Display text on LCD using Serial Protocol on an 8051 microcontroller

Name: Nirvan Tamhane

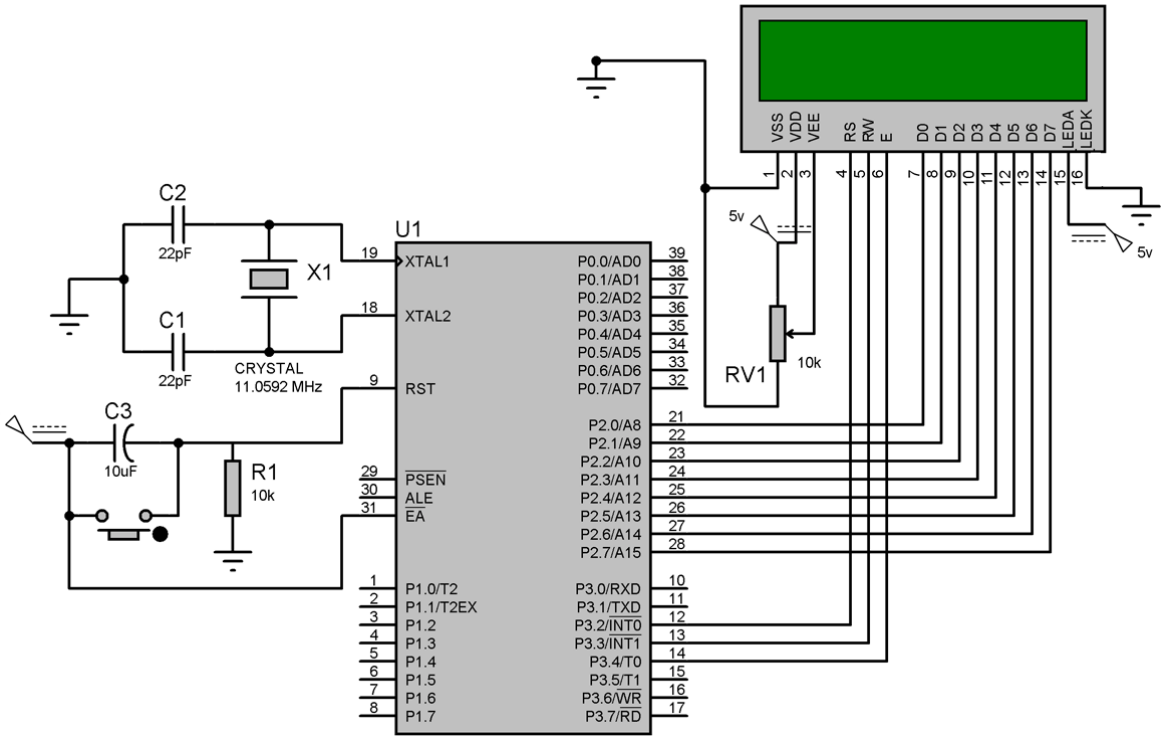
Aim: Write a program to transmit a string “HELLO” serially

Theory:

Serial communication is mostly used for transmitting and receiving the signal. The 8051 microcontroller is consisting of Universal Asynchronous Receiver Transmitter (UART) used for serial communication. The signals are transmitted and received by the Rx and Tx pins of microcontroller.

The UART take individual bytes of data and sends the individual bits in a sequential manner. The registers are used for collecting and storing the data inside a memory. UART is based on half-duplex protocol. Half-duplex means transferring and receiving the data, but not at the same time.

Schematic:



Code:

```
#include <reg51.h>
unsigned char bdata EP_DATA;
sbit lsb=EP_DATA^0;
sbit msb=EP_DATA^7;
sbit SDA = P3^7;
sbit SCL = P3^6;
unsigned char rec[12];
void i2c_start(void);
void i2c_stop(void);
void i2c_send (unsigned char);
void i2c_send_byte(unsigned char addr,unsigned char dataa);
void i2c_send_string(unsigned char addr,unsigned char *s);
unsigned char i2c_read(void);
unsigned char i2c_read_byte(unsigned char addr);
unsigned char i2c_read_string(unsigned char addr);
void ser_init();
void tx(unsigned char send);
void tx_str(unsigned char *s);
void i2c_start(void)
{
SDA=1;
SCL=1;
SDA=0;
SCL=0;
}
void i2c_send (unsigned char send)
{
unsigned char i;
EP_DATA=send;
for(i=0;i<=7;i++) {
SDA=msb;
SCL=1;
SCL=0;
EP_DATA=EP_DATA<<1;
}
while(SDA!=0);
SCL=1;
SCL=0;
}
unsigned char i2c_read(void)
{
unsigned char i;
lsb=SDA;
for(i=0;i<=7;i++) {
EP_DATA=EP_DATA<<1;
lsb=SDA;
SCL=1;
```

```
SCL=0;
}

if(EP_DATA==13) {
SDA=1;
SCL=1;
SCL=0;
SDA=0;
i2c_stop();
return(EP_DATA);
}
SDA=0;
SCL=1;
SCL=0;
SDA=1;
return(EP_DATA);
}
void i2c_stop(void)
{
SDA=0;
SCL=1;
SDA=1;
SCL=0;
}
void i2c_send_byte(unsigned char addr,unsigned char dataa)
{
i2c_start();
i2c_send(0xa0);
i2c_send(addr);
i2c_send(dataa);
i2c_stop();
}
unsigned char i2c_read_byte(unsigned char addr)
{
unsigned char rec;
i2c_start();
i2c_send(0xa0);
i2c_send(addr);
i2c_start();
i2c_send(0xa1);
rec=i2c_read();
i2c_stop();
return rec;
}
void i2c_send_string(unsigned char addr,unsigned char *s)
{
i2c_start();
i2c_send(0xa0);
i2c_send(addr);
while(*s) {
```

```
i2c_send(*s++);
}
i2c_stop();
}
unsigned char i2c_read_string(unsigned char addr)
{
unsigned char i;
i2c_start();
i2c_send(0xa0);
i2c_send(addr);
i2c_start();
i2c_send(0xa1);
for(i=0;i<10;i++) {
rec[i]=i2c_read(); }
i2c_stop();
return rec;
}
void ser_init()
{
SCON=0x50;
TMOD|=0x20;
TH1=0xFD;
TL1=0xFD;
TR1=1;
}
void tx(unsigned char send)
{
SBUF=send;
while(TI==0);
TI=0;
}
void tx_str(unsigned char *s)
{
while(*s)
tx(*s++);
}

int main()
{
#unsigned char data[12];
ser_init();
i2c_send_string(0x00,"EmbeTronicX");
i2c_read_string(0x00);
tx_str(rec);
while(1);
}
```

Conclusion:

In this experiment I interfaced 8051 and EEPROM interfacing using I2C. 8051 doesn't have built in I2C therefore externally it is achieved. EEPROM also supports I2C and I read and

also written on using the I2C using 8051. Data was successfully read and written on the EEPROM.