Automation of Irrigation System using Android Technology

Nikhil Bawane¹, Prashant Kumar Gautam², Ashish Dewase³, Vishabsh Mishra³, Snehal S. Golait³
Department of Computer Technology
PCE, Nagpur, Maharashtra, India

Abstract:
Agricultural technology is a growing field that culminates to the utilization of vast terrestrial expanses. The managerial work involved in the land is proportional to its size. Generally most of the irrigation systems are manually operated and these techniques are being replaced with automated techniques that suggest an automated concept of irrigation to use the water effectively. Automated Irrigation system is implemented either based on the soil water content or based on the user input via Short Message Service systems. The first method is a secluded irrigation system where the farmer is not relayed information about the irrigation status and causes inefficiency in usage of water due to user issuing command without factoring the soil condition. Resulting from the perpetual rising of population, modern techniques are developed to control the system. This paper proposes a novel technique to present a automated irrigation system that analyze the conditions of agricultural land in a real-time manner and provide a rapid supervisory control through Android application.

Keywords: android, automation, irrigation, arduino, optimization.

I. INTRODUCTION

The traditional techniques are being replaced with semi-automated and automated techniques implemented either based on the soil humidity or based on GSM systems. These methods have improved farming and crop management many times over, however they have their limitations. Some of the methods cannot be handled remotely and, thus, cannot provide timely information. Other methods that provide information on-the-go, are limited by the latency in communication. In our current times we are seeing a rapid growth in various technological advancements and as such, modern techniques are introduced to control the agricultural systems. This paper aims to present an automated irrigation system that analyse the conditions of soil in a real-time manner and provide a rapid supervisory control through Android application. A combination of humidity, moisture, pH, and light and temperature sensor is used to find the soil and environmental conditions. These sensors are connected to the Arduino which is used here for sending information via serial communication to the server's software. All the sensor data is periodically captured and stored in a database by the server. The Android app is developed using Java to interface with server. The application presents the data to the user in the form of statistics and graphs. The user is also capable of controlling actuators components of the system like motors and pumps, through the applications interface and can switch them on or off, remotely. In addition to providing a system interface, the client android app also provides a notification system for alarm conditions.

II. RELATED WORK

A paper [1] describes a system of drip irrigation using Internet of Things (IoT). This system creates a network using usb to serial connection to the server. The server connects to the internet and through this the server side data can be retrieved. The sensor data, such as moisture content and pH level of the soil, is also displayed on an LCD screen that is located at the network site. In [2] we see a “Garden Bot” proposition that optimizes regulation of water in plants using data from sensors. This system uses a Raspberry Pi for the server and Android application to display status information remotely. In [3] a GSM based home automation system is described that was developed using the Google App Inventor for Android. App Inventor is a visual programming platform for developing mobile app without need to write any code. Instead it provides blocks of interlocking components to design the app’s functions. In another paper [4], using an Android device, messages containing commands and information is sent to a GSM module that is parsed by a controller. 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. In paper [5] we see an irrigation system that utilizes a Wireless Sensor Network. This interesting system uses the ZigBee protocol for the wireless communication, with sensor nodes made with Arduino and Xbee wireless shield connected with sensors. It also uses a raspberry pi to provide the server, database and web interface.

III. AUTOMATED IRRIGATION SYSTEM

A. Proposed architecture used for automated irrigation

Architecture of the automated system is as shown in the figure. The system consists of three parts:

a. The Android Client application,
b. The Arduino controller node and,
c. The Web Server.

Figure. 1. Architecture of System
The Arduino controller and motors are connected using a relay circuit module. The sensors are connected directly to the Arduino interface. The web server is designed on .NET technology for monitoring and control the irrigation. Devices which have networking components will be used here like PC, laptop etc.

1) Client Application

The client application is built upon Android technology using the Android software development kit and Android Studio. Upon opening the application the user is asked for authentication. If the user does not have their credentials they can ask to be registered through a link provided. Once authentication is validated, the user is navigated to another screen that shows the information of the currently connected system at site of installation. The information from the sensors is displayed as well as controls for the irrigation system. The user can choose to toggle the irrigation system on/off or schedule it to be run at a later system. These commands override the scheduling decisions made by the server and should be used as per the judgment of the user.

2) Controller node

a) Sensor node

Sensor node used here to obtain the soil parameters. It is designed using Arduino UNO prototyping board. It consists of soil moisture and soil temperature sensors, and data cache for temporary storage. It will detect the soil variables typically at 10 second intervals and signal back to the server via serial Bluetooth communication. There are two configurations in which the sensor node can be used. The sensor node can be singularly connected to the irrigation system, providing a central microprocessor. It consists of connecting it to a computer. The sensor uses a Arduino UNO prototyping board. It consists of rino UNO prototyping board. It consists of

b) Actuator node

The actuator node is the parts that which we want to control using our system. It consists of the motors, pumps, lights, etc. and these are connected to the 2 channel 5v/12v relay module. This relay module is an electronic switch that is controlled using voltage values. When a high voltage (+5v or bit value 1) is passed to the module, the switch is turned on. When voltage is low or is removed (0v or bit value 0), the switch is turned off. The relay module is connected to the Arduino and allows controlling the actuators without damaging itself.

c) Bluetooth module

The Bluetooth module we are using is the HC-05 module. This is an easy to use Bluetooth Serial Communications module, designed for simple and hassle-free wireless connection setup. The module uses Bluetooth V2.0+EDR allowing up to 3Mpbs data speed. The module does not require any special Bluetooth drivers. In order to transmit or receive data with an Arduino, only the serial read/write data commands have to be used.

d) Sensors

i. Soil temperature sensor

The temperature sensor used here is NTC digital/analog thermometer. The sensor requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –20°C to +105°C. Output in digital form in the range 0V to 5V, adjustable trigger level from preset. Output in analog is over 0V to 5V based on temperature on ambient. The sensor uses a LM393 comparator. The LM393 is a dual differential comparator; this means that it accepts 2 inputs for comparison. This essentially allows the sensor to output both analog and digital signal at the same time.

i. Soil moisture sensor

The module consists of, detection prongs, and analog to digital convertor circuit. It has two outputs, digital and analog. It can output signal types, digital as well as analog, both at the same time. The sensor will detect the moisture of the soil that it has been inserted in. If the water moisture contents are low the module’s output will be high otherwise the output will remain as an unchanged value. This moisture sensor has a pair of prongs used to pass a current into the soil. It then reads that resistance across the soil between the pair of prongs to get a reading. This reading is detected by the Arduino analog input in the range of 0 – 1023. If the soil moisture content of the soil being observed is more, the electricity will conduct freely, indicating a low resistance value. If the moisture content is low, the soil will conduct lesser electricity. Thus, indicating a higher level of resistance value.

3) Web Server

The web server is built using Microsoft IIS, SQL server and .NET technologies. The Internet Information Services (IIS) server is used to host the web application interface for the android client application. This uses ASP.NET to create a RESTful API. The SQL server is used to create the database. This database contains the user registrations table, the sensor and other components tables, and the data history logs. All data from the sensors is time-logged in the database. This data is very important for the optimization of irrigation patterns. The link between the server and the Arduino controller is programmed on the .NET framework using c# programming language. Microsoft Visual Studio is an IDE which helps to write code for the .NET framework. The server contains the optimization algorithm which essentially calculates mean values for each season of the year. This is further tuned to the time of day.

B. Introduction of Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

Figure 2. Arduino UNO R3
C. Introduction to Android

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touch screen mobile devices such as smartphones and tablets. It is a highly popular platform with around 1.5 billion monthly active users (as of September 2015). It provides a very user friendly experience and a centralized application store from where users can easily install apps. Android applications are written using the Android software development kit, in the Java programming language. There also exists an IDE called the Android Studio, based on the open source IntelliJ IDEA, which is a developer friendly tool for Android application development. Through the application programming interface (API) for Android, developers can access the phones basic functions like telephony, internet access, Bluetooth etc., as well as it’s in-built sensors’ data and GPS location.

IV. IRRIGATION SYSTEM OPERATION

A. At the irrigation side

The sensor node is connected to the controller which also connects to the actuator node. The sensor node contains one soil moisture and soil temperature sensors. The moisture and temperature together allows to estimate the soil condition and help decide whether the irrigation system should activated or not. The programming on the Arduino board is such way that after every minute sensor node sends soil parameter data to server via serial communication. The server receives the information and logs it into the database. Based on the soil condition and previously logged information, the server controller decides whether or not to start irrigation and for how long to irrigate. The temperature sensor allows knowing the temperature of the soil. We require this information in order to know whether drop in soil moisture content was caused due to evaporation, or that the crop has properly hydrated itself. If the moisture was used by the crop then the next irrigation cycle can be started at a further time. The actuator node consists of the parts of the irrigation system that we need to control like motors and pumps. These are known as the actuators, since they are the moving mechanical parts of the complete system. The actuators are connected to relay module which is connected to the Arduino. The relay module allows the Arduino to safely control the actuators (which work on a different voltage range).

B. At the Client side

The Android app is developed using Java to interface with server. The app provides the user interface to authenticate the user identity for automation of irrigation system. The user login credentials are stored in a database and is retrieved to verify the identity of registered users. The users upon successful login send the sensor data request to server and the requested data is retrieved from the sensor database to identify the water level in plants. The client app running on Android technology is capable of retrieving this data from the server. The application presents the received data to the user in the form of statistics and graphs. The user is also capable of controlling actuator components of the system, like motors and pumps, through the applications interface and can switch them on or off, remotely. In addition to providing a system interface, the client android app also provides a notification system. When the microcontroller detects a conditional event, like soil moisture dropping below a threshold, an alarm is received by the server. The server processes this information and pushes a notification to the user and an action that can be performed. In case the action times out, a final default action will be performed. In this case, the irrigation system will be activated.

V. CONCLUSION

This paper designs the automated irrigation system using Android technology. In this we have used Arduino as a middleman controller which allows collecting the sensor information from sensor node continuously. By providing the interface on Android phone, user can easily monitor the system. This interface also provides an array of functions like scheduling overrides, data history, action log etc. in a secure manner. We have a .NET server that stores data in a database, performs the decision making based on the information stored in database, monitors the activity from the Arduino, manages scheduling overrides that are set from the android application, and also provides the information to the application. The system determines proper irrigation pattern by analyzing the soil variables. These patterns correspond to publicly available information about soils in the particular area. This complete process smartly reduces water consumption.

VI. REFERENCES


