Multi-Interface LCD board

Basic Description:

The Multi-Interface LCD board is designed to display information on the LCD from three different protocol interfaces. The three protocols that can be used to communicate with the LCD are parallel, serial, and SPI (serial peripheral interface).

The protocol is determined automatically by the LCD board microprocessor. One thing to remember is that two protocols CANNOT be used at the same time. Only one protocol will write to the LCD at a time. Also remember that the Multi-Interface LCD board was not designed to allow data or information to be read back from the LCD.

The Multi-Interface LCD board is powered by +9V to +12V and ground. The parallel interface consists of 8 data pins and 3 control lines. Three pins are used for the serial interface. There is a 4-pin connection for the SPI interface.

Once power is applied to the LCD board, the LCD will display “Alive and well” for 5 seconds. If there is no display, check that power is applied correctly. If there is still no display, take the board to your lab instructor for repair.

No matter which protocol is being used to interface to the LCD, it must be initialized previous to normal use. The initialization sequence can be found in the data sheet for the LCD, KS0066U 16COM/40SEG Drivers and Controller for DOT MATRIX LCD. The data sheet can be found at G:\EET\Eet309\KS0066U.pdf.

Parallel Interface:

The parallel interface controls the LCD via 8 data pins and 3 control lines. The control lines used are Enable (E) and Register Select (RS).

RS is used to signal when a command or data is being sent to the LCD (Table 1). The Enable is used to signal when the information is ready to be interpreted by the LCD.

RS	Operation
L	Instruction write operation
H	Data write operation

Table 1. Operations according to RS
(L = low; H = high)

The timing diagram for writing information to the LCD (Figure 1), the timing characteristics for writing data (Table 2), and the table of instructions for writing commands (Table 3) were taken from the data sheet for the KS0066U 16COM/40SEG Drivers and Controller for DOT MATRIX LCD and are as follows:

Mode	Characteristic	Symbol	Min	Typ	Max	Unit
Write Mode	E Cycle Time	t_c	1000	-	-	ns
	E Rise / Fall Time	$t_R t_F$	-	-	25	
	E Pulse Width(High,Low)	t_w	450	-	-	
	R/W and RS Setup Time	t_{su1}	60	-	-	
	R/W and RS Hold Time	t_{H1}	20	-	-	
	Data Setup Time	t_{su2}	195	-	-	
	Data Hold Time	t_{H2}	10	-	-	

Table 2. AC and timing characteristics for write operations

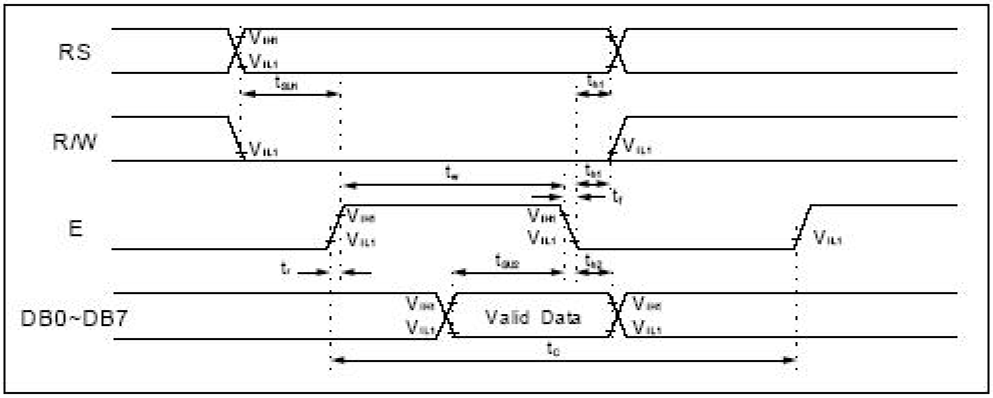


Figure 1. Timing Diagram for Write operations

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution time (fosc= 270 kHz)
	Instruction Code											
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.53 ms
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.	1.53 ms
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.	39 µs
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.	39 µs
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.	39 µs
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:5×11dots/5×8 dots)	39 µs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 µs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 µs
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.	0 µs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 µs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 µs

* "-": don't care

Table 3. Instruction Table

Serial Interface:

The serial interface controls the LCD via 3 pins – transmit, receive, and ground. The LCD board uses standard RS232 serial communication.

The serial interface takes away the need to manually control the Enable and RS line. The LCD board microprocessor interprets the data sent to it serially into data and control line instructions for the LCD in the usual manner for a serially-driven LCD.

Because the serial interface only allows data to be sent, the LCD board processor needs a way to tell the difference between command data and display data. That is handled by sending a special data code to the LCD board processor to signal that command data follows. It is called the instruction indicator or 0xFE.

To send a command, send the instruction indicator first, i.e. send_char(0xFE), and then send the command, i.e. send_char(0x14) for shift cursor to the left. To send data, simply send the data, i.e. send_char(0x35) for the number 5.

Here is an example of what the code to use this interface looks like:

```
send_charb(0xfe);     //instruction indicator
delay_ms(1);
send_charb(0x01);    //clear display instruction
delay_ms(2);
send_charb('H');      //Displays "Hi"
delay_ms(1);
send_charb('i');
delay_ms(1);
```

Note: The code is using the Serial interface from an ATMEL Mega16.

SPI Interface:

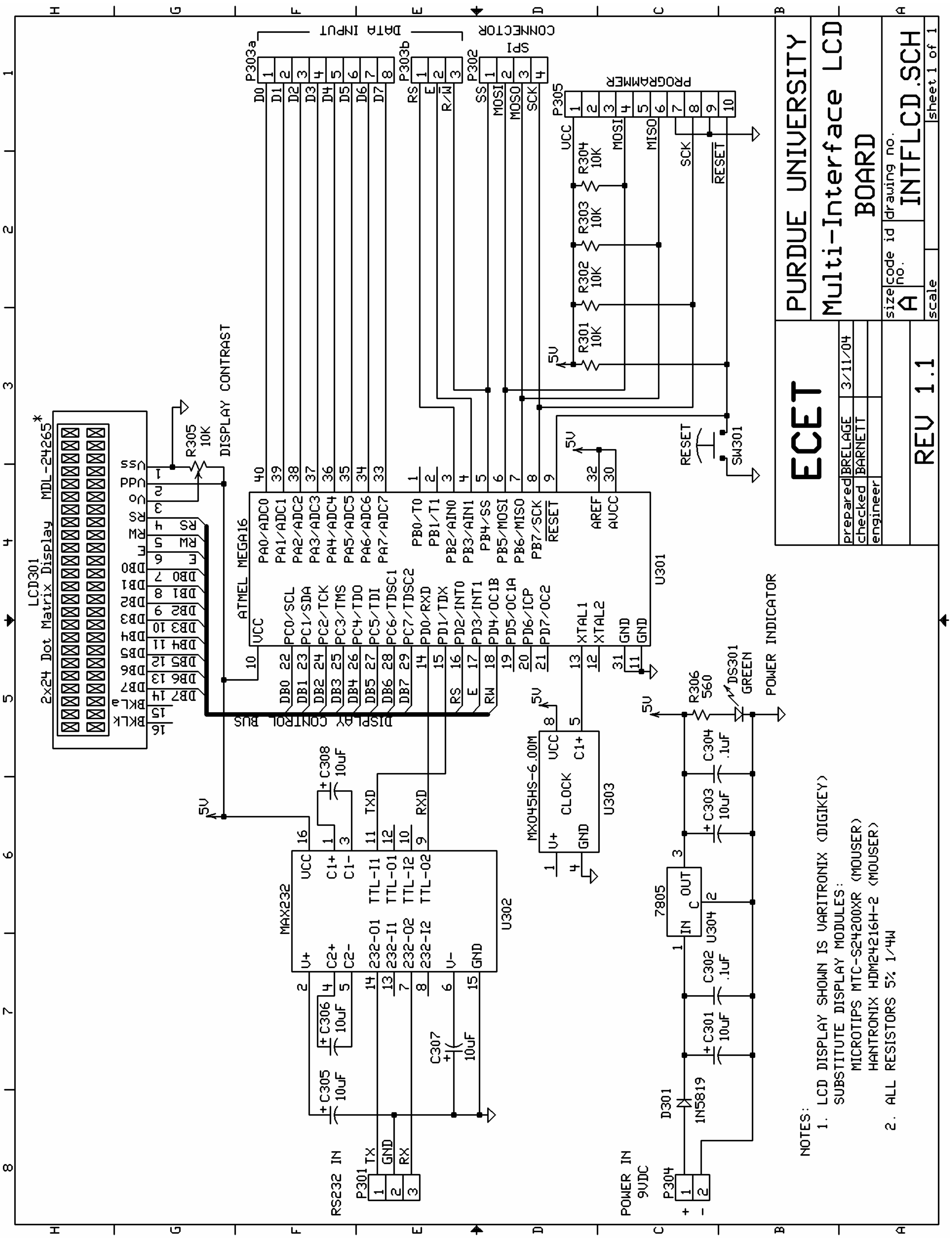
The SPI interface is very similar to the serial communications described above. However, in SPI there are 4 pins – SS, MOSI, MISO, and SCK. The LCD board SPI communication follows the basic SPI standard.

Again, as in the serial interface, the LCD board processor takes the data sent to it via the SPI interface and transfers it into data and control line instructions. As before an instruction indicator is needed to differentiate between command instructions and display data. For simplicity the same instruction indictor is used in the SPI interface, 0xFE.

Here is an example of what the code to use this interface looks like:

```
PORTB.4 = 0;     //clear SS on slave for comm
SPDR=0xfe;       //instruction indicator
delay_ms(1);
SPDR=0x01;      //clear display indicator
delay_ms(2);
SPDR='H';        //Displays "HI"
delay_ms(1);
SPDR='i';
delay_ms(1);
```

Note: The code is using the SPI interface from an ATMEL Mega16.


PURDUE UNIVERSITY
Multi-Interface LCD BOARD
ECET
prepared BRELAGE 3/11/04
checked BARNETT
engineer
size A
code id no.
drawing no. INTFLCD.SCH
scale
sheet 1 of 1
REV 1.1
NOTES:
1. LCD DISPLAY SHOWN IS VARITRONIX (DIGIKEY)
SUBSTITUTE DISPLAY MODULES:
MICROTIPS MTC-S24200XR (MOUSER)
HANTRONIX HDM24216H-2 (MOUSER)
2. ALL RESISTORS 5% 1/4W
LCD301
2x24 Dot Matrix Display MDL-24265*
ATMEL MEGA16
U301
MAX232
U302
MX045HS-6.00M
CLOCK
U303
7805
U304
DATA INPUT
CONNECTOR
SPI
PROGRAMMER
DISPLAY CONTRAST
DISPLAY CONTROL BUS
POWER INDICATOR
RS232 IN
POWER IN
9VDC

ASCII TABLE

		MSB							
		0000	0001	0010	0011	0100	0101	0110	0111
LSB	0000	NUL	DLE	SP	0	@	P		p
	0001	SOH	DC1	!	1	A	Q	a	q
	0010	STX	DC2		2	B	R	b	r
	0011	ETX	DC3	#	3	C	S	c	s
	0100	EOT	DC4	$	4	D	T	d	t
	0101	ENQ	NAK	%	5	E	U	e	u
	0110	ACK	SYN	&	6	F	V	f	v
	0111	BEL	ETB		7	G	W	g	w
	1000	BS	CAN	(	8	H	X	h	x
	1001	HT	EM	)	9	I	Y	I	y
	1010	LF	SUB	*	:	J	Z	j	z
	1011	VT	ESC	+	;	K	[	k	{
	1100	FF	FS	,	<	L	\	l	|
	1101	CR	GS		=	M	]	m	}
	1110	SO	RS	.	>	N	^	n	~
	1111	SI	US		?	O	-	o	DEL