



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PIC16F877-Liquid Crystal Display(LCD)

Dr.G.Arthy

Assistant Professor

Department of EEE

SNS College of Engineering



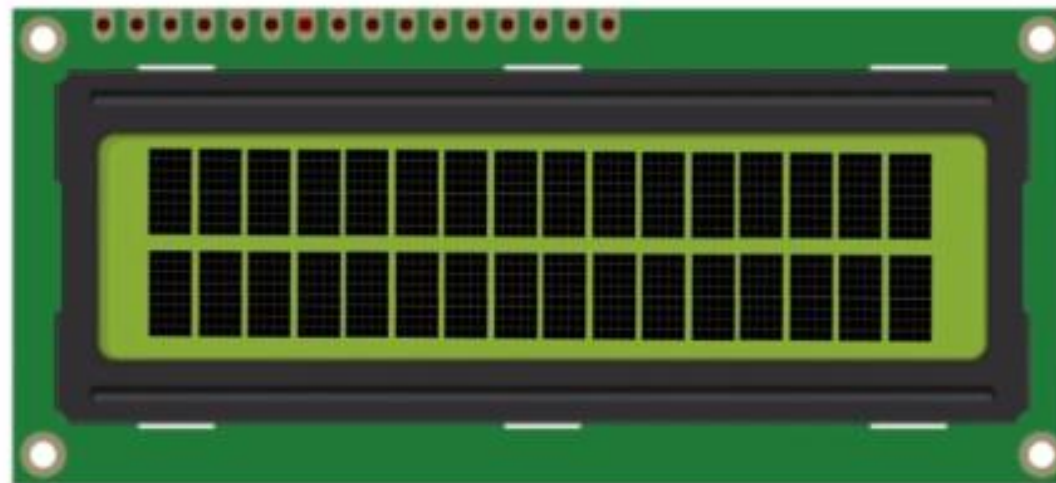
LCD

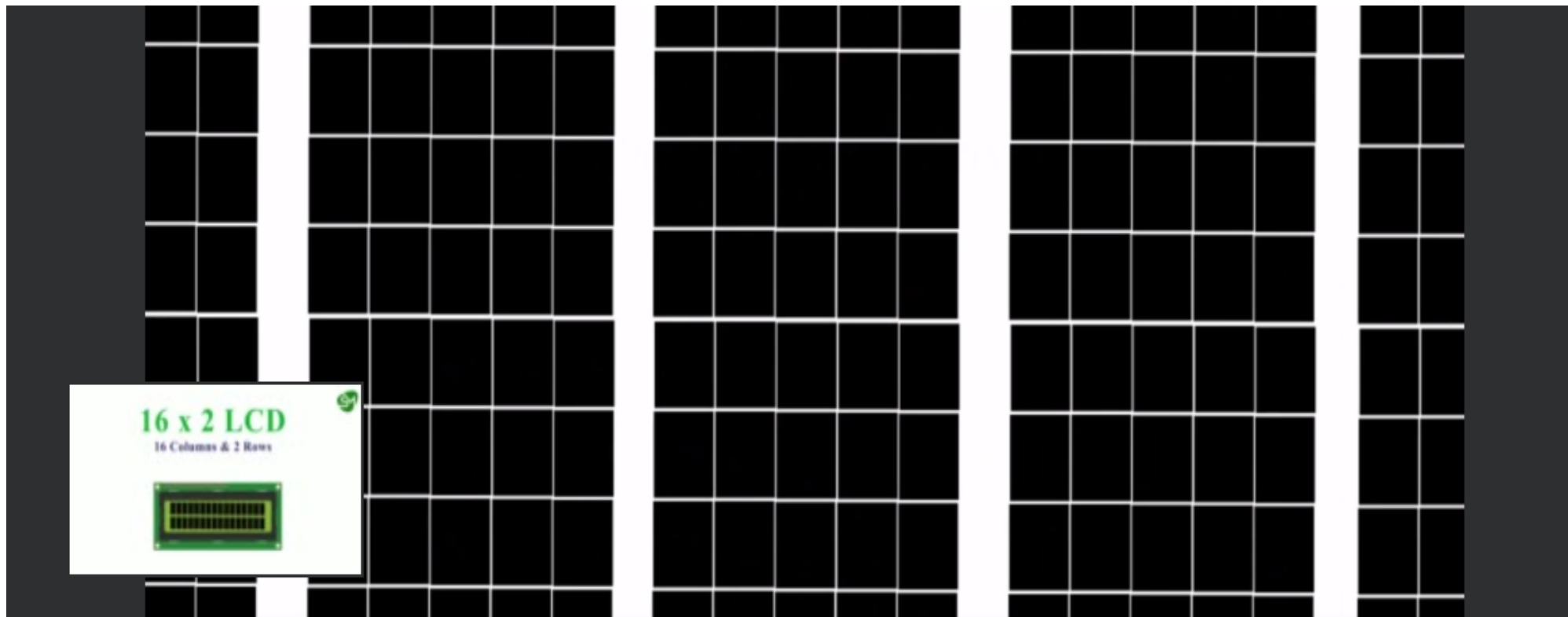
(liquid crystal display)



16 x 2 LCD

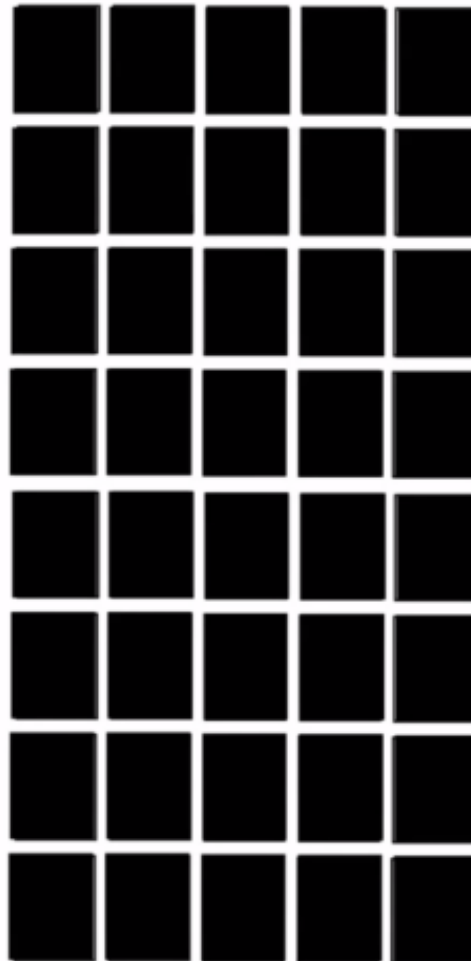
16 Columns & 2 Rows





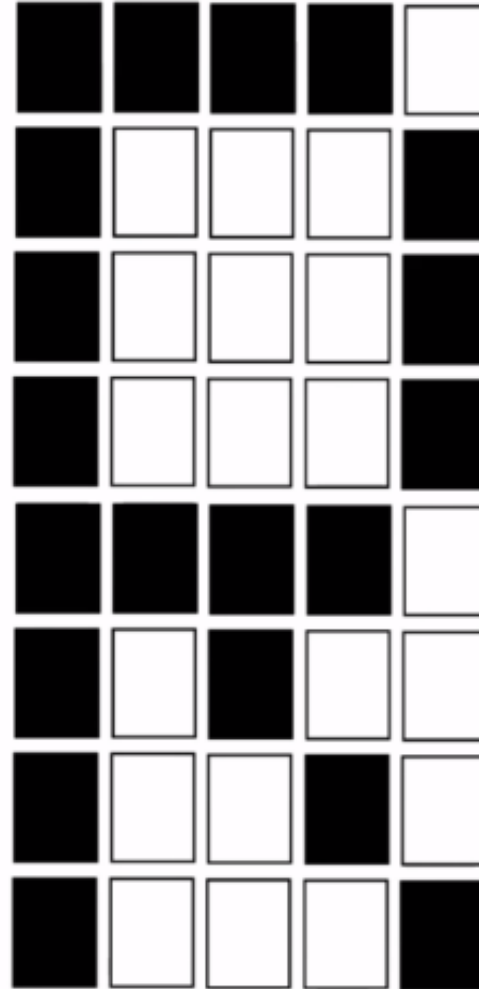


5 x 8
dot matrix display





**If we want to print
character
“R”**





LCD Controller



- For enabling dot matrix for each character a controller is attached with every LCD display.
- HD44780 – type of controller from Hitachi.



CONTROLLER

HD44780

INSTRUCTION REGISTER

DATA REGISTER

PASS THE ADDRESS AT WHICH WE ARE GOING TO PRINT

PASS THE DATA WHICH WE WANT TO PRINT



- In LCD, address starts from 0x80 for row 1
- For Row 2 address starts from 0xC0



LCD Command Codes

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 to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

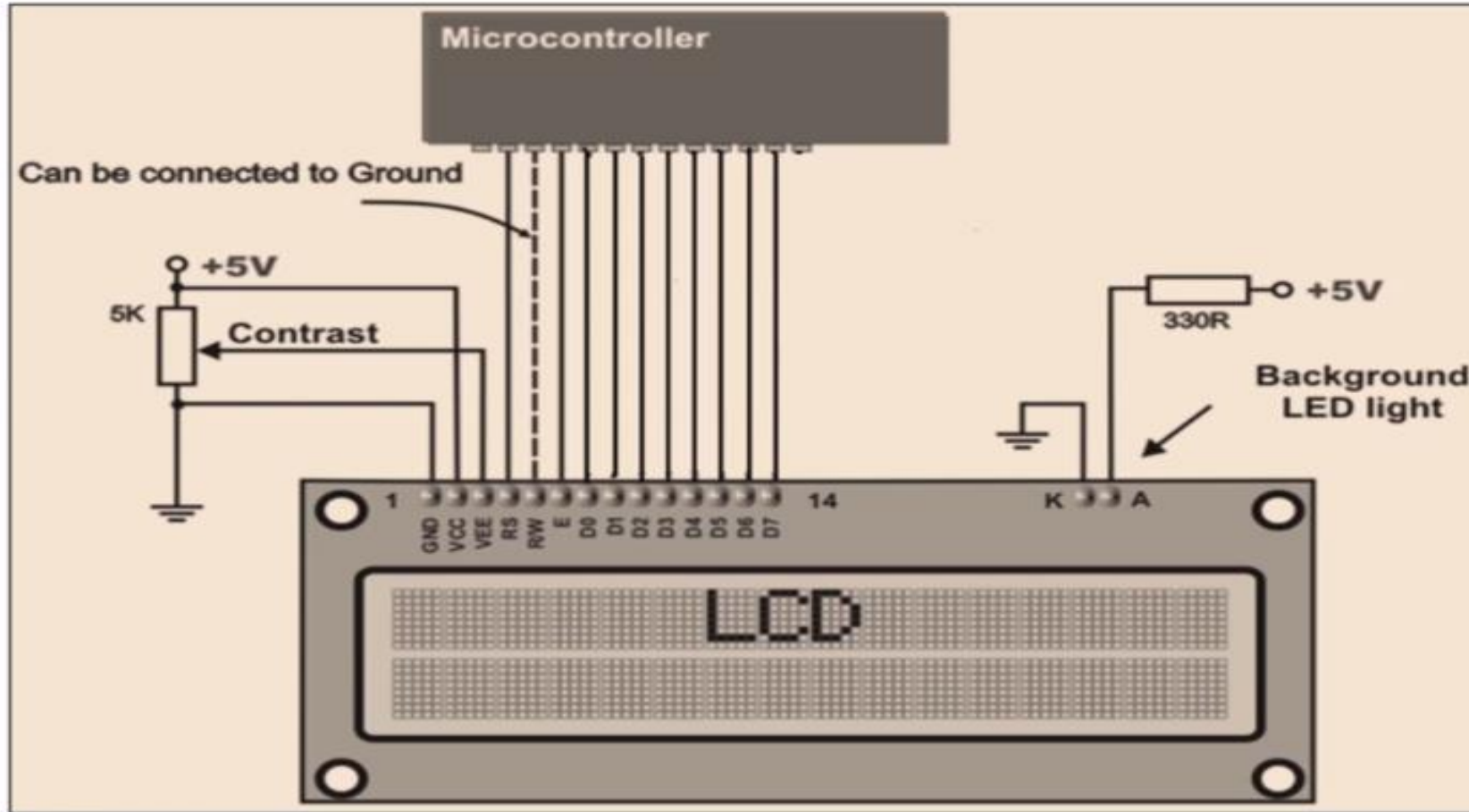
PINOOTS OF LCD

if $RS = 0$, Data provided will be stored in data register of LCD
if $RS = 1$, Data provided will be stored in instruction register of LCD

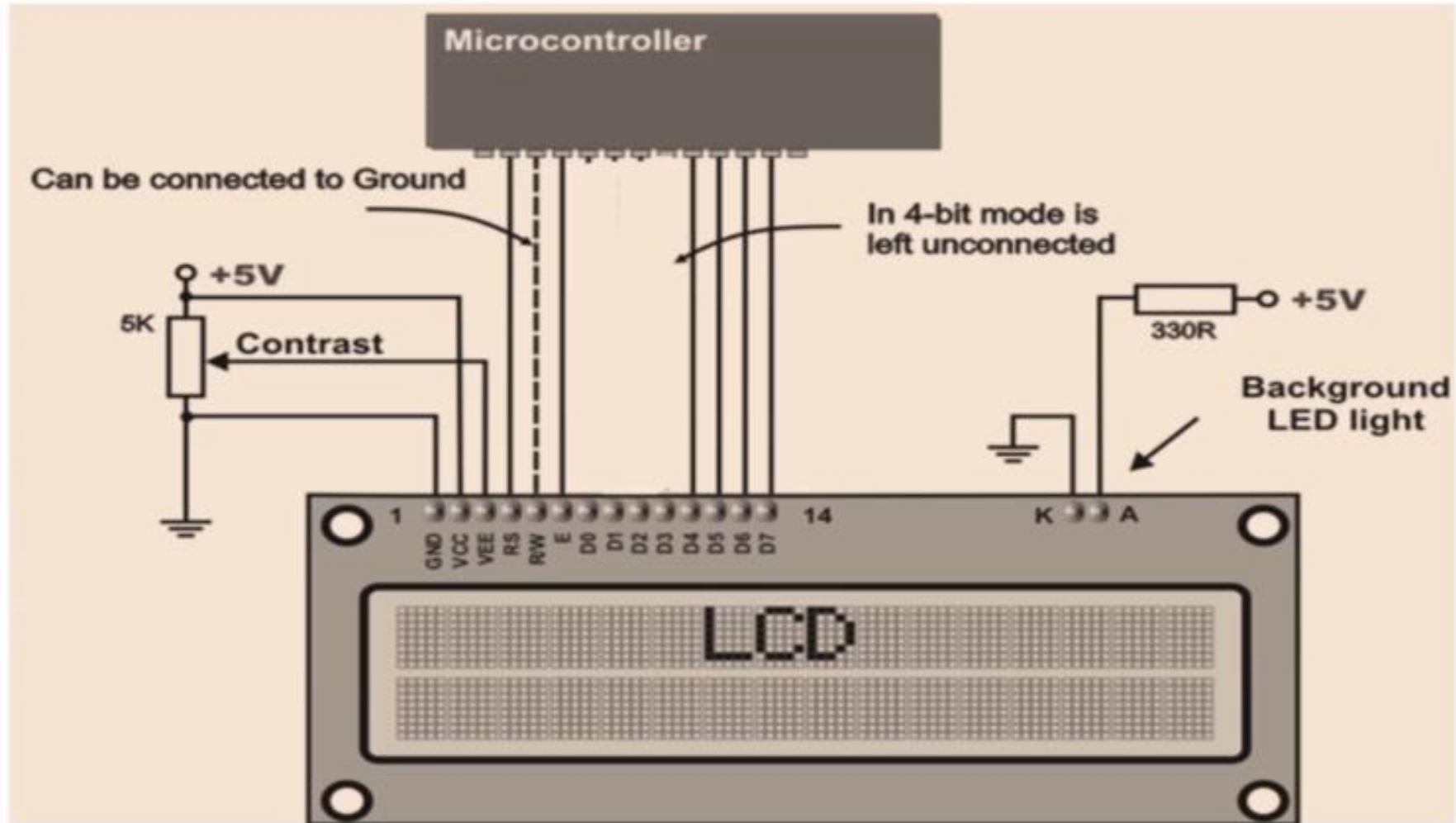
if $RW = 0$, Write operation is performed in LCD
if $RW = 1$, Read operation is performed in LCD

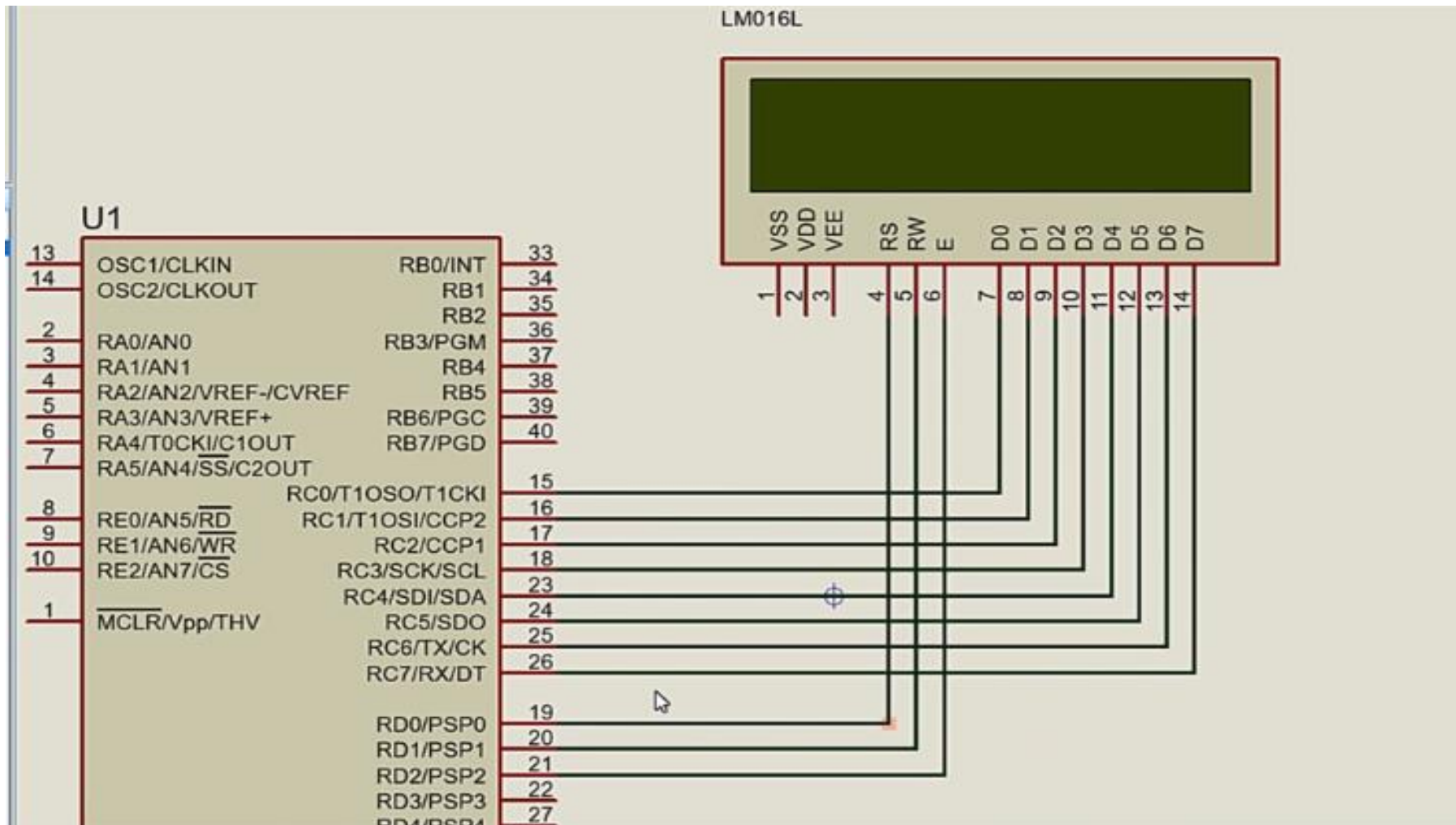


8 bit



4-bit







CODE



```
#include<16F877A.h>
•#use delay(clock=20000000)
•#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
•void busycheck();
•unsigned int array[19]={"Vi Microsystems"};
•unsigned char array1[] = {"VPUT-01"};
•unsigned char array2[] = {"Embedded Trainer"};
•unsigned char a,i,b;
```

•



```

• busycheck ();
•     output_d (0x01);           /* clear display
•     output_high (PIN_B3);
•     output_low (PIN_B3);
•
• busycheck ();
•     output_d (0x0c);           /* cursor on */
•     output_high (PIN_B3);
•     output_low (PIN_B3);
•
•
•     busycheck ();
•     output_d (0x82);           /* starting addre
•     output_high (PIN_B3);
•     output_low (PIN_B3);

```

LCD Command Codes

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 to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix



```
• for (i=0;i<15;i++)
•     {
•         busycheck();
•         output_d(0x01);
•         output_low(PIN_B1);
•         output_high(PIN_B1);
•         b=array[i];
•         output_d(b);
•         output_high(PIN_B3);
•         output_low(PIN_B3);
•     }
•     busycheck();
•     output_d(0xc6);           /* starting address */
•     output_high(PIN_B3);
•     output_low(PIN_B3);
```



```
•for (i=0;i<7;i++)
•    {
•        busycheck();
•        output_d(0x01);
•        output_low(PIN_B1);
•        output_high(PIN_B1);
•        b=array1[i];
•        output_d(b);
•        output_high(PIN_B3);
•        output_low(PIN_B3);
•    }
•busycheck();
•    output_d(0x96);          /* starting address */
•    output_high(PIN_B3);
•    output_low(PIN_B3);
•
•for (i=0;i<16;i++)
•    {
•        busycheck();
•        output_d(0x01);
•        output_low(PIN_B1);
•        output_high(PIN_B1);
```



```
• output_d(b);
•
•     output_high(PIN_B3);
•     output_low(PIN_B3);
•
•     }
•
•
•
•     busycheck();
•     output_d(0xd4);           /* starting address */
•     output_high(PIN_B3);
•     output_low(PIN_B3);

for (i=0;i<20;i++)
•     {
•         busycheck();
•         output_d(0x01);
•         output_low(PIN_B1);
•         output_high(PIN_B1);
•         output_d('-');
•         output_high(PIN_B3);
•         output_low(PIN_B3);
•     }
```



```
stop:
    goto stop;
}

void busycheck()
{
    output_d(0x02);
    output_low(PIN_B1);
    output_high(PIN_B1);
    delay_ms(2);
busy:
    output_high(PIN_B3);
output_low(PIN_B3);
    a=input_d();
    if ((a&0x80)==0x80)
        goto busy;
    output_d(0x0);
    output_low(PIN_B1);
    output_high(PIN_B1);
    delay_us(10);
}
```