AUTOMATIC IRRIGATION SYSTEM

Computer Interface

4th Year Computer Engineering

Team ID: D7

Member	Sec.
Mohamed Nasser El-Sayed	4

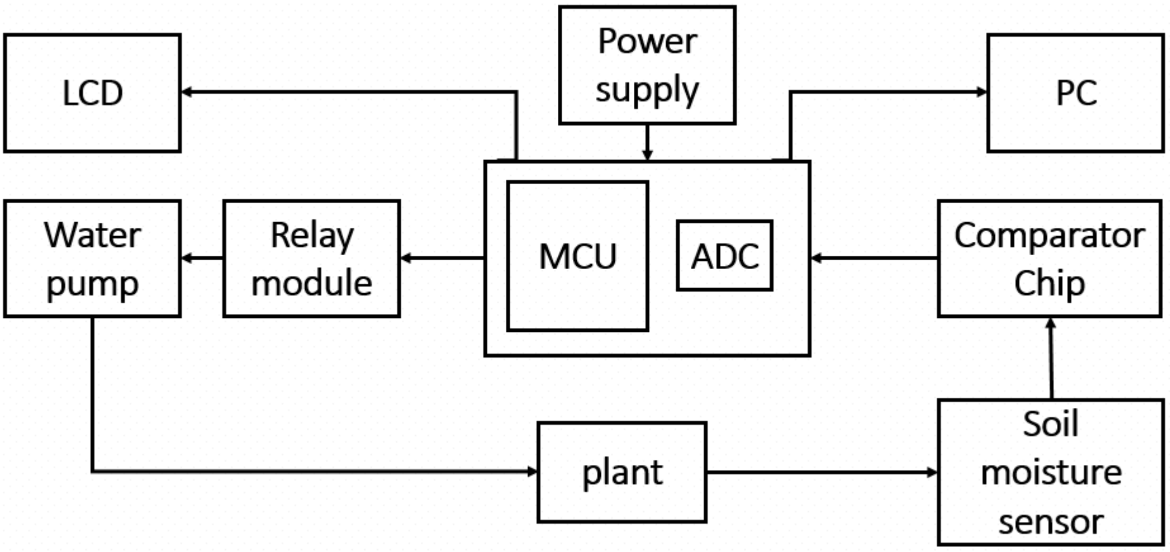
Project Objective:

Water shortage is one of the major problem in the world. Agriculture is one such field where water is required in high quantity. Wastage of water is a major problem in agriculture.so we make automatic irrigation, for some reasons:

1. Saves water.
2. Improves growth.
3. Saves time and effort.
4. Eliminates the manual operation of opening or closing valves.
5. Saves energy, where irrigation process starts and stops exactly when required, thus optimizing energy requirements.
6. Doing the irrigation work in extremely odd weather conditions, hard work of repeated assembly and will get rid of poisonous reptiles. The system, which is designed, will help the farmers to do the irrigation process in night also. The system designed do not requires the physical presence of the farmers during irrigation in the fields.

System Block Diagram:

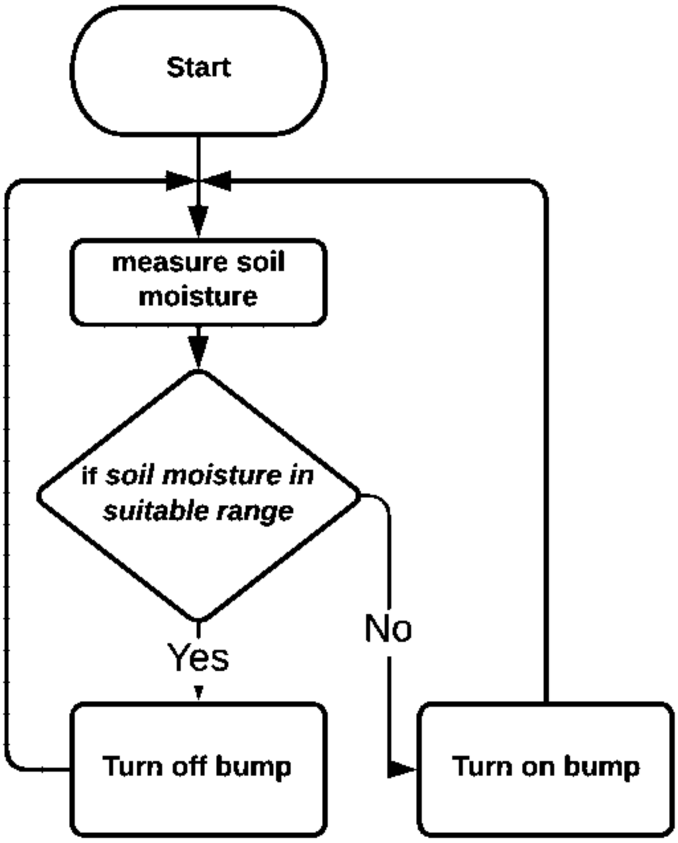
1. Block Diagram:



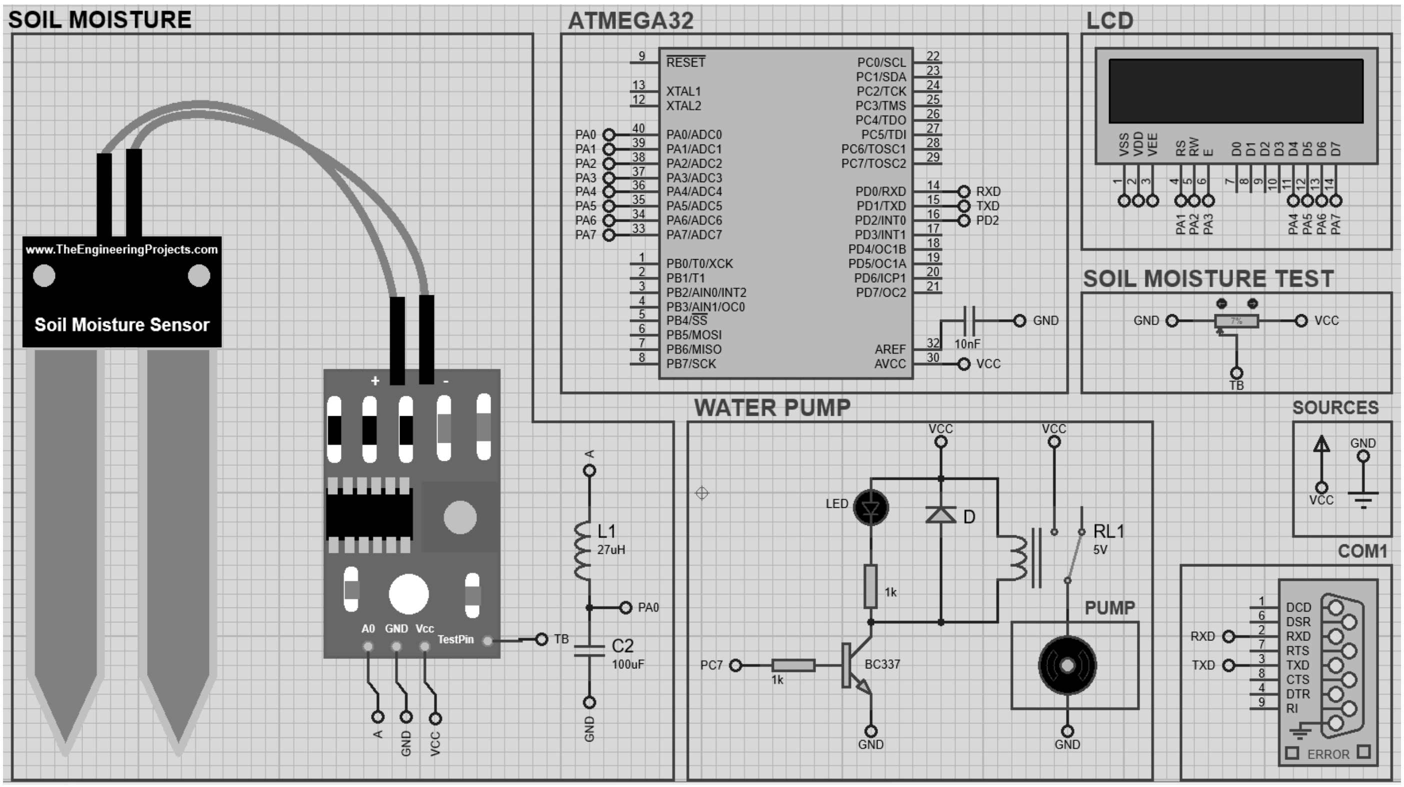
2. Block Diagram Description:

The system provides us to measure the moisture of soil by **soil moisture sensor** which is attached to **MCU** (we use ATMEGA32) via **comparator chip**. Then according to followed algorithm the MCU controls the **water bump** via **relay module** where the water pump connects to soil and water tank. And the soil moisture is displayed on **LCD**. Then the MCU is communicate with **PC** via USART (communication protocol) where the PC displays the soil moisture value and the water pump state (ON or OFF).

3. Flow chart:



Schematic Diagram (Proteus Circuit Diagram):



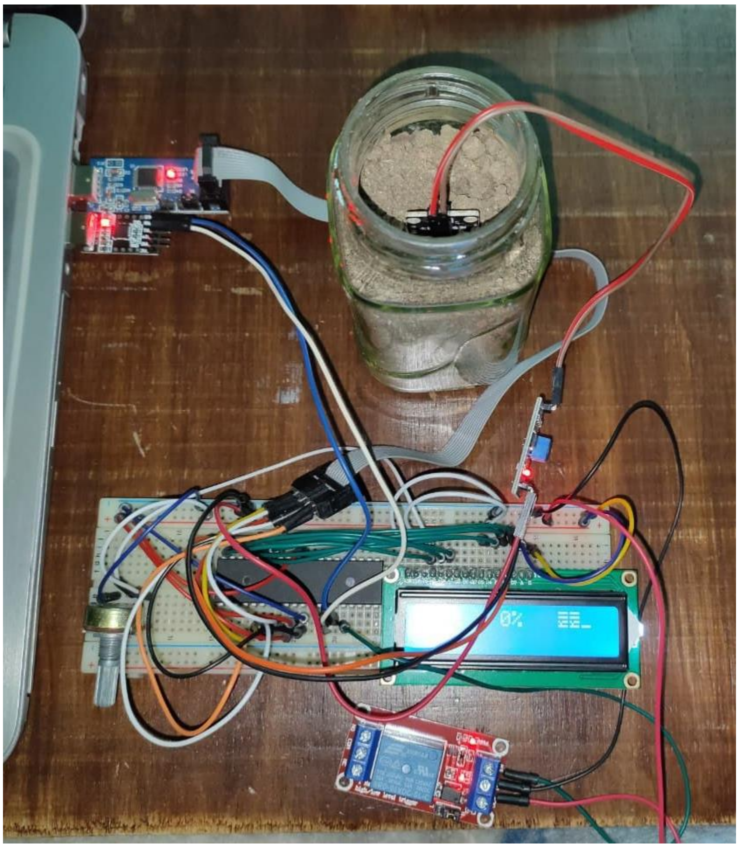
List of Components:

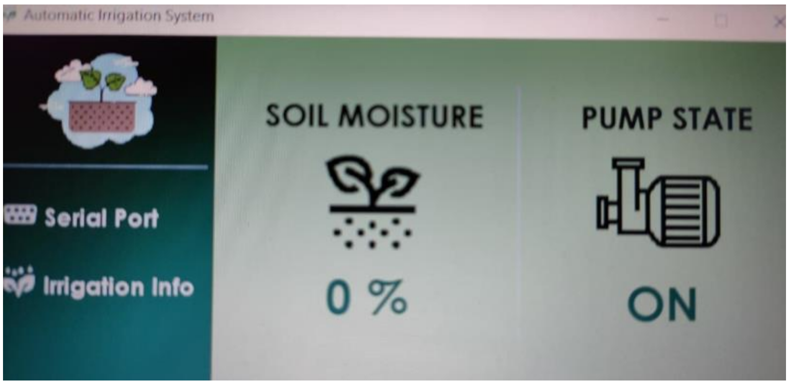
SN	Item Type	Item Code Name	Purpose	Quantity
1	MCU	ATMEGA32		1
2	Soil moisture module	-	To Measure the soil moisture.	1
3	LCD 16*2	LM016L	To display soil moisture.	1
4	Relay Module	-	To drive the 12 v water pump which the MCU cannot drive.	1

5	Water Pump 12v	AD20P-1230C	To supply the soil with water from water tank.	1
6	USB to TTL converter	CP2102	To connect the MCU with PC via USART.	1
7	potentiometer	-	To control the contrast of LCD.	1
8	jumpers	-	To connect between different component.	-
9	USBASP AVR Programmer	-	To burn the hex file of code on flash memory of MCU.	1

Real-Time Hardware Photo:

Low soil moisture:





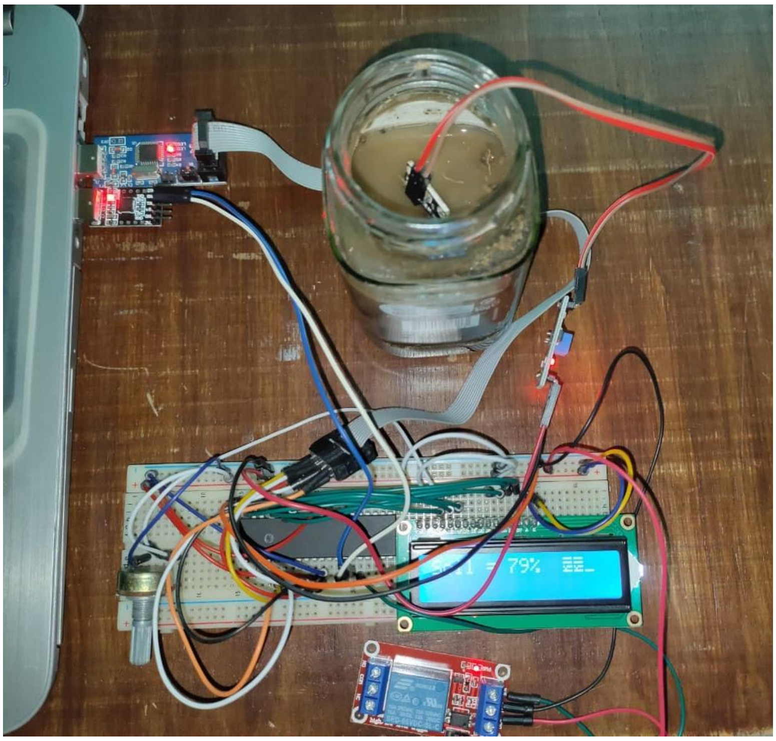
SOIL MOISTURE

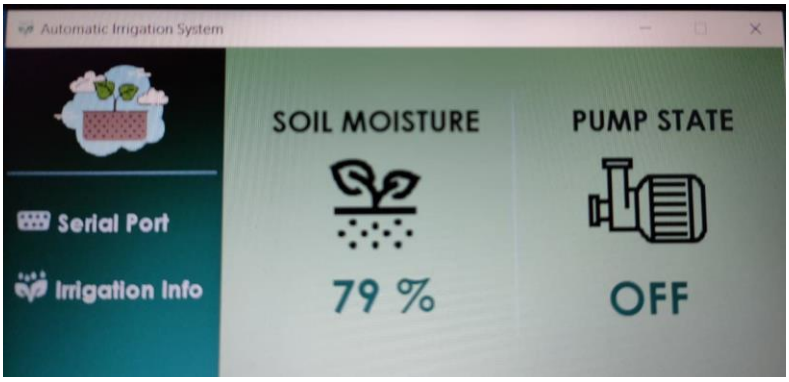
0 %

PUMP STATE

ON

High soil moisture:





SOIL MOISTURE

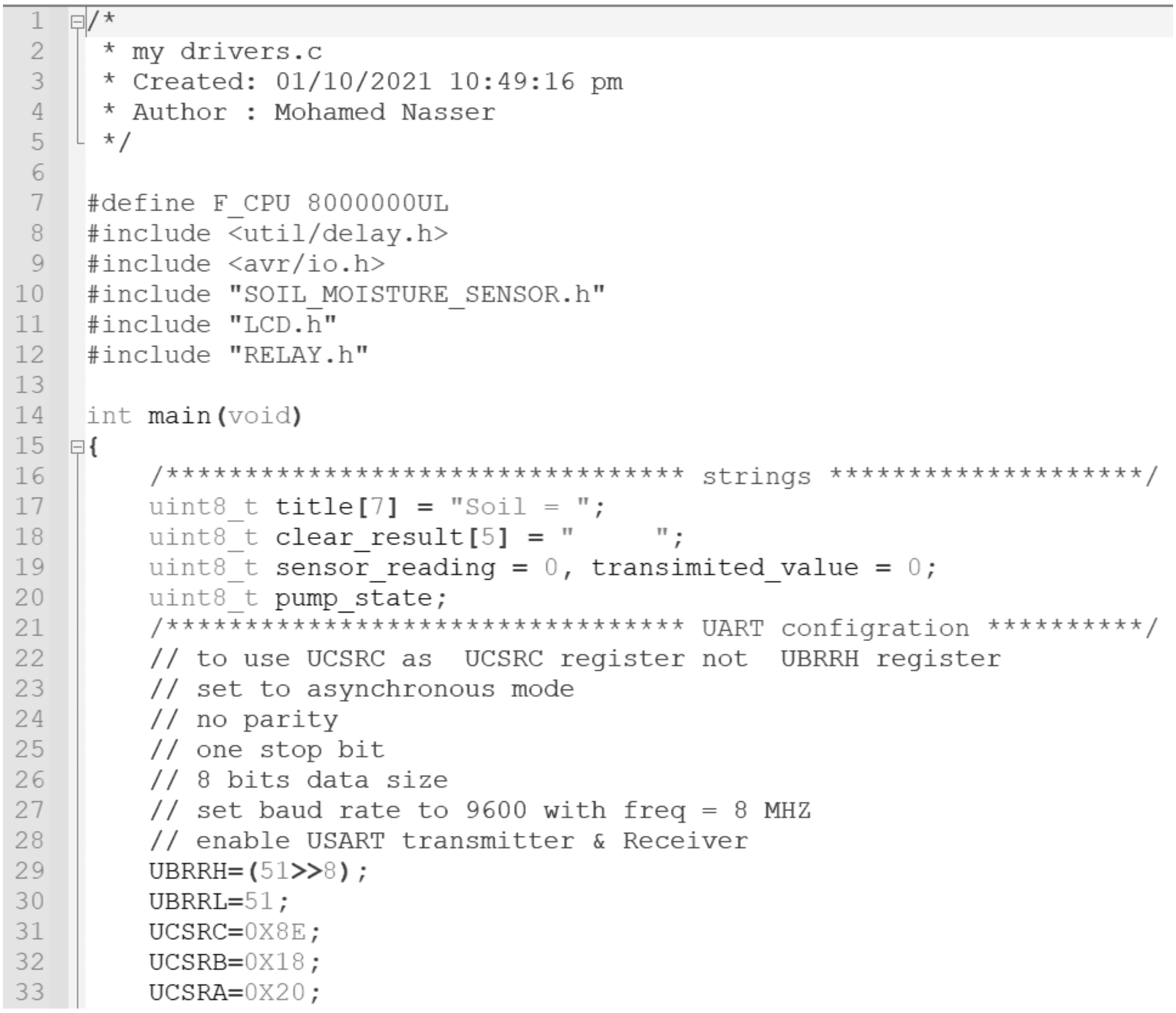
79 %

PUMP STATE

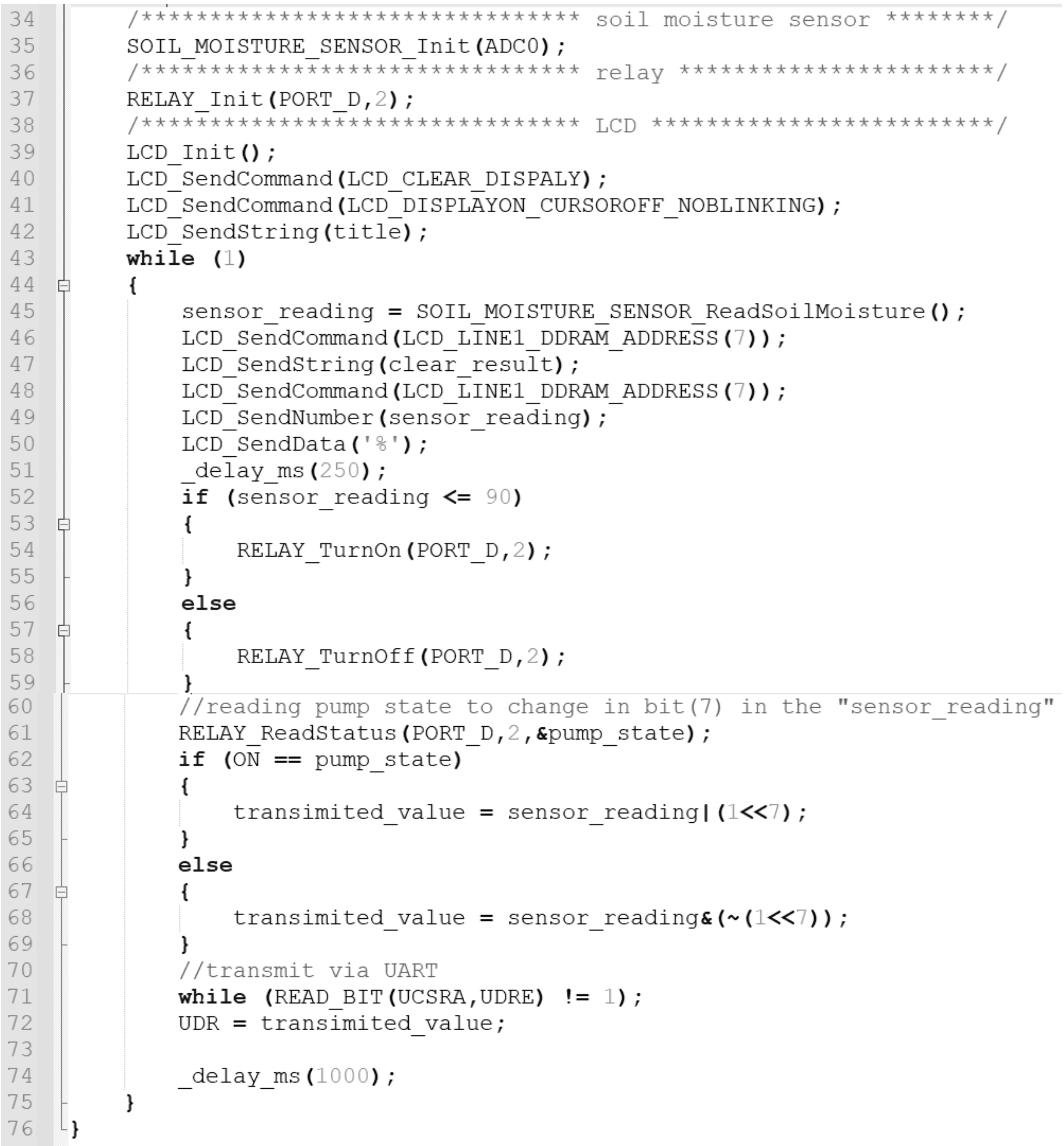
OFF

Source Code:

1. Hardware-side source code:

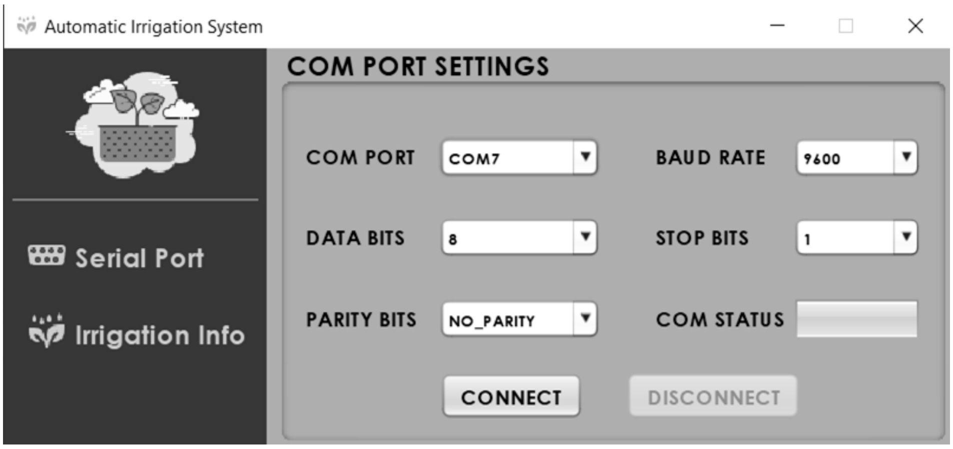


```c
/*
 * my drivers.c
 * Created: 01/10/2021 10:49:16 pm
 * Author : Mohamed Nasser
 */

#define F_CPU 8000000UL
#include <util/delay.h>
#include <avr/io.h>
#include "SOIL_MOISTURE_SENSOR.h"
#include "LCD.h"
#include "RELAY.h"

int main(void)
{
    /********************************* strings *********************/
    uint8_t title[7] = "Soil = ";
    uint8_t clear_result[5] = "     ";
    uint8_t sensor_reading = 0, transimited_value = 0;
    uint8_t pump_state;
    /********************************* UART configration ***********/
    // to use UCSRC as  UCSRC register not  UBRRH register
    // set to asynchronous mode
    // no parity
    // one stop bit
    // 8 bits data size
    // set baud rate to 9600 with freq = 8 MHZ
    // enable USART transmitter & Receiver
    UBRRH=(51>>8);
    UBRRL=51;
    UCSRC=0X8E;
    UCSRB=0X18;
    UCSRA=0X20;
```



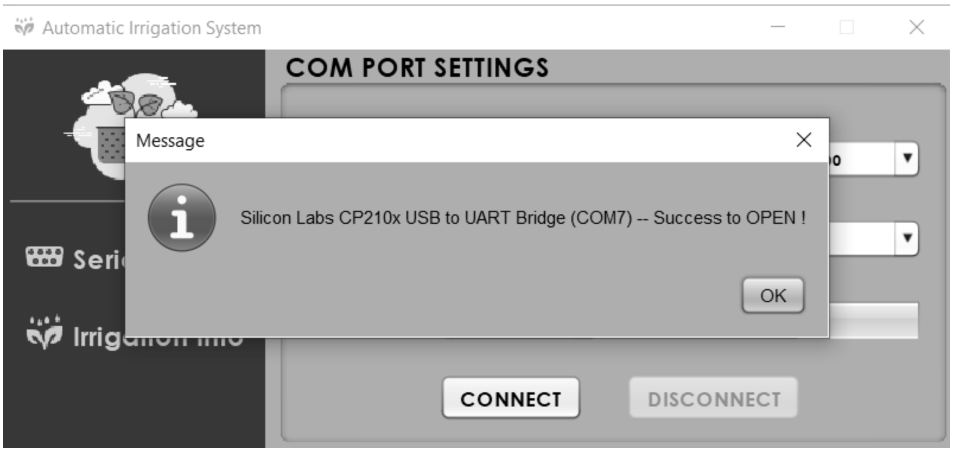
```
    /******************************** soil moisture sensor ********/
    SOIL_MOISTURE_SENSOR_Init(ADC0);
    /******************************** relay **********************/
    RELAY_Init(PORT_D,2);
    /******************************** LCD ************************/
    LCD_Init();
    LCD_SendCommand(LCD_CLEAR_DISPALY);
    LCD_SendCommand(LCD_DISPLAYON_CURSOROFF_NOBLINKING);
    LCD_SendString(title);
    while (1)
    {
        sensor_reading = SOIL_MOISTURE_SENSOR_ReadSoilMoisture();
        LCD_SendCommand(LCD_LINE1_DDRAM_ADDRESS(7));
        LCD_SendString(clear_result);
        LCD_SendCommand(LCD_LINE1_DDRAM_ADDRESS(7));
        LCD_SendNumber(sensor_reading);
        LCD_SendData('%');
        _delay_ms(250);
        if (sensor_reading <= 90)
        {
            RELAY_TurnOn(PORT_D,2);
        }
        else
        {
            RELAY_TurnOff(PORT_D,2);
        }
        //reading pump state to change in bit(7) in the "sensor_reading"
        RELAY_ReadStatus(PORT_D,2,&pump_state);
        if (ON == pump_state)
        {
            transimited_value = sensor_reading|(1<<7);
        }
        else
        {
            transimited_value = sensor_reading&(~(1<<7));
        }
        //transmit via UART
        while (READ_BIT(UCSRA,UDRE) != 1);
        UDR = transimited_value;

        _delay_ms(1000);
    }
}
```

2. PC-side source code (java code):



COM PORT SETTINGS

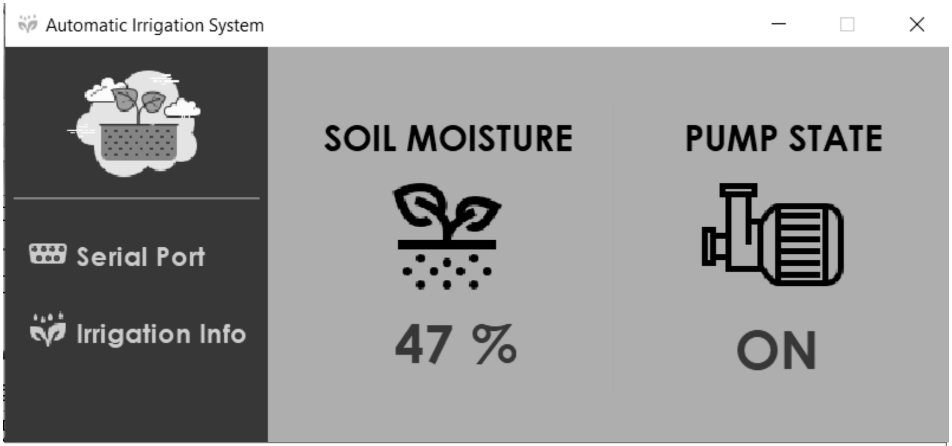
COM PORT: COM7
BAUD RATE: 9600
DATA BITS: 8
STOP BITS: 1
PARITY BITS: NO_PARITY
COM STATUS



COM PORT SETTINGS

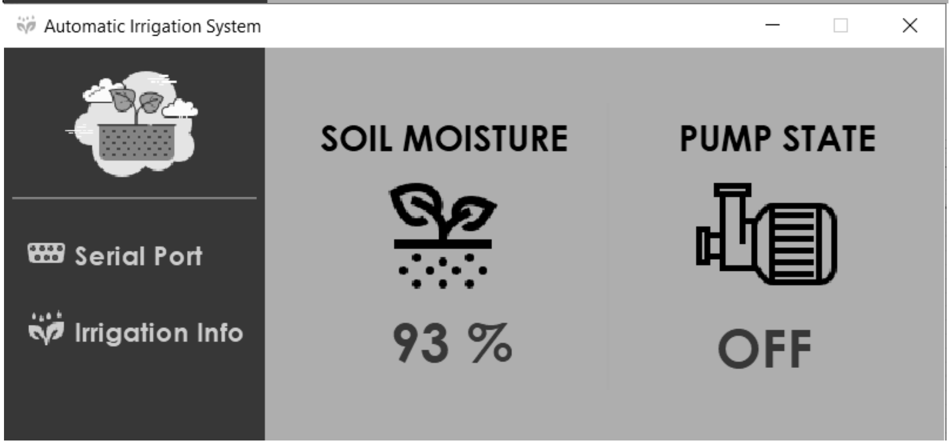
Message

Silicon Labs CP210x USB to UART Bridge (COM7) -- Success to OPEN !



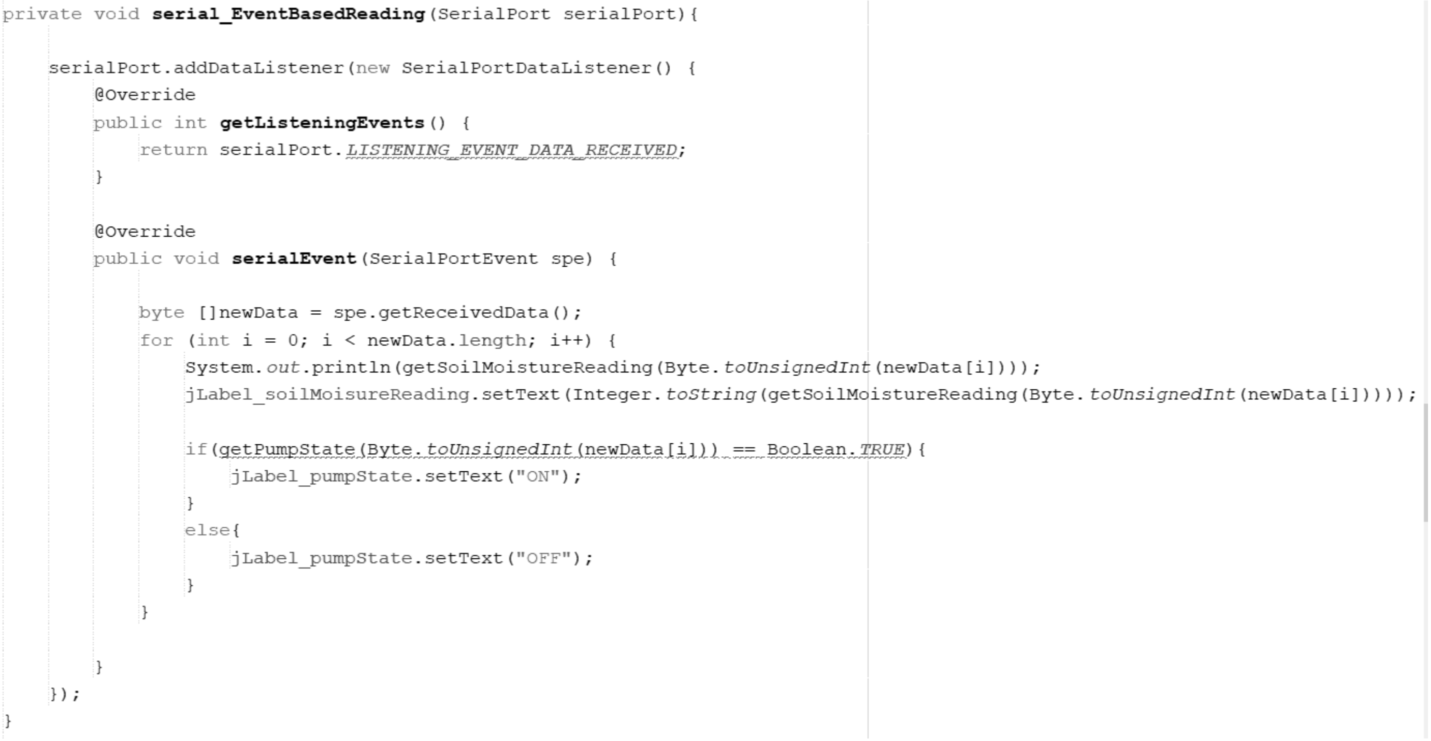
SOIL MOISTURE: 47 %

PUMP STATE: ON



SOIL MOISTURE: 93 %

PUMP STATE: OFF



```
private void serial_EventBasedReading(SerialPort serialPort){

    serialPort.addDataListener(new SerialPortDataListener() {
        @Override
        public int getListeningEvents() {
            return serialPort.LISTENING_EVENT_DATA_RECEIVED;
        }

        @Override
        public void serialEvent(SerialPortEvent spe) {

            byte []newData = spe.getReceivedData();
            for (int i = 0; i < newData.length; i++) {
                System.out.println(getSoilMoistureReading(Byte.toUnsignedInt(newData[i])));
                jLabel_soilMoisureReading.setText(Integer.toString(getSoilMoistureReading(Byte.toUnsignedInt(newData[i]))));

                if(getPumpState(Byte.toUnsignedInt(newData[i])) == Boolean.TRUE){
                    jLabel_pumpState.setText("ON");
                }
                else{
                    jLabel_pumpState.setText("OFF");
                }
            }

        }
    });
}
```