ITESO

Universidad Jesuita
de Guadalajara

PRÁCTICA 4

“LCD, puerto serial y teclado”

FUNDAMENTOS DE MICROPROCESADORES Y MICROCONTROLADORES

INTEGRANTES:

Lilia Arceli Lobato Martínez	IE706937
Jorge Karim Naciff Maldonnat	IE708501
Gustavo Gutiérrez Iñiguez	IE709414

19-ABRIL-2018

INTRODUCCIÓN

El objetivo principal de esta práctica es armar una conexión entre el microprocesador y otros dispositivos como una pantalla LCD, la cual trabaja con un código ascii, pero diferente al que utilizan las computadoras comunes, este es un código ascii japonés. Al mismo tiempo se debe hacer uso del puerto serial para mandarlo a otros dispositivos mediante un módulo Bluetooth o mediante el módulo Rs-232.

Este programa debe de obtener el dato de un teclado matricial, el cual para evitar muchos posibles errores utilizamos un decodificador de teclado matricial. Con ello, el micro debe de procesar la información recibida y mandarla a la pantalla. Asimismo debe de contar con boton de reset, un botón que sirva como Alt, para introducir caracteres en ascii y otro para enviar la información por el serial.

La pantalla sirve mediante códigos que podemos separar entre instrucciones para la pantalla y datos para la pantalla, eso es mediante una configuración que se debe de tomar en cuenta tanto en el cableado como en el código.

Como codificador utilizamos un MM74C922, es un codificador CMOS para una matriz de 16, es decir, admite 4 filas y 4 columnas, lo necesario para codificar un teclado matricial a la perfección.

DESARROLLO TEÓRICO

Pantalla LCD

Un LCD de dos líneas de 16 caracteres de matriz de 5x7 puntos. El controlador incluye un generador de caracteres en ROM con 192 caracteres y la posibilidad de definir otros 8 más en RAM. Además tiene una RAM de datos del display donde se almacena el mensaje a visualizar de 80 bytes, de los cuales se visualizan 32(2x16).

Patillaje

SEÑAL	PIN	FUNCIÓN
DB0-3	8-5	Nibble bajo del bus de datos
DB4-7	4-1	Nibble alto del bus de datos
E	9	Señal de habilitación (Enable) del LCD
R/W	10	Selección lectura escritura 1 Leer | 0 Escribir
RS	11	Selección registro instrucción datos 0 IR | 0 DR
V_{LS}	12	Tensión de contraste, unir al cursor del potenciómetro
V_{DD}	14	Tensión de alimentación, 5v

V_{SS}	13	Conectar a GND

Operaciones

El controlador tiene dos registros, el de instrucciones (IR) y el de datos (DR), que son seleccionador mediante la señal RS. Estos registros pueden leerse o escribirse según indique la señal R/W, de modo que son posibles 4 operaciones diferentes:

0	0	Escribir en IR: Borrar Display, etc., o modificar AC.
0	1	Leer el IR: Leer el AC(DB0-6) y el Busy Flag (DB7).
1	0	Escribir en DR: Escribir en DD RAM o CG RAM
1	1	Leer el DR: Leer de DD RAM o CG RAM

Escribir en el IR: Se envían instrucciones y también para escribir una nueva dirección en el registro AC (Address Counter), que es el registro encargado de apuntar, tanto a la DD RAM (RAM de Datos del Display) como a la CG RAM (RAM de Generador de Caracteres)

Leer del IR: Permite la lectura del registro AC, del cual sólo son válidos los 7 bits de menos peso DB0-6, el bit de mayo peso, el DB7, informa del estado del Busy Flag o indicador del Display Ocupado.

Escribir en el DR: Permite escribir en DD RAM o CG RAM allí donde se encuentre apuntado el registro AC.

Leer del DR: Permite leer de DD RAM o CG RAM allí donde apunta el AC

Busy Flag (BF)

A "1" indica que el LCD está ocupado realizando operaciones internas y no puede aceptar nuevas instrucciones. Hay que esperar que el Busy Flag valga "0" para enviarle la siguiente instrucción.

Contador de Dirección (AC)

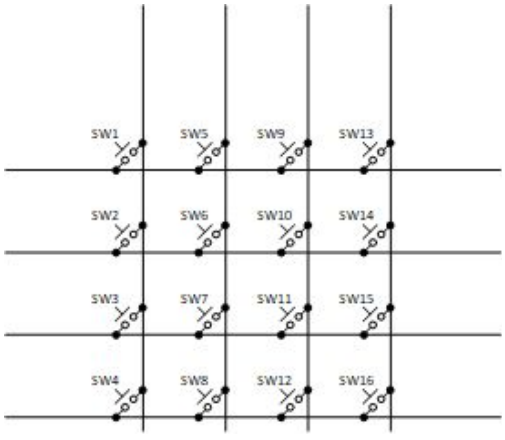
Indica la dirección donde serán leídos o escritos los datos sobre DD RAM o CG RAM. Este registro puede modificarse realizando una escritura en el IR.

Además, cuando se escriben o se leen datos en/de DD RAM o CG RAM, el AC se incrementa o decrementa de manera automática de acuerdo con el Modo de Entrada (Entry Mode Set).

RAM de Datos del Display (DD RAM): Tiene una capacidad de 80 bytes, 40 para cada línea; lógicamente solo 32 de los 80 bytes podrán ser visualizados a la vez, aunque desplazando (shift) el display podrán irse visualizando todos los caracteres escritos.

Generador de caracteres en ROM (CG ROM): Tiene 192 caracteres, en matrices de 5x7 puntos.

Generador de caracteres en RAM (CG RAM): Permite definir 8 caracteres, cuyos códigos van desde el 00 al 08, o desde el 09 al 0f.



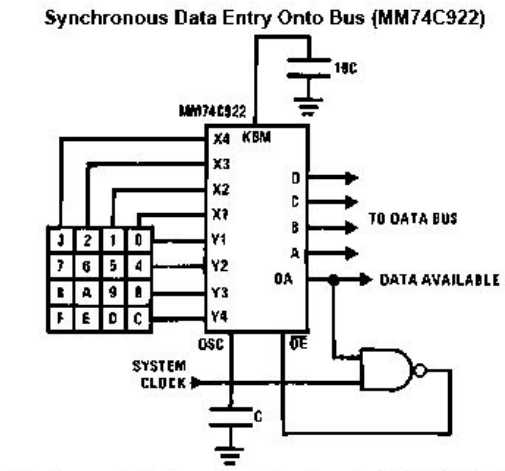
Teclado Matricial

Un teclado matricial es aquel que tiene organizadas las conexiones de las teclas en forma de n columnas y m renglones, tal como una matriz.

Se hacen de este modo, ya que con n + m líneas de conexión se pueden leer n x m teclas. Por ejemplo, un teclado de 16 botones se puede leer con tan solo 8 líneas. Sin embargo, esto incrementa un poco la complejidad del

algoritmo de lectura ya que es necesario hacer hasta n lecturas m veces para encontrar la tecla presionada, es decir una lectura por tecla.
Para facilitar la lectura utilizamos un codificador MM74C922

Codificador MM74C922



Los codificadores de teclado MM74C922 y MM74C923 CMOS proporcionan la lógica necesaria para codificar completamente un teclado matricial.
Tiene un circuito de supresión de rebotes interno por lo que la salida de datos a nivel alto, cuando se ha hecho una entrada de teclado válido, se muestra "limpia". Los datos de salida disponibles regresan a un nivel bajo cuando la tecla ingresada se ha liberado, incluso si otra de las teclas está oprimida. El datos disponible volverá a alto para indicar la aceptación de la nueva tecla después de un período de supresión de rebotes normales.

DESARROLLO PRÁCTICO

El Código en ensamblador que desarrollamos ha sido el siguiente:

```
;P2.0 ---------- Enable LCD
;P2.1 ---------- RS LCD (0 Instrucción, 1 Dato)
;P3.4 ---------- ALTBTN (0 Si no está presionaldo alt, 1 si es que si)

E           EQU  P2.0
RS          EQU  P2.1
ALTBTN      EQU P3.4
DATAAV      EQU P3.2

ORG 0000H
LJMP INICIO

ORG 0003H
JMP TECLADO

ORG 0013H
```

```
JMP SEND ;NUDES


                ORG 0040H
INICIO:     MOV TMOD,#21H     ; MODO 0 TMP 1
                MOV IE,#85H       ; HABILITA INTERRUPCIÓN EXT0
                MOV SCON,#40H
                MOV TH1,#0FDH
                MOV TL1,#-3D
                SETB TR1
                ACALL INIT        ; LLAMA A INICIALIZAR LCD
                ACALL EQUIPO9     ; PRESENTACIÓN DEL EQUIPO
                MOV P1,#0H        ; PUERTO 1 EN 0 DESPUES DE EQUIPO9
                MOV R5,#0H        ; CONTADOR DE CARACTERES EN LCD EN 0
                MOV R6,#0H        ; CONTADOR DE DOS NUMEROS EN ALT EN 0
                MOV R0,#0H        ; VARIABLE DONDE SE ALMACENA DATO A
IMPRIMIR EN 0
                MOV R4,#60H
                MOV R2,#60H
                MOV R3,#60H
                MOV R1,#40H

////////////////////////////////////////////////////////////////////////////////
; TABLA DE VALORES DE ASCII A HEXADECIMAL
////////////////////////////////////////////////////////////////////////////////
                MOV 30H, #48D     ;1
                MOV 31H, #49D     ;2
                MOV 32H, #50D     ;3
                MOV 33H, #51D     ;A
                MOV 34H, #52D     ;4
                MOV 35H, #53D ;5
                MOV 36H, #54D     ;6
                MOV 37H, #55D     ;B
                MOV 38H, #56D     ;7
```

```
                MOV 39H, #57D       ;8
                MOV 3AH, #65D       ;9
                MOV 3BH, #66D       ;C
                MOV 3CH, #67D;F
                MOV 3DH, #68D       ;0
                MOV 3EH, #69D       ;E
                MOV 3FH, #70D       ;D
                SJMP $

INIT:       ACALL WAIT          ; TIEMPO DE ESPERA TÍPICO
                MOV A, #38H         ; 2 LÍNEAS, MATRÍZ DE 5*8
                ACALL MANDAR        ; ENVÍA LA INSTRUCCIÓN A LA LCD
                MOV A, #38H         ; 2 LÍNEAS, MATRÍZ DE 5*8
                ACALL MANDAR        ; ENVÍA LA INSTRUCCIÓN A LA LCD
                MOV A, #38H         ; 2 LÍNEAS, MATRÍZ DE 5*8
                ACALL MANDAR        ; ENVÍA LA INSTRUCCIÓN A LA LCD

REINICIO:   MOV A, #01H         ; LIMPIA LA LCD
                ACALL MANDAR        ; ENVÍA LA INSTRUCCIÓN A LA LCD
                MOV A, #0FH         ; DISPLAY, CURSOR Y PARPADEO PRENDIDOS
                ACALL MANDAR        ; ENVÍA LA INSTRUCCIÓN A LA LCD
                MOV A, #80H         ; CURSOR LÍNEA 1, POSICIÓN 1 (DDRAM DIRECCIÓN
EN 0)
                ACALL MANDAR        ; ENVÍA LA INSTRUCCIÓN A LA LCD
                RET                 ; RETORNA A DONDE SE LLAMÓ

////////////////////////////////////////////////////////////////////////////////////////////

WAIT:       MOV TL0, #0B6H  ; PARTE INFERIOR (057 PARA 25ms)
                MOV TH0, #03CH      ; PARTE SUPERIOR (09E PARA 25ms
                SETB TR0            ; INICIALIZA EN 1 PARA INICIAR CONTEO
                JNB TF0,$           ; 50ms
                CLR TF0             ; LIMPIA BANDERA DE DESBORDE
```

```
                RET                     ; RETORNA A DONDE SE LLAMÓ

////////////////////////////////////////////////////////////////////////////////

MANDAR:         CLR RS          ; RS MODO INSTRUCCIÓN
                ACALL WAIT      ; TIEMPO DE ESPERA TÍPICO
                CLR E           ; PREPARANDO EL...
                ACALL WAIT      ; TIEMPO DE ESPERA TÍPICO
                SETB E          ; ...ENABLE
                ACALL WAIT      ; TIEMPO DE ESPERA TÍPICO
                MOV P1, A       ; ACUMILADOR A PUERTO 1 (LCD)
                ACALL WAIT      ; TIEMPO DE ESPERA TÍPICO
                CLR E           ; FLANCO DE BAJADA ENABLE (MANDA
INSTRUCCIÓN)
                ACALL WAIT      ; ESPERA TÍPICA
                RET                     ; RETORNA A DONDE SE LLAMÓ

////////////////////////////////////////////////////////////////////////////////

ESCRIBE:    SETB RS             ; RS EN MODO DE DATO
                ACALL WAIT      ; TIEMPO DE ESPERA TÍPICO
                CLR E           ; PREPARANDO EL...
                ACALL WAIT      ; TIEMPO DE ESPERA TÍPICO
                SETB E          ; ...ENABLE
                ACALL WAIT      ; ESPERA TÍPICA
                MOV P1, A       ; ACUMULADOR A PUERTO 1 (LCD)
                ACALL WAIT      ; ESPERA TÍPICA
                CLR E           ; FLANCO DE BAJADA ENABLE (MANDA
INSTRUCCIÓN)
                ACALL WAIT      ; ESPERA TÍPICA
                RET                     ; RETORNA A DONDE SE LLAMÓ

////////////////////////////////////////////////////////////////////////////////
```

```
EQUIPO9:    MOV A,#69D          ; E
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#81D          ; Q
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#85D          ; U
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#73D          ; I
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#80D          ; P
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#79D          ; O
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#32D          ; " "
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV A,#57D          ; 9
            ACALL ESCRIBE       ; ESCRIBE EN LCD
            MOV R7,#20D         ; REPETICIONES DE 50ms

LGK:        MOV TL0, #0H   ; PARTE INFERIOR
            MOV TH0, #0H   ; PARTE SUPERIOR
            SETB TR0            ; INICIALIZA EN 1 PARA INICIAR CONTEO
            JNB TF0,$           ; 50ms
            CLR TF0                     ; LIMPIA BANDERA DESBORDAMIENTO
            DJNZ R7, LGK   ; EJECUTA 20 VECES
            ACALL REINICIO      ; REINICIA PANTALLA
            RET                         ; RETORNA A DONDE SE LLAMÓ

////////////////////////////////////////////////////////////////////////////////////////////

TECLADO:    CJNE R5,#10H,PLLENA     ; COMPARA QUE LA PRIMERIA LÍNEA NO ESTE
LLENA
```

```
                MOV  A,#0C0H            ; SI SE LLENA SE CAMBIA A LA SIGUIENTE LÍNEA
                ACALL MANDAR            ; MANDA INSTRUCCIÓN A LCD

PLLENA:         CJNE R5,#20H,ELCD       ; COMPARA QUE LA SEGUNDA LÍNEA NO ESTE LLENA
                MOV A,#01H              ; CUANDO ESTA LLENA BORRA TODO
                ACALL MANDAR            ; MANDA INSTRUCCIÓN A LCD
                MOV R5,#0H              ; LIMPIA CONTADOR DE CARACTERES
                RETI                    ; SALE DE LA INTERRUPCIÓN DEL TECLADO

////////////////////////////////////////////////////////////////////////////////////////

ELCD:           JB ALTBTN,ALT           ; VERIFICA QUE NO ESTÉ PRESIONADO EL PUSHBTN ALT
                MOV A,P0                ; MUEVE DATOS DEL TECLADO AL ACUMULADOR
                ADD A,#30H              ; LO POSICIONA EN SU DECODIFICACIÓN CORRESPONDIENTE
                MOV R0,A                ; GUARDA ACUMULADOR EN R0 PARA USARLO COMO APUNTADOR
                MOV A,@R0               ; APUNTA AL VALOR EN LA DIRECCION DONDE ESTÁ R0
                ACALL TLCD              ; LLAMA A TECLADO LCD (TLCD)
                MOV @R1,A
                INC R1
                RETI                    ; SALE DE LA INTERRUPCIÓN DEL TECLADO

////////////////////////////////////////////////////////////////////////////////////////

TLCD:           SETB RS                 ; RS EN MODO DE DATO
                ACALL WAIT              ; ESPERA TÍPICA
                CLR E                   ; PREPARANDO EL...
```

```
                ACALL WAIT              ; ESPERA TÍPICA
                SETB E                  ; ...ENABLE
                ACALL WAIT              ; ESPERA TÍPICA
                MOV P1,A                ; ACUMULADOR A PUERTO 1 (LCD)
                ACALL WAIT              ; ESPERA TÍPICA
                CLR E                   ; FLANCO DE BAJADA ENABLE
(MANDA INSTRUCCIÓN)
                ACALL WAIT              ; ESPERA TÍPICA
                CLR RS                  ; RS EN MODO INSTRUCCIÓN
                ACALL WAIT              ; ESPERA TÍPICA
                INC R5                  ; INCREMENTA CONTADOR DE
CARACTERES
                RET                     ; RETORNA A DONDE SE
LLAMÓ

////////////////////////////////////////////////////////////////////////////////

ALT:        JNB DATAAV,ALT
                MOV A,P0                ; MUEVE DATOS DEL TECLADO AL
ACUMULADOR
                CJNE R6,#01H,PD         ; MANDA A RUTINA PARA GUARDAR
LOS DOS VALORES PRESIONADOS
                ADD A,B                 ; SUMA PRIMER VALOR
(PARTE ALTA) CON SEGUNDO (PARTE BAJA)
                MOV R6,#0H              ; PONE EN 0 EL COMPARADOR
                ACALL TLCD              ; LLAMA A TECLADO LCD (TLCD)
                MOV @R1,A
                INC R1
                RETI                    ; SALE DE LA INTERRUPCIÓN DEL
TECLADO

////////////////////////////////////////////////////////////////////////////////

PD:             SWAP A                  ; MANDA LA PARTE BAJA DE
"A" A LA PARTE ALTA
```

```
                MOV B,A                         ; MUEVE "A" A LA VARIABLE
"B"
                INC R6                          ; PONE EN 1 EL COMPARADOR
                RETI                            ; SALE DE LA INTERRUPCIÓN DEL
TECLADO

////////////////////////////////////////////////////////////////////////////////

SEND:       MOV A,R1
                SUBB A,#40H
                JZ SALIR
                MOV R0,#40H
                MOV 60H,A
                SETB TR1

SER:        MOV SBUF,@R0
                ACALL WAIT
                JNB TI,$
                ACALL WAIT
                CLR TI
                ACALL WAIT
                INC R0
                DJNZ 60H,SER
                CLR TR1
                MOV R1,#40H
                ACALL REINICIO

SALIR:      RETI

////////////////////////////////////////////////////////////////////////////////

END
```

El código generado en Hex80 es el siguiente:

```
:03000000020040BB
:02000300211FBB
:02001300216E5C
:100040007589217 5A885759840758DFD758BFDD2D4
:100050008E119811EB7590007D007E0078007C6019
:100060007A607B607940753030753131753232752 8
:10007000333375343475353575363675373775 38ED
:1000800038753939753A41753B42753C43753D4485
:10009000753E45753F4680FE11B3743811C1743802
:1000A00011C1743811C1740111C1740F11C1748070
:1000B00011C122758AB6758C3CD28C308DFDC28DF3
:1000C00022C2A111B3C2A011B3D2A011B3F59011F5
:1000D000B3C2A011B322D2A111B3C2A011B3D2A056
:1000E00011B3F59011B3C2A011B322744511D674A7
:1000F0005111D6745511D6744911D6745011D67455
:100100004F11D6742011D6743911D67F14758A0018
:10011000758C00D28C308DFDC28DDFF111A622BD11
:10012000100474C011C1BD2007740111C17D0032DB
:1001300020B425E5802430F8E6313EF70932D2A11B
:1001400011B3C2A011B3D2A011B3F59011B3C2A0E4
:1001500011B3C2A111B30D2230B2FDE580BE010979
:1001600025F07E00313EF70932C4F5F00E32E994F5
:0701700040601D7840F560BE
:10017700D28E869911B33099FD11B3C29911B30884
:0A018700D560EFC28E794011A63258
:00000001FF
```

EXPLICACIÓN DEL DESARROLLO

Lo primero fue conseguir el material del cual, algunos componentes se nos fueron prestados por la institución. Dentro de los materiales utilizados fueron:

- 1 Microcontrolador AT89S52
- 1 Display LCD 16x2
- 1 Teclado matricial
- Encoder MM79C922
- Cristal de cuarzo de 11.059 Mhz
- 2 Protoboards
- Modulo Bluetooth HC-06
- Resistencias y capacitores
- 3 Push buttons

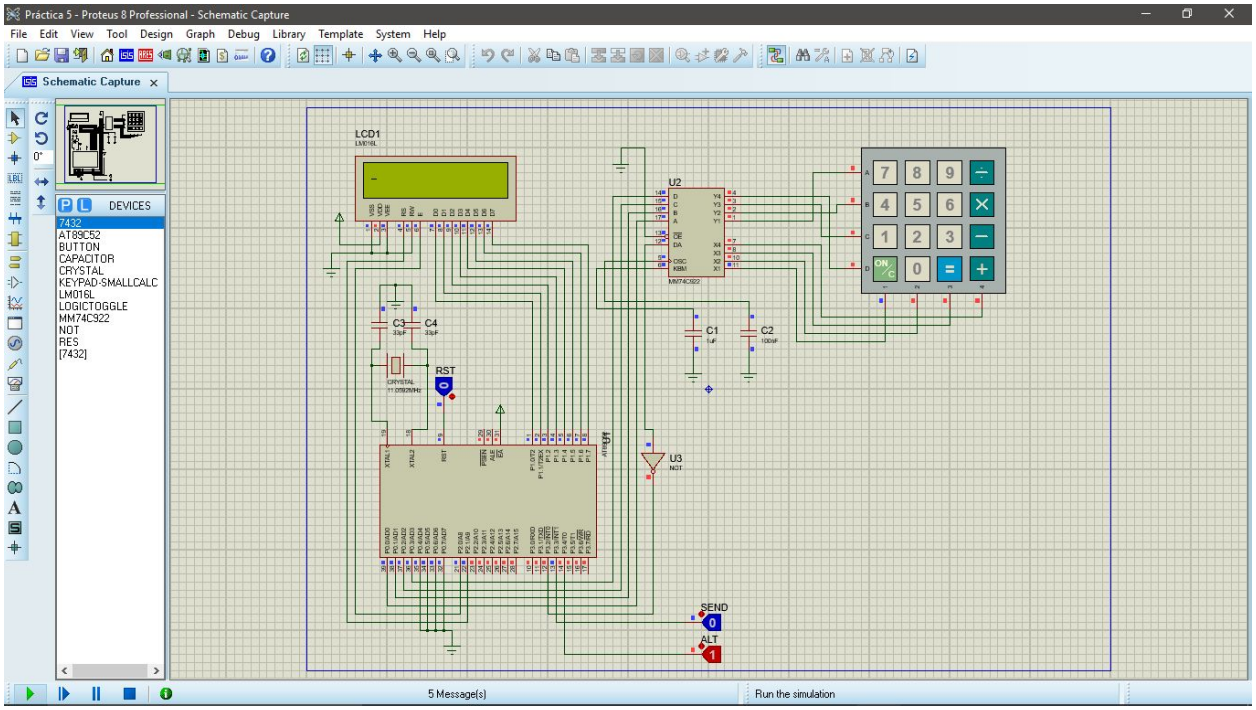
Con esta práctica tuvimos que comenzar con el proceso de diseño que no era necesario en las prácticas pasadas, ya que se nos daba un esquemático de cómo debe ser el cableado de los componentes; ahora no fue el caso, por lo que fue de nuestra cuenta decidir qué puertos íbamos a usar, qué componentes iban conectados a qué parte y sobre todo saber explicar el por qué de las cosas que hicimos.

Ya habíamos trabajado un poco con el teclado matricial en diseño digital, por lo que teníamos un poco de experiencia en los problemas que podía presentar, como los rebotes y el reacomodo del mismo. Por ello decidimos utilizar un codificador especial para este proceso, que toma las filas y las columnas del teclado y nos permite clasificarlo como una matriz, además lidiando con el problema de los rebotes, ya que cuenta con un debouncer propio.

Para poder utilizar la pantalla, en nuestro programa utilizamos una rutina que espera 50 ms, lo suficiente y sobrado para que no haya problemas con que la pantalla no lea datos o instrucciones.

Al momento de mandar el dato por el puerto serial, se nos presentó el caso que no manda el mismo dato en la pantalla como en el puerto, ya que estos trabajan con diferentes códigos ascii, pero tomando en cuenta este detalle, podemos mandar los valores correctos por el serial tomando en cuenta su valor en hexadecimal.

DIAGRAMA DE BLOQUES



PRINCIPALES DIFICULTADES

Es un error en conceptos teóricos lo que nos detuvo por unas horas. Al imprimir en la pantalla un valor Hexadecimal, mandar este por bluetooth e imprimirlo en la pantalla de un celular, los caracteres ASCII varían pero el código en hexadecimal se mantenía.

Para entender este problema nos pusimos a investigar y nos encontramos que el código ASCII se divide en "caracteres imprimibles" y "Alfabeto extendido". A continuación explicamos cada uno:

USASCII code chart

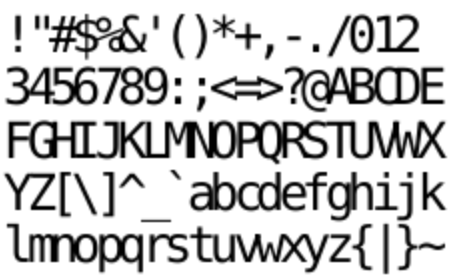
Bits b_7 b_6 b_5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 ↓ Row / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Caracteres imprimibles:

es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno. Fue creado en 1963 por el Comité Estadounidense de Estándares como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía. Más tarde, en 1967, se incluyeron las minúsculas, y se redefinieron algunos códigos de control.

En la actualidad define códigos para 32 caracteres no imprimibles, de los cuales la mayoría son caracteres de control que tienen efecto sobre cómo se procesa el texto, más otros 95 caracteres imprimibles que les siguen en la numeración (empezando por el carácter espacio).

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión. A menudo se llama incorrectamente ASCII a varios códigos de caracteres de 8 bits que extienden el ASCII con caracteres propios de idiomas distintos al inglés, como el estándar ISO/IEC 8859-1 justo este es el error que encontramos. La pantalla trabaja con un ASCII japonés mientras que el celular trabaja sobre un ASCII mexicano, teóricamente, si el idioma del celular se cambia a japonés, se verían los mismos caracteres.



ASCII Extendido:

A medida que la tecnología informática se difundió a lo largo del mundo, se desarrollaron diferentes estándares y las empresas desarrollaron muchas variaciones del código ASCII para facilitar la escritura de lenguas diferentes al inglés que usan alfabetos latinos.

Así que, cualquier juego de caracteres de 8 bits, en el cual los códigos 32 a 126 (0x20 a 0x7E) coinciden con los caracteres imprimibles de ASCII, así como los códigos de control de 8 a 13 pero con caracteres de 127 en adelante distintos, se les denomina "Alfabeto Extendido"

Las codificaciones de ASCII extendido utiliza además parte o la totalidad de los códigos superiores a 128 para codificar caracteres adicionales a los caracteres imprimibles ASCII.

APRENDIZAJES OBTENIDOS

Gracias a esta práctica tenemos un mayor entendimiento sobre como hacer debug en hardware así como a utilizar los instrumentos de laboratorio para asegurar que los datos que queremos mandar realmente sean los que se están mandando por el 8051. De igual forma, entendemos cómo utilizar los recursos en materia de componentes para solucionar problemas de forma rápida.

Nos obligamos a realizar un código que fuera entendible y utilizara la memoria del micro de manera eficiente.

Nos fue necesario tener que diseñar nuestro propio circuito en su totalidad, cosa que no habíamos hecho en las anteriores prácticas y que es en donde de verdad nos damos cuenta qué es lo que sabemos y qué es lo que nos hace falta.

CONCLUSIONES

Lilia Lobato

La práctica me ayudó a terminar de comprender el uso de Registros como apuntadores y la importancia de hacer un buen plan de cableado desde el inicio. Con el problema que tuvimos respecto a la interpretación del código ASCII entendí que los conocimientos prácticos no son más importantes que los conocimientos teóricos y que no se puede lograr un trabajo completo si en alguno de los dos está deficiente. Es la práctica que más me ha gustado, al inicio se veía complicada pero conforme fuimos separando en secciones más pequeñas y resolviendo estas la práctica fluyó.

Jorge Karim

Esta práctica me sirvió bastante para lo que es la programación ya que no la había practicado mucho anteriormente con esta pude agarrar la práctica necesaria para poder contribuir de gran parte a mi equipo, en la parte del cableado no hubo ninguna complicación más que uno que otro error al principio, me sirvió bastante la práctica para poder aprender como utilizar la pantalla LCD, el teclado matricial y su respectiva codificación.

Gustavo Gutierrez

Cada vez que comenzamos una práctica nos damos cuenta que lo que sabemos de el microprocesador es muy poco comparado con todo lo que puede hacer, y no es malo, sino que es lo que debemos de estudiar para por lo menos completar nuestro trabajo. Sin embargo, esto de estudiar todo lo que no comprendemos nos da también las bases para hacer cosas mucho más grandes, cosa que ya nos hemos estado dando cuenta con los proyectos que cada equipo está planeando hacer, en nuestro caso la máquina expendedora, en la que haber aprendido bien cómo funciona la pantalla es fundamental, así como aprender cómo es que trabajan los motores y saber aplicar otros circuitos como el detector de monedas.

BIBLIOGRAFÍA

https://ascii.cl/es/

https://es.wikipedia.org/wiki/ASCII_extendido

https://www.taringa.net/posts/info/950198/Codigos-ASCII-caracteres-diferentes-simbolos.html

http://www.elcodigoascii.com.ar/codigos-ascii/dos-puntos-codigo-ascii-58.html

http://www.datasheetcatalog.com/info_redirect/datasheets/228/243043_DS.pdf.shtml

http://ee-classes.usc.edu/ee459/library/datasheets/MM74C922.pdf

Apuntes de clase