International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Smart Irrigation System using Arduino with Solar Power

Saleh Elkelani Babaa
Systems Engineering Department
Military Technological College
PO Box 262, Postal Code 111
Muscat, Oman

Muneer Ahmed
Systems Engineering Department
Military Technological College
PO Box 262, Postal Code 111
Muscat, Oman

Babatunde Samuel Ogunleye
Systems Engineering Department
Military Technological college
PO Box 262, Postal Code 111
Muscat, Oman

Salim Ahmed Al-Jahdhami
Systems Engineering Department
Military Technological College
PO Box 262, Postal Code 111
Muscat, Oman

Shahid Ali Khan
Systems Engineering Department
Military Technological College
PO Box 262, Postal Code 111
Muscat, Oman

John Regan Pillai
Systems Engineering Department
Military Technological College
PO Box 262, Postal Code 111
Muscat, Oman

***Abstract*— This paper deals with the innovative technology in considering the various ways to irrigate the agricultural land using solar power. Since the agriculture plays the significant role in improving the country's economy, an improvement should be applied in order to increase the productivity and expand the quality of crops. So, this project signifies a smart Auto-irrigation system by using soil moisture sensors is connected to the Arduino Uno which act as a controller and a global System for mobile communication which is used to transmit and receive the data between the controller and user. The study is conducted in practically and has achieved the objectives.**

Keywords—Soil Moisture sensors, Arduino Uno, GSM, Solar Power

I. INTRODUCTION

The increasing demand for energy, the constant decline in existing sources of fossil fuels and the growing alarm regarding environment pollution, have pushed mankind to discover new non-conventional, renewable energy resources such as solar, wind energy, etc. for the production of electrical energy [1]. Since Oman receives sunlight all 12 months of a year. Hence utilizing it in the different fields is a clever idea. Solar energy is the most ample source of energy in the world. Photovoltaic generation is an effective approach for using the solar energy. Photovoltaic pumping systems (PVPs) are easy to be installed in any place and they require less maintenance maybe every 5 to 10 years intervals so that reduce the overall cost, just the cost of the photovoltaic cells and the other equipment needed which consider as the initial capital cost [1] [2]. Also, they doesn't require non-renewable source of power to operate because the dependent of solar power. Since the solar pumps doesn't relies to the diesel or oil so they consider as a silent and clean system with no air pollution or noise [3]. So, this such system is good for people who live in faraway from the water and electricity networks and even for those who live in cities [4].

The cost of solar panels has been continuously decreasing which encourages its usage in various sectors. One of the applications of this technology is used in irrigation systems for farming [5]. Solar powered irrigation system can be an appropriate alternative for farmers in the present state of energy crisis in Oman and other countries [1]. In this paper we propose an smart irrigation system using solar power which drives water pumps to pump water from bore well to a tank and the outlet valve of tank is automatically regulated using Arduino UNO, GSM and moisture sensor to control the flow rate of water from the tank to the irrigation field which optimizes the use of water [6].

An Arduino-based automated irrigation system use Android smart phone for remote control, is suggested by A.N. Arvindan and Keerthika. D [7]. They said that this system will be economical and easy to use. This system consist of Arduino Uno processor gets its input voltage signal from the soil moisture sensors which measure the moisture content in the soil. The Arduino compare the data came from the sensors with predetermined threshold value. The Arduino connected to the Android smart phone by a wireless via HC-05 module. The received data in the Android smart phone is displayed on the user interface (UI). The UI provide easy remote control of irrigation for the user involve switching ON and OFF [7].

Srishti and Rawal [8], suggested that making smart agricultural by using IoT (Internet of Things) technologies. The idea of their project has three concerns, firstly it involve a sound alarm used to detect any intruder into the farm weather it was a human or animal. Secondly, the project was to prevent the crops from damage during rain. The rain water also being recycled for irrigation efficiency. Thirdly, was for smart irrigation. The operations performed by interfacing sensors, Wi-Fi module and GSM module. This project intend to solve many problems like reducing wastage of water, human effort and give the farmer updated information about the field through his mobile device. The smart irrigation system is firmware based. Figure 4, show the project system configuration [8].

A. Methodology

In order to have good irrigation system, the specification of the water pump should satisfy the required land area which is being irrigated. So, initially we should calculate the land area for irrigation and then select the suitable water pump to irrigate on a particular area. Also, with respect to the solar

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

panel, it should be able to generate the suitable amount of power to operate the water pump and the control system [9] [10]-[20].

First, the system implemented to record humidity data using soil sensors as shown in the threshold of the system humidity Figure 1, and control in Figure 2 is programmed to operate the system. If the soil is dry more than 800 then the soil sensor will send signal to Arduino to turn on using mobile phone the pump to irrigate and if it is wet less than 400, the pump will be turned off.

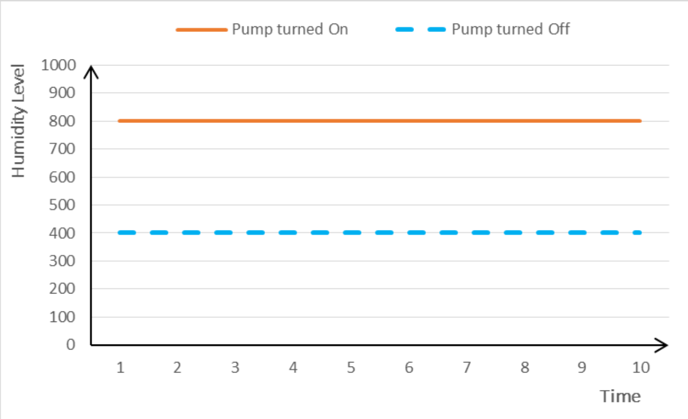

Figure 1, Thresholds of the system humidity

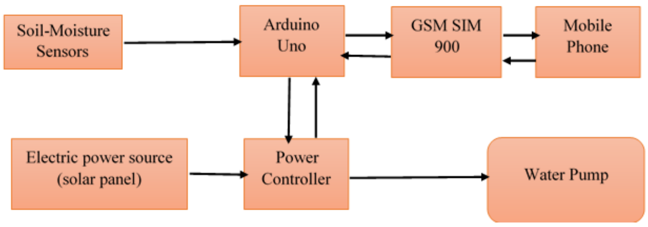

Figure 2, Flow chart of smart irrigation system

II. COMPONENTS

A. Soil Moisture Sensor

Shown in Figure 3, soil moisture sensors usually refer to sensors that estimate volumetric water content. This device used to convert the physical parameters in to an electric signals. The function of this sensor is to sense the content of the moisture in the soil.

Figure 3, Soil Moisture Sensor

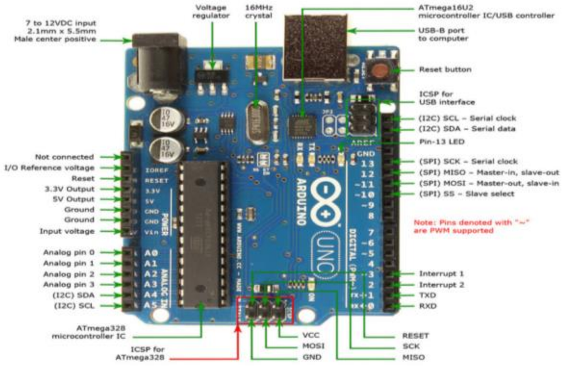

Figure 4, Arduino Uno

B.Arduino Uno

The Arduino Uno shown in Figure 4, is an open source electronic platform with a microcontroller board based on ATmeage328, and integrated development environment is provided by Arduino project based on a programming language called processing. It supports low-power consumption, huge documentation, large libraries and highly portable.

C. GSM SIM 900 shield for Arduino

GSM shown in Figure 5, is stand for Global System for Mobile Communication. The research standards for GSM is 802.11. GSM Sim 900 is a digital transmitting and receiving device used for open air in cellular mobiles. In this project it used to make a communication between user (farmer) and the system for long distance to control the

Figure 5, GSM SIM 900 shield for Arduino

D. Inter – Integrated Circuit (IIC or I2C)

Figure 6, is a serial computer bus. It is a small piece used to connect lower-speed peripheral ICs to processors and microcontrollers. In this project it has been used to connect LCD to Arduino.

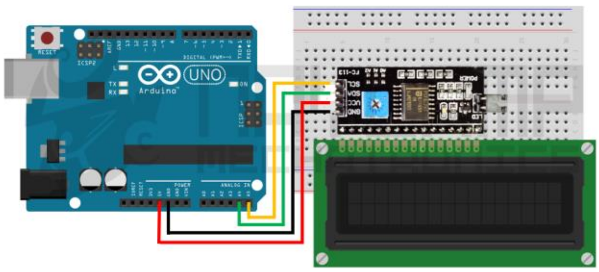

Figure 6, I2C connected to LCD and Arduino.

E. Solar Panel

Solar power panel as shown in figure 7, is a device used to convert the sunlight and temperature directly to electrical power. In this project the PV panel used is (185*250*15 mm) size with (5 W), Used MPPT controller to generate the maximum power [21].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Figure 7, Solar Panel (5W) (185 x 200x15 mm).

F. Dc Water Pump

Figure 8 is a DC water pump used to pump the water from one point to another. With (DC 12V) (4.2 W) (Qmax: 204L/H).

Figure 8, 12V DC Water Pump

G. Relay

It is an electrically operated device, used in this project to control the operation of the water pump, shown in Figure 9.

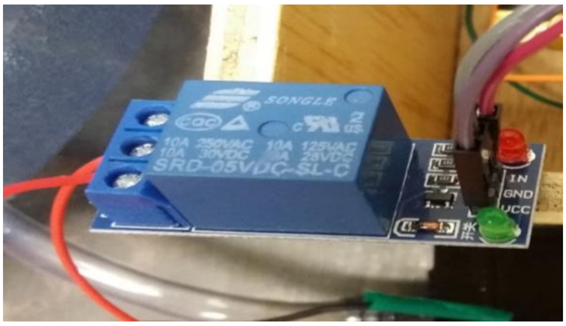

Figure 9, Relay

III. IMPLEMENTATIONS AND RESULTS

The smart irrigation system has built as shown in figure 10, illustrate the final stage with all components of the project; solar panel, plants area size, electronic board with control, relay switch, smart phone, LCD, water pump, moisture sensors, and GSM for Arduino. The system specification measurement has been considered as shown in table 1. The control algorithm has been programmed in Arduino Uno and was uploaded to microcontroller as shown in Appendix. The developed smart control to enable the irrigation system to automatic start or stop the water pump when the moister content reaches presented thresholds in Figure1.

As can be seen in the appendix, the commands represent the libraries which must be mentioned firstly using the command (include). There are three libraries, one for the SIM900 component, second for the call function, third for the LCD which connected to the I2C. Then some main functions came like the (call GSM call). The next commands which start with (unsigned int) represent the integral values but not the negatives. For example the command (unsigned int attempt = 3) mean that there are three attempts for the system to call the registered user. The followed commands which start with (Boolean) it indicates that there are two statements rather True or False (1 or 0 in binary).

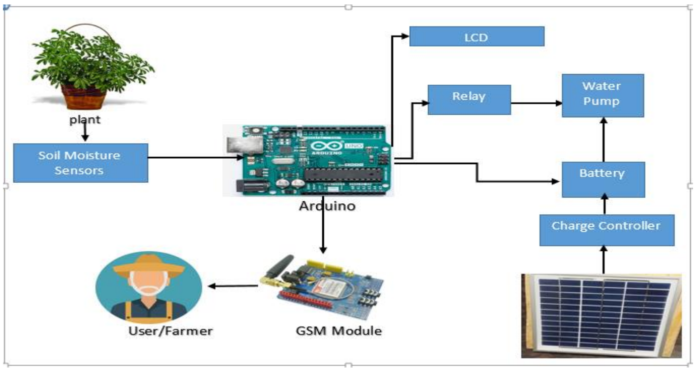

Figure 10, Experimental setup of smart irrigation system

Table 1. Shows the specification of the designed system

No.	Component	Specification
1	Solar Panel	5W
2	Water Pump	12V DC
3	Size of plants land	0.4 m^2
4	Arduino	5 V
5	GSM SIM900 shield for Arduino	9V, Require SIM CARD,
6	Soil Moisture Sensors	4 sensors
7.	LCD	I2C
* The system could irrigate land area of 2 m^2		

Other part of the program shows the statement of the auto-irrigation process, "if" function has been used. Here there are 6 cases between OR Gate and each case depend on two values from two different sensors, so if there is one of the cases is achieved (true), the "if"' function will be satisfied, so the irrigation process is True mean the irrigation is ON.

The main component in this system is the Arduino, which consider as the brain of the system. It provide the logical functions of the system. Its programmed to implement two cases, automatic irrigation and semi-manually controlled by the user through its mobile phone. The Arduino require 5 volt gained from the solar panel. The four soil moisture sensors have a moisture values from 0 to 1023, indicate extremely wet, extremely dry respectively. They provide four different values of moisture content of the soil for the same land area but at different positions. Then send these data to the Arduino which will compare these values with the threshold values determined by the user at the programming stage. The threshold set as follow: If two sensors provide moisture more than 800, pump turned ON, when all moister sensors reach value less than 400, the pump will turn OFF.

In case when two out of four sensors record humidity data more than 800 (soil is dry), the Arduino should turn ON the system automatically to irrigate the land until the four sensors record humidity values less than 400 (soil is wet). Then all sensors indicate that the soil is wet and the Arduino also indicate that in the LCD screen and will turn the pump OFF.

In case, if the system is controlled by the user. So, when two out of four sensors record humidity data more than 800 (soil is very dry), the Arduino should call the farmer through GSM SIM900, then the farmer should call back the system to turn the water pump ON which will irrigate the land as shown in Figure 11. When the four sensors record humidity values less than 400 (soil is very wet) then the controller will stop the system automatically.

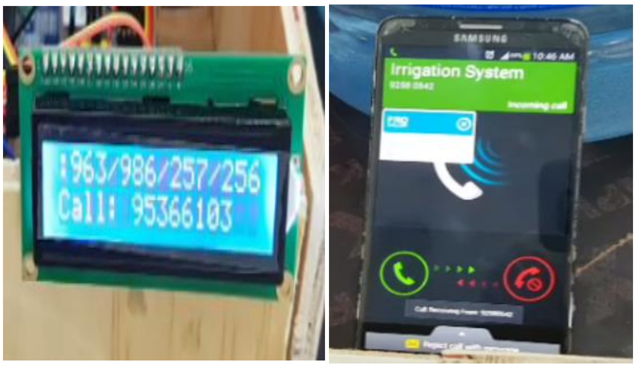

Figure 11, LCD Screen show the sensors readings and the system call the user, and the call appear in the user's phone.

A. Clay soil permeability

The table 2 and the graph below show the recorded moisture data of the clay soil recorded by the four soil moisture sensors every each five seconds. The last column in the table illustrate the average humidity value for the four sensors which has been calculated and represented in figure 12.

Table 2. Humidity value with respect to time recorded by the sensors for Clay soil.

Time(s)	Sensor1	Sensor2	Sensor3	Sensor4	Mean
0	980	983	988	989	985
5	983	983	988	989	985.75
10	813	893	895	915	879
15	800	887	879	912	869.5
20	800	891	813	912	854
25	780	883	813	765	810.25
30	741	859	723	500	705.75
35	745	514	723	401	595.75
40	716	303	679	260	489.5
45	310	335	676	260	395.25
50	261	288	278	249	269

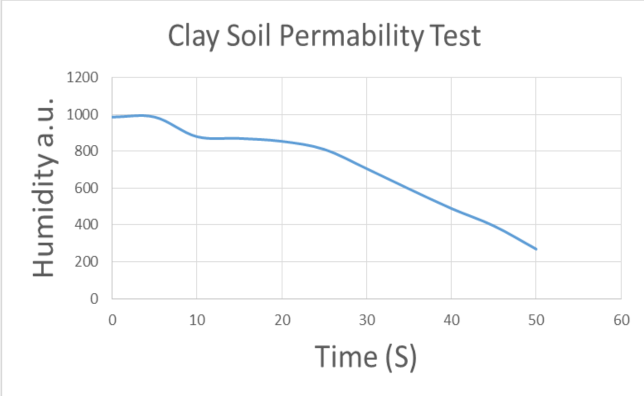

Figure 12, The Average value of humidity recorded by the four sensors in Clay soil.

B. Sand Soil permeability

The table 3 and the graph below show the recorded moisture data of the sand soil recorded by the four soil moisture sensors every each five seconds. The last column in the table illustrate the average humidity value for the four sensors which has been calculated and represented in the Figure 13.

Table 3. Humidity value with respect to time recorded by the sensors for Sand soil.

Time(s)	Sensor1	Sensor2	Sensor3	Sensor4	Mean
0	988	982	989	990	987.25
5	988	982	982	989	985.25
10	988	988	979	986	985.25
15	989	989	950	987	978.75
20	989	988	911	987	968.75
25	989	989	500	987	866.25
30	942	942	500	950	833.5
35	537	931	467	933	717
40	326	827	374	527	513.5
45	323	443	374	444	396
50	324	373	376	398	367.75

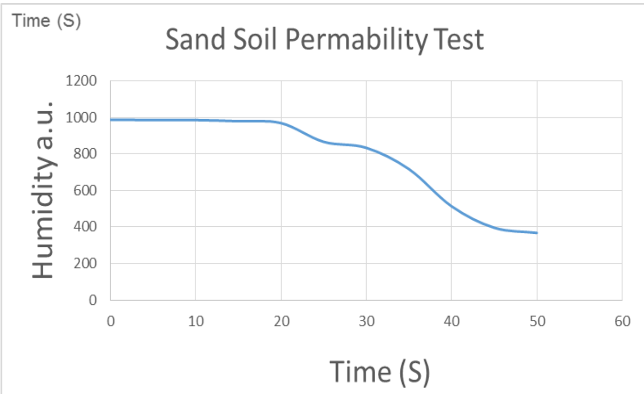

Figure 13, The Average value of humidity recorded by the four sensors in Sand soil.

C. Coarse soil permability

The table 4 below shows the recorded moisture data of the sand soil recorded by the four soil moisture sensors every each five seconds. The last column in the table illustrate the average humidity value for the four sensors which has been calculated and represented in the Figure 14.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Table 4. Humidity value with respect to time recorded by the sensors for coarse soil.

Time(s)	Sensor1	Sensor2	Sensor3	Sensor4	Mean
0	980	989	989	987	986.25
5	980	989	989	987	986.25
10	988	987	989	988	988
15	980	987	988	988	985.75
20	988	989	989	987	988.25
25	988	988	949	987	978
30	980	965	813	987	936.25
35	800	812	612	832	764
40	656	612	401	590	5645
45	440	366	359	428	398.25
50	280	265	277	265	271.75

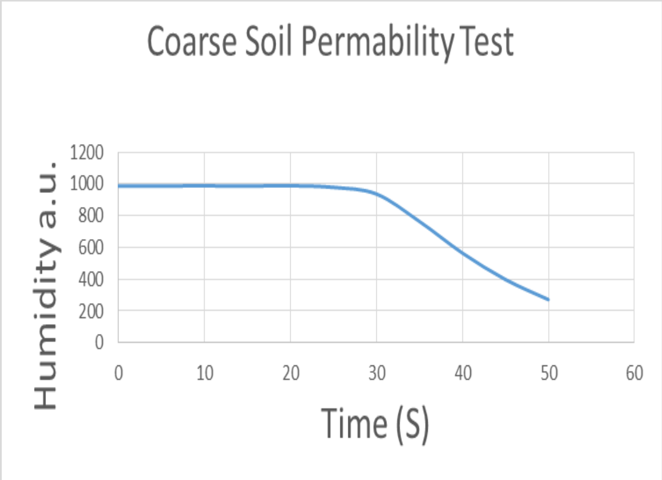

Figure 14, The Average value of humidity recorded by the four sensors in coarse soil.

IV. CONCLUSION

The aim of this project to save time, money and water consumption, by providing smart control irrigation system using friendly solar power. This is an important study in energy and environmental sector.

The irrigation control system was designed, executed, and have achieved the research aims:

- Sense the moisture of the soil through the soil moisture (humidity) sensors.
- Display the humidity data provided by the sensors in LCD screen.
- Arduino is programmed and connected with mobile phone to automatically control irrigation system.
- To contact the user via mankind smart phone when the soil is dry or moist to automatically turn ON and OFF the water pump in order to irrigate the plants land.

This smart control irrigation system is beneficial in places where there are shortage of water, absence of electrical grid and huge farming lands.

A. Future Work

Since the scope of the project was using the water pumping system for irrigation and control this system by microcontroller, there are extra things could be added to improve this project. A water level sensors could be added to irrigate the land with the suitable amount of water. Also, noise sensor could be added to protect the farm field from the animals or even human get closer to the farm by producing a noisy sound as an alarm or buzzer.

Another idea should be added in this project is to add an electronic gate valves in order if there are multi farms need to be irrigated individually at different times controlled by one system shown in Figure 15. Each land area should has its own sensors connected to the controller as an input. Also, the electronic gate valves should be connected to the controller as an output.

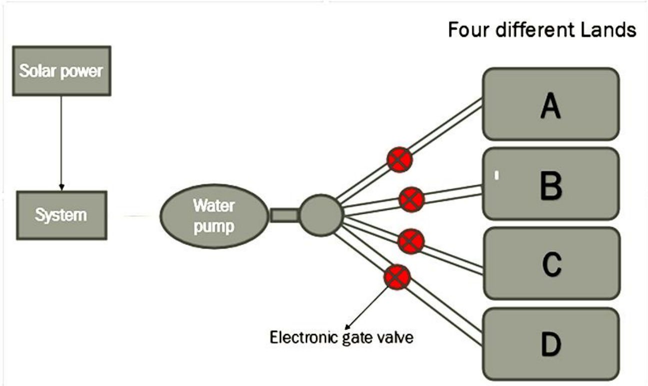

Figure. 15, Irrigation system control with 4 different lands.

Another recommendation should be considered in this project is that the farmer should have the option of controlling the water pump in case of crops is damaged, so no need for irrigation even the soil is dry because there is no plants. Also, improve this system to include the other functions rather than irrigation only. We could add seeding, ploughing and fertilizing functions to be implemented automatically whether using Arduino or PLC as a controller.

Moreover, develop using the GSM Module for remote control. For example, use this technology to remotely switch ON/OFF the AC in home or Open/Close the door of the home and other functions in order to make a smart house.

REFERENCES

[1] Geoffrey, G., Dieu, M. , Pierre, N. and Aimable, T. (2015) Design of Automatic Irrigation System for Small Farmers in Rwanda. Agricultural Sciences, **6**, 291-294. doi: 10.4236/as.2015.63029.

[2] S. Babaa, M. Armstrong, "Novel control strategy for photovoltaic systems based on irradiation" in International Renewable Energy Congress (IREC 2017), 8th IEEE International Conference.

[3] S. Babaa, M. Armstrong, and V. Pickert, "High efficiency standalone photovoltaic system using adaptive switching of an interleaved boost converter," in Power Electronics, Machines and Drives (PEMD 2012), 6th IET International Conference on, pp. 1-7.

[4] EMCON (2006): Feasibility Assessment for the Replacement of Diesel Water Pumps with Solar Water Pumps. Final Report. Windhoek:Ministry of Mines and Energy, Namibia **URL**: [https://sswm.info/node/4539 **]**.

[5] United States Departments of Agriculture (October 2006) "Design of Small Photovoltaic (PV) Solar-Powered Water Pump Systems" Technical Note No. 28 PORTLAND, OREGON.

[6] Marcuzzo, F. and Wendland, E. (2014) The Optimization of Irrigation Networks Using Genetic Algorithms. Journal of Water

Resource and Protection, **6**, 1124-1138. doi: 10.4236/jwarp.2014.612105.

[7] A. N. Arvindan and D. Keerthika, "Experimental investigation of remote control via Android smart phone of arduino-based automated irrigation system using moisture sensor," 2016 3rd International Conference on Electrical Energy Systems (ICEES), Chennai, 2016, pp. 168-175.

[8] Srishti Rawal,"IOT based Smart Irrigation System" International Journal of Computer Applications (0975 – 8887) Volume 159 – No 8, February 2017.

[9] Himasailaja.V[1], Needhidasan.S[2] (2017)''SMART IRRIGATION FOR SMART CITIES A SYSTEM NEEDED FOR INDIA-A REVIEW '' International Journal of Pure and Applied Mathematics. Volume 119 No. 17 2018, 253-262.

[10] Ingram, W, Saeb, L. , Sadr, S. , Hygate, R. and Memon, F. (2018) A Preliminary Investigation of Smart Rural Water Distribution Systems in the Gambia. Journal of Water Resource and Protection, **10**, 395-407. doi: 10.4236/jwarp.2018.104021.

[11] Ogidan, O, Onile, A. and Adegboro, O. (2019) Smart Irrigation System: A Water Management Procedure. Agricultural Sciences, **10**,25-31. doi: 10.4236/as.2019.101003.

[12] S. Babaa, M. Armstrong, "High efficient interleaved boost converter with novel switch adaptive control in photovoltaic systems" in Power Electronics, Drive Systems and technologies (PEDSTC 2017), 8th IEEE International Conference.

[13] Babaa, S. "Novel Switch Adaptive Control to Improve the Efficiency of Boost Converter in Photovoltaic Systems". Journal of Power and Energy Engineering, 5,1-14. 2017. doi: 10.4236/jpee.2017.57001

[14] Thompson, K. and Kadiyala, R. (2013) Protecting Water Quality and Public Health Using a Smart Grid. Computational Water, Energy, and Environmental Engineering, **2**, 73-80. doi: 10.4236/cweee.2013.22B013.

[15] Amala, S. and Aleru, C. (2016) Bacteriological Quality of Swimming Pools Water in Port Harcourt Metropolis. Natural Science, **8**, 79-84. doi: 10.4236/ns.2016.83010.

[16] Babaa, S, Murr, G. , Mohamed, F. and Pamuri, S. "Overview of Boost Converters for Photovoltaic Systems". Journal of Power and Energy Engineering, 6, 16-31. 2018, doi: 10.4236/jpee.2018.64002.

[17] Al-Mashakbeh, H. (2017) The Influence of Lithostratigraphy on the Type and Quality of Stored Water in Mujib Reservoir-Jordan. Journal of Environmental Protection, **8**, 568-590. doi: 10.4236/jep.2017.84038.

[18] Zeng, S. , Li, J. , Schumann, R. and Smart, R. (2013) Effect of pH and Dissolved Silicate on the Formation of Surface Passivation Layers for Reducing Pyrite Oxidation. Computational Water, Energy, and Environmental Engineering, **2**, 50-55. doi: 10.4236/cweee.2013.22B009.

[19] Alkarrami, F, Iqbal, T. , Pope, K. and Rideout, G. (2020) Dynamic Modelling of Submersible Pump Based Solar Water-Pumping System with Three-Phase Induction Motor Using MATLAB. Journal of Power and Energy Engineering, **8**, 20-64. doi: 10.4236/jpee.2020.82002.

[20] S. Khader and A. Daud, "PV-Grid Tie System Energizing Water Pump," Smart Grid and Renewable Energy, Vol. 4 No. 5, 2013, pp. 409-418. doi: 10.4236/sgre.2013.45047.

[21] Babaa, S., Armstrong, M. and Pickert, V. "*Overview of Maximum Power Point Tracking Control Methods for PV Systems*". *Journal of Power and Energy Engineering*, 2, 59-72. 2014. doi: 10.4236/jpee.2014.28006.

APPENDIX

```
// Libraries used in the program
#include "SIM900.h"
#include "call.h"
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x3F); // Initialization of LCD with address in hex
3F
CallGSM call; // Initialization of GSM function named "call"
//Variables declaration
unsigned int attempt = 3; // call attempt 3X
unsigned int bilang = 0; // 10sec waiting time before the next call
unsigned int val0; // Moisture sensor1
unsigned int val1; // Moisture sensor2
unsigned int val2; // Moisture sensor3
unsigned int val3; // Moisture sensor4
boolean called_already = false; //to know if it has called already or is calling
at the moment
boolean wait_before_call = true; //waiting for an answer/reply call from the
owner
boolean irrigate = false; //indicator of irrigation is whether on/off
boolean tito_done = false; //if 3 attempts of calling is done
boolean automatic = true; //switch if manual or automatic turning on of
irrigation/water pump

void setup()
  {
   pinMode(3, OUTPUT); // LED indicator for irrigation
   pinMode(4, INPUT); // switch pin for automatic/manual
   digitalWrite(3, irrigate); // LED indicator is initialized to be off (false)
   Serial.begin(9600); lcd.begin(16,2); //Initialization of Serial
communication of Arduino. LCD initialization 16X2 display
   Serial.println("Initializing"); lcd.print("Initializing");
   if (gsm.begin(9600)) // Initialization of the GSM function/kit
   {
     Serial.println("Status=READY"); //if GSM successfully initialized
     lcd.setCursor(0,1); lcd.print("Status=READY");
   }
   else
   {
    Serial.println("Status=IDLE"); //if GSM did not initialized
    lcd.setCursor(0,1); lcd.print("Status=IDLE");
   }
  }
void loop()
  {
   val0 = analogRead(0); val1 = analogRead(1); val2 = analogRead(2); val3
= analogRead(3); //Analog value of the 4 sensors (1023 max value)
   Serial.println("Moisture:"); Serial.print(val0); Serial.print("\t");
Serial.print(val1); Serial.print("\t"); Serial.print(val2); Serial.print("\t");
Serial.println(val3); //Serial display
   lcd.clear(); //clear previous LCD display
   lcd.setCursor(0,0); lcd.print(":"); lcd.print(val0); lcd.print("/");
lcd.print(val1); lcd.print("/"); lcd.print(val2); lcd.print("/"); lcd.print(val3);
//LCD display
   if(digitalRead(4)) //switch for automatic/manual
    {
     if ((val0>800 && val1>800) || (val0>800 && val2>800) || (val0>800
&& val3>800) || (val1>800 && val2>800) || (val1>800 && val3>800) ||
(val2>800 && val3>800)) //if statement for any of the 2 sensors are both
more than 800 value
       {
        digitalWrite(3, irrigate);
        if(!tito_done)
          {
           while (called_already)
             {
              switch (call.CallStatus())
                {
                 case CALL_NONE: // Nothing is happening
                 Serial.println("Waiting for a call"); lcd.setCursor(0,1);
lcd.print("Waiting a call");
                 val0 = analogRead(0); val1 = analogRead(1); val2 =
analogRead(2); val3 = analogRead(3);
                 Serial.println("Moisture:"); Serial.print(val0);
Serial.print("\t"); Serial.print(val1); Serial.print("\t"); Serial.print(val2);
Serial.print("\t"); Serial.println(val3);
                 lcd.setCursor(0,0); lcd.print(":"); lcd.print(val0);
lcd.print("/"); lcd.print(val1); lcd.print("/"); lcd.print(val2); lcd.print("/");
lcd.print(val3);
                 if (val0<700 && val1<700 && val2<700 && val3<700)
                   {
                    tito_done = true;
                    goto luwas;
                   }

                   if (bilang == 10)
```

```
                  {
                   wait_before_call = true;
                   called_already = false;
                   bilang = 0;
                  }
                 bilang++;
                 break;
                 case CALL_INCOM_VOICE : // Yes! Someone is calling us
                 Serial.println("Welcome");
                 Serial.println("Pump is ON"); lcd.setCursor(0,1);
lcd.print("Pump is now ON");
                 irrigate = true;
                 tito_done = true;
                 goto luwas;
                }
               delay(1000);
              }
              if ((!called_already) && wait_before_call)
               {
                if (attempt)
                 {
                  attempt = attempt - 1;
                  wait_before_call = false;
                  called_already = true;
                  call.Call("95366103");
                  Serial.println("Now calling 95366103"); lcd.setCursor(0,1);
lcd.print("Call: 95366103");
                  delay(10000);
                 }
                else
                 {
                  attempt = 3;
                  Serial.println("Pump is ON"); lcd.setCursor(0,1);
lcd.print("Pump is now ON");
                  irrigate = true;
                  tito_done = true;
                  goto luwas;
                 }
               }
              }
      else
        {
         if(irrigate) { Serial.println("Pump switch ON"); lcd.setCursor(0,1);
lcd.print("Pump switch ON"); }
         else { Serial.println("Pump is CLOSED"); lcd.setCursor(0,1);
lcd.print("Pump is CLOSED"); }
        }
      luwas:;
    }
   else
    {
      if(irrigate) { Serial.println("Pump switch ON"); lcd.setCursor(0,1);
lcd.print("Pump switch ON"); }
      else { Serial.println("Pump is OFF"); lcd.setCursor(0,1);
lcd.print("Pump is CLOSED"); }

      if (val0<400 && val1<400 && val2<400 && val3<400)
       {
        Serial.println("Pump is OFF"); lcd.setCursor(0,1); lcd.print("Pump is
CLOSED");
        irrigate = false;
        tito_done = false;
        bilang = 0;
        called_already = false;
        wait_before_call = true;
       }
      digitalWrite(3, irrigate);
     }
   delay(1000);
  }
 else
  {
    if ((val0>800 && val1>800) || (val0>800 && val2>800) || (val0>800
&& val3>800) || (val1>800 && val2>800) || (val1>800 && val3>800) ||
(val2>800 && val3>800)) //if statement for any of the 2 sensors are both
more than 800 value
     {
      irrigate = true; //signals the irrigation is on
      digitalWrite(3, irrigate); //LED indicator for pump is on
Serial.println("Pump switch ON"); lcd.setCursor(0,1); lcd.print("Pump
switch ON"); //display
     }
    else //if values of any of 2-sensors are not more than 800
     {
      if(irrigate) { Serial.println("Pump switch ON"); lcd.setCursor(0,1);
lcd.print("Pump switch ON"); }
      else { Serial.println("Pump is OFF"); lcd.setCursor(0,1);
lcd.print("Pump is CLOSED"); }

      if (val0<400 && val1<400 && val2<400 && val3<400)
       {
        Serial.println("Pump is OFF"); lcd.setCursor(0,1); lcd.print("Pump
is CLOSED");
        irrigate = false;
       }
      digitalWrite(3, irrigate);

      //reinitialization of variable for manual mode
      tito_done = false;
      bilang = 0;
      called_already = false;
      wait_before_call = true;
     }

    delay(1000);
   }
  }
```