como salidas, a su vez el display tiene pines de control RS (selección de registro) y E (habilita registro) entonces estos pines en el micro también son de salida, el pin RW (lectura/escritura) del display está conectado a GND, porque solo lo utilizo para escribir. En resumen, se tiene 10 pines de salida y 4 como entrada, como utilizo el micro QY4A, tengo todos los pines del puerto A y B ocupados.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Pines Display
GND	VCC	Vo (contraste)	RS	RW	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	resistencia a Vcc	GND	
			PTA4	GND	PTA5	PTB0	PTB1	PTB2	PTB3	PTB4	PTB5	PTB6	PTB7	PINES DEL MICRO SALIDA		
			5		4	15	14	11	10	7	6	3	2			
										PTA0	PTA1	PTA2	PTA3	Pines DEL MICRO ENTRADA		
										13	12	9	8			



QY4A

☐ : Entradas

☐ : Salidas



Display LCD: Pines y funcionamiento

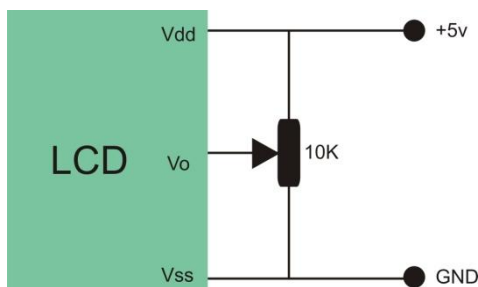
El fin de utilizar este display mostrando los estados sería para ubicarlos en el centro de un hogar e indicar por ejemplo que luz se encuentra encendida y si el usuario cree que no es necesario que este encendida, la apaga de forma remota y sin necesidad de ir hasta el lugar para ver si está o no encendida.

Entonces dependiendo si tiene un 0 o 1 en la entrada del micro, el display mostrara el resultado de eso.

El display LCD que utilizo es de 20 carácter por 4 líneas, conectado en modo de 8bits, el controlador interno que tiene es Hitachi 44780 (más común). En la siguiente tabla se explica la función de cada pin para esta clase de display LCD:

Pin	Símbolo	Función	
1	Vss	Tierra (0V)	
2	Vdd	Alimentación +5V	
3	Vo	Contraste	
4	RS	Selector de Dato/Instrucción	
5	R/W	Selector de Lectura/Escritura	1:Lectura 0: Escritura
6	E	Habilitación del módulo	
7	DB0	Bus de datos	
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	LED (+)	Ánodo del Led de luz de fondo	
16	LED (-)	Cátodo del Led de luz de fondo	

Los 3 primeros pines son de alimentación, es fácil definir el contraste mediante la siguiente conexión:



Pines del 7 al 14: bus de datos, por donde mandaremos la información al LCD.

Pin 6: Enable habilita (1) o deshabilita (0) el módulo. Encendiéndolo y apagándolo le indicamos al LCD que vamos a realizar una operación sobre él y éste pasa a leer el resto de los pines para recibir la orden.

Pin 5: R/W indica si se va a hacer una operación de escritura (0) o lectura (1). Como solo voy a escribir, este pin está conectado a tierra.



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

El pin 4: RS selecciona el registro del LCD que vamos a usar, el LCD tiene dos registros de 8 bits, un registro de instrucciones (0) y otro de datos (1), seleccionando el de instrucciones indicamos al LCD que lo que hay en el bus de datos (pines desde 7 a 14) es una instrucción, y seleccionando el registro de datos indicamos al LCD que lo que tenemos en el bus de datos es un carácter cuyo destino es la DDRAM o la CGRAM que son las dos zonas de memoria del LCD.

La DDRAM (Data Display RAM) es una zona de memoria donde se almacenan los caracteres que se van a representar en la pantalla, su capacidad es de 80 caracteres, 20 por línea. Cada vez que se escribe un dato en ésta se apunta a la siguiente posición.

Las posiciones de los caracteres en la memoria cuando se usan 4 líneas de pantalla no son consecutivas, de la línea 1 pasa a la línea 3, cosa que tendremos que tener en cuenta si queremos escribir caracteres de manera secuencial.

Posición en memoria de cada carácter en la pantalla																			
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53
14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67

En la CGRAM podemos dibujar nuestros propios caracteres, se pueden almacenar hasta 4 de 5x10 puntos y hasta 8 de 5x8 puntos.

La CGROM genera los caracteres disponibles para representar en la pantalla a partir de un código de 8 bits.

Cuando tengamos un 1 en el bit RS, los 8 bits del bus de datos representarán uno de los caracteres siguientes.

Upper 4 Bits		Lower 4 Bits																
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000		CG RAM (1)			0	1	P	`	P					-	9	E	α	p
xxxx0001	(2)			!	1	A	Q	a	q			°	7	4		ä	q	
xxxx0010	(3)			"	2	B	R	b	r			「	イ	ツ	×	ρ	θ	
xxxx0011	(4)			#	3	C	S	c	s			」	ウ	〒	ε	∞		
xxxx0100	(5)			\$	4	D	T	d	t			、	エ	ト	μ	Ω		
xxxx0101	(6)			%	5	E	U	e	u			・	オ	ナ	1	ö	Ü	
xxxx0110	(7)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ	
xxxx0111	(8)			'	7	G	W	g	w			ア	キ	ヌ	ラ	q	π	
xxxx1000	(1)			(8	H	X	h	x			イ	ク	ネ	リ	7	×	
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	ル	リ	4		
xxxx1010	(3)			*	:	J	Z	j	z			エ	コ	ハ	レ	j	≠	
xxxx1011	(4)			+	;	K	[k	[オ	サ	ヒ	ロ	×	π	
xxxx1100	(5)			,	<	L	¥	l	¥			カ	シ	フ	ワ	φ	π	
xxxx1101	(6)			-	=	M]	m]			ユ	ズ	ハ	ン	も	÷	
xxxx1110	(7)			.	>	N	^	n	^			ヨ	セ	ホ	°	π		
xxxx1111	(8)			/	?	O	_	o	_			ッ	リ	マ	°	ö	■	



Con un 0 en RS tendremos que mandar uno de los siguientes comandos por los 8 bits del bus de datos.

Instruction	Instruction Code										Description	Execution time (fosc=270Khz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	—	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and blinking of cursor (B) on/off control bit.	39μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	—	—	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39μs
Function Set	0	0	0	0	1	DL	N	F	—	—	Set interface data length (DL:8-bit/4-bit), numbers of display line (N:2-line/1-line)and, display font type (F:5×11 dots/5×8 dots)	39μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43μs

Clear Display: Borra el módulo LCD y coloca el cursor en la primera posición (dirección 00).

Return Home: Coloca el cursor en la posición de inicio (dirección 00) y hace que el display comience a desplazarse desde la posición original. El contenido de la memoria RAM de datos de visualización (DD RAM) permanece invariable.

Entry Mode Set: Establece la dirección de movimiento del cursor y especifica si la visualización se va desplazando a la siguiente posición de la pantalla o no. Estas operaciones se ejecutan durante la lectura o escritura de la DDRAM o CGRAM. Para visualizar normalmente poner el bit S=0.

Display ON/OFF Control: Activa o desactiva al display (D), el mensaje aparece (se hace visible) cuando D = 1 y desaparece (invisible) cuando D = 0. El cursor aparece (es visible) cuando C = 1 y desaparece (invisible) cuando C = 0. Los datos contenidos en pantalla no se ven afectados por operaciones en este bit. Cuando B = 1 y C = 1 el cursor aparece alternando con el texto en ese segmento con la matriz con todos sus pixeles encendidos en intervalos regulares de 0,4 segundos. Cuando B = 1 y C = 0, obtenemos la activación de un cubo negro (todos los pixeles de la matriz activos) con intervalos regulares de 0,4 segundos. En resumen con B = 0 el cursor no parpadea y con B = 1 parpadea.



Cursor or Display Shift: Desplaza el cursor con respecto al mensaje hacia la derecha o izquierda sin la necesidad de escribir o leer datos del display. Esta función se emplea normalmente para modificar sólo algunos datos en pantalla evitándonos la necesidad de limpiarla (borrarla) y escribir nuevamente todo su contenido.

S/C	R/L	Efecto
0	0	Desplaza el cursor hacia la izquierda (disminuye el contador de direcciones en una unidad).
0	1	Desplaza el cursor hacia la derecha (aumenta el contador de direcciones en una unidad)
1	0	Desplaza el cursor, junto con el mensaje, hacia la izquierda.
1	1	Desplaza el cursor, junto con el mensaje, hacia la derecha.

Function Set: Establece el tamaño del bus de datos (DL), número de líneas del display (N) y tipo de carácter (F).

DL: Establece el número de bits de interconexión. Los datos son transportados en 8 bits (DB7 - DB0) cuando DL = 1, y en 4 bits (DB7 - DB4) cuando DL = 0. Cuando los datos viajan en 4 bits, es preciso efectuar dos operaciones en vez de una. En una primera instancia viajan los bits más significativos y, en una última, viajan los menos significativos.

N: Establece el número de líneas de la pantalla, N= 1 pantalla de 4 líneas, N= 0 pantalla de 2 líneas.

F: Establece el tipo de matriz, F=1 Caracteres 5x10 puntos, 0 caracteres 5x8 puntos.

Set CGRAM ADDRESS: El módulo LCD además de tener definidos todo el conjunto de caracteres ASCII, permite al usuario definir 4 u 8 caracteres gráficos. La composición de estos caracteres se va guardando en una memoria llamada CGRAM con capacidad para 64 bytes. Cada carácter gráfico definido por el usuario se compone de 16 u 8 bytes que se almacenan en sucesivas posiciones de la CGRAM.

Set DDRAM ADDRESS: Los caracteres o datos que se van visualizando, se van almacenando en una memoria llamada DD RAM para de aquí pasar a la pantalla.

Read BUSY FLAG and ADDRESS: Permite la lectura del bit 7 que funciona como busy flag ó bandera de ocupado. Cuando BF = 1 el sistema está ocupado en alguna operación interna y no acepta ninguna instrucción hasta que BF = 0, o sea que se desocupe. Cualquier instrucción enviada por error no será leída, por lo que será nula.

WRITE DATA TO RAM: Mediante este comando se escribe en la memoria DDRAM los datos que se quieren presentar en pantalla y que serán los diferentes códigos ASCII de los caracteres a visualizar.

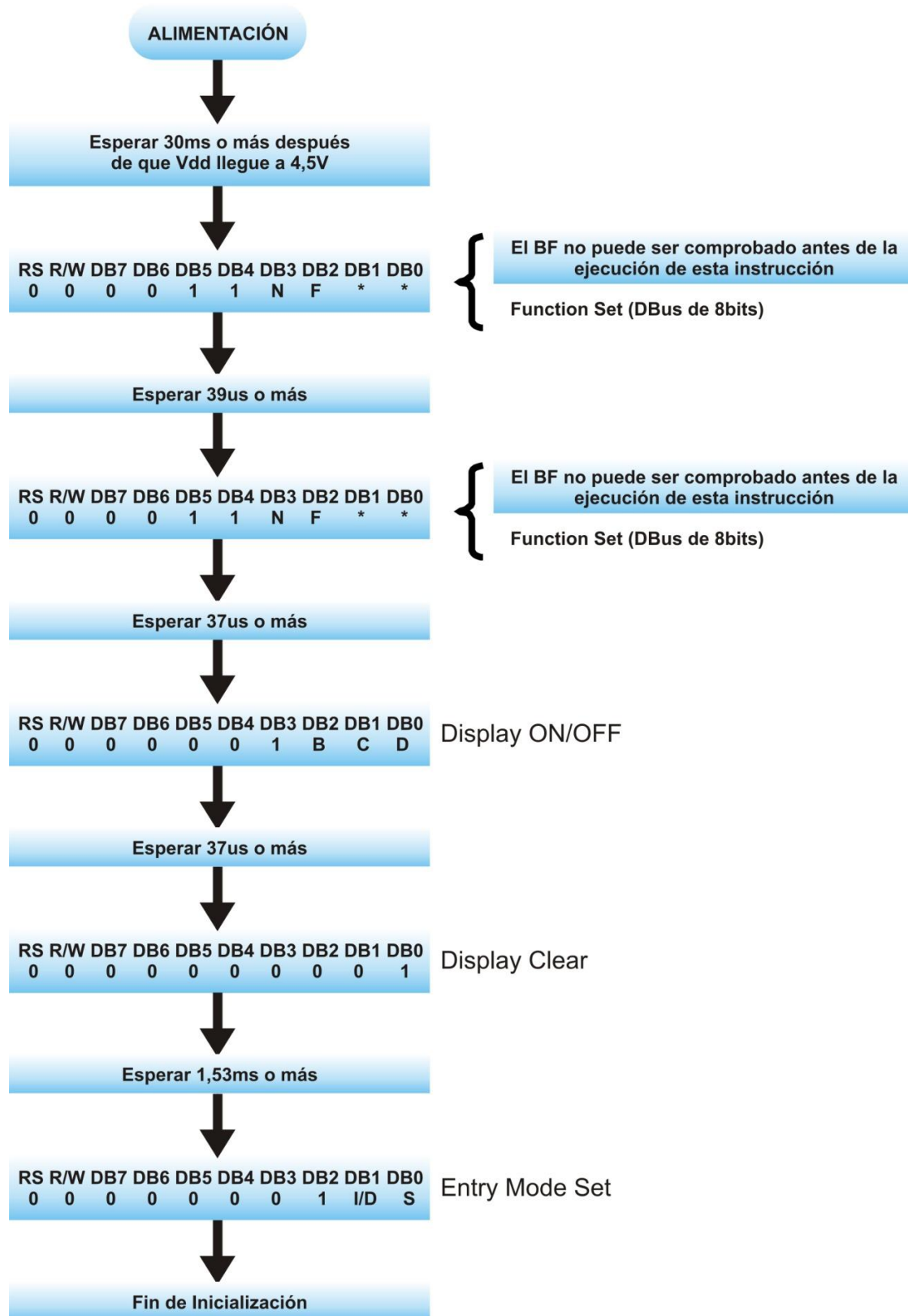
Igualmente se escribe en la memoria CGRAM los diferentes bytes que permiten confeccionar caracteres gráficos a gusto del usuario.

READ DATA FROM RAM: Mediante este comando se lee de la memoria DDRAM los datos que haya almacenados y que serán los códigos ASCII de los caracteres almacenados.

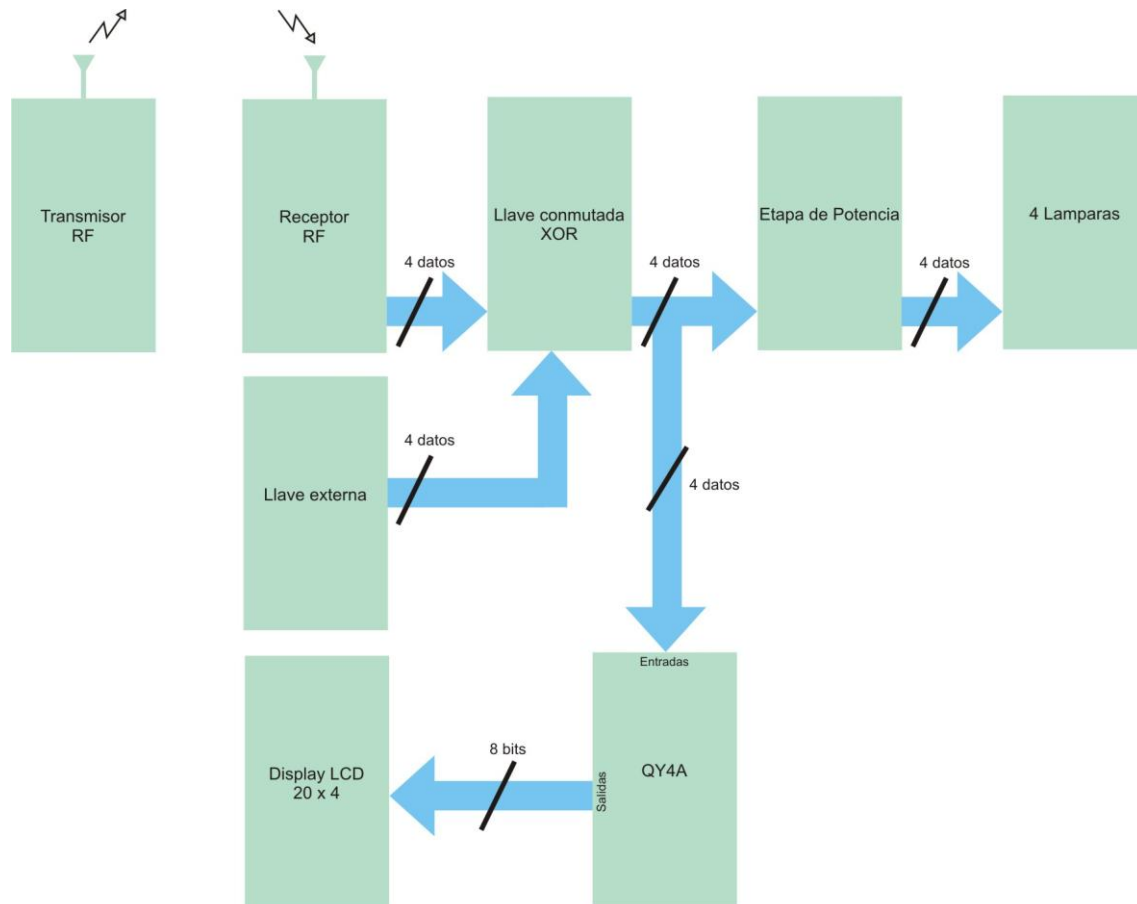
Igualmente se lee de la memoria CGRAM los diferentes bytes con los que se ha confeccionado un determinado carácter gráfico.

Inicialización del LCD

Cada vez que se conecta la alimentación al módulo debe ser ejecutado un procedimiento para iniciar el LCD, el cual consiste en enviar al módulo un grupo inicial de instrucciones para configurar el modo de operación. El procedimiento a seguir teniendo un bus de 8 bits es el siguiente:



Breve explicación del circuito general y el LCD

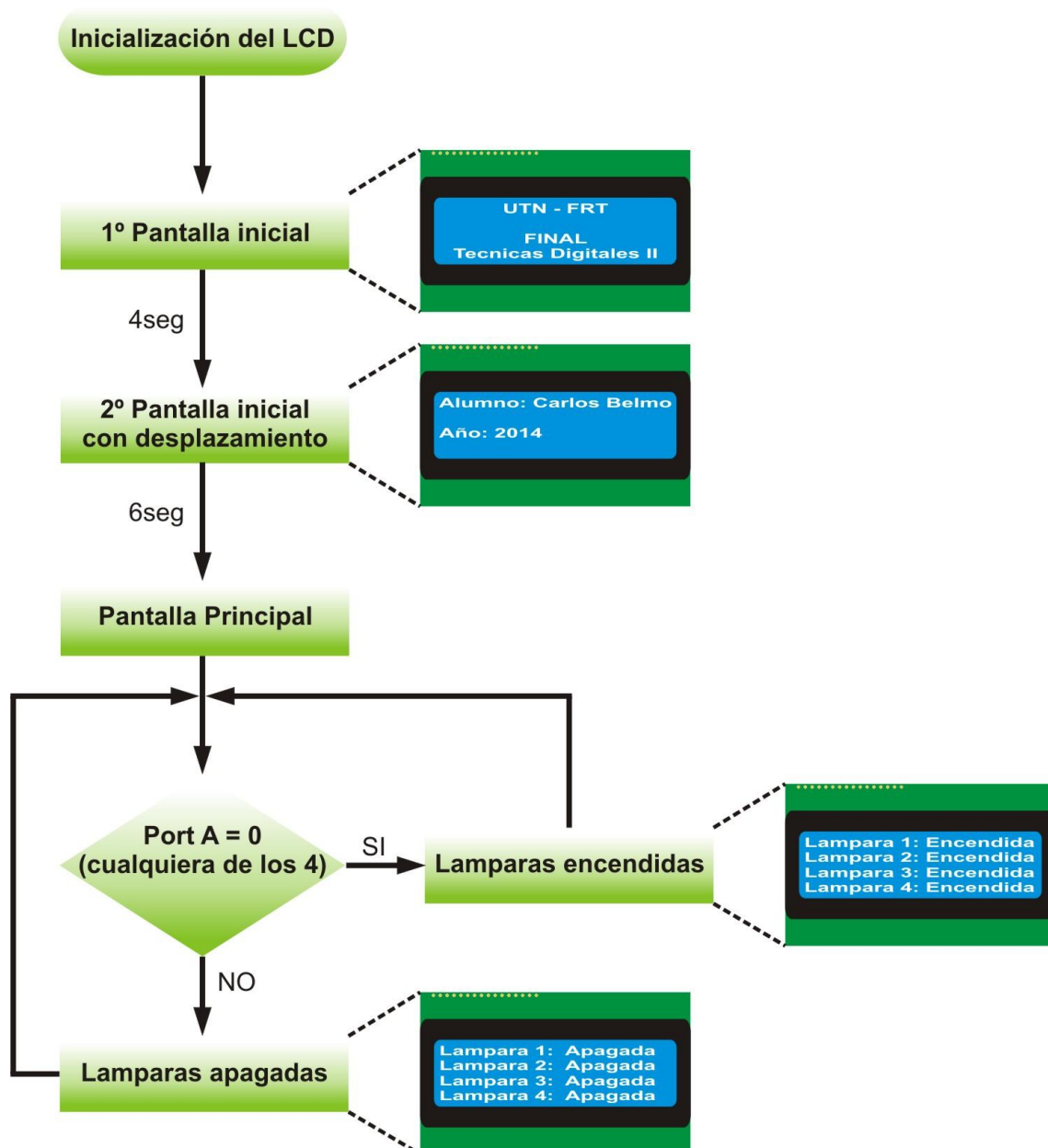


Si se activa cualquier llave del transmisor (o de las llaves externas) esto se compara en la llave conmutada, es decir si el transmisor envía un 1(nivel alto) y la llave externa envía un 0(nivel bajo) a la salida de la XOR tenemos un 1(nivel alto), este nivel va hacia los Darlington, teniendo un nivel bajo a la salida de estos, como 4 están conectados a relees y estos a su vez a 5V, el rele se activa, cerrando el circuito y encendiendo las lámparas. En los otros 4 Darlington también va a haber estados bajos (siempre que haya estado altos en XOR), y estos se conectan a la entrada del micro, este detalle es importante para tener en cuenta cuando se haga el programa que tome información de las entradas, el micro entenderá que cuando haya un 0 en sus entradas, las lámparas están ENCENDIDAS. Al final de la carpeta se adjunta el circuito completo del proyecto, para entender mejor su funcionamiento.

El display LCD contara con 3 pantallas, 2 iniciales y 1 principal.

En las iniciales se mostrara: UTN-FRT y el nombre de la materia (Técnicas Digitales II), en la otra pantalla mi nombre (Carlos Belmonte) con un desplazamiento de derecha a izquierda y el año (2014).

En la pantalla principal: Cada lámpara (en este caso 4) y sus respectivos estados (Encendida-Apagada).



Programación

La programación del micro QY4A para controlar el LCD la realice en assembler y con el programa WinIde Development Environment.

A continuación explico las partes más importante del programa y al final adjunto todo el programa, el cual está grabado en el micro. También se adjuntan al final las partes importantes del datasheet del micro QY4A.

Mapa de memoria: Lo primero que se hace es configurar las direcciones de memoria y de vectores que tiene el micro.



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

RAM_START	EQU \$0080	;Dirección de Inicio de la RAM
RAM_END	EQU \$00FF	;Dirección de Final de la RAM, 128Bytes de RAM para QY4A
FLASH_START	EQU \$EE00	;Dirección de Inicio de la FLASH
FLASH_END	EQU \$FDFF	;Dirección de Final de la FLASH, 4KBytes de RAM para QY4A
VECTORS	EQU \$FFDE	;Vectores de interrupción

Definir nombres a los pines de entrada/salida: Ciertos pines del puerto A fueron definidos con etiquetas "EQU" para su reconocimiento más rápido en el programa.

E	EQU 5	;PTA5 como salida, conectado al display
RS	EQU 4	;PTA4 como salida, conectado al display
IN_1	EQU 0	;PTA0 como entrada, estado de la lámpara
IN_2	EQU 1	;PTA1 como entrada, estado de la lámpara
IN_3	EQU 2	;PTA2 como entrada, estado de la lámpara
IN_4	EQU 3	;PTA3 como entrada, estado de la lámpara

Variables RAM y archivo include: Defino todas las variables RAM que utilizo, en este caso defino 2 auxiliares, pero no las utilizo. El archivo INCLUDE tiene todas las definiciones para el micro QY4A.

```
$BASE 10T                ;Base decimal por Default
$include 'QTQY_Registers.inc' ;Definiciones de Registros p/QY4A

*****VARIABLES DE RAM*****

      ORG RAM_START
aux1   ds      1
aux2   ds      1
```

Inicio de los registros del micro:

```
START:      ORG FLASH_START

            CLR OSCSC      ;Oscilador interno a 4 MHz
            RSP             ;Inicialización del micro, reset al Stack Pointer
            CLRA           ;Limpiar registro acumulador A
            CLRH           ;Limpiar registro índice H
            CLRX           ;Limpiar registro índice X
            CLC            ;Borra el bit de Carry
```

Config 1 y Config2: Los registros de configuración se utilizan en la inicialización de las diversas opciones. Los registros de configuración pueden escribirse una vez después de cada reinicio.

```
      mov     #%00001001,CONFIG1
;           \\ \\ \\ \\      b0: COPD= 1 COP deshabilitado
;           \\ \\ \\ \\      b1: STOP= 0 instrucción STOP deshabilitada
;           \\ \\ \\ \\      b2: SSREC= 0 Recup. de modo STOP 4096 ciclos
;           \\ \\ \\ \\      b3: LVITRIP= 1 LVI opera en 5V
;           \\ \\ \\ \\      b4: LVIPWRD= 0 LVI habilitado
;           \\ \\ \\ \\      b5: LVIRSTD= 0 Reset desde LVI habilitado
;           \\ \\ \\ \\      b6: LVISTOP= 0 LVI deshabilitado en STOP
;           \\ \\ \\ \\      b7: COPRS= 0 COP reset long cycle
      NOP      ;Hace de delay para asegurar la configuración
```



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

NOP		;Hace de delay para asegurar la configuración
mov	##10000000,CONFIG2	
;	\\ \\ \\ \\	b0: RSTEN= 0 Función inactiva en pin de reset
;	\\ \\ \\ \\	b1: OSCENINSTOP=0 Oscilador deshab en STOP
;	\\ \\ \\ \\	b2: Sin función
;	\\ \\ \\	b3: Sin función
;	\\ \\	b4: Sin función
;	\\	b5: Sin función
;	\\	b6: IRQEN= 0 Función de interrupción inactiva
;	\\	b7: IRQPUD= 1 Pull up interna desconecta
NOP		;Hace de delay para asegurar la configuración
NOP		;Hace de delay para asegurar la configuración

Pines de entrada/salida: Después de asignar una etiqueta a cada pin, se configura si el pin va a ser entrada o salida mediante el DDRA y DDRB colocando un 1 si es de salida y 0 si es de entrada.

mov	##FF,DDRB	; Puerto B todo como salida
mov	##00110000, DDRA	; PTA5 y PTA4 son salidas.
		; PTA3, PTA2, PTA1 Y PTA0 son entradas.
CLR PTA		; Limpia el puerto A
CLR PTB		; Limpia el puerto B
JSR RETARDO		; Salta a subrutina de retardo, para dar tiempo
		; y así asegurar la configuración

Programa para el manejo del display: Hasta acá se definió lo que hace el micro en cada arranque, ahora se explicara cómo se inicializa el display LCD como también las diferentes pantallas que presenta.

Aclaración: Durante la programación hice varios llamados a subrutinas para no cargar mucho el programa principal, ya que al compilar con Winlde salía error si estaba con demasiadas instrucciones este, por eso durante la explicación del programa, si hay un salto a subrutina, se la explicara debajo de esas líneas, lo que no quiere decir que lleve ese orden, en el programa original las subrutinas van al final.

Inicialización del LCD

INICIO_LCD:		;Prepara al LCD para modo 8BITS
LDA ##00111000		;Function Set: DL=1, bus de 8bits
		N=1, 4 líneas en pantalla
		F=0 matriz de 5x8
JSR ESCRIBE_IR		; Subrutina para mandar instrucciones al LCD
LDA ##00001100		;Display ON/OFF: D=1, se activa el display
		C=0, sin cursor
		B=0, sin parpadeo
JSR ESCRIBE_IR		; Subrutina para mandar instrucciones al LCD
LDA ##00000001		;Clear Display



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)

Alumno: Belmonte Carlos

Fecha: Mayo/2014

JSR ESCRIBE_IR ; Subrutina para mandar instrucciones al LCD

LDA #%00000110 ; Entry Mode Set: I/D=1, incrementa el cursor
; S=0, visualización normal

JSR ESCRIBE_IR ; Subrutina para mandar instrucciones al LCD

LDA #%00001100 ; Display ON/OFF

JSR ESCRIBE_IR

Rutina de mensajes en pantallas del display

MSJ_INICIO:

LDA #\$01 ; CLEAR DISPLAY

JSR ESCRIBE_IR

LDA #\$0C ; DISPLAY ON

JSR ESCRIBE_IR

LDA #%11000000 ; SET DDRAM \$40. Posición 0 de la 2da fila

;Aclaración: Los 7 primeros bits de LDA, generan el nº 40 en hex, el 8vo bit indica que se trabaja sobre la dirección de la DDRAM;

JSR LINEA_VACIA ; Línea vacía en la segunda fila

;Aclaración: Coloco líneas vacías, porque cuando no las ponías el mensaje en pantalla aparecía con todos los puntos rellenos de la matriz;

LDA #%10000111 ; SET DDRAM \$07. Posición 7 de la 1ra fila

JSR UTN_FRT ; Salta a subrutina que tiene el mensaje UTN-FRT

LDA #%10011100 ; SET DDRAM \$1C. Posición 8 de la 3ra fila

JSR FINAL ; Salta a subrutina que tiene el mensaje FINAL

;Aclaración: Como la forma de ingresar la frase "UTN-FRT" y "FINAL" son idénticas, solo se explicara una sola frase, ya que vale el mismo procedimiento para ambas; "LINEA VACIA" tiene el mismo procedimiento, solo que en vez de cargar letras se cargan "espacios" en el display

LDA #%11010100 ; SET DDRAM \$54. Posición 0 de la 4ta fila

JSR TDII ; Salta a subrutina que tiene el mensaje TDII

JSR RETARDO ; \

JSR RETARDO ; \

JSR RETARDO ; | Demora para pasar a la siguiente pantalla

JSR RETARDO ; /

JSR RETARDO ; /

LDA #\$01 ; CLEAR DISPLAY

JSR ESCRIBE_IR

LDA #%11000000 ; SET DDRAM \$40. Posición 0 de la 2da fila

JSR LINEA_VACIA ; Línea vacía en la segunda fila

LDA #%11010100 ; SET DDRAM \$54. Posición 0 de la 4ta fila

JSR LINEA_VACIA ; Línea vacía en la cuarta fila

LDA #%10010100 ; SET DDRAM \$14. Posición 0 de la 3ra fila

JSR AÑO ; Salta a subrutina que tiene el mensaje AÑO



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)

Alumno: Belmonte Carlos

Fecha: Mayo/2014

```
LDA #10000000 _____ ; SET DDRAM $00. Posición 0 de la 1ra fila
JSR ALUMNO _____ ; Salta a subrutina que tiene el mensaje ALUMNO
JSR CARLOS_BELMONTE _____ ; Salta a subrutina que tiene el mensaje BELMONTE
```

;Aclaración: La forma de ingresar la frase AÑO y ALUMNO es idéntica, pero la forma de ingresar BELMONTE tiene un agregado de desplazamiento de derecha a izquierda, así que se explicara más adelante;

```
JSR RETARDO _____ ; \
JSR RETARDO _____ ; \
JSR RETARDO _____ ; | Demora para pasar a la siguiente pantalla
JSR RETARDO _____ ; /
JSR RETARDO _____ ; /
```

```
LDA #$01 _____ ;CLEAR DISPLAY
JSR ESCRIBE_IR _____
```

Pantalla principal: después de los mensajes de inicio, el display muestra permanentemente esta pantalla.

LAMPARAS:

```
LDA #10000000 _____ ; SET DDRAM $00. Posición 0 de la 1ra fila
JSR LAMPARA_1 _____ ; Salta a subrutina que tiene el mensaje Lampara1
JSR ESTADO_1 _____ ; Salta a subrutina que controla estados (ON/OFF)
```

;Aclaración: La forma de escribir el mensaje LAMPARA es igual que los otros métodos de escritura, lo que varía en este caso son los estados (encendida/apagada), solo se explicara 1, porque el procedimiento es igual para las 4;

```
LDA #11000000 _____ ; SET DDRAM $40. Posición 0 de la 2da fila
JSR LAMPARA_2 _____ ; Salta a subrutina que tiene el mensaje Lampara2
JSR ESTADO_2 _____ ; Salta a subrutina que controla estados (ON/OFF)
```

```
LDA #10010100 _____ ; SET DDRAM $14. Posición 0 de la 3ra fila
JSR LAMPARA_3 _____ ; Salta a subrutina que tiene el mensaje Lampara3
JSR ESTADO_3 _____ ; Salta a subrutina que controla estados (ON/OFF)
```

```
LDA #11010100 _____ ; SET DDRAM $54. Posición 0 de la 4ta fila
JSR LAMPARA_4 _____ ; Salta a subrutina que tiene el mensaje Lampara4
JSR ESTADO_4 _____ ; Salta a subrutina que controla estados (ON/OFF)
```

```
JSR RETARDO _____
bra LAMPARAS _____ ; Salta siempre a LAMPARAS, se repite el ciclo
```

Subrutina para mandar instrucciones al LCD:

ESCRIBE_IR:

```
BCLR RS,PTA _____ ; Se pone en 0 para poder mandar instrucciones
BSET E,PTA _____ ; Se habilita el modulo/registros
STA PTB _____ ; Envía al PTB el valor del acumulador A
JSR DEMORA _____ ; Subrutina de retardo para cambio de nivel de E
BCLR E,PTA _____ ; Se deshabilita el modulo/registros
JSR DEMORA _____ ; Subrutina de retardo para cambio de nivel de E
BCLR RS,PTA _____ ; Se pone en 0 para poder mandar instrucciones
```



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

RTS

; Retorna de subrutina

Subrutina para mandar datos (carácter) al LCD:

ESCRIBE_DR:

BSET RS,PTA

; Se pone en 1 para poder mandar datos

;Aclaración: Si nos fijamos en la tabla de "Código de Instrucciones" del LCD, vemos que para poder escribir (mandar caracteres) RS=1 y RW=0;

BSET E,PTA

; Se habilita el modulo/registros

STA PTB

; Envía al PTB el valor del acumulador A

JSR DEMORA

; Subrutina de retardo para cambio de nivel de E

BCLR E,PTA

; Se deshabilita el modulo/registros

JSR DEMORA

; Subrutina de retardo para cambio de nivel de E

BCLR RS,PTA

; Se pone en 0 para poder mandar instrucciones

RTS

; Retorna de subrutina

Lámparas y sus estados: Solo se explica el procedimiento para una lámpara ya que para el resto es idéntico. También solo se explican unas cuantas letras, ya que el procedimiento es el mismo para las letras restantes que forman la palabra.

LAMPARA_1

JSR ESCRIBE_IR

; Subrutina para mandar instrucciones al LCD

LDX #\$12

; Carga en X el valor 12 (hexadecimal)

LDA TABLA,X ;-----;L

; Carga en A el dato almacenado en TABLA, que
; corresponde al valor de X

JSR ESCRIBE_DR

; Subrutina para mandar Carácter

;Aclaración: Genero una TABLA desde \$00 - \$32, donde están almacenado los bits de los caracteres, el procedimiento es el mismo para todas las letras que se quieran usar;

LDX #\$14

LDA TABLA,X ;-----;a

JSR ESCRIBE_DR

LDX #\$2D

LDA TABLA,X ;-----;m

JSR ESCRIBE_DR

LDX #\$1B

LDA TABLA,X ;-----;p

JSR ESCRIBE_DR

LDX #\$01

LDA TABLA,X ;-----;nro. 1

JSR ESCRIBE_DR

LDX #\$27

LDA TABLA,X ;-----;dos puntos

JSR ESCRIBE_DR

; Termina con este salto, sino borra la ultima letra

RTS

ESTADO_1



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

	BRCLR IN_1,PORTA,enc1	; Si hay un 0 salta a la subrutina enc1
	NOP	
	BRSET IN_1,PORTA,apa1	; Si hay un 1 salta a la subrutina apa1
	RTS	
enc1		; Subrutina que tiene el mensaje ENCENDIDA
	JSR ENCENDIDA	
	RTS	
apa1		; Subrutina que tiene el mensaje APAGADA
	JSR APAGADA	
	RTS	

Subrutina con mensaje “Encendida/Apagada”: este mensaje es el mismo para todas las lámparas, es decir que todas saltan a esta misma subrutina.

ENCENDIDA

```
LDX #$24
LDA TABLA,X ;-----;espacio
JSR ESCRIBE_DR
LDX #$0F
LDA TABLA,X ;-----;E
JSR ESCRIBE_DR
LDX #$28
LDA TABLA,X ;-----;n
JSR ESCRIBE_DR
LDX #$15
LDA TABLA,X ;-----;c
JSR ESCRIBE_DR
RTS
```

; Aclaración: Como el modo de cargar el mensaje es el mismo que el explicado anteriormente, solo se muestran a modo de ejemplo unas cuantas letras del mensaje;

APAGADA

```
LDX #$24
LDA TABLA,X ;-----;espacio
JSR ESCRIBE_DR

LDX #$24
LDA TABLA,X ;-----;espacio
JSR ESCRIBE_DR

LDX #$2E
LDA TABLA,X ;-----;A
JSR ESCRIBE_DR

LDX #$1B
LDA TABLA,X ;-----;p
JSR ESCRIBE_DR

LDX #$14
LDA TABLA,X ;-----;a
JSR ESCRIBE_DR
```



RTS

Mensajes UTN-FINAL-TDII-ALUMNO-AÑO: Todos estos mensajes tienen la misma forma de ingreso que el anterior explicado, por eso solo se detallaran las 3 letras de UTN.

```
UTN_FRT
    JSR ESCRIBE_IR                ; Subrutina para mandar instrucciones al LCD
    LDX #$0D                     ; Carga en X el valor 0D
    LDA TABLA,X                  ;U      ; Carga en A el dato almacenado en TABLA, que
                                   ; corresponde al valor de X
    JSR ESCRIBE_DR                ; Subrutina para mandar Carácter
;Aclaración: Genero una TABLA en orden desde $00 - $32, donde están almacenado los bits de los
caracteres, el procedimiento es el mismo para todas las letras que se quieran usar;

    LDX #$0C                     ; Carga en X el valor 0C
    LDA TABLA,X                  ;T      ; Carga en A el dato almacenado en TABLA
    JSR ESCRIBE_DR                ; Subrutina para mandar Carácter

    LDX #$13                     ; Carga en X el valor 13
    LDA TABLA,X                  ;N      ; Carga en A el dato almacenado en TABLA
    JSR ESCRIBE_DR                ; Subrutina para mandar Carácter
    RTS
```

Como se puede ver hasta acá, el envío de mensajes al display es bastante sencillo y repetitivo, hay que tener primeramente en cuenta cuantas letras se va a mandar por línea y en qué posición del LCD se va a encontrar la primera letra de ese mensaje, una vez planteado eso, se envía las instrucciones que ejecutan lo que queremos.

Desplazamiento del mensaje: Siguiendo la idea anteriormente mencionada, una forma para mover el mensaje sería variar la posición en que se encuentra la letra (ya sea una posición más o una posición menos) y dándole un pequeño delay para que se vea el efecto de desplazamiento. En el mensaje “Carlos Belmonte” se sigue esta idea pero se deja constante una posición del LCD y lo que se mueven son las letras, es decir si hay una “C” en el estado siguiente habrá una “a” letra por letra.

Como el proceso es repetitivo, solo se explicaran unas cuantas letras.

```
CARLOS_BELMONTE
    JSR ESCRIBE_IR                ; Subrutina para mandar instrucciones al LCD
    LDX #$24                     ; Carga en X el valor 24
    LDA TABLA,X ;-----;ESPACIO ; Carga en A el dato almacenado en TABLA, que
                                   ; corresponde al valor de X
    JSR ESCRIBE_DR                ; Subrutina para mandar Carácter

    LDX #$0E                     ; Carga en X el valor 0E
    LDA TABLA,X ;-----;C
    JSR ESCRIBE_DR

    LDX #$14                     ; Carga en X el valor 14
    LDA TABLA,X ;-----;a
```



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

JSR ESCRIBE_DR

LDX #\$1C

LDA TABLA,X ;-----;r

JSR ESCRIBE_DR

JSR RETARDO_____ ; \

JSR RETARDO_____ ; | Demora para ver el desplazamiento

JSR RETARDO_____ ; /

LDA #%10000111_____ ; SET DDRAM \$07. Posición 7 de la 1ra fila

; Aclaración: Después del espacio, se encontraba la letra C, ahora después del espacio se encuentra la letra a;

JSR ESCRIBE_IR

LDX #\$24

LDA TABLA,X ;-----;ESPACIO

JSR ESCRIBE_DR

LDX #\$14

LDA TABLA,X ;-----;a

JSR ESCRIBE_DR

LDX #\$1C

LDA TABLA,X ;-----;r

JSR ESCRIBE_DR

LDX #\$19

LDA TABLA,X ;-----;l

JSR ESCRIBE_DR

; Aclaración: Al desaparecer la letra C, aparece al final la letra l;

JSR RETARDO

JSR RETARDO

JSR RETARDO

LDA #%10000111_____ ; SET DDRAM \$07. Posición 7 de la 1ra fila

JSR ESCRIBE_IR

LDX #\$24

LDA TABLA,X ;-----;ESPACIO

JSR ESCRIBE_DR

LDX #\$1C

LDA TABLA,X ;-----;r

JSR ESCRIBE_DR

LDX #\$19

LDA TABLA,X ;-----;l

JSR ESCRIBE_DR

LDX #\$1A



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

LDA TABLA,X ;-----;o

JSR ESCRIBE_DR

; Aclaración: Al desaparecer la letra a, aparece al final la letra o;

RTS

Retardos: El primero sirve para visualización de los mensajes, mientras que el segundo sirve para cambiar de estado entre envió de datos o de comandos.

RETARDO

PSHA	;[2]	Inserta A en el Stack, para guardar los datos que habían
PSHH	;[2]	Inserta H en el Stack, para guardar los datos que habían
PSHX	;[2]	Inserta X en el Stack, para guardar los datos que habían
LDA #\$FF	;[2]	Carga en A el valor 255

DELAY

LDHX #\$00EF	;[3]	Carga en H:X el valor 239
--------------	------	---------------------------

LOOP1

AIX #-1	;[2]	Al valor de H:X se le suma (-1)
CPHX #0	;[3]	Compara el valor de H:X con el valor 0
BNE LOOP1	;[3]	Salta a LOOP1 si no es igual
DECA	;[1]	Decrementa y salta si no es cero
BNE DELAY	;[3]	Salta a DELAY si no es igual
PULX	;[2]	Saca X en el Stack
PULH	;[2]	Saca H en el Stack
PULA	;[2]	Saca A en el Stack
RTS	;[4]	

Como el micro trabaja a 4Mhz, el Fbus=1Mhz, entonces Ciclo T= 1µseg

Tiempo total= Tciclo x Nº ciclos= (((3+3+2) x239)+1+3+3) x255)+8+6+4) x 1µseg ≈ 0,5seg

DEMORA:

PSHA	[2]	Inserta A en el Stack, para guardar los datos que habían
LDA #\$FF	[2]	Carga en A el valor 255

DELAY2:

DECA	[1]	Decrementa y salta si no es cero
BNE DELAY2	[3]	Salta a DELAY2 si no es igual
PULA	[2]	Saca A en el Stack
RTS	[4]	

Como el micro trabaja a 4Mhz, el Fbus=1Mhz, entonces Ciclo T= 1µseg

Tiempo total= Tciclo x Nº ciclos= (((1+3) x255)+2+4+2+2) x 1µseg ≈ 1mseg

Tabla de caracteres: se define una tabla en donde se encuentran todos los caracteres utilizados. Los valores en binario representan los valores de cada letra en el display, y los valores en hexadecimal indican en qué posición se van ubicando esos valores en el micro.

TABLA:

DB %00110000;0	X = \$00
DB %00110001;1	X = \$01
DB %00110010;2	X = \$02
DB %00110011;3	X = \$03
DB %00110100;4	X = \$04



Técnicas Digitales II (FINAL)

Profesora: Ing. A.M. Frenzel
Alumno: Belmonte Carlos

Tema: Encendido-Apagado de lámparas mediante control remoto (RF)
Fecha: Mayo/2014

DB %00110101;5	X = \$05
DB %00110110;6	X = \$06
DB %00110111;7	X = \$07
DB %00111000;8	X = \$08
DB %00111001;9	X = \$09
DB %01010000;P	X = \$0A
DB %01010010;R	X = \$0B
DB %01010100;T	X = \$0C
DB %01010101;U	X = \$0D
DB %01000011;C	X = \$0E
DB %01000101;E	X = \$0F
DB %01000110;F	X = \$10
DB %01001000;H	X = \$11
DB %01001100;L	X = \$12
DB %01001110;N	X = \$13
DB %01100001;a	X = \$14
DB %01100011;c	X = \$15
DB %01100111;g	X = \$16
DB %01101001;i	X = \$17
DB %01101010;j	X = \$18
DB %01101100;l	X = \$19
DB %01101111;o	X = \$1A
DB %01110000;p	X = \$1B
DB %01110010;r	X = \$1C
DB %01110011;s	X = \$1D
DB %01110101;u	X = \$1E
DB %01100010;b	X = \$1F
DB %01110001;q	X = \$20
DB %01101000;h	X = \$21
DB %01101011;k	X = \$22
DB %01111010;z	X = \$23
DB %00100000;ESPACIO	X = \$24
DB %01110111;w	X = \$25
DB %01100101;e	X = \$26
DB %00111010;dos puntos	X = \$27
DB %01101110;n	X = \$28
DB %01000010;B	X = \$29
DB %01100100;d	X = \$2A
DB %01001010;J	X = \$2B
DB %01110110;v	X = \$2C
DB %01101101;m	X = \$2D
DB %01000001;A	X = \$2E
DB %01001001;l	X = \$2F
DB %01000100;D	X = \$30
DB %01110100;t	X = \$31
DB %11101110;ñ	X = \$32