NAME: ADETORO MAYOWA SOLA

COURSE: ELECTRICAL/ELECTRICAL ENGINEERING

MAT NO: 18/ENG04/005

CONCEPTUALIZATION

In this design and implementation of such a project, steps have to be followed and these steps are the software development processes. In this initial stage, there is the **conceptualization**

We have to create a software that will be able to overcome the major challenges on the ABUAD FARM. We all know the major challenges are ‘’lack of modernization’’, ‘’poor infrastructure’’, and ‘’finance’’. This software will determine the temperature of the soil. After determining the temperature of the soil, we need to determine the moisture of the soil.

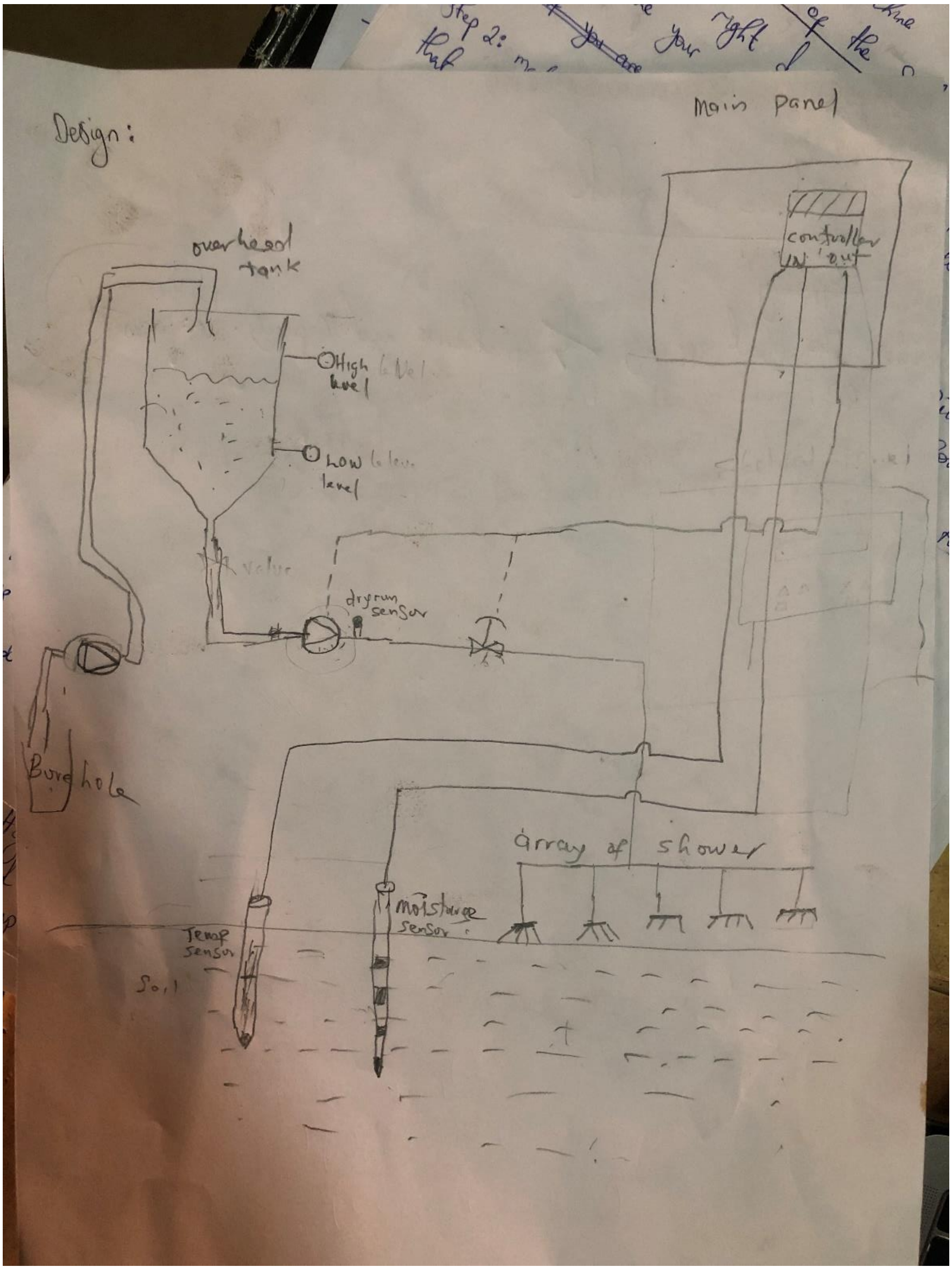
The software is going to be easy to operate. The time interval for the water system is to regulate the water flow on the farm. The alarm goes with the water system, if there is no sufficient water in the tank, the alarm will go off. For the safety of the software and ensure it is not misused a password will be enabled for the system. This is a general overview of the application.

SPECIFICATION.

- Water tank 1000litres capacity tank
- Temperature sensor of 0-100C
- pump of 2880 revolution per minute, 415volts A/C, stainless steel pump head with adequate mounting function with it.

- Temperature to current transmitter with output 4-20mmA.
- Soil moisture sensor
  Input voltage – 3.3-5V
  Output voltage- 0-4.2V d/c
  Input current 35mmA
  Output signal (both analog & digital)
- Controller (ABB Controller CM30)
  220-volt a/c power supply, 2 resistive input, 1 current input, 2 analog output (0-10V d/c and 4-20mmA d/c)
- Power supply: siemens power supply 415v A/C input and 24v d/c output 40A current
- Password: for administrator must include alphanumeric, upper- and lower-case character.
- Cable: flexible cable for wiring 1.5mm^2, motor 2.5mm^2 4 core cable, 1.5mm^2 screen cable for analog
- Contactor: k44E contactor, auxiliary contactor, 2 main contactor.
- Push button for start and stop, lamp for running, lamp for overload.
- Level probe: 2 level probes.
- 5/2 solenoid valve with actuator

DESIGN



There will be a source of water preferably from borehole that will fill water into the tank. This tank is equipped with high- and low-level sensor. At the outlet of the tank there is a pump and on/off valve.

There is an array of shower that will spray water into the soil as commanded by the controller.

There are two sensors buried under soil:

- Temperature sensor
- Moisture sensor

There will be an electrical panel where process controller is mounted and all electrical component mounted.

Moisture and temperature sensor are wired to the input of the controller while digital control of the pump and outlet valve.

IMPLEMENTATION.

All input is converted from analog to digital standard of control which is 4-20mmA. We have to handle the safety part of the program and those are dry run sensor and low/high level sensor. The pump must not run if the dry run sensor does not detect water. When water is below low level the discharge pump should stop. If the high-level sensor is on the underground pump should stop to prevent pump from running day. There should be alarm when the overhead tank is empty. The temperature and moisture state of the soil, the result of the measurement sent to the controller and compare with the set point values on the controller. If the temperature and moisture is higher than set point value set on the controller the supply pump is turn on and the outlet valve is opened to wet the soil. Measurement continues until the measured value are now below set point value on the controller, the valve will close and the pump will stop.

This continues until the operator stops the operation.

TESTING & DEBUGGING

This will do by simulation; all output is individually tested by sending a simulate output signals to them and observe their response. All input signals are also simulated either by using decade box for resistive circuit or signal generators. If all the different sections are independently tested and simulated and they are ok, then we can run the entire project with water under close monitor.

RELEASE AND UPDATE.

The plant efficiency, effectiveness and capacity is then communicated and handed over to end user like a certificate, which must be confirmed & signed off.

HARDWARE FOR CONTROL OF WATER.

You have to mount temperature sensor at the depth you want it to be. You will be wire it down to the controller. Then the control side on the controller you will set upper and lower limit and hysteresis. When the soil temperature is 40C, the control will start the pump through the controller that will water the farm produce. Thereafter, as you are watering it the temperature will start decreasing, when it gets to 30C you will set on the controller so the pump will stop. Hysteresis is the difference between high and low limit.

SOFTWARE FOR CONTROL OF WATER FOR ALARM.

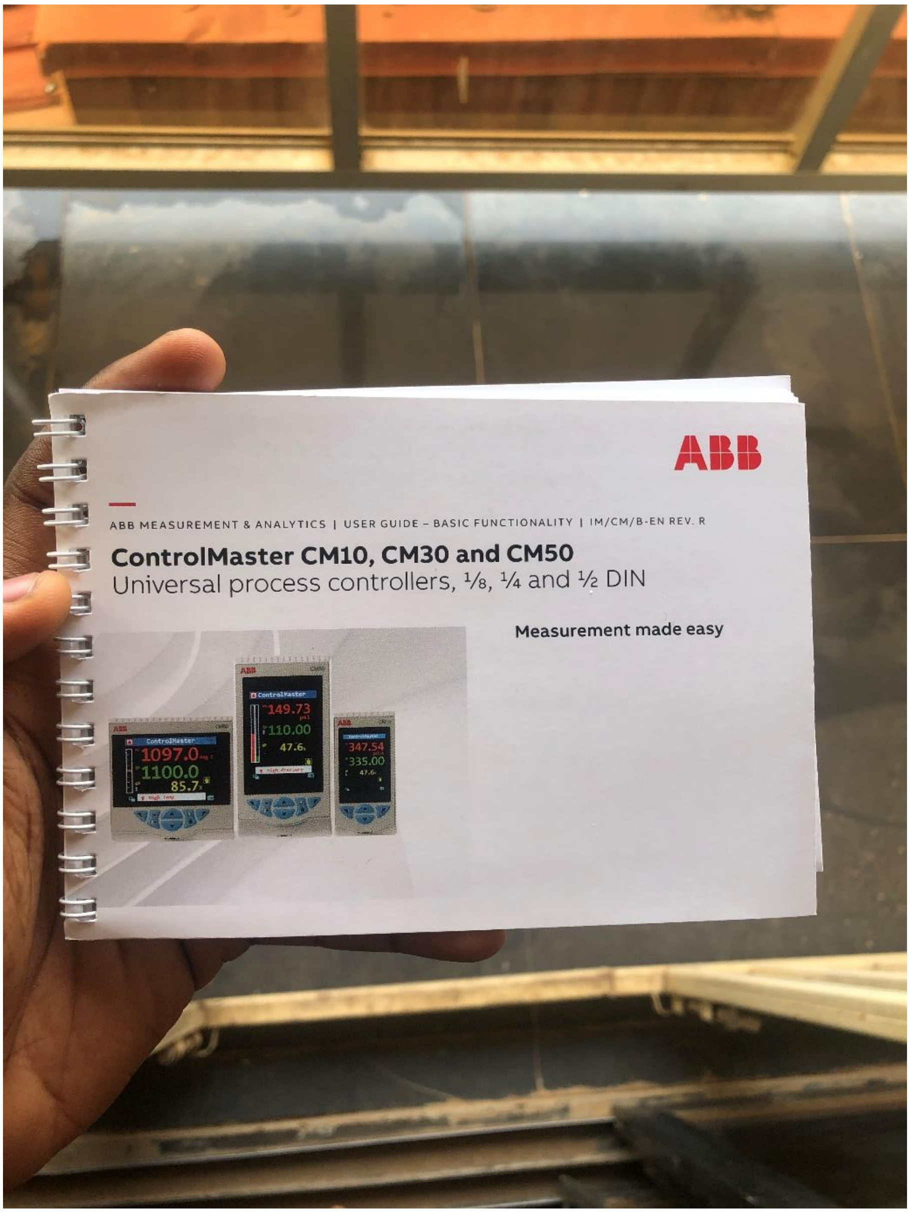
Configure the temperature controller such that you set the high level, low level measurement. Set pump restart differential.
For the tank they will be two level probe high and low level.
When the overhead tank is full and the high-level probe has cut off the supply pump that will pump water into the overhead

tank stops when it reaches high level for few seconds. Watering pump starts when soil temperature is on or above 40C, it will then stop when the temperature is 30C with some time delay from restarting.

For alarm limit there will be another level sensor called low-low level mounted on the tank, this stop the pump from preventing it from dry running. There will be an alert lamp/siren that will alert the operator to take necessary action.

Hardware features.

- Electrical pump: the pump responsible for taking water from the tank to wet the soil.
- Contactor: electromagnetic device for switching on the pump.
- Temperature controller: is the programmable logic controller that read in the input of all sensors, process it in accordance to set of rules(program) and command the output function.
- Temperature/ moisture sensor: is an input to the controller
- Level probe: for level control inside the tank.



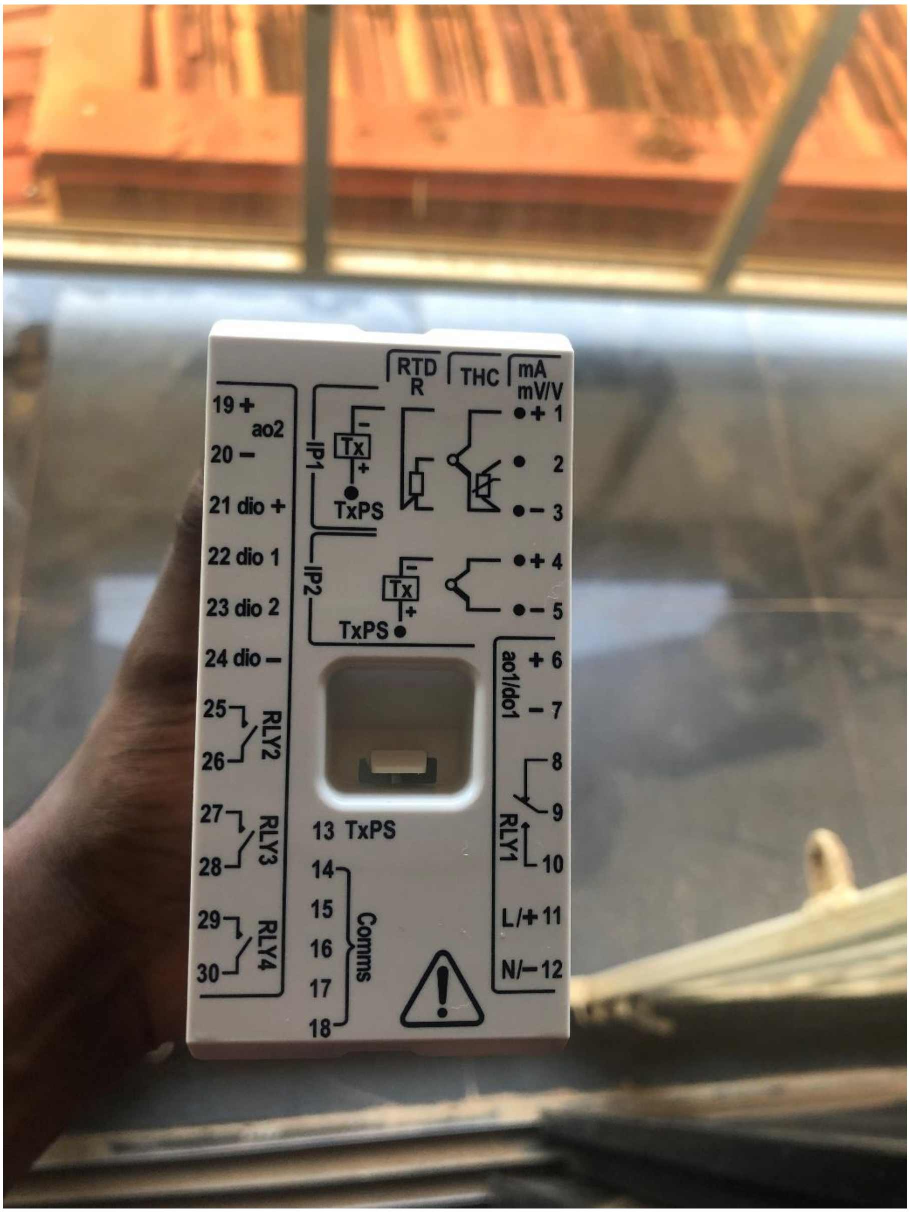
ABB

ABB MEASUREMENT & ANALYTICS | USER GUIDE – BASIC FUNCTIONALITY | IM/CM/B-EN REV. R

ControlMaster CM10, CM30 and CM50

Universal process controllers, ⅛, ¼ and ½ DIN

Measurement made easy


RTD
R
THC
mA
mV/V
19 +
ao2
20 −
21 dio +
22 dio 1
23 dio 2
24 dio −
25
26
RLY2
27
28
RLY3
29
30
RLY4
IP1
Tx
−
+
TxPS
IP2
Tx
+
TxPS
+ 1
2
− 3
+ 4
− 5
+ 6
ao1/do1
− 7
8
9
RLY1
10
L/+ 11
N/− 12
13 TxPS
14
15
16
17
18
Comms

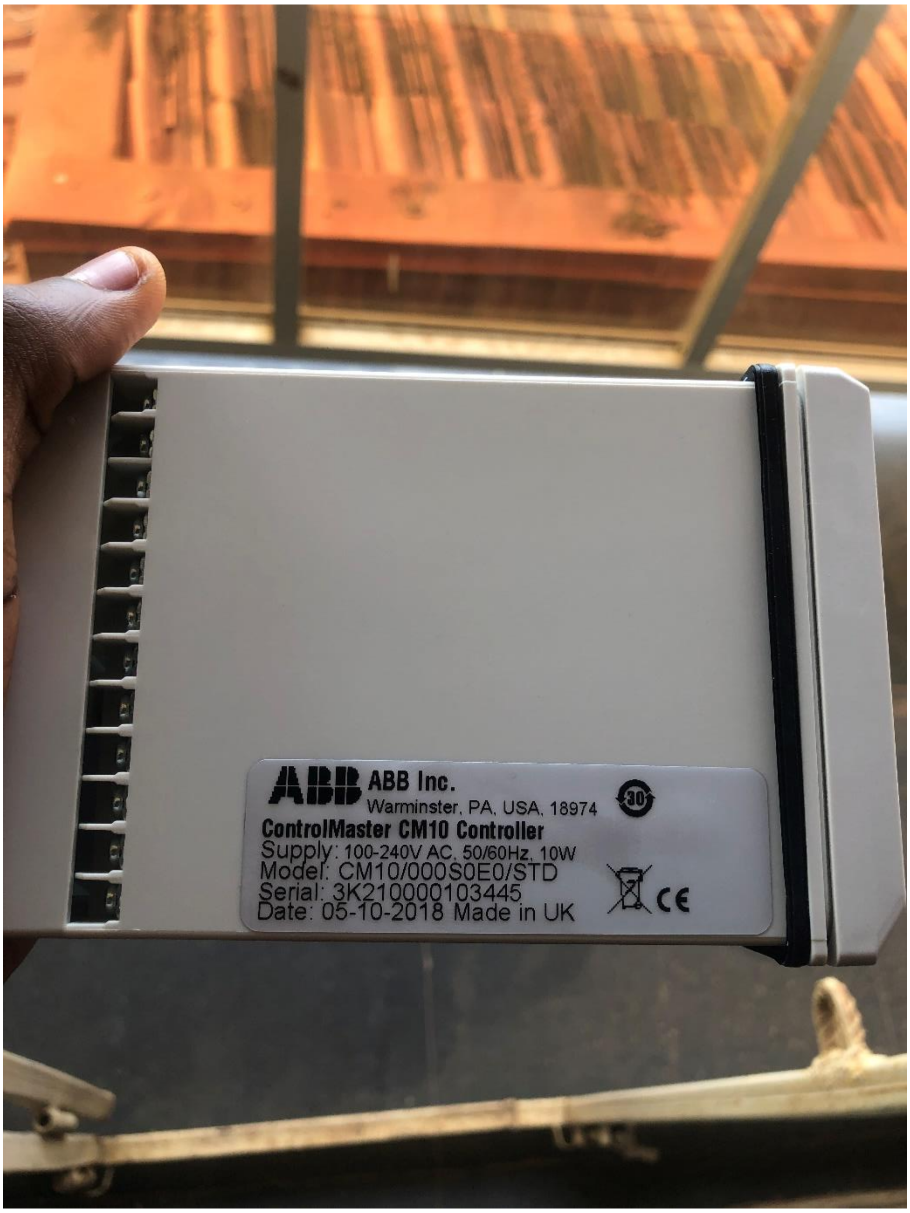


ABB ABB Inc.
Warminster, PA, USA, 18974

ControlMaster CM10 Controller

Supply: 100-240V AC, 50/60Hz, 10W

Model: CM10/000S0E0/STD

Serial: 3K210000103445

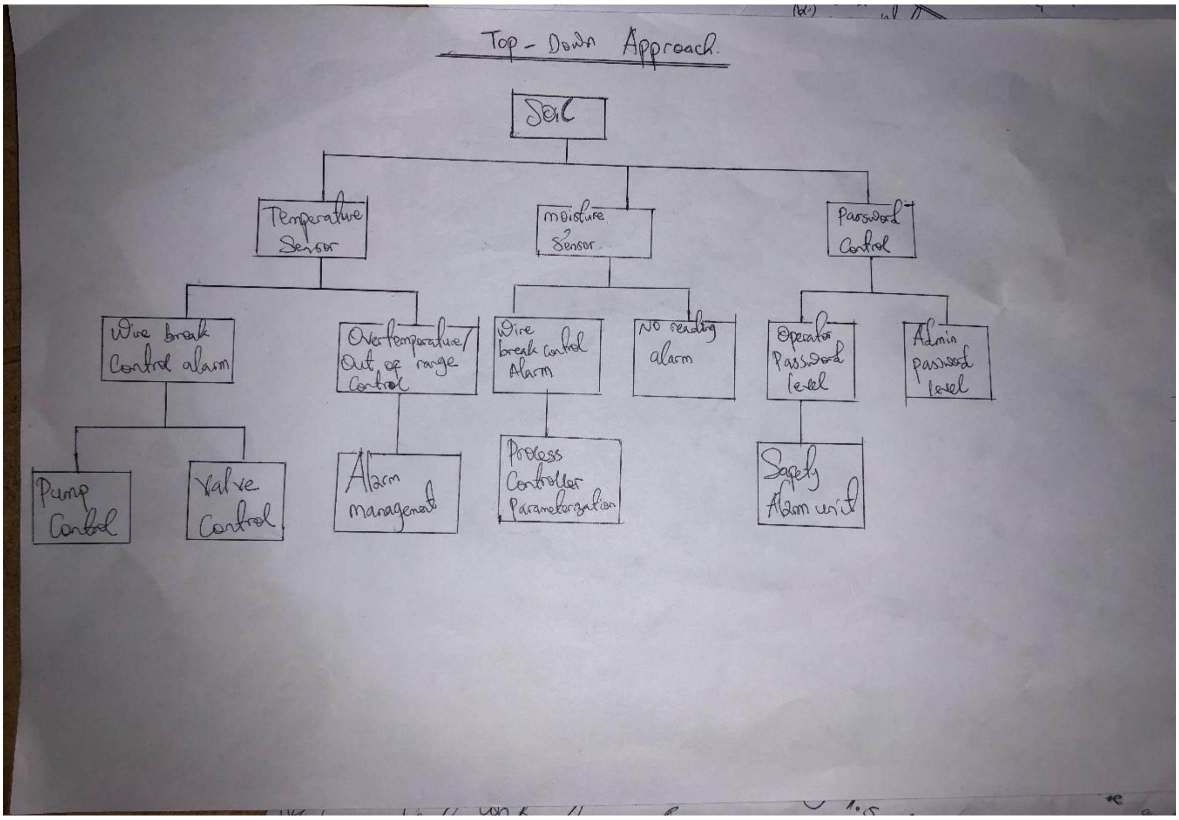
Date: 05-10-2018 Made in UK

Software features

This is a generic controller. It is used in control system engineering which include temperture, pressure and volumetric control.

- Access control: primarily for assigning password level for different user management (user, operator, technician and administrator password)

Top-down approach.



ALGORITHM

Start

Read temperature of the soil

Print finish reading 20 data

Calculate average temperature ADC value & convert to Celsius

Read soil moisture output voltage and convert to ADC value

Print finish reading 20 data

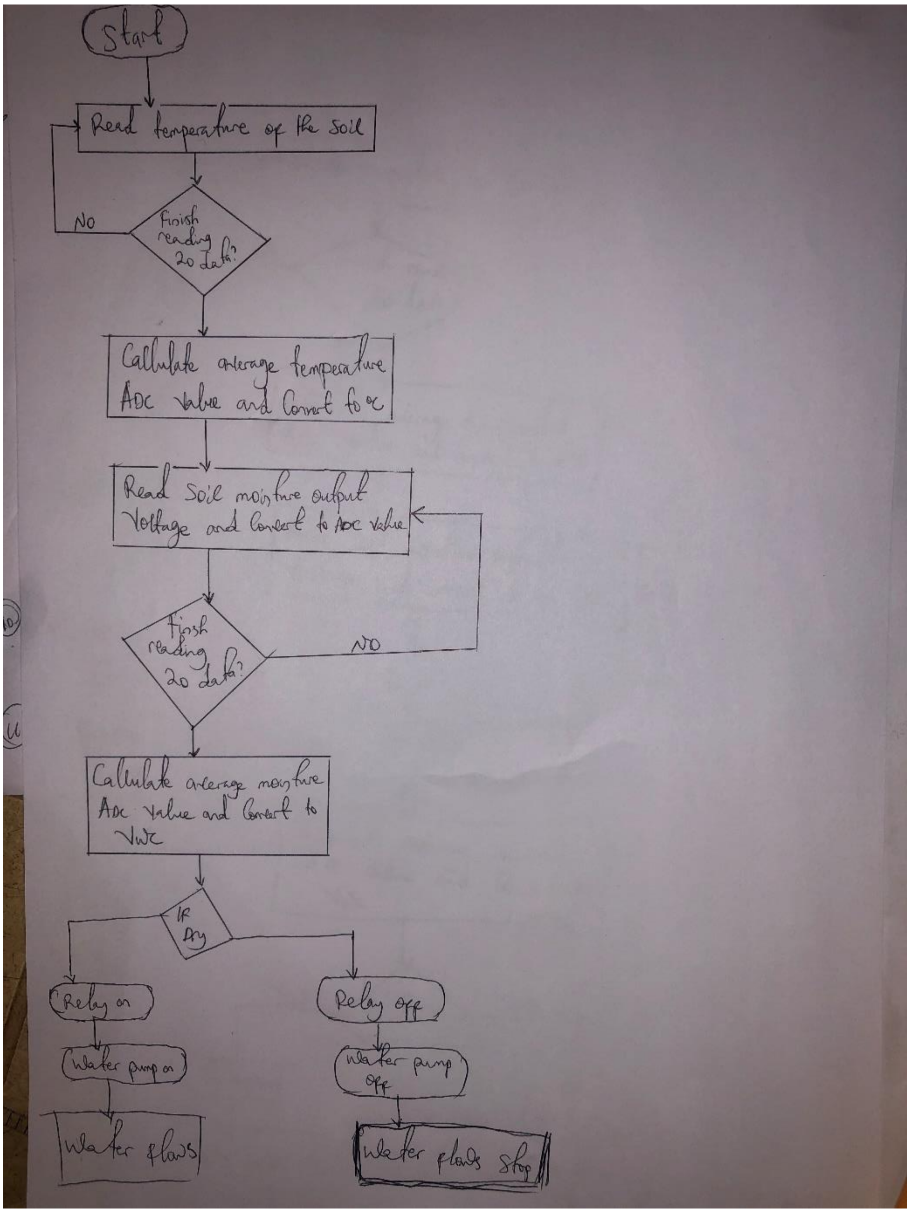
Calculate average moisture ADC value and convert to VWC

Read if there is no sufficient water

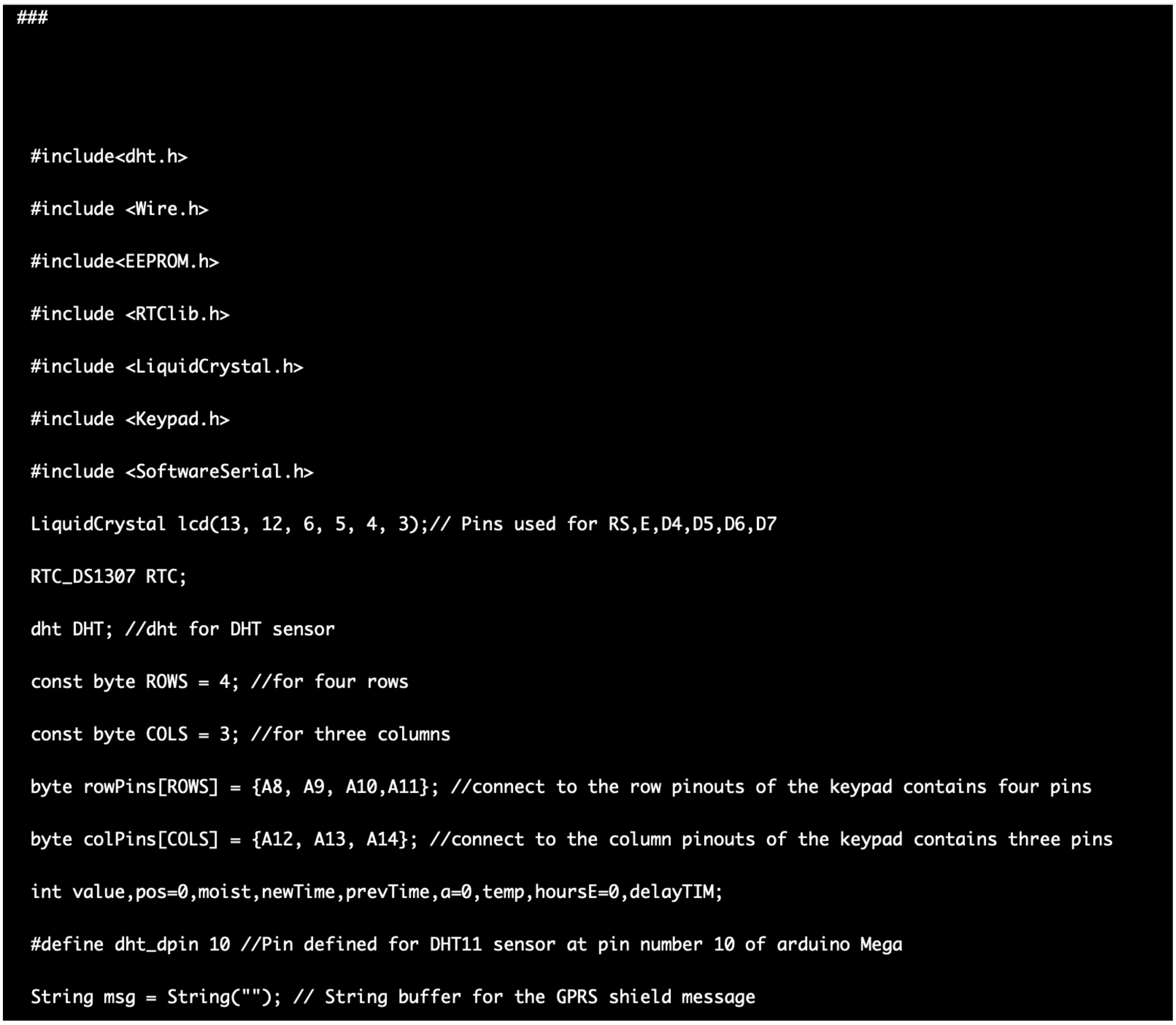
If dry > wet

```
Print dry no
Else
Print wet yes
Stop
```

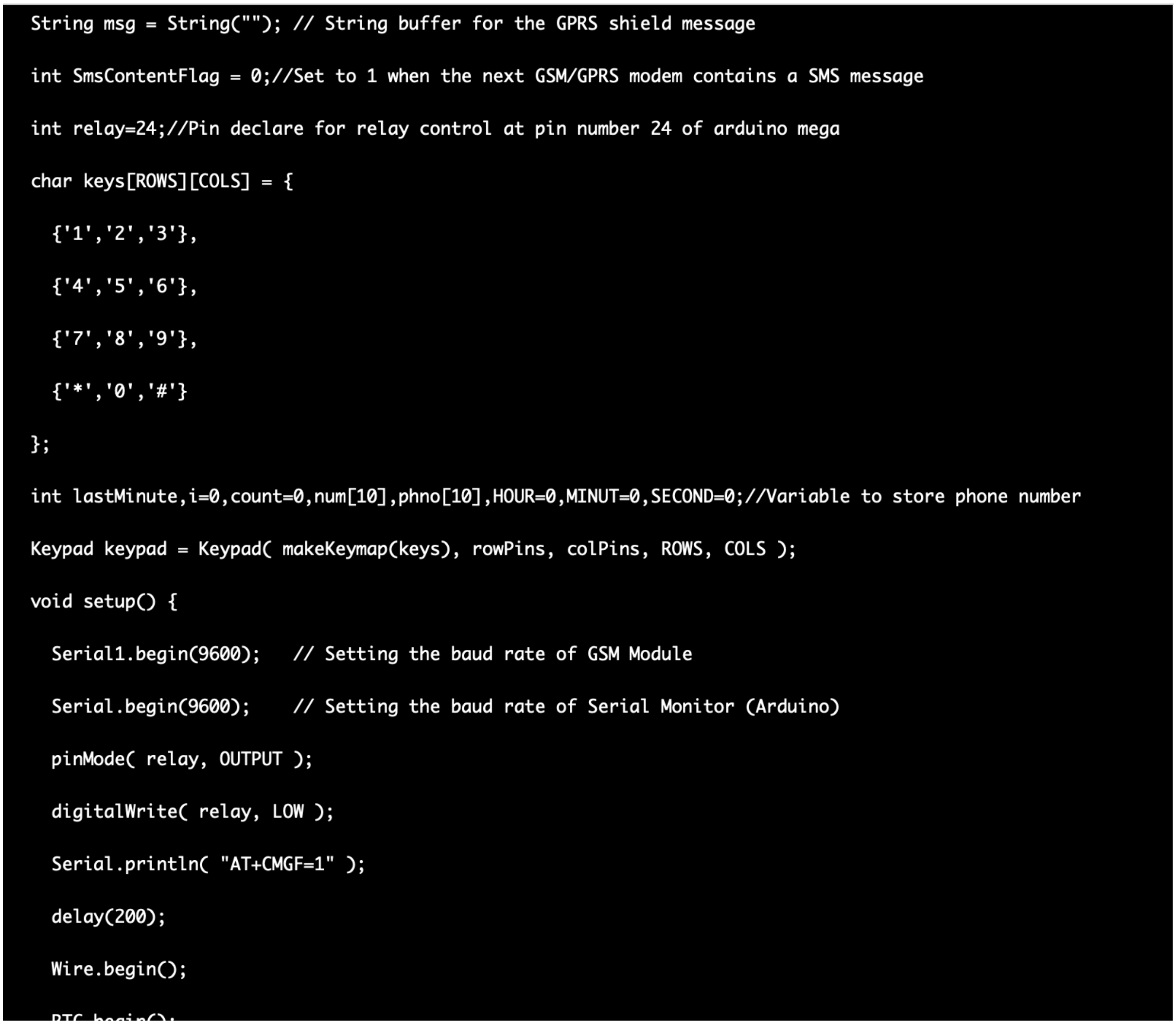
FLOWCHART


Start
Read temperature of the soil
No
Finish reading 20 data?
Callulate average temperature ADC value and Convert to °C
Read Soil moisture output Voltage and Convert to ADC value
Finish reading 20 data?
NO
Callulate average moisture ADC value and Convert to VWC
If Dry
Relay on
Relay off
Water pump on
Water pump off
Water flows
Water flows stop

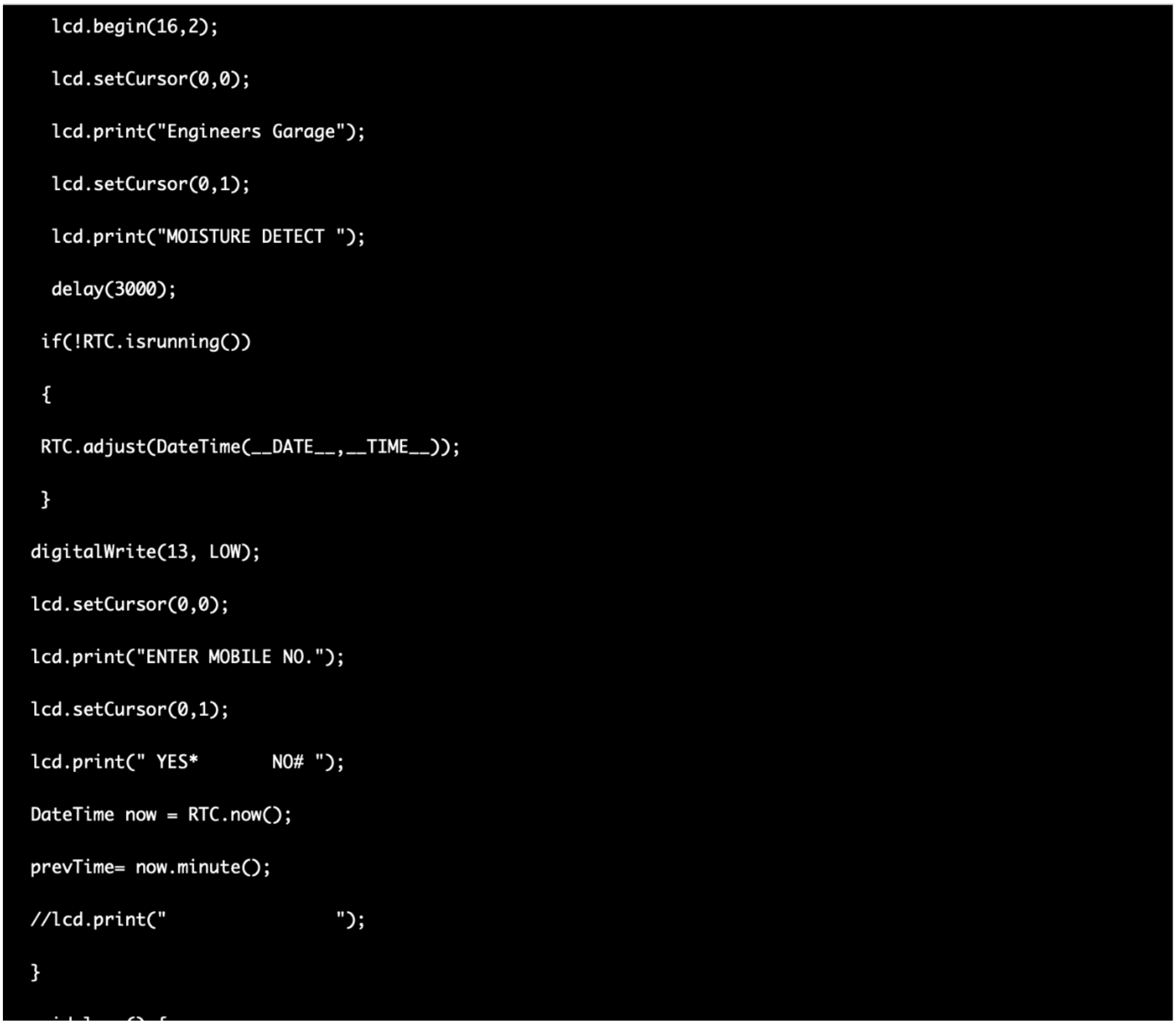
IRRIGATION SYSTEM CODE



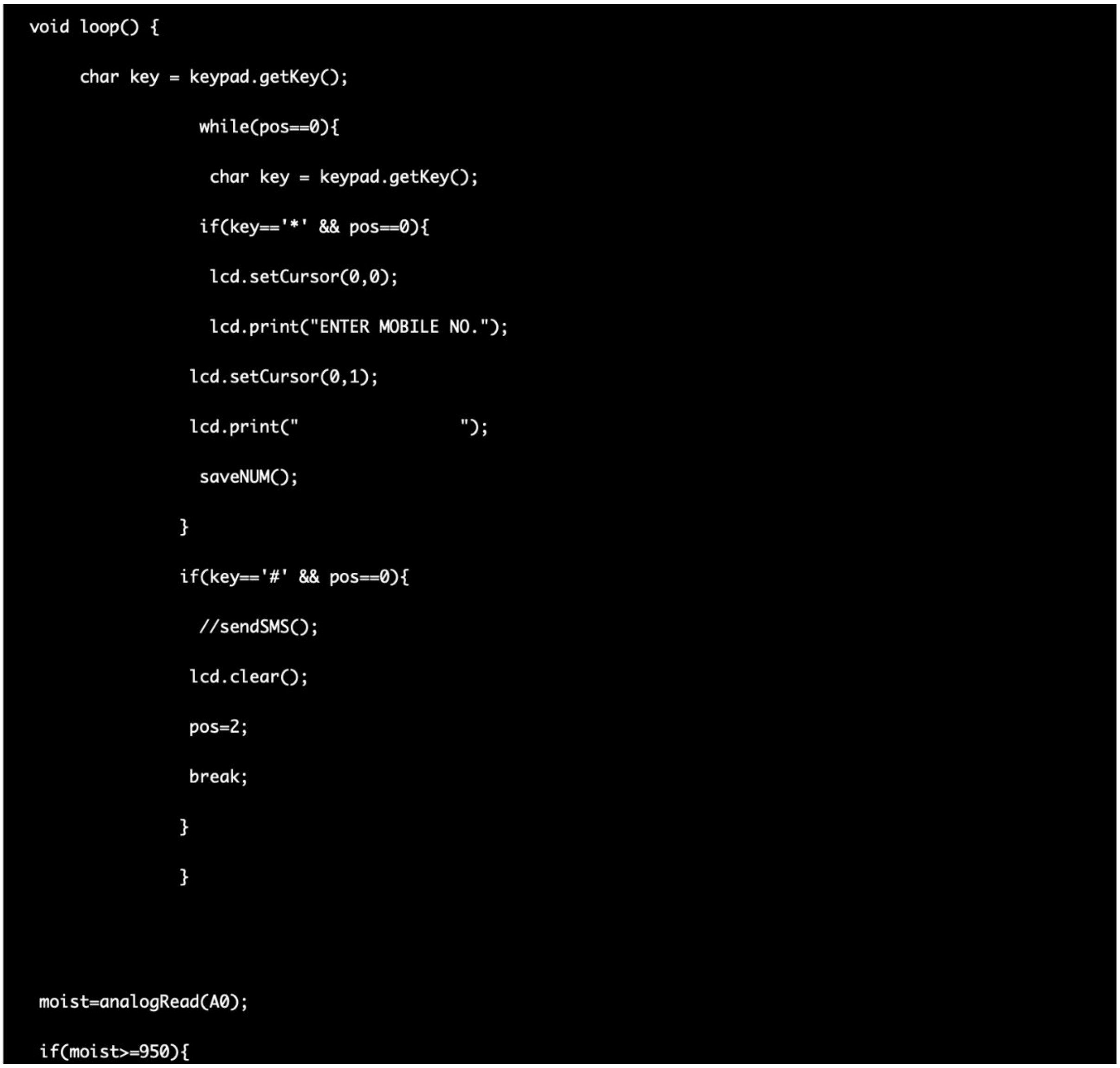
```
###



#include<dht.h>

#include <Wire.h>

#include<EEPROM.h>

#include <RTClib.h>

#include <LiquidCrystal.h>

#include <Keypad.h>

#include <SoftwareSerial.h>

LiquidCrystal lcd(13, 12, 6, 5, 4, 3);// Pins used for RS,E,D4,D5,D6,D7

RTC_DS1307 RTC;

dht DHT; //dht for DHT sensor

const byte ROWS = 4; //for four rows

const byte COLS = 3; //for three columns

byte rowPins[ROWS] = {A8, A9, A10,A11}; //connect to the row pinouts of the keypad contains four pins

byte colPins[COLS] = {A12, A13, A14}; //connect to the column pinouts of the keypad contains three pins

int value,pos=0,moist,newTime,prevTime,a=0,temp,hoursE=0,delayTIM;

#define dht_dpin 10 //Pin defined for DHT11 sensor at pin number 10 of arduino Mega

String msg = String(""); // String buffer for the GPRS shield message
```



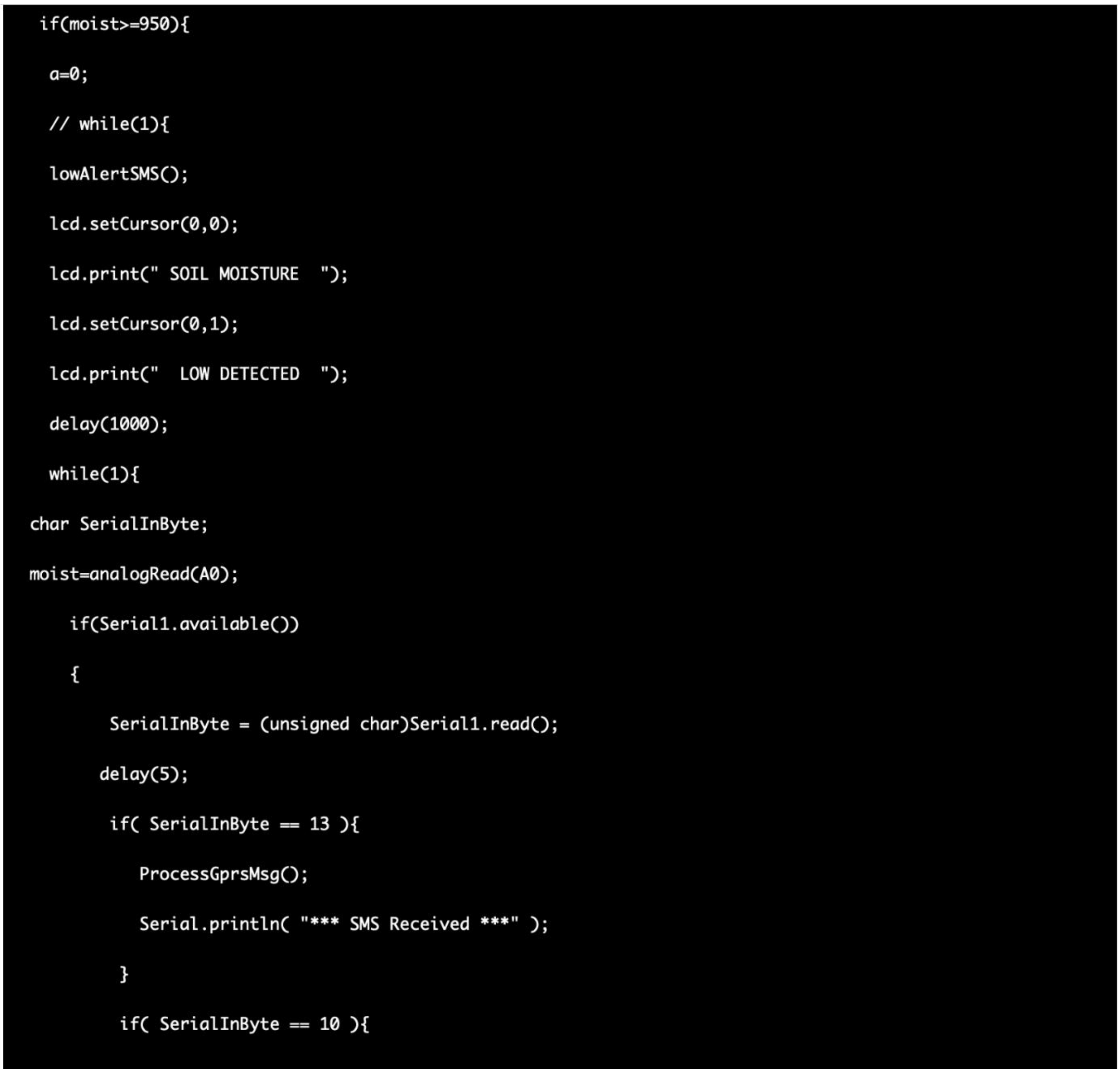
```
String msg = String(""); // String buffer for the GPRS shield message

int SmsContentFlag = 0;//Set to 1 when the next GSM/GPRS modem contains a SMS message

int relay=24;//Pin declare for relay control at pin number 24 of arduino mega

char keys[ROWS][COLS] = {

  {'1','2','3'},

  {'4','5','6'},

  {'7','8','9'},

  {'*','0','#'}

};

int lastMinute,i=0,count=0,num[10],phno[10],HOUR=0,MINUT=0,SECOND=0;//Variable to store phone number

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup() {

  Serial1.begin(9600);   // Setting the baud rate of GSM Module

  Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)

  pinMode( relay, OUTPUT );

  digitalWrite( relay, LOW );

  Serial.println( "AT+CMGF=1" );

  delay(200);

  Wire.begin();

  RTC.begin();
```



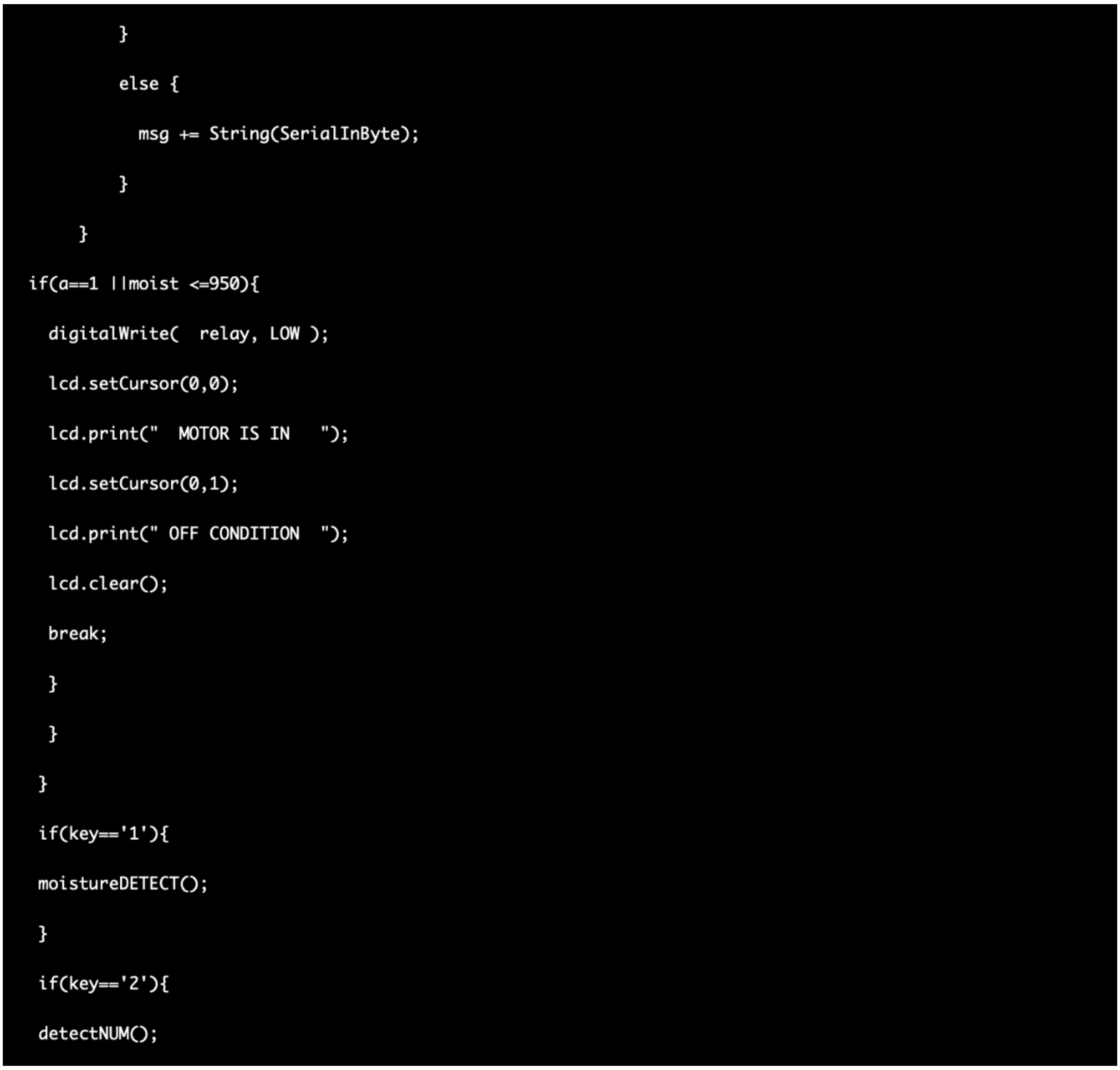
```
  lcd.begin(16,2);

  lcd.setCursor(0,0);

  lcd.print("Engineers Garage");

  lcd.setCursor(0,1);

  lcd.print("MOISTURE DETECT ");

  delay(3000);

 if(!RTC.isrunning())

 {

 RTC.adjust(DateTime(__DATE__,__TIME__));

 }

digitalWrite(13, LOW);

lcd.setCursor(0,0);

lcd.print("ENTER MOBILE NO.");

lcd.setCursor(0,1);

lcd.print(" YES*       NO# ");

DateTime now = RTC.now();

prevTime= now.minute();

//lcd.print("                ");

}
```



```
void loop() {

     char key = keypad.getKey();

                 while(pos==0){

                  char key = keypad.getKey();

                 if(key=='*' && pos==0){

                  lcd.setCursor(0,0);

                  lcd.print("ENTER MOBILE NO.");

                lcd.setCursor(0,1);

                lcd.print("                ");

                 saveNUM();

               }

               if(key=='#' && pos==0){

                 //sendSMS();

                lcd.clear();

                pos=2;

                break;

               }

               }



 moist=analogRead(A0);

 if(moist>=950){
```



```
 if(moist>=950){
  a=0;
  // while(1){
  lowAlertSMS();
  lcd.setCursor(0,0);
  lcd.print(" SOIL MOISTURE  ");
  lcd.setCursor(0,1);
  lcd.print("  LOW DETECTED  ");
  delay(1000);
  while(1){
char SerialInByte;
moist=analogRead(A0);
    if(Serial1.available())
    {
        SerialInByte = (unsigned char)Serial1.read();
       delay(5);
        if( SerialInByte == 13 ){
           ProcessGprsMsg();
           Serial.println( "*** SMS Received ***" );
         }
         if( SerialInByte == 10 ){
```



```
         }
         else {
           msg += String(SerialInByte);
         }
     }
if(a==1 ||moist <=950){
  digitalWrite(  relay, LOW );
  lcd.setCursor(0,0);
  lcd.print("  MOTOR IS IN   ");
  lcd.setCursor(0,1);
  lcd.print(" OFF CONDITION  ");
  lcd.clear();
  break;
  }
  }
 }
 if(key=='1'){
 moistureDETECT();
 }
 if(key=='2'){
 detectNUM();
```