

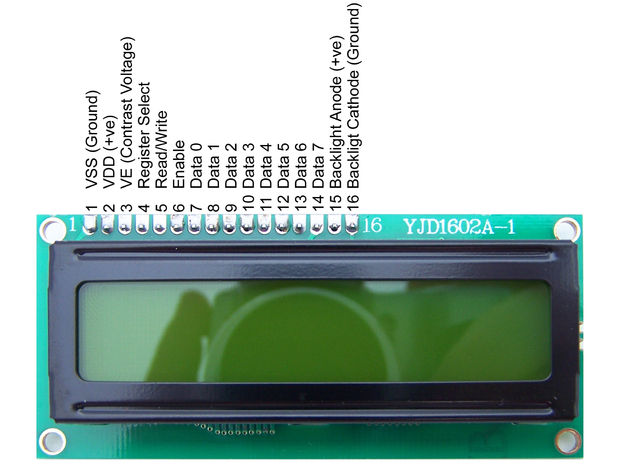
**DISPLAY**

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments](http://www.engineersgarage.com/content/seven-segment-display) and other multi segment [LED](http://www.engineersgarage.com/content/led)s. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even [custom characters](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](http://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a [LCD](http://www.engineersgarage.com/insight/how-lcd-works).

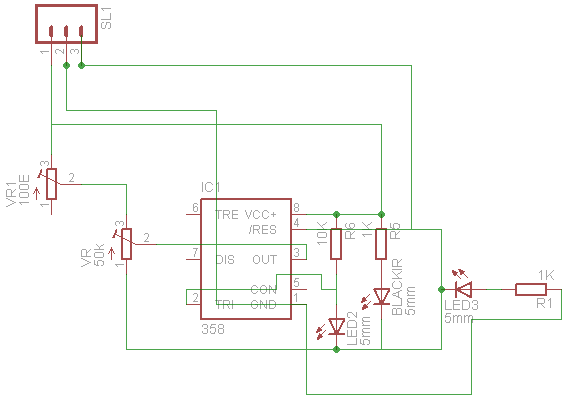
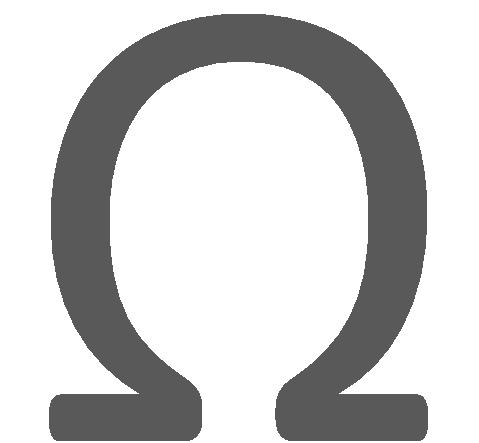
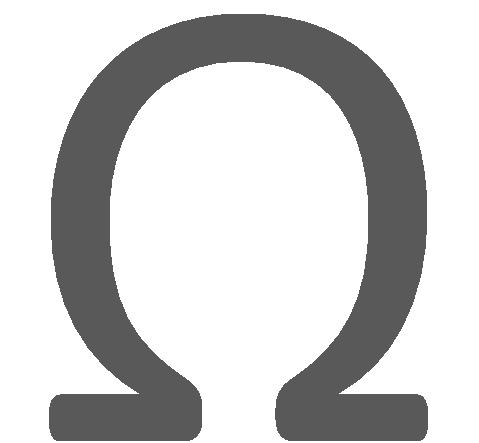
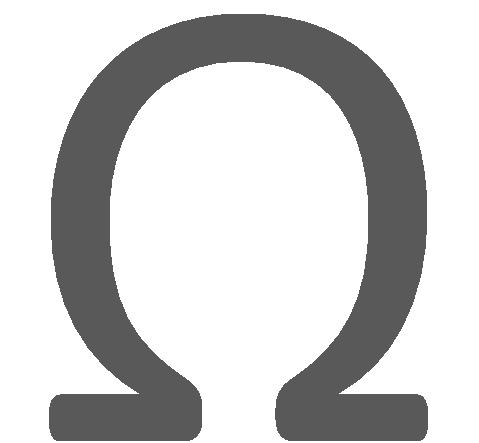
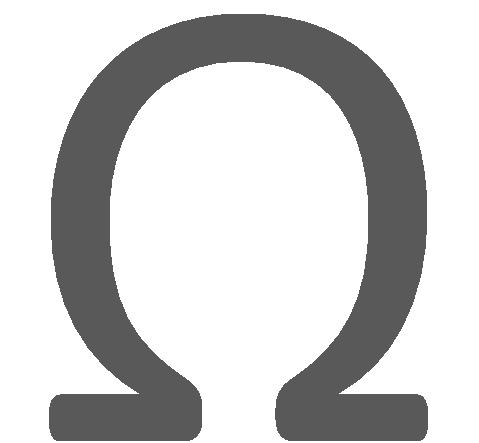
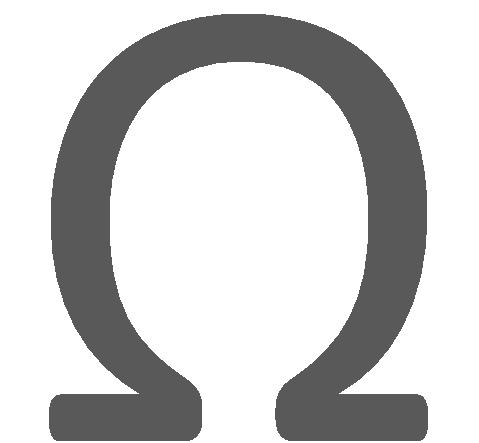
### Pin Diagram:



**IR MODULE**

The transmitter part of the sensor project is an [Infrared (IR) Led](http://engineersgarage.com/content/infrared-led) which transmits continuous IR rays to be received by an IR receiver. The output of the receiver varies depending upon its reception of IR rays. Since this variation cannot be analyzed as such, therefore this output can be fed to a comparator. Here operational amplifier (op-amp) of [LM 339](http://engineersgarage.com/content/ic-lm339) is used as comparator.

When the IR receiver does not receive signal the potential at the inverting input goes higher than that that at non-inverting input of the comparator (LM 339). Thus the output of the comparator goes low and the [LED](http://engineersgarage.com/content/led) does not glow .When the IR receiver receives signal the potential at the inverting input goes low. Thus the output of the comparator (LM 339) goes high and the LED starts glowing. [Resistor](http://engineersgarage.com/content/resistor) R1 (100), R2 (10k) and R3 (330 ) are used to ensure that minimum 10 mA current passes through the IR LED, photodiode and normal LED, respectively. Resistor VR2 (preset=5k) is used to adjust the output. Resistor VR1 (preset=10k) is used to set the sensitivity of the circuit. Read more about [IR sensor](http://www.engineersgarage.com/articles/infrared-sensors) here.



#include<reg52.h> //including sfr registers for ports of the controller

#include<lcd.h>

#define on 0

#define off 1

int scan=1;

//LCD Module Connections

sbit RS = P0^0;

sbit EN = P0^1;

sbit D0 = P2^0;

sbit D1 = P2^1;

sbit D2 = P2^2;

sbit D3 = P2^3;

sbit D4 = P2^4;

sbit D5 = P2^5;

sbit D6 = P2^6;

sbit D7 = P2^7;

//for first segment

int count\_1=0;

int runtime\_1 = 15;

sbit R1 = P1^7;

sbit G1 = P1^6;

sbit module1\_1 = P3^6;

sbit module1\_2 = P3^7;

//for second segment

int count\_2=0;

int runtime\_2 = 15;

sbit R2 = P1^5;

sbit G2 = P1^4;

sbit module2\_1 = P3^5;

sbit module2\_2 = P3^4;

//for third segment

int count\_3=0;

int runtime\_3 = 15;

sbit R3 = P1^3;

sbit G3 = P1^2;

sbit module3\_1 = P3^1;

sbit module3\_2 = P3^0;

//for fourth segment

int count\_4=0;

int runtime\_4 = 15;

sbit R4 = P1^0;

sbit G4 = P1^1;

sbit module4\_1 = P3^2;

sbit module4\_2 = P3^3;

//

//End LCD Module Connections

void Delay(int a)

{

int j;

int i;

for(i=0;i<a;i++)

{

for(j=0;j<100;j++)

{

}

}

}

char toChar(int a)

{

switch(a)

{

case 1:

return '1';

break;

case 2:

return '2';

break;

case 3:

return '3';

break;

case 4:

return '4';

break;

case 5:

return '5';

break;

case 6:

return '6';

break;

case 7:

return '7';

break;

case 8:

return '8';

break;

case 9:

return '9';

break;

case 0:

return '0';

break;

}

}

void main()

{

int i,j,k,l;

char c;

Lcd8\_init();

Lcd8\_Set\_Cursor(1,1);

Lcd8\_Write\_String("Traffic Light");

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Control");

R1=off;

G1=off;

R2=off;

G2=off;

Delay(100);

R3=off;

G3=off;

R4=off;

G4=off;

Delay(100);

R1=on;

R2=on;

R3=on;

R4=on;

Delay(100);

G1=off;

G2=off;

G3=off;

G4=off;

Delay(100);

while(1)

{

switch(scan)

{

case 1:

// scan first segment

if(count\_1 <= 0)

{

runtime\_1=15;

if( module1\_1 == 1)

{

runtime\_1=30;

}

if(module1\_2 == 1)

{

runtime\_1=60;

}

count\_1=runtime\_1;

}

R1=off;

G1=on;

count\_1--;

Lcd8\_Clear();

i=count\_1;

j=i%10;

k=i/10;

l=i/100;

k=k-l\*10;

Lcd8\_Set\_Cursor(1,3);

c=toChar(j);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(1,2);

c=toChar(k);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(2,1);

c=toChar(l);

Lcd8\_Write\_Char(c);

if(runtime\_1 == 60)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg1:60 Secs");

}

else if(runtime\_1 == 30)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg1:30 Secs");

}

else

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg1:15 Secs");

}

if(count\_1 == 0)

{

G1=off;

count\_1=0;

scan=2;

R1=on;

}

break;

case 2:

// scan second segment

if(count\_2 <= 0)

{

runtime\_2=15;

if( module2\_1 == 1)

{

runtime\_2=30;

}

if(module2\_2 == 1)

{

runtime\_2=60;

}

count\_2=runtime\_2;

}

R2=off;

G2=on;

count\_2--;

Lcd8\_Clear();

i=count\_2;

j=i%10;

k=i/10;

l=i/100;

k=k-l\*10;

Lcd8\_Set\_Cursor(1,3);

c=toChar(j);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(1,2);

c=toChar(k);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(2,1);

c=toChar(l);

Lcd8\_Write\_Char(c);

if(runtime\_2 == 60)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg2:60 Secs");

}

else if(runtime\_2 == 30)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg2:30 Secs");

}

else

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg2:15 Secs");

}

if(count\_2 == 0)

{

G2=off;

count\_2=0;

scan=3;

R2=on;

}

break;

case 3:

// scan third segment

if(count\_3 <= 0)

{

runtime\_3=15;

if( module3\_1 == 1)

{

runtime\_3=30;

}

if(module3\_2 == 1)

{

runtime\_3=60;

}

count\_3=runtime\_3;

}

R3=off;

G3=on;

count\_3--;

Lcd8\_Clear();

i=count\_3;

j=i%10;

k=i/10;

l=i/100;

k=k-l\*10;

Lcd8\_Set\_Cursor(1,3);

c=toChar(j);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(1,2);

c=toChar(k);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(2,1);

c=toChar(l);

Lcd8\_Write\_Char(c);

if(runtime\_3 == 60)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg3:60 Secs");

}

else if(runtime\_3 == 30)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg3:30 Secs");

}

else

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg3:15 Secs");

}

if(count\_3 == 0)

{

G3=off;

count\_3=0;

scan=4;

R3=on;

}

break;

case 4:

// scan fourth segment

if(count\_4 <= 0)

{

runtime\_4=15;

if( module4\_1 == 1)

{

runtime\_4=30;

}

if(module4\_2 == 1)

{

runtime\_4=60;

}

count\_4=runtime\_4;

}

R4=off;

G4=on;

count\_4--;

Lcd8\_Clear();

i=count\_4;

j=i%10;

l=i/100;

k=k-l\*10;

c=toChar(j);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(1,2);

c=toChar(k);

Lcd8\_Write\_Char(c);

Lcd8\_Set\_Cursor(2,1);

c=toChar(l);

Lcd8\_Write\_Char(c);

if(runtime\_4 == 60)

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg4:60 Secs");

}

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg4:30 Secs");

}

else

{

Lcd8\_Set\_Cursor(2,1);

Lcd8\_Write\_String("Seg4:15 Secs");

}

if(count\_4 == 0)

{

G4=off;

count\_4=0;

scan=1;

R4=on;

}

break;

}

Delay(700);

}

}



Part Value Quantity

LCD 16\*2 1

VR 20K 1

module IR MODULE 8

BRIDGE rectifier 1

C1 33pf 2

C2 470uf 1

C3 .01uf 2

CRYSTAL 11.0592 mhz 1

IC1 AT89S51 1

IC2 7805TV 1

JP1 TRANSFORMER 1

LED 5mm 1

R1 10k 1

R2 560E 1

RN10 10 k 1

Switch1 on/off 1

Switch2 push switch 1

