**ABSTRACT**

The greenhouse based modern agriculture industries are the recent requirement in every part of the world. In this technology, the humidity and temperature of plants are precisely controlled. Due to the variable atmospheric circumstances these conditions sometimes may vary from place to place in large farmhouse, which makes very difficult to maintain the uniformity at all the places in the farmhouse manually. It is observed that for the first time an android phone-control the Irrigation system, which could give the facilities of maintaining uniform environmental conditions are proposed. The Android Software Development Kit provides the tools and Application Programmable Interface necessary to begin developing applications on the Android platform using the Java programming language. Mobile phones have almost become an integral part of human life serving multiple needs of humans. This application makes use of the GPRS [General Packet Radio Service] feature of mobile phone as a solution for irrigation control system. GSM (Global System for Mobile Communication) is used to inform the user about the exact field condition. The information is passed onto the user request in the form of SMS.

**TABLE OF CONTENTS**

**TITLE PAGE NO**

Introduction

Objective

Existing System

Proposed System

Hardware Requirements

Software Requirements

Individual Contribution

Scope

Controlling Motor and Solenoid Valves

Interfacing Soil Moisture Sensor

Interfacing Water level Sensor

SIM900A GSM Module interfacing

Android Application and UI

Conclusion

References

**1.INTRODUCTION**

“**Smart Farm Automated Irrigation Control System Using Smartphone**” implements the emerging applications of GSM, Wireless sensor networks and Android together to provide an Irrigation system which can completely helps the farmer to remotely monitor, control and supply water to the crops by sending messages in the Irrigation field with the help of Smartphone interface. This is a cost efficient and effortless system which reduces the wastage of water and electricity. This system is scalable and even supports the larger number of control points.

Drip irrigation is artificial method of supplying water to the roots of the plant. It is also called micro irrigation. In past few years there is a rapid growth in this system. The user communicates with the centralized unit through SMS. The centralized unit communicates with the system through SMS which will be received by the GSM with the help of the SIM card. The GSM sends this data to ARM7which is also continuously receives the data from sensors in some form of codes. After processing, this data is displayed on the LCD. Thus in short whenever the system receives the activation command from the subscriber it checks all the field conditions and gives a detailed feedback to the user and waits for another activation command to start the motor. The motor is controlled by a simple manipulation in the internal structure of the starter. The starter coil is indirectly activated by means of a transistorized relay circuit. When the motor is started, a constant monitoring on soil moisture and water level is done & once the soil moisture is reached to sufficient level the motor is automatically turned off & a massage is send to subscriber that the motor is turned off. The water level indicator indicates three levels low, medium, high and also empty tank.

1. **EMBEDDED C**

**Embedded C** is a set of language extensions for the [C Programming language](https://en.wikipedia.org/wiki/C_Language) by the [C Standards committee](https://en.wikipedia.org/wiki/SC22) to address commonality issues that exist between C extensions for different [embedded systems](https://en.wikipedia.org/wiki/Embedded_systems). Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as [fixed-point arithmetic](https://en.wikipedia.org/wiki/Fixed-point_arithmetic), multiple distinct [memory banks](https://en.wikipedia.org/wiki/Memory_bank), and basic [I/O](https://en.wikipedia.org/wiki/Input/output) operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

1. **ARDUINO**

**Arduino** is common term for a software company, project, and user community that designs and manufactures computer [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware), [open-source software](https://en.wikipedia.org/wiki/Open-source_software), and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller)-based kits for building digital devices and interactive objects that can sense and control physical devices.

The project is based on microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog [I/O](https://en.wikipedia.org/wiki/I/O) pins that can interface to various expansion boards (termed *shields*) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus ([USB](https://en.wikipedia.org/wiki/USB)) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment)(IDE) based on a programming language named [*Processing*](https://en.wikipedia.org/wiki/Processing_(programming_language)), which also supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B).

The first Arduino was introduced in 2005, aiming to provide a low cost, easy way for novices and professionals to create devices that interact with their environment using [sensors](https://en.wikipedia.org/wiki/Sensor) and [actuators](https://en.wikipedia.org/wiki/Actuator). Common examples of such devices intended for beginner hobbyists include simple [robots](https://en.wikipedia.org/wiki/Robot), [thermostats](https://en.wikipedia.org/wiki/Thermostat), and motion detectors.

Arduino boards are available commercially in preassembled form, or as [do-it-yourself](https://en.wikipedia.org/wiki/Do-it-yourself) kits. The hardware design specifications are openly available, allowing the Arduino boards to be produced by anyone. [Adafruit Industries](https://en.wikipedia.org/wiki/Adafruit_Industries) estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

1. **JAVA**

**Java** is a general-purpose [computer programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of[computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). As of 2016, Java is one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity), particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (which has since been [acquired by Oracle Corporation](https://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.

The original and [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) Java [compilers](https://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](https://en.wikipedia.org/wiki/Library_(computing)) were originally released by Sun under proprietary licences. As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun relicensed most of its Java technologies under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License). Others have also developed alternative implementations of these Sun technologies, such as the [GNU Compiler for Java](https://en.wikipedia.org/wiki/GNU_Compiler_for_Java) (bytecode compiler), [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath) (standard libraries), and [IcedTea](https://en.wikipedia.org/wiki/IcedTea)-Web (browser plugin for applets).

The latest version is Java 8, which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.

1. **ANDROID**

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android and took over its development work (as well as its development team).The following Fig.2 shows the major components of the Android operating system. The main advantage of adopting Android is that it offers a unified approach to application development and their applications should be able to run on numerous different devices, as long as the devices are powered using Android

**Applications:**

These are applications written in Java. Some of basic applications include calendar, email client, SMS program, maps, making phone calls, accessing the Web browser, accessing your contacts list and others.

**Application Framework:**

This is the skeleton or framework which all android developers have to follow. The developers can access all framework APIs an manage phone’s basic functions like resource allocation, switching between processes or programs, telephone applications, and keeping track of the phone’s physical location.

**Libraries:**

This layer consists of Android libraries written in C, C++, and used by various systems. These libraries tell the device how to handle different kinds of data and are exposed to Android developers via Android Application framework. Some of these libraries includes media, graphics, 3D, SQL, web browser library etc. The Android runtime layer which includes set of core java libraries and DVM (Dalvik Virtual Machine) is also located in same layer.

**Runtime Android:**

This layer includes set of base libraries that are required for java libraries. Every Android application gets its own instance of DVM. Dalvik has been written so that a device can run multiple VMs efficiently and it executes files in executable (.Dex) optimized for minimum memory.

**Kernel Linux:**

This layer includes Android’s memory management programs, security settings, power management software and several drivers for hardware, file system access, networking and inter-process-communication. The kernel also acts as an abstraction layer between hardware and the rest of the software stack.

1. **GSM MODULE**

At present the GSM module is used for Remote Control activities such as Gate Control, Temperature Control etc. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB) for computer. The MODEM is the soul of such modules. They generate, transmit or decode data from a cellular network, for establishing communication between the cellular network and the computer. These are manufactured for specific cellular network (GSM/UMTS/CDMA) or specific cellular data standard (GSM/UMTS/GPRS/EDGE/HSDPA) or technology (GPS/SIM). They use serial communication to interface with the user and need Hayes compatible AT (Attention) commands for communication with the computer (any microprocessor or microcontroller system).

An RS-232 port was once a standard feature of a personal computer for connections to modems, printers, mice, data storage, un-interruptible power supplies, and other peripheral devices. However, the limited transmission speed, relatively large voltage swing, and large standard connectors motivated development of the universal serial bus which has displaced RS-232 from most of its peripheral interface roles. Many modern personal computers have no RS-232 ports and must use an external converter to connect to older peripherals. Some RS-232 devices are still found especially in industrial machines or scientific instruments.

**OBJECTIVIES**

* Designing an embedded system which helps the farmer to supply water and control the gate valves
* To provide a simple and efficient irrigation application for the user to control the system
* To automatically power on or off the system in the low water level
* To keep track of moisture and humidity in the soil
* To acknowledge the user about the status of each control given to the system
* To reduce the Wastage of water and electricity

**EXISTING SYSTEM**

* Motor can be switched on or off remotely
* User have to make a call or send a manual message each time to switch on or off the motor
* No provision for the individual gate valve controlling
* Not possible to monitor the system
* No provision for detecting the moisture and humidity in the soil

**PROPOSED SYSTEM**

* Motor can be switched on or off remotely either manually or automatically
* All the gate valves in the farm can be controlled individually
* No manual messages or calls required
* Water level indicator to detect the water level in the tank or Water well
* Soil sensors in the farm fields to measure the humidity and moisture of the Soil for better irrigation control
* Notification by the system after each and every input given by the user

**HARDWARE REQUIREMENTS**

* Arduino UNO R3 Board
* SIMcom SIM900A GSM Module
* 0.5” Submersible pump
* 0.5” Solenoid Valve
* KG003 Soil Sensor
* ZC09200 Water Level Sensor
* Android Smartphone
* Dual-core processor
* 4 GB of memory
* 150 GB of internal storage
* Network interface card
* Windows 7, Windows 8, or Windows 8.1

**SOFTWARE REQUIREMENTS**

* Windows 7 OS
* Arduino IDE 1.6.7
* Android Studio Canary Build: 2.0
* Android OS 4.3 Jellybean or higher version
* Flow code v6
* Java
* AndroidSDK

**MODULE:**

**ARDUINO PROGRAMMING:**

#include <SoftwareSerial.h>   
char inchar; // Will hold the incoming character from the GSM shield  
const int moisture = 0; // Analog pin 0  
const int water = 2; // Analog pin 2  
int val,value,flag=0;  
const int temperature = 1; // Analog pin 1  
SoftwareSerial SIM900(5, 6);  
int led1 = 9;  //Motor  
int led2 = 10; //Solenoid Valve 1  
int led3 = 11; //Solenoid Valve 2  
int led4 = 12; //Solenoid Valve 3  
int led5 = 13; //Solenoid Valve 4  
void setup()  
{  
  Serial.begin(19200);  
  // set up the digital pins to control  
  pinMode(led1, OUTPUT);  
  pinMode(led2, OUTPUT);  
  pinMode(led3, OUTPUT);  
  pinMode(led4, OUTPUT);  
  pinMode(led5, OUTPUT);  
  digitalWrite(led1, LOW);  
  digitalWrite(led2, LOW);  
  digitalWrite(led3, LOW);  
  digitalWrite(led4, LOW);  
  digitalWrite(led5, LOW);  
  
  // wake up the GSM shield  
  SIM900power();   
  SIM900.begin(19200);  
  delay(20000);  // give time to log on to network.  
  SIM900.print("AT+CMGF=1\r");  // set SMS mode to text  
  delay(100);  
  SIM900.print("AT+CNMI=2,2,0,0,0\r");   
  // blurt out contents of new SMS upon receipt to the GSM shield's serial out  
  delay(100);  
  Serial.println("Ready...");  
}  
  
void SIM900power()  
// software equivalent of pressing the GSM shield "power" button  
{  
  digitalWrite(9, HIGH);  
  delay(1000);  
  digitalWrite(9, LOW);  
  delay(7000);  
}  
  
void loop()   
{  
  //If a character comes in from the cellular module...  
  if(SIM900.available() >=0)  
   
  {  
   value=analogRead(water);  
   delay(100);  
   if (value<=480 && flag==1)  
   {  
                SIM900.print("AT+CMGF=1\r");                                                        // AT command to send SMS message  
                delay(100);  
                SIM900.println("AT + CMGS = \"+919600514508\"");                                     // recipient's mobile number, in international format  
                delay(100);  
                SIM900.println("Water level: 0mm - Empty!");   
                delay(100);  
                SIM900.println((char)26);  
                digitalWrite(led1, LOW);  
                digitalWrite(led2, LOW);  
                digitalWrite(led3, LOW);  
                digitalWrite(led4, LOW);  
                digitalWrite(led5, LOW);  
                flag=0;  
   }  
   if(flag==1)  
   {  
                int soilvalue = analogRead(moisture);  
                SIM900.print("AT+CMGF=1\r");                                                        // AT command to send SMS message  
                delay(100);  
                SIM900.println("AT + CMGS = \"+919600514508\"");                                     // recipient's mobile number, in international format  
                delay(100);  
                    if (soilvalue >= 820)  
                    {  
                    SIM900.println("Soil full of water");  
                    SIM900.println("Humidity Level : ");  
                    SIM900.println(soilvalue);  
                    delay(100);  
                    SIM900.println((char)26);  
                    }  
      
   }  
     
            if(inchar=='w')  
            {  
                value=analogRead(water);  
                delay(100);  
                SIM900.print("AT+CMGF=1\r");                                                        // AT command to send SMS message  
                delay(100);  
                SIM900.println("AT + CMGS = \"+919600514508\"");                                     // recipient's mobile number, in international format  
                delay(100);  
                if (value<=480)  
                {   
                SIM900.println("Water level: 0mm - Empty!");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>480 && value<=530)  
                {   
                SIM900.println("Water level: 0mm to 5mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>530 && value<=615)  
                {   
                SIM900.println("Water level: 5mm to 10mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>615 && value<=660)  
                {   
                SIM900.println("Water level: 10mm to 15mm");   
                delay(100);  
                SIM900.println((char)26);  
                }   
                else if (value>660 && value<=680)  
                {   
                SIM900.println("Water level: 15mm to 20mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>680 && value<=690)  
                {   
                SIM900.println("Water level: 20mm to 25mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>690 && value<=700)  
                {   
                SIM900.println("Water level: 25mm to 30mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>700 && value<=705)  
                {   
                SIM900.println("Water level: 30mm to 35mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else if (value>705)  
                {   
                SIM900.println("Water level: 35mm to 40mm");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                 
            }  
              
              
            if(inchar=='m')  
                  { Serial.println("Success");  
                  int soilvalue = analogRead(moisture);  
                  SIM900.print("AT+CMGF=1\r");                                                        // AT command to send SMS message  
                  delay(100);  
                  SIM900.println("AT + CMGS = \"+919600514508\"");                                     // recipient's mobile number, in international format  
                  delay(100);  
                    if (soilvalue >= 820)  
                    {  
                    SIM900.println("Soil full of water");  
                    SIM900.println("Humidity Level : ");  
                    SIM900.println(soilvalue);  
                    delay(100);  
                    SIM900.println((char)26);  
                    }  
                    else if (soilvalue >= 616  && soilvalue < 820)  
                    {  
                    SIM900.println("High humidity and Soil is Wet");  
                    SIM900.println("Humidity Level : ");  
                    SIM900.println(soilvalue);  
                    delay(100);  
                    SIM900.println((char)26);  
                    }    
                    else if (soilvalue >= 411 && soilvalue < 615)  
                    {  
                    SIM900.println("Moderate humidity");  
                    SIM900.print("Humidity Level : ");  
                    SIM900.println(soilvalue);  
                    delay(100);  
                    SIM900.println((char)26);  
                    }      
                    else if (soilvalue >= 251 && soilvalue < 410)  
                    {  
                    SIM900.println("Low humidity");  
                    SIM900.print("Humidity Level : ");  
                    SIM900.println(soilvalue);  
                    delay(100);  
                    SIM900.println((char)26);  
                    }  
                    else if (soilvalue >= 1 && soilvalue < 250)  
                    {  
                    SIM900.println("Dry soil");  
                    SIM900.print("Humidity Level : ");  
                    SIM900.println(soilvalue);  
                    delay(100);  
                    SIM900.println((char)26);  
                    }  
                    else if (soilvalue = 0)  
                    {  
                    SIM900.println("Please insert the sensor into soil");  
                    SIM900.println((char)26);  
                    }  
                      
                  }  
  
             if(inchar=='t')  
              {  
                val = analogRead(temperature);  
                float mv = ( val/1024.0)\*3300;  
                float cel = mv/10;  
                float farh = (cel\*9)/5 + 32;  
                SIM900.print("AT+CMGF=1\r");                                                        // AT command to send SMS message  
                delay(100);  
                SIM900.println("AT + CMGS = \"+919600514508\"");                                     // recipient's mobile number, in international format  
                delay(100);  
                SIM900.println("TEMPRATURE is ");  
                SIM900.print(cel);  
                SIM900.println("\* celsius");  
                delay(1000);  
                SIM900.println((char)26);  
                 
              }  
  
                     
             if (inchar=='#')  //Valve Control Code  
                  {  
                  delay(10);  
                  inchar=SIM900.read();   
                  value=analogRead(water);  
                delay(100);  
                SIM900.print("AT+CMGF=1\r");                                                        // AT command to send SMS message  
                delay(100);  
                SIM900.println("AT + CMGS = \"+919600514508\"");                                     // recipient's mobile number, in international format  
                delay(100);  
                if (value<=480)  
                {   
                SIM900.println("Water level: 0mm - Empty!");   
                delay(100);  
                SIM900.println((char)26);  
                }  
                else  
                {  
                  if (inchar=='a')  
                    {  
                     delay(10);  
                     inchar=SIM900.read();  
                            if (inchar=='0')  
                            {  
                              digitalWrite(led1, LOW);  
                              flag=0;  
                            }   
                            else if (inchar=='1')  
                            {  
                              digitalWrite(led1, HIGH);  
                              flag=1;  
                            }  
                            delay(10);  
                            inchar=SIM900.read();   
                            if (inchar=='b')  
                            {  
                              inchar=SIM900.read();  
                              if (inchar=='0')  
                              {  
                                digitalWrite(led2, LOW);  
                              }   
                              else if (inchar=='1')  
                              {  
                                digitalWrite(led2, HIGH);  
                              }  
                              delay(10);  
                              inchar=SIM900.read();   
                              if (inchar=='c')  
                              {  
                                inchar=SIM900.read();  
                                if (inchar=='0')  
                                {  
                                  digitalWrite(led3, LOW);  
                                }   
                                else if (inchar=='1')  
                                {  
                                  digitalWrite(led3, HIGH);  
                                }  
                                delay(10);  
                                inchar=SIM900.read();   
                                if (inchar=='d')  
                                {  
                                  delay(10);  
                                  inchar=SIM900.read();  
                                  if (inchar=='0')  
                                  {  
                                    digitalWrite(led4, LOW);  
                                  }   
                                  else if (inchar=='1')  
                                  {  
                                    digitalWrite(led4, HIGH);  
                                  }  
                                  delay(10);  
                                  inchar=SIM900.read();   
                              if (inchar=='e')  
                              {  
                                inchar=SIM900.read();  
                                if (inchar=='0')  
                                {  
                                  digitalWrite(led5, LOW);  
                                }   
                                else if (inchar=='1')  
                                {  
                                  digitalWrite(led5, HIGH);  
                                }  
                                delay(10);  
                                }  
                              }  
                              }  
                              SIM900.println("AT+CMGD=1,4"); // delete all SMS  
                            }  
                          }  
    }  
   }  
  }  
}

**INTERFACING SOIL MOISURE SENSOR**

* Soil moisture sensors measure the volumetric water content indirectly by using some properties of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content
* The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity
* Soil moisture sensor values are calculated as:
  + Dry Soil – 0 to 250
  + Low humidity – 250 to 500
  + More Humidity – 500 to 700
  + Wet Soil - >700

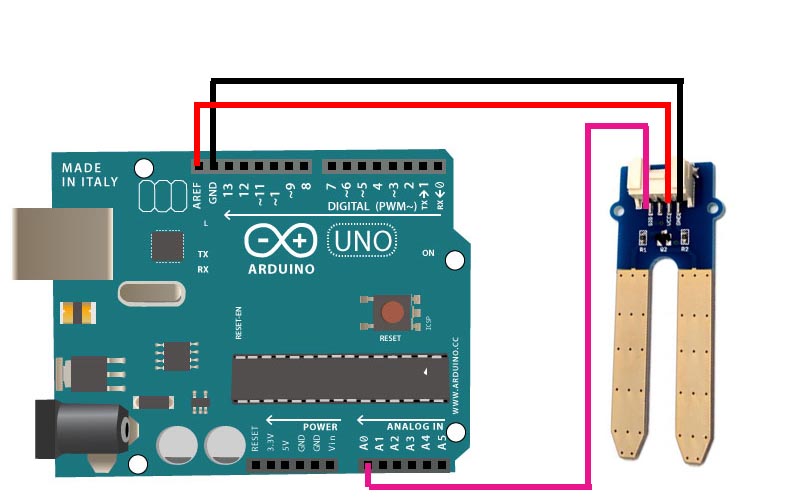


Fig 1. Soil moisture sensor with arduino

**INTERFACING WATERLEVEL SENSOR**

* Water Level sensors detect the level of water inside a container or in its natural form (a river or a lake). The level measurement can be either continuous or point values.
* This module alerts
* The analog values can be converted to height as given below:
  + 0mm - 480
  + 10mm - 615
  + 20mm - 680
  + 30mm - 700
  + 40mm - 710

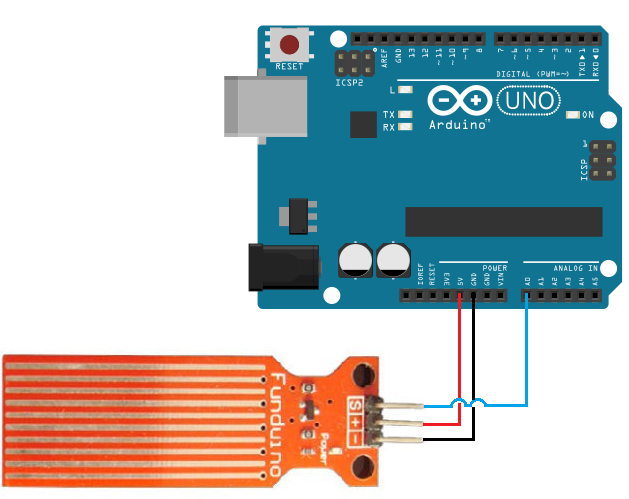


Fig 2. Water level sensor with Arduino

The model consists of a series of tanks arranged one below the other. The volume of the tanks is in descending order. Water flows from the top tank through outlets at the bottom. Three tanks or tropic levels chosen for the model is the optimum number required to analyze the effect of top down and bottom up controls. Each tank has two outlets, outlet A and outlet B. Each outlet has the water flow through it regulated by means of valves. These valves are controlled by floats in the tanks. The system used microcontroller to automate the process of water pumping in an over-head tank storage system and has the ability to detect the level of water in a tank, switch on/off the pump accordingly and display the status on an LCD screen.



**GSM MODULE INTERFACING**

* SIM900A GSM Module is used to send and receive the text commands
* It operates on the 900 MHz frequency band
* This module is responsible for external wireless communication and it receives commands from the user through text message and pass it to the particular module
* It also alerts the user about the water level, soil moisture and power on state, etc.,

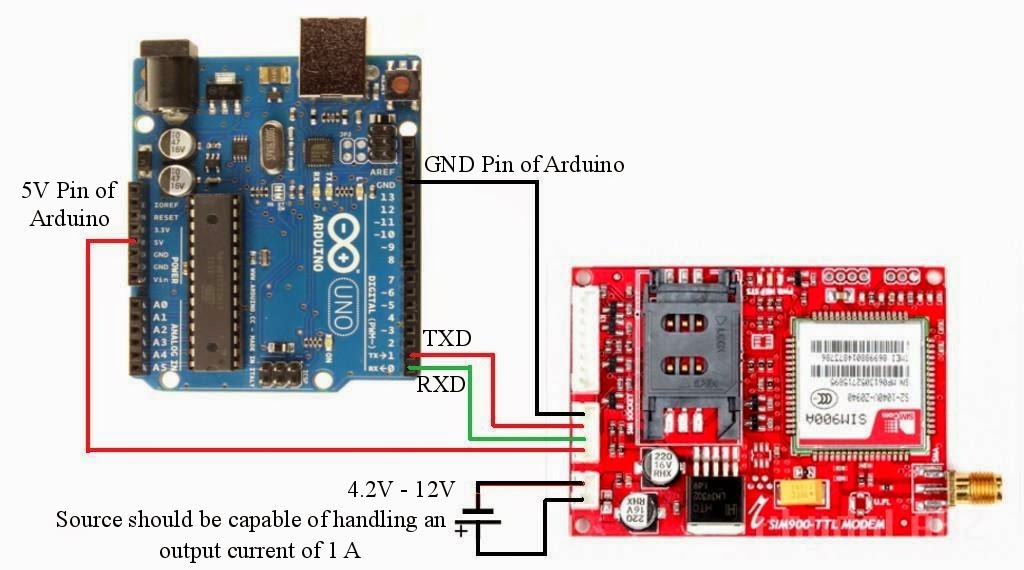


Fig 3. GSM Module with Arduino

**STRUCTURE OF GSM MODULE**



**Software Implementations**

**Android Software development kit** Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit. The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing. Android applications are packaged in.apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains.dex files (compiled byte code files called Dalvik executable), resource files, etc...

**ANDROID PROGRAM:**

**loginsuccessActivity.java**

package com.example.sun\_demo;

import android.app.Activity;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import android.telephony.SmsManager;

import android.view.View;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.TextView;

import android.widget.Toast;

public class loginsuccessActivity extends Activity{

CheckBox one,two,three,four;

Button w,t,m;

TextView welcome;

TextView p\_number;

String v1,v2,v3,v4;

String b1,b2,b3;

String phone\_number;

String i\_username;

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.login\_success);

//welcome message settings..

welcome=(TextView)findViewById(R.id.wmeg);

i\_username=getIntent().getStringExtra("username");

welcome.setText("Welcome "+i\_username);

phone\_number=getIntent().getStringExtra("phonenumber");

p\_number=(TextView)findViewById(R.id.p\_num);

p\_number.setText("("+phone\_number+")");

// checkbox setting message

one=(CheckBox)findViewById(R.id.one);

two=(CheckBox)findViewById(R.id.two);

three=(CheckBox)findViewById(R.id.three);

four=(CheckBox)findViewById(R.id.four);

//buttong setting message

w=(Button)findViewById(R.id.w);

t=(Button)findViewById(R.id.t);

m=(Button)findViewById(R.id.m);

w.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

b1="w";

if(one.isChecked())

{

v1=one.getText().toString();

}

if(two.isChecked())

{

v2=two.getText().toString();

}

if(three.isChecked())

{

v3=three.getText().toString();

}

if(four.isChecked())

{

v4=four.getText().toString();

}

send\_sms(b1);

}

});

t.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

b2="t";

if(one.isChecked())

{

v1=one.getText().toString();

}

if(two.isChecked())

{

v2=two.getText().toString();

}

if(three.isChecked())

{

v3=three.getText().toString();

}

if(four.isChecked())

{

v4=four.getText().toString();

}

send\_sms(b2);

}

});

m.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

b3="m";

if(one.isChecked())

{

v1=one.getText().toString();

}

if(two.isChecked())

{

v2=two.getText().toString();

}

if(three.isChecked())

{

v3=three.getText().toString();

}

if(four.isChecked())

{

v4=four.getText().toString();

}

send\_sms(b3);

}

});

}

public void send\_sms(String value)

{

String meg\_data =value+" "+v1+" "+v2+" "+v3+" "+v4;

//Getting intent and PendingIntent instance

Intent intent=new Intent(getApplicationContext(),loginsuccessActivity.class);

intent.putExtra("username", i\_username);

intent.putExtra("phonenumber", phone\_number);

PendingIntent pi=PendingIntent.getActivity(getApplicationContext(), 0, intent,0);

//Get the SmsManager instance and call the sendTextMessage method to send message

SmsManager sms=SmsManager.getDefault();

sms.sendTextMessage(phone\_number, null, meg\_data, pi,null);

Toast.makeText(getApplicationContext(), "Message Sent successfully!",

Toast.LENGTH\_LONG).show();

}

}

**data.java**

package com.example.sun\_demo;

public class Data {

String uname,phonenumber,city,address;

public String getUname() {

return uname;

}

public void setUname(String uname) {

this.uname = uname;

}

public String getPhonenumber() {

return phonenumber;

}

public void setPhonenumber(String phonenumber) {

this.phonenumber = phonenumber;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

**MainActivity.java**

package com.example.sun\_demo;

import java.util.ArrayList;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.view.Menu;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends Activity {

EditText username,password;

Button login;

TextView newuser;

String u\_name;

String u\_password;

static ArrayList<Data> values;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

username=(EditText)findViewById(R.id.username);

password=(EditText)findViewById(R.id.pasword);

login=(Button)findViewById(R.id.login);

newuser=(TextView)findViewById(R.id.newuserss);

login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(username.getText().length()==0 || password.getText().length()==0)

{

Toast.makeText(getApplicationContext(),"fill out all fields",Toast.LENGTH\_LONG).show();

}

else

{

u\_name=username.getText().toString();

u\_password=password.getText().toString();

for(Data t:values)

{

if(t.getUname().equalsIgnoreCase(u\_name) && t.getAddress().equalsIgnoreCase(u\_password))

{

Intent i=new Intent(MainActivity.this,loginsuccessActivity.class);

i.putExtra("username", t.getUname().toString());

i.putExtra("phonenumber",t.getPhonenumber());

startActivity(i);

}

}

}

}

});

newuser.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

values=new ArrayList<Data>();

Intent intent=new Intent(MainActivity.this,RegisterActivity.class);

startActivity(intent);

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

}

**RegisterActivity .java**

package com.example.sun\_demo;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class RegisterActivity extends Activity {

EditText uname,number,city,address;

Button register;

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.register);

uname=(EditText)findViewById(R.id.reusername);

number=(EditText)findViewById(R.id.phonenumber);

city=(EditText)findViewById(R.id.city);

address=(EditText)findViewById(R.id.readdres);

register=(Button)findViewById(R.id.register);

register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

if(uname.getText().length()==0 || number.getText().length()==0 || city.getText().length()==0 || address.getText().length()==0)

{

Toast.makeText(getApplicationContext(), "fill out all fields",Toast.LENGTH\_LONG).show();

}

else

{

Data da=new Data();

da.setUname(uname.getText().toString());

da.setCity(city.getText().toString());

da.setPhonenumber(number.getText().toString());

da.setAddress(address.getText().toString());

MainActivity.values.add(da);

Toast.makeText(getApplicationContext(), "created", Toast.LENGTH\_LONG).show();

Intent i=new Intent(RegisterActivity.this,MainActivity.class);

startActivity(i);

}

}

});

}

}

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.sun\_demo"

android:versionCode="1"

android:versionName="1.0" >

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="18" />

<uses-permission android:name="android.permission.SEND\_SMS"/>

<uses-permission android:name="android.permission.RECEIVE\_SMS"/>

<application

android:allowBackup="true"

android:icon="@drawable/empicon"

android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity

android:name="com.example.sun\_demo.MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name="com.example.sun\_demo.RegisterActivity"></activity>

<activity android:name="com.example.sun\_demo.loginsuccessActivity"></activity>

</application>

</manifest>

**MainActivity.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity" >

<EditText

android:id="@+id/username"

android:hint="user name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true"

android:layout\_marginRight="39dp"

android:layout\_marginTop="113dp"

android:ems="10" />

<EditText

android:id="@+id/pasword"

android:hint="password"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/username"

android:layout\_centerVertical="true"

android:ems="10"

android:inputType="textPassword" >

<requestFocus />

</EditText>

<Button

android:id="@+id/login"

style="?android:attr/buttonStyleSmall"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/pasword"

android:layout\_below="@+id/pasword"

android:layout\_marginTop="39dp"

android:text="Login" />

<TextView

android:id="@+id/newuserss"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/login"

android:layout\_alignRight="@+id/pasword"

android:text="new user"

android:textAppearance="?android:attr/textAppearanceLarge" />

</RelativeLayout>

**loginsuccessActivity.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<CheckBox

android:id="@+id/one"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="52dp"

android:layout\_marginTop="93dp"

android:text="@string/VALVE1" />

<CheckBox

android:id="@+id/two"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/one"

android:layout\_below="@+id/one"

android:layout\_marginTop="22dp"

android:text="@string/VALVE2" />

<CheckBox

android:id="@+id/three"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/two"

android:layout\_below="@+id/two"

android:layout\_marginTop="16dp"

android:text="@string/VALVE3" />

<CheckBox

android:id="@+id/four"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/three"

android:layout\_below="@+id/three"

android:layout\_marginTop="24dp"

android:text="@string/VALVE4" />

<Button

android:id="@+id/w"

style="?android:attr/buttonStyleSmall"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/four"

android:layout\_below="@+id/four"

android:layout\_marginLeft="54dp"

android:layout\_marginTop="20dp"

android:text="send w" />

<Button

android:id="@+id/t"

style="?android:attr/buttonStyleSmall"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/w"

android:layout\_alignRight="@+id/w"

android:layout\_below="@+id/w"

android:text="send t" />

<Button

android:id="@+id/m"

style="?android:attr/buttonStyleSmall"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/t"

android:layout\_below="@+id/t"

android:text="send m" />

<TextView

android:id="@+id/wmeg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/w"

android:layout\_alignParentTop="true"

android:layout\_marginTop="16dp"

android:text="welcome\_message"

android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView

android:id="@+id/p\_num"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignRight="@+id/wmeg"

android:layout\_below="@+id/wmeg"

android:layout\_marginRight="24dp"

android:layout\_marginTop="16dp"

android:text="TextView" />

</RelativeLayout>

**RegisterActivity.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<EditText

android:id="@+id/reusername"

android:hint=" user name "

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true"

android:layout\_marginRight="31dp"

android:layout\_marginTop="46dp"

android:ems="10" />

<EditText

android:id="@+id/phonenumber"

android:hint="phone number"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/reusername"

android:layout\_below="@+id/reusername"

android:layout\_marginTop="54dp"

android:ems="10"

android:inputType="phone" />

<EditText

android:id="@+id/city"

android:hint=" user city "

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/phonenumber"

android:layout\_below="@+id/phonenumber"

android:layout\_marginTop="57dp"

android:ems="10" >

<requestFocus />

</EditText>

<EditText

android:id="@+id/readdres"

android:hint="password"

android:inputType="textPassword"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/city"

android:layout\_below="@+id/city"

android:layout\_marginTop="57dp"

android:ems="10" />

<Button

android:id="@+id/register"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="22dp"

android:text="Register" /

</RelativeLayout>

**REFERENCES**

[1]<https://ccafs.cgiar.org/publications/adaptation-measures-agricultural-systems-messages-sbsta-44-agriculture-workshops#.VxDp7baY61I>

[2]Hu ddar, S.R.; Gowri, S.; Keerthana, K.; Vasanthi, S.; Rupanagudi, S.R., " Novel algorithm for segmentation and automatic identification of pests on plants using image processing, " Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on , vol., no., pp.1,5, 26-28 July 2012

[3]Agricultural Products in India http://www. agriculturalproductsindia. com/agro/introduction. html

[4]ICT in Indian Agriculture-Disseminating Information to Farmers http://129. 3. 20. 41/eps/get/papers/0503/0503002. pdf

[5]Pithoragarh-Taking ICT to Grass Roots, District Informatics, April 2009 http://www. nic. in/sites/upload-files/nichome/files/pdf/Pithoragarh. spdf

[6]Google Android http://www. android. com/

[7]Socio-Economic Impact of Mobile Phones on Indian Agriculture http://www. mobileactive. org/files/file-uploads/Impact%20of%20P hones%20on%20Indian%20Agriculture. pdf

[8]Rekha P, Lekshmi G.S and ManeeshaV.Ramesh,"Inegrated Wireless Sensor Network for Smart Sesame Farming in India", Elsevier, 2012.

[9] AWATI J. S, PATIL V. S,"Automatic Irrigation Control by using wireless sensor networks",Journal of Exclusive Management Science, 2012.

[10] Laith Ali Abdul-Rahaim, Ahmed Mohammed Ali Ali,"Remote Wireless Automation and Monitoring of Large Farm using wireless sensors networks and Internet", IJCSET, 2015.

[11] Lopez RJA, Sotoa F, Suardiaza F, Sancheza P, Iborraa A,Verab JA, “Wireless Sensor Networks for precision horticulture in Southern Spain”, Computer Electronics Agriculture, 68: 25-35.

[12] MahirDursun and SemihOzden, “A wireless application of drip irrigation automation supported by soil moisturesensors”, Scientific Research and Essays Vol. 6(7),Academic Journals,2011.

[13] Aniket Hade, Dr. M.K. Sengupta, "Automatic Control of Drip Irrigation System & Monitoring Of Soil by Wireless", IOSR-JAVS, 2014.

**CONCLUSION**

In farming Temperature, Humidity and Waterlevel are the most essential parameters. The growth of crops is mainly depending on these three parameters. Currently farmers don't have any system which will show real-time levels of these parameters. Even farmer don't know when humidity is increased or waterlevel increased in his green house, because of it cropproduction gets affected. The proposed system is going to monitor these changes periodically and take an action automatically or pretend the required action to the farmer. System will have a provision to visualize the graphical representation of all the streaming data from the green house. Later on farmer can operate the devices from remote location by using its smartphone.