An automation of soil moisture control by Ardunio using soil Moisture sensor

[1]JAGANMOHANRAO.TARRA, [2]K.SUNEEL GOUTHAM

[1,2]ASSITANT PROFESSOR
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI, SRIKAKULAM, AP,INDIA.

***Abstract*: In this paper an automation of soil moisture control by ardunio using soil Moisture sensor. This automatic irrigation system senses the moisture content of the soil and automatically switches the pump when the power is on. A proper Usage of irrigation system is very Necessary because the main reason is the Shortage of land reserved water due to lack of rain, spontaneous use of water as A result a large amount of water goes waste. For this reason, we use this Automatic plant watering and soil moisture monitoring system is very useful in all Climatic conditions. India is the agriculture based on country our most of Peoples are completely depended on the agriculture harvesting.**

Agriculture is a Source of Employment of majority Indians and has great impact on the economy Of the Country. In dry area or In case of lacking rainfall, irrigation becomes difficult. So, it needs to be automated for proper watering a plant and handled remotely by framer. When soil goes dry pump will start watering. The aim of the Implementation is to reduce water use and automatic irrigation can be used for Save time and low Power monitor device. The aim of the implementation this Project was to the Demonstrate that automatic plant irrigation can be used to reduce water use, and Save your time.

***Index Terms*: Ardunio, Sensor**

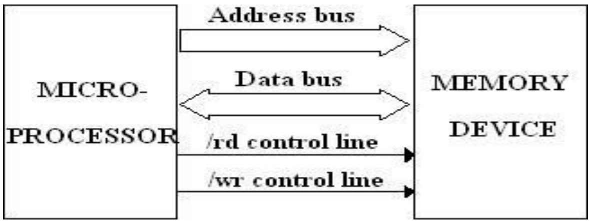
1. *Introduction-*

The first recognizably modern embedded system was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT instrumentation Laboratory. The first mass- produced embedded system was the automatics D-17 guidance Computer for the Minuteman (missile), released in 1961. It was built from transistor logic and had a hard disk for main memory. Since these early applications in the 1960s, embedded systems have come down in price. There has also been an enormous rise in processing power and functionality. For example the first microprocessor was the Intel 4004, which found its way into calculators and other small systems, but required external memory and support chips

In 1978 National Engineering Manufacturers Association released the standard for a programmable microcontroller. The definition was an almost any Computer –based controller. They included single board Computers, numerical controllers and sequential controllers in order to perform event-based instructions. By the mid-1980s, many of he previously external system components had been integrated into the same chip as the processor, resulting in integrated circuits called microcontrollers, and wide spread use of embedded systems became feasible. Presently, a lot of varieties of embedded systems are available at very low costs.

2. *Overview of Embedded System Architecture-*

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the ‘firmware’. The operating system runs above the hardware, and the application software runs above the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system. For small appliances such as remote control units, air conditioners, toys etc., there is no need for an operating system and you can write only the software specific to that application. For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run for a long time you don’t need to reload new software**.**



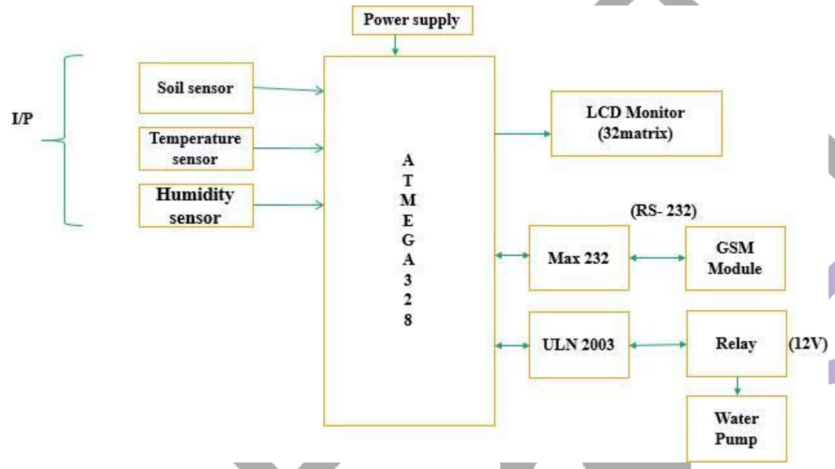
Communication between microprocessor and memory device

3. ANALYSIS-

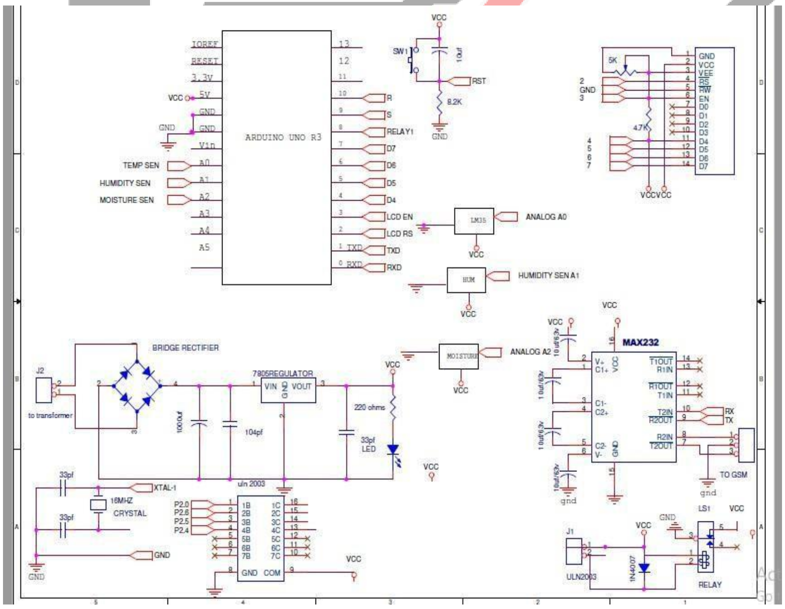
The Agriculture is the unquestionably the largest livelihood provider in India. With rising population, there is a need for increased agricultural production. In order to support greater production in farms, the requirement of the amount of fresh water used in irrigation also rises. Currently, agriculture accounts 83% of the total water consumption in India . Unplanned use of water inadvertently results in wastage of water. This suggests that there is an urgent need to develop systems that prevent water wastage without imposing pressure on farmers.

Over the past 15 years, farmers started using computers and software systems to organize their financial data and keep track of their transactions with third parties and also monitor their crops more effectively. In the Internet era, where information plays a key role in people's lives, agriculture is rapidly becoming a very data intensive industry where farmers need to collect and evaluate a huge amount of information from a diverse number of devices (eg sensors, faming machinery etc.) in order to become more efficient in production and communicating appropriate information.

With the advent of open source Arduino boards along with cheap moisture sensors, it is viable to create devices that can monitor the soil moisture content and accordingly irrigating the fields or the landscape as a when needed. The proposed system makes use of microcontroller ATMEGA328P on Arduino Uno platform and IOT which enable farmers to remotely monitor the status of sprinklers installed on the farm by knowing the sensor values thereby, making the farmers' work much easier as they can concentrate on other farm activities..



4.BLOCK DIAGRAM-

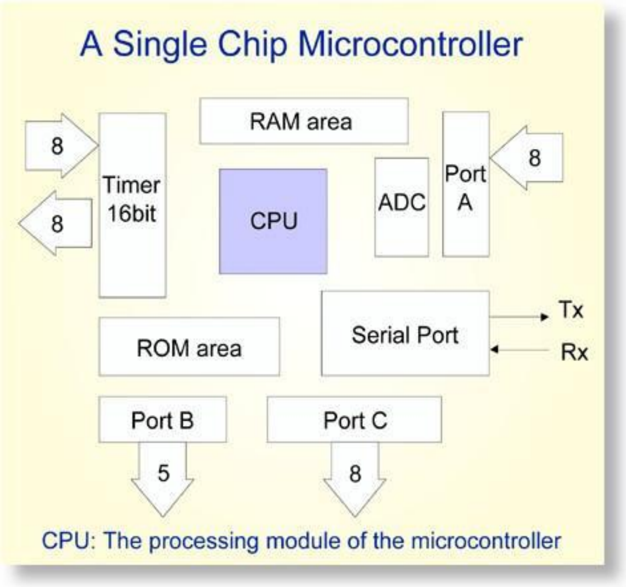


5. Description of the Circuit-

In the circuit description we will be discussing about how the interfacings of the modules are done and connections established between the microcontrollers. The heart of the circuit is the microcontroller and the program in it will be the key for functioning. Some of the important components used are Relays, LED’s, Max 232, Motor and LCD.

Now coming to the description of individual modules and their importance. Firstly, there are two modes in the system one is Command through SMS automated mode and other is Manual mode. Initially the system is in auto mode by default are activated which takes action automatically and intimates the user through SMS. We can set Manual mode using the keypad connected to port2 which acts as input for key to change into other mode. In order to establish communication between GSM and microcontro

6. MICRO CONTROLLERS-

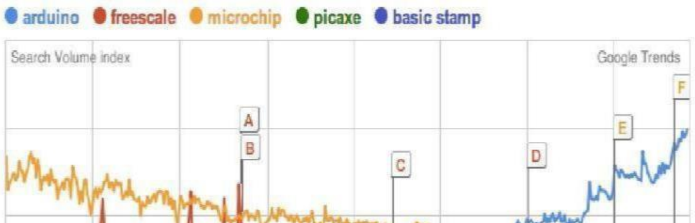

A Single Chip Microcontroller
RAM area
8
Timer 16bit
8
CPU
ADC
Port A
8
Tx
Rx
ROM area
Serial Port
Port B
Port C
5
8
CPU: The processing module of the microcontroller

Integration of Functionality-
Microcontrollers sometimes are called single-chip computers because they have on-chip memory and I/O circuitry and other circuitries that enable them to function as small standalone computers without other supporting circuitry.
Field Programmability, flexibility-
Microcontrollers often use EEPROM or EPROM as their storage device to allow field programmability so they are flexible to use. Once the program is tested to be correct then large quantities of microcontrollers can be programmed to be used in embedded systems.
Easy to Use
Assembly language is often used in microcontrollers and since they usually follow RISC Architecture, the instruction set is small. The development package of microcontrollers Often.
7.ARDUINO-

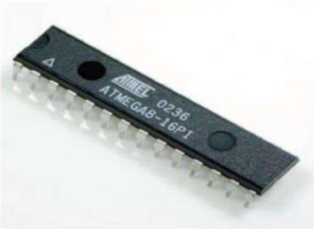
Arduino is an open source electronics prototyping platform based on flexible, easy-to-use hardware and software. It’s intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. It’s an open-source physical computing platform based on a microcontroller board, and a development environment for writing software for the board. The name “Arduino” is reserved by the original makers. However, clone Arduino designs often have the letters “duino” on the end of their name, for example, Fredonia or DFRduino. The software for programming your Arduino is easy to use and also freely available for Windows, Mac, and LINUX computers at no cost.

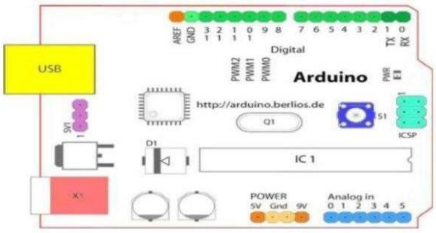
8.Competitors-

Before Arduino, the largest players in the design/hobbyist market segment were the PIC microcontroller family (made by Microchip) and the BASIC Stamp (made by Parallax). Since the introduction of the Arduino, other large companies have tried to enter the hobbyist market, including Texas Instruments, and even Microsoft. However, the open-sourced tools of the Arduino and the size of its community are large barriers for new platforms to overcome.



9.HARDWARE: ARDUINO Board Layout-



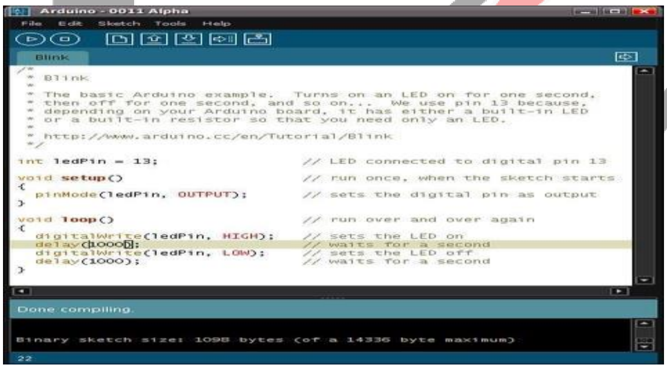


10. SOFTWARE MODULE-

The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command. Although building on command-line is possible if required with some third-party tools such as Ino

The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need define two functions to make a runnable program:

setup () – a function run once at the start of a program that can initialize settings
loop () – a function called repeatedly until the board powers off



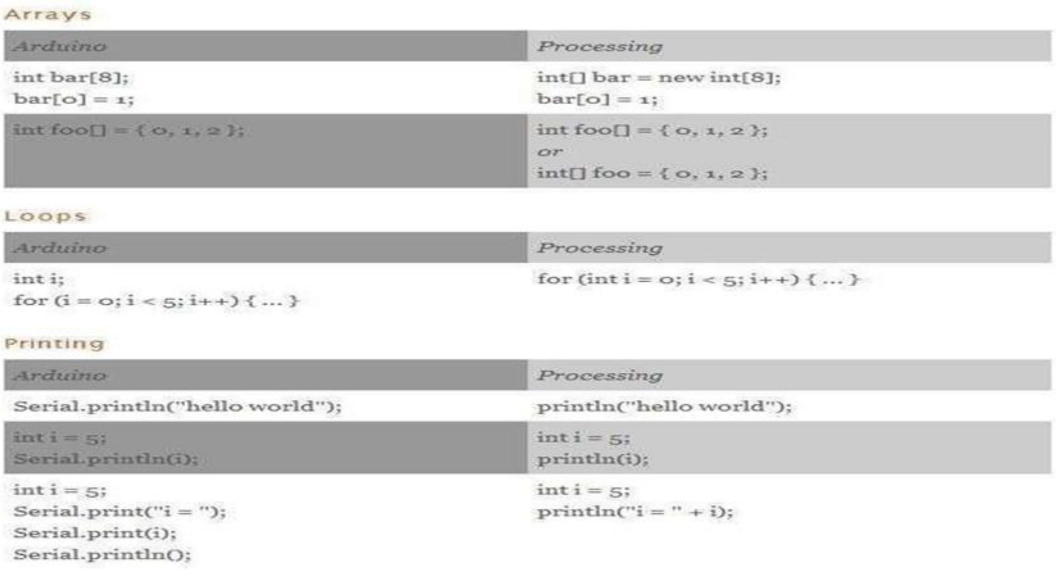
```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

```
Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)
22
```

```
digital Write (LED_PIN, HIGH); // turn on the LED

delay (1000);                  // wait one second (1000 milliseconds)

digital Write (LED_PIN, LOW); // turn off the LED

delay (1000);                  // wait one second

}

#define LED_PIN 13
```

```
void setup () {

pin Mode (LED_PIN, OUTPUT);                    // enable pin 13 for digital output

}

void loop () {
```

- void
- Boolean
- char (0 – 255)
- byte - 8-bit data (0 – 255)
- int - 16-bit data (32,767 - -32,768)
- long – 32-bit data (2,147,483,647 to -2,147,483,648)
- float



Arrays

Arduino	Processing

int bar[8];
bar[0] = 1;
	int[] bar = new int[8];
	bar[0] = 1;

int foo[] = { 0, 1, 2 };	int foo[] = { 0, 1, 2 };
	or
	int[] foo = { 0, 1, 2 };

Loops

Arduino	Processing

int i;
for (i = 0; i < 5; i++) { ... }
	for (int i = 0; i < 5; i++) { ... }

Printing

Arduino	Processing
Serial.println("hello world");	println("hello world");

int i = 5;
Serial.println(i);
	int i = 5;
	println(i);

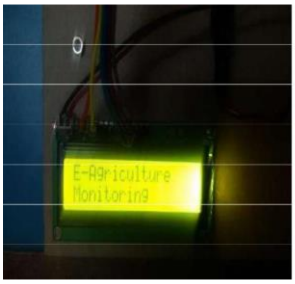
int i = 5;
Serial.print("i = ");
Serial.print(i);
Serial.println();
	int i = 5;
	println("i = " + i);

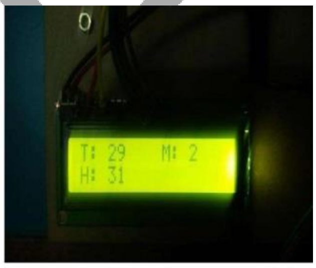
*11.**SOURCE CODE-***

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);//rs,en,data pins d4 -d7 unsigned int TEMP; unsigned int MOISURE; unsigned int HUM;
int val;
int TEMP1;
const int buzzer=6; const int motor=7; int led=4;
int temp=0,i=0; char str[30];
const          int    S1E=3;    L1const          int       S2E=4;L2constint S3E=5;L3
int a=0; int b=0
int c=0;              int d=0;
int ddd=0;///water int dt=0;///temp int dh=0;///hum
int dm=0;///moisure int d1=0;///WL1
int d2=0;///WL2
int S11E=0; int S12E=0; int S13E=0;
int S1Ealert=1; // variable for reading the pushbutton status int S2Ealert=1; // variable for reading the pushbutton status int
S3Ealert=1; // variable for reading the pushbutton status
void setup()
lcd.begin(16,2);
Serial.begin(9600);
pinMode(buzzer, OUTPUT);
     pinMode(S1E, INPUT);
    pinMode(S2E, INPUT);
   pinMode(S3E, INPUT);
   digitalWrite(buzzer, LOW);
  digitalWrite(motor, LOW);
lcd.clear(); lcd.setCursor(0,0);lcd.print("IOT Based"); lcd.setCursor(0,1);lcd.print("Smart");delay(5000); lcd.clear();
lcd.setCursor(0,0);lcd.print("E-Agriculture ");

pinMode( motor,OUTPUT);
lcd.setCursor(0,0);lcd.print("Arduino uno r3"); lcd.setCursor(0,1);lcd.print("GPRS, SIM800L ");delay(5000);
lcd.clear();gsm_init();lcd.clear();
lcd.clear()
lcd.setCursor(0,0); lcd.
print("OPEN SMS APP "); lcd.setCursor(0,1); lcd.print("PRESS THE BUTTON");
digitalWrite(buzzer, LOW); digitalWrite(motor, LOW); }
void serialEvent()

{

while(Serial.available())

{
//#S.motoron*

//#S.motoroff*
if(Serial.find("#S."))

{
digitalWrite(led, HIGH); delay(1000); digitalWrite(led, LOW); while (Serial.available()) {
char inChar=Serial.read(); str[i++]=inChar; if(inChar=='*')

{
temp=1; return;

}

}

}

void loop()

{
digitalWrite(buzzer, LOW); if(temp==1)

{
check(); temp=0; i=0;


delay(1000);


}

lcd.clear();
```

```
TEMP = analogRead(0);float T=( TEMP/1024.0)*5000; TEMP1= T/10;
lcd.setCursor(0,0);lcd.print("T:");lcd.setCursor(3,0);lcd.print(TEMP1);delay(200);if(TEMP1

<50){dt=0;delay(100);} if(TEMP1>50){

dt=dt+1; if(dt==2)

{

lcd.clear();send_gprs();delay(500);

delay(200);digitalWrite(buzzer,
          HIGH);delay(200);digitalWrite(buzzer
LOW);delay(200);digitalWrite(buzzer, HIGH); delay(200);digitalWrite(buzzer, LOW);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");lcd.setCursor(0,1);lcd.print("TEMP ALERT");
Serial.print("AT\r\n");delay(2000);Serial.print("AT\r\n");delay(2000);
Serial.println("AT+CMGF=1");delay(400);Serial.println("AT+CMGS=\"9032504177\"");de
l                ay(400);
//                  use your 10 digit cell no. here lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer,                          HIGH);delay(200);digitalWrite(buzzer, HIGH);
delay(200);digitalWrite(buzzer, LOW); Serial.println("Over Temperature\n");delay(100);
Serial.print("Temp=");delay(100);Serial.print(TEMP1);delay(500);Serial.write(26);delay(50 0);
Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+C M GF=1");delay(1000);}}
MOISURE = analogRead(1); MOISURE= MOISURE/4;MOISURE=256-MOISURE;
lcd.setCursor(9,0);lcd.print("M:");
lcd.setCursor(12,0);lcd.print(MOISURE);delay(200);if(MOISURE<150){dm=0;delay(100);

}
if{(MOISURE>150)

dm=dm+1; if(dm==2)

{
lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer,
         HIGH);delay(200);digitalWrite(buzzer
       HIGH);delay(200);digitalWrite(buzzer, LOW);
delay(200);digitalWrite(buzzer,
        HIGH);delay(200);digitalWrite(buzzer
,                LOW);delay(200);digitalWrite(buzzer, HIGH); delay(200);digitalWrite(buzzer,
LOW);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");lcd.setCursor(0,1);lcd.print("MOISURE ALERT");



Serial.print("AT\r\n");delay(2000);Serial.print("AT\r\n");delay(2000);
Serial.println("AT+CMGF=1");delay(400);Serial.println("AT+CMGS=\"9032504177\"");de l ay(400); // use your 10 digit cell
no. here
Serial.println("MOISURE ALERT\n");delay(100);Serial.write(26);delay(100);
Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+C M GF=1");delay(1000);}}
HUM= analogRead(2);HUM =HUM/2;
lcd.setCursor(0,1);lcd.print("H:");lcd.setCursor(3,1);lcd.print(HUM);delay(2500);if(HUM<3
0){dh=0;delay(100);}
if(HUM>30)

{

dh=dh+1; if(dh==2)

{

lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer,

        HIGH);delay(200);digitalWrite(buzzer

,delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer

LOW);delay(200);digitalWrite(buzzer, HIGH); delay(200);digitalWrite(buzzer, LOW);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");lcd.setCursor(0,1);lcd.print("HUMIDITY ALERT");
Serial.print("AT\r\n");delay(2000);Serial.print("AT\r\n");delay(2000);


Serial.println("AT+CMGF=1");delay(400);Serial.println("AT+CMGS=\"9032504177\"");de l ay(400);
Serial.println("HUMIDITY ALERT\n");delay(100);Serial.write(26);delay(100);

Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+C

M                GF=1");delay(1000);}}

}


void send_gprs()


{
HIGH);delay(200);digitalWrite(buzzer,LOW);

lcd.clear();lcd.print("GPRS SENDING"); boolean test7_flag=1;

while(test7_flag){Serial.print("AT+HTTPPARA=\"URL\",\"http://rpihealth.com/iot_agri_fe il d/put_data.php");
Serial.print("?temp=");Serial.print(TEMP1);
if((HUM >= 0) & (HUM <= 9))
{Serial.print("&hum=00");Serial.print(HUM);} if((HUM >= 10) & (HUM <= 99))

{Serial.print("&hum=0");Serial.print(HUM);}

if((HUM >= 99) & (HUM <= 255))
{Serial.print("&hum=");Serial.print(HUM);}

if((MOISURE >= 0) & (MOISURE <= 9)) {Serial.print("&mos=00");Serial.print(MOISURE);} if((MOISURE >= 10) &
(MOISURE <= 99))

{Serial.print("&mos=0");Serial.print(MOISURE);} if((MOISURE >= 99) & (MOISURE <= 255))
Serial.print("&mos=");Serial.print(MOISURE);} Serial.print("\"");Serial.print("\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test7_flag=0;}delay(1000);} lcd.clear();lcd.print("SENT
COMPLETED");delay(10000);lcd.clear();

lcd.clear();lcd.print("ACTION");

boolean                    test8_flag=1;while(test8_flag){Serial.print("AT+HTTPACTION=0\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test8_flag=0;}delay(1000);}

lcd.clear();lcd.print("SEND OK");delay(2000); delay(2000);delay(2000);

}void gsm_init()

{lcd.clear();lcd.


at_flag=1;while(at_flag){Serial.println("AT");while(Serial.available()>0){if(Serial.find("OK ")) at_flag=0;}delay(1000);}
lcd.clear();lcd.print("GSM
CONNECTED");delay(1000);lcd.clear();
lcd.print("ECHO");
boolean echo_flag=1; while(echo_flag)
{Serial.println("ATE0");
while(Serial.available()>0){if(Serial.find("OK"))echo_flag=0;}delay(1000);}
lcd.clear(); lcd.print("Echo OFF");delay(1000);lcd.clear();


lcd.print("Finding Network..");

boolean         net_flag=1;while(net_flag){Serial.println("AT+CPIN?");          while(Serial.available()>0){if(Serial.find("+CPIN:
READY"))net_flag=0;}delay(1000);} lcd.clear();lcd.print("Network Found..");
lcd.setCursor(0,1);lcd.print("GSM NETWORK OK");delay(2000);lcd.clear();
/            lcd.clear();lcd.print("TEST MESS");

boolean                                                    test_flag=1;while(test_flag){Serial.println("AT+CMGF=1");

while(Serial.available()>0){if(Serial.find("OK"))test_flag=0;}delay(1000);}

lcd.clear();lcd.print("TEST


lcd.clear();lcd.print("AT+CGATT");

boolean test1_flag=1;while(test1_flag){Serial.println("AT+CGATT=1");
while(Serial.available()>0){if(Serial.find("OK"))test1_flag=0;}delay(1000);} lcd.clear();lcd.print("AT+CGATT=1");delay(1000);
lcd.clear();lcd.print("GPRS START");
boolean
test2_flag=1;while(test2_flag){Serial.print("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"\r\n") ;
while(Serial.available()>0){if(Serial.find("OK"))test2_flag=0;}delay(1000);} lcd.clear();lcd.print("GPRS
START1");delay(1000);
/ lcd.clear();lcd.print("GPRS MAIN");
boolean test4_flag=1;while(test4_flag){Serial.print("AT+SAPBR=1,1\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test4_flag=0;}delay(1000);} lcd.clear();lcd.print("GPRS FIND");delay(1000);
lcd.clear();lcd.print("HTTP STARTS");
boolean test5_flag=1;while(test5_flag){Serial.print("AT+HTTPINIT\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test5_flag=0;}delay(1000);} lcd.clear();lcd.print("HTTP
STARTS1");delay(1000);
lcd.clear();lcd.print("HTTP STARTS");
boolean
test6_flag=1;while(test6_flag){Serial.print("AT+HTTPPARA=\"CID\",1\r\n");
```

12.RESULT





***References*:**

[1] Dr. Narayan G. Hegde, “Water Scarcity and Security in India”, BAIF Development Research Foundation, Pune.

[2] Marvin T. Batte, “Changing computer use in agriculture: evidence from Ohio”, Computers and Electronics in Agriculture, Elsevier science publishers, vol. 47, 1–13, 2005

[3] Csótó, Magyar, “Information flow in agriculture – through new channels for improved effectiveness”, Journal of Agricultural Informatics 1 (2), 25–34, 2010

[4] Jin Shen, Song Jingling, Han Qiuyan and Yang Yan, “A Remote Measurement and Control System for Greenhouse Based on GSM-SMS”, Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference

[5] Indu Gautam and S.R.N Reddy, “Innovative GSM based Remote Controlled Embedded System for Irrigation”, International Journal of Computer Applications Vol. 47 – No.13,
June 2012

[6] R.Suresh, S.Gopinath, K.Govindaraju,T.Devika,