

MIT ACADEMY OF ENGINEERING, Alandi (D)

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

School of Electrical Engineering

AY: 2020 - 2021

ET233 Microcontroller and Interfacing
PRACTICAL NO.--3

Aim: LCD Interfacing with PIC 18F4520 and
Embedded C Program

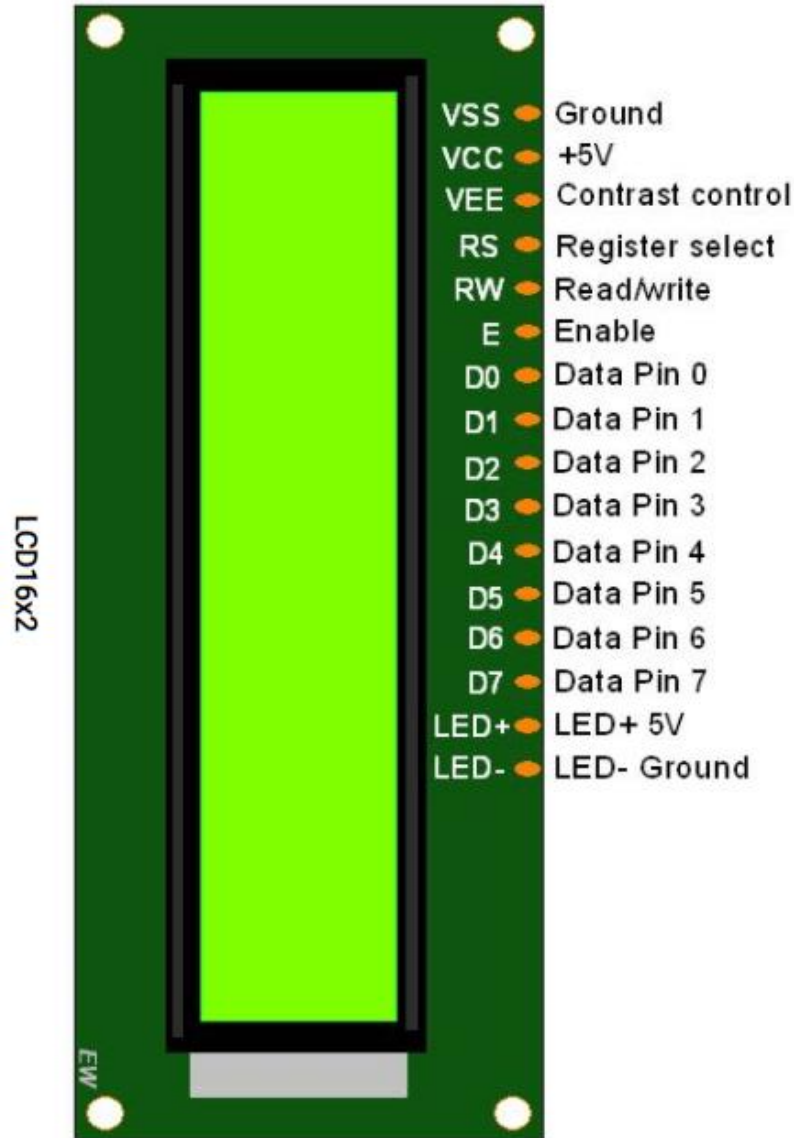
5/18/2021 Savita Pawar

Difference between LED, 7-segment display and LCD

- LED & 7-segment display require to send continuous data through MC/MP
- LED & 7-segment display hardware is complex to display message
- LCD once programmed it will display message continuously. MC/MP port pins will be free.
- LCD hardware is less complex

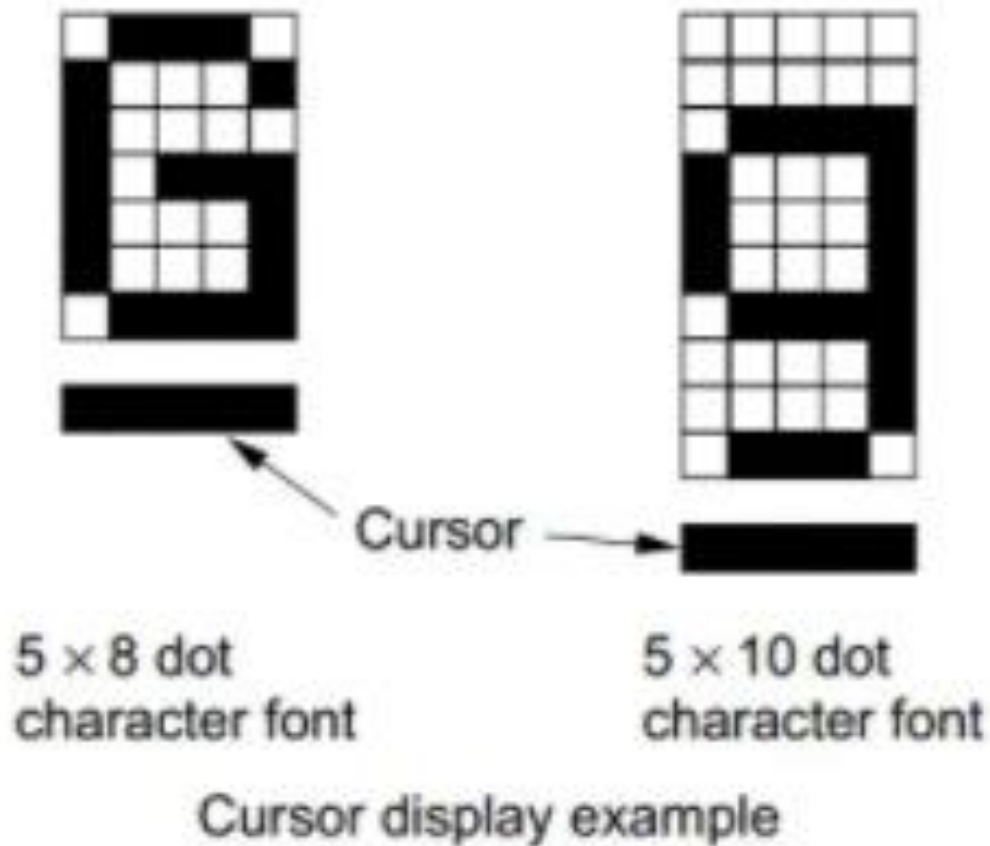
16 x 2 LCD

Pin Description



Pin	Symbol	I/O	Description
1	V_{SS}	--	Ground
2	V_{CC}	--	+5 V power supply
3	V_{EE}	--	Power supply to control contrast
4	RS	I	RS = 0 to select command register, RS = 1 to select data register
5	R/W	I	R/W = 0 for write, R/W = 1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

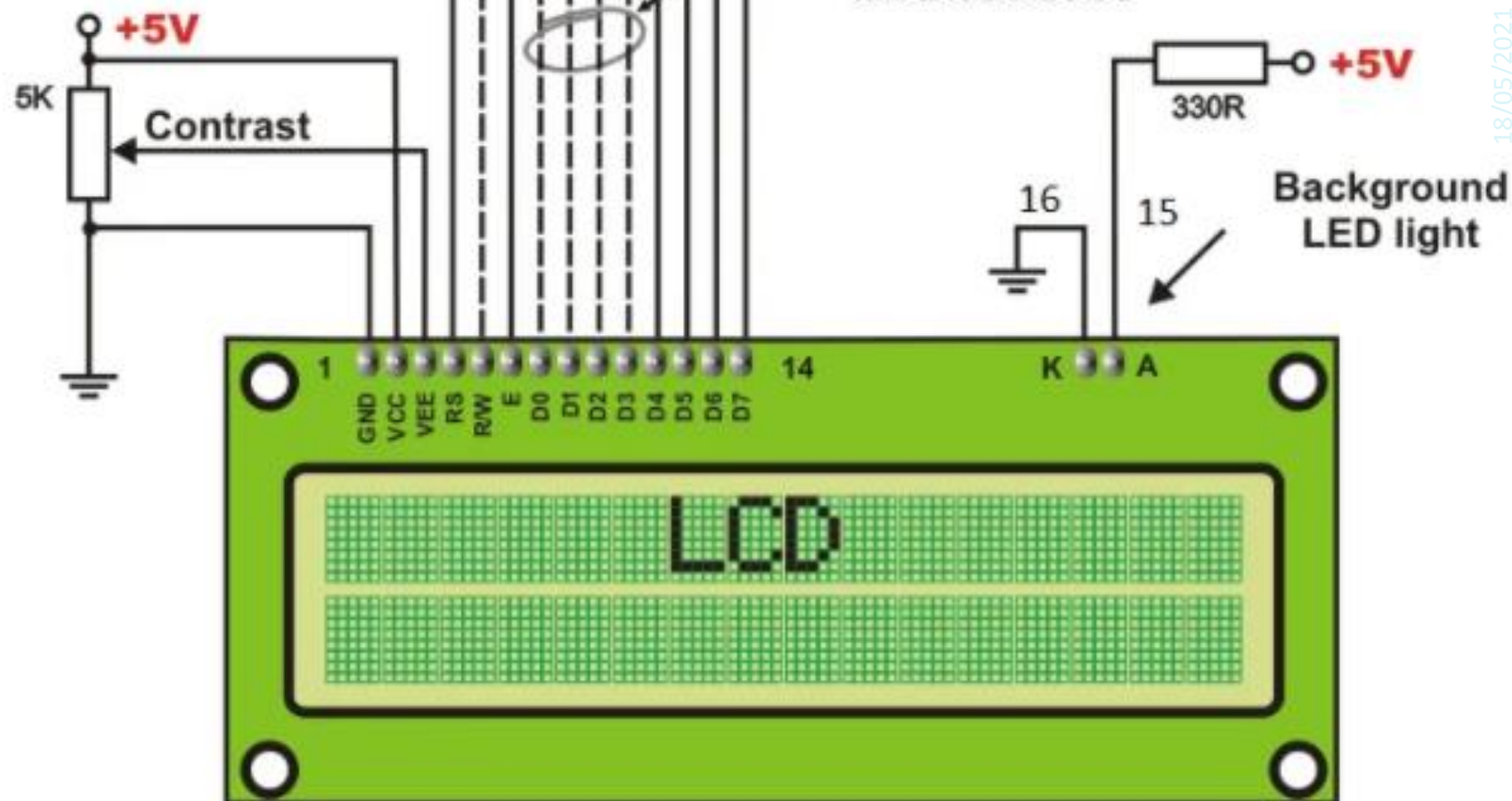
Character Display



Microcontroller

Can be connected to Ground

In 4-bit mode is left unconnected



LCD Instructions \ Commands

Command	Code										Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear Display	0	0	0	0	0	0	0	0	0	1	82 μ s~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	40 μ s~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	40 μ s

- I/D = 1 Increment I/D = 0 Decrement
- S = 1 Accompanies Display Shift

LCD Instructions \ Commands

Command	Code										Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	40µs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	40µs

- D = 1 Display on
 - C = 1 Cursor on
 - B = 1 Blinking on
 - S/C = 1 Display Shift
 - R/L = 1 Shift to the Right
- D = 0 Display off
 - C = 0 Cursor off
 - B = 0 Blinking off
 - S/C = 0 Cursor Shift
 - R/L = 0 Shift to the Left

LCD Instructions \ Commands

Command	Code										Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Function Set	0	0	0	0	1	DL	N*	F	*	#	40µs

- DL = 1 8 bit DL = 0 4 bit
- N = 1 1 line N = 0 1 Line
- F = 1 5 x 10 dots F = 0 5 x 7 dots

LCD Instructions \ Commands

Command	Code										Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Read Busy Flag & Address	0	1	BF	AC							1 μ s

- BF = Busy flag
- **4 ways to program LCD**
 - 1) 8 bit Delay method
 - 2) 4 bit Delay method
 - 3) 8 bit Busy flag check
 - 4) 4 bit Busy flag check

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

LCD Address command

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

16 x 2 LCD	80	81	82	83	84	85	86 through 8F
	C0	C1	C2	C3	C4	C5	C6 through CF

Algorithm

- Pin Declaration
- PORT direction declaration in Main program
- Initialize LCD as 8 bit, 5 x7 dots, line 0 (Function set command)
- Give Clear Screen command
- Give Entry mode set command
- Give Display on/off, cursor on/off command
- Give Starting address of display
- Send data in ASCII form

Program

```
#include<PIC18F4520.h>
#pragma config OSC = HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF, LVP=OFF
#define ldata PORTD
#define lcd_rs PORTCbits.RC0
#define lcd_rw PORTCbits.RC1
#define lcd_en PORTCbits.RC2
```

```
void LCD_Init()
{
    lcdcmd(0x38); /* Function set command*/
    msdelay(15);
    lcdcmd(0x01); /*clear display screen*/
    msdelay(15);
    lcdcmd(0x0c); /*display on cursor off*/
    msdelay(15);
    lcdcmd(0x06); /* (shift cursor to right)*/
    msdelay(15);
}
```

```
void lcdcmd (unsigned char cmd)
{
    ldata = cmd;
    lcd_rs = 0;
    lcd_rw = 0;
    lcd_en = 1;
    msdelay(5);
    lcd_en = 0;
}
```

```
void lcddata (unsigned char data1)
{
    ldata = data1;
    lcd_rs = 1;
    lcd_rw = 0;
    lcd_en = 1;
    msdelay(5);
    lcd_en = 0;
}
```



```
void msdelay (int a)
{
    unsigned int b,c;
    for(b=0;b<a;b++)
    for(c=0;c<200;c++);
}
```

```
Void main()
{
    TRISC = 0;
    TRISD = 0;
    void LCD_Init();
    lcdcmd(0x80);
    msdelay(15);
    lcddata('M');
    msdelay(15);
    while(1);
}
```