AVR

DDRB=0xFF; // set LCD data port as output

DDRD=0xE0; // set LCD signals (RS, RW, E) as out put

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LCD PROGRAM STARTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

#define LCD\_DATA PORTB // port B is selected as LCD data port

#define ctrl PORTD // port D is selected as LCD command port

#define en PD7 // enable signal is connected to port D pin 7

#define rw PD6 // read/write signal is connected to port D pin 6

#define rs PD5 // register select signal is connected to port D pin 5

void init\_LCD(void)

{

LCD\_cmd(0x38); // initialization in 8bit mode of 16X2 LCD

*\_delay\_ms*(1);

LCD\_cmd(0x01); // make clear LCD

*\_delay\_ms*(1);

LCD\_cmd(0x02); // return home

*\_delay\_ms*(1);

LCD\_cmd(0x06); // make increment in cursor

*\_delay\_ms*(1);

LCD\_cmd(0x80); // “8” go to first line and “0” is for 0th position

*\_delay\_ms*(1);

LCD\_cmd(0x0C);

return;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*sending command on LCD\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void LCD\_cmd(unsigned char cmd)

{

LCD\_DATA = cmd; // data lines are set to send command

PORTD &= ~(1<<rs); // RS sets 0

PORTD &= ~(1<<rw); // RW sets 0

PORTD |= (1<<en); // make enable from high to low

*\_delay\_ms*(100);

PORTD &= ~(1<<en);

return;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*write data on LCD\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void LCD\_write(unsigned char data)

{

LCD\_DATA= data; // data lines are set to send command

PORTD |= (1<<rs); // RS sets 1

PORTD &= ~(1<<rw); // RW sets 0

PORTD |= (1<<en); // make enable from high to low

*\_delay\_ms*(100);

PORTD &= ~(1<<en);

return ;

}