DESIGN AND IMPLEMENTATION OF AN ANDROID BASED TOMATO IRRIGATION SYSTEM

By

Blessed Machengete (R14662E)

Submitted in partial fulfilment of the requirement for the degree of

Bachelor of Science (Hons) Telecommunications

Department of Applied Physics and Telecommunications in the

Faculty of Science and Technology at the

Midlands State University

GWERU

June, 2017

Supervisors: Mr G. Manjengwa and Dr Gora

ABSTRACT

The advent of technology has made life easier, smooth and smarter. Smart phones are being applied in all aspects of life, be it Medical field, Military, Aviation or Agriculture. Agriculture is one of the key drivers of Zimbabwe`s Economy. The majority of Zimbabweans venture into farming for both personal sustenance as well as sustenance of the nation as a whole. Small farming projects are being carried out in the backyards of urban areas. People have resorted to growing Potatoes, Beans, Peas, Carrots, Tomatoes, cabbages and many other vegetables for personal consumption as well as for commercial purposes. Most of these people do this on a part time basis. This project seeks to devise a simple low cost android based tomato irrigation system. The user will be able to control the Irrigation system using an android application on a smart phone. This project is developed by using Arduino Uno, SIM900 GSM module, LCD display, soil moisture sensors and water tank level sensor. The android application makes use of the GSM technology to send commands to the irrigation system. GSM network is widespread in Zimbabwe and have almost reached every corner of the country. This makes the Android application reliable and cheap to use.

DECLARATION

I, **Blessed Machengete**, hereby declare that I am the sole author of this thesis. I authorize the Midlands State University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature ________________________________ Date ____________________________

APPROVAL

This dissertation/thesis entitled **Design and Implementation of an Android Based Tomato Irrigation System** by **Blessed Machengete** meets the regulations governing the award of the degree of **BSC Telecommunications Honours** of the **Midlands State University**, and is approved for its contribution to knowledge and literal presentation.

Supervisor ……………………

Mr G. Manjengwa

Date ……………………………

Supervisor ……………………

Mr Gora

Date ……………………………

ACKNOWLEDGEMENTS

I would like to thank the almighty God for giving me the privilege to successfully complete this project. I would also like to acknowledge my Supervisors Mr G Manjengwa and Mr Gora for their unwavering support and guidance. I would also like to thank my workmates at Liquid Telecom for their support during the tenure of the project. Last but not least, Special thanks goes to my life partner Lesley Machengete for being the pillar of my strength.

LIST OF FIGURES

ABBREVIATIONS

GSM	Global System for Mobile Communications
LCD	Liquid Crystal Display
ZIM ASSET	Zimbabwe Agenda for Sustainable Socio-Economic Transformation
SIM	Subscriber Identity Module
SMS	Short Message Services
LED	Light Emitting Diode
PC	Personal Computer
USB	Universal Serial Bus
MCU	Microcontroller Unit

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Research Background

Zimbabwe is an Agro-based country where 80% of the population derive their livelihood from agriculture. The greater percentage of this portion are people from the rural areas. There has been an increase in people from the urban areas concentrating on agriculture as well. This is being done on small to medium-sized gardens to provide a diversified diet, improved nutrition for families, and supplemental income. A lot of them are concentrating on farming, potatoes, tomatoes, peas, paper, and other vegetables.

Agriculture is one of the key drivers for projected growth in the ZIM-ASSET programme with an estimated growth of 4% from 2017 into 2018 [1].

Table 1: Growth targets for Zim Asset

Sector	2013	2014	2015	2016	2017	2018
	Proj. %	Proj. %	Proj. %	Proj. %	Proj. %	Proj. %
Agriculture, hunting and fishing	-1.3	9.0	5.1	7.0	8.0	12.5

Fig 1.1 Zim Asset growth targets [1]

Part of the key success factors that the ZIM-ASSET programme will depend on the following, in relation to Agriculture:

I. Scientific research and development.
II. Rehabilitation, upgrading and development of key infrastructure and utilities comprising power generation, roads, aviation and water.

The expansion and enhancement of irrigation projects will also enhance the anticipated growth within the agricultural sector. One of the strategies that can be put in place is working on the technology that supports the irrigation projects. About 16% of farmed land is irrigated. This accounts for about 40% of productivity. Close to 80% of the water used in the world is for irrigation and there is a huge portion of that 80% that is going to waste due to inefficient irrigation methods. We are at about 600,000,000 acres of irrigated land in the world. Most horticulture crops are sold fresh with 80 – 90% of their weight being water. [2]

1.2 Problem Statement

Tomatoes are the most popular vegetables grown in the home garden. This is so because, apart from being an essential part of every meal, they are relatively easy to grow. However, this does not mean that they grow without care. One of the most crucial parts of their care is knowing how much water they need. [3] They require an even supply of water throughout the season, an irregular water supply will cause the tomatoes to develop problems such as:

- Blossom end rot
- Stunted growth
- Reduced fruit production
- Susceptibility to pests
- Root loss
- Sub quality fruit

The amount of water requirements vary with the soil but, a general baseline is considered to be at least one inch of rain or irrigation water per week for a steady growth. [4] In the hotter, drier parts of the country, their needs can go up to two inches of water per week during the summer months. An inch of water measures out to about 60 gallons for each 100 square feet of garden. Tomatoes like moisture, but overwatering is harmful. You not only waste water, but soggy soil will prevent roots from getting the air they need. The traditional irrigation system for tomatoes requires human intervention to monitor moisture, temperatures, and light intensity amongst other things. With the advent of smart phones everything is evolving around them these days.

Applications are being developed to manage day to day human activities. With the use of an android based application we can control our tomato irrigation system remotely without any human intervention.

1.3 Aim

The project seeks to design and implement an android based Tomato irrigation system. The application will integrate, a microcontroller, a soil moisture sensor, tank water level detector, a pump to feed water into the water tank and another pump to feed water into the irrigation pipes. Collected information from the soil moisture sensor and the water level detector will trigger actions via the microcontroller.

1.4 Objectives

The objectives of this research are:

- To design and implement an android based Tomato irrigation system that gathers soil moisture as well as monitoring water levels in the tank supplying irrigation water.
- To design an android based Tomato irrigation system that initiates actions to stabilise anomalies in the conditions and communicate these actions via the android app. The initiation of actions can either be automatic or manual.

1.5 Justification

- The system will automate the process of monitoring soil conditions for the Tomatoes as well as rectifying them if need be.
- The system will communicate actions and events to the user through the android application, thereby eliminating need for user`s physical presence.
- The system will cut on labour costs.
- The system will optimise conditions for plant growth thereby increasing production.
- The system will cut on wastage of water. The sensors will help regulate the amount of water required to water the plants.

1.6 Scope

The scope of the project is to monitor the:

- Soil moisture.
- Water tank levels.

The scope is also to pump water into the tank and to the irrigation pipes. The application initiates watering of plants and refilling of water tank, automatically or manually

Fig 1.3 below illustrates a block diagram representing the android based irrigation system. Whilst Fig 1.4 illustrates a complete overview of the android based irrigation system.

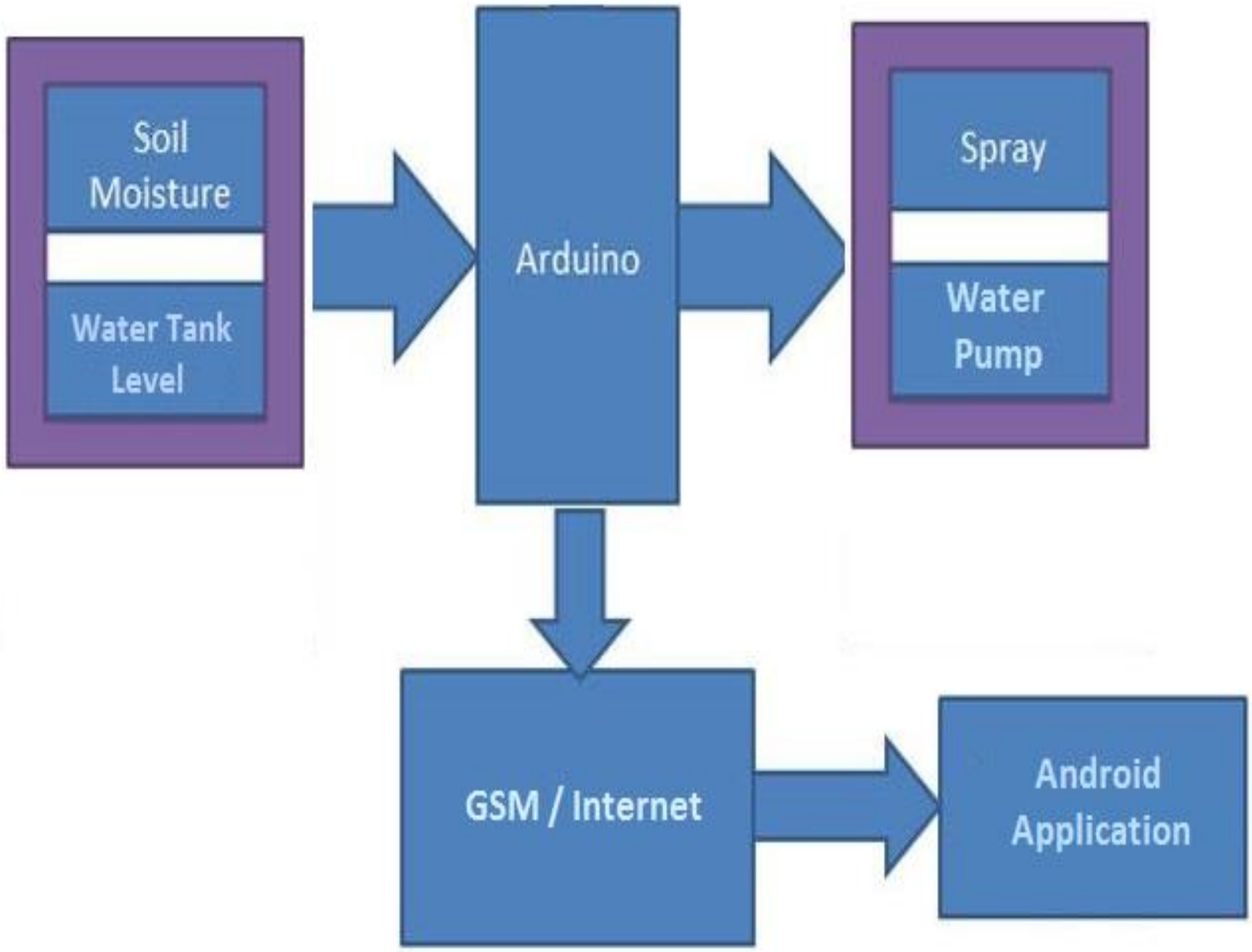

Fig 1.2 Android Based Irrigation system Block diagram

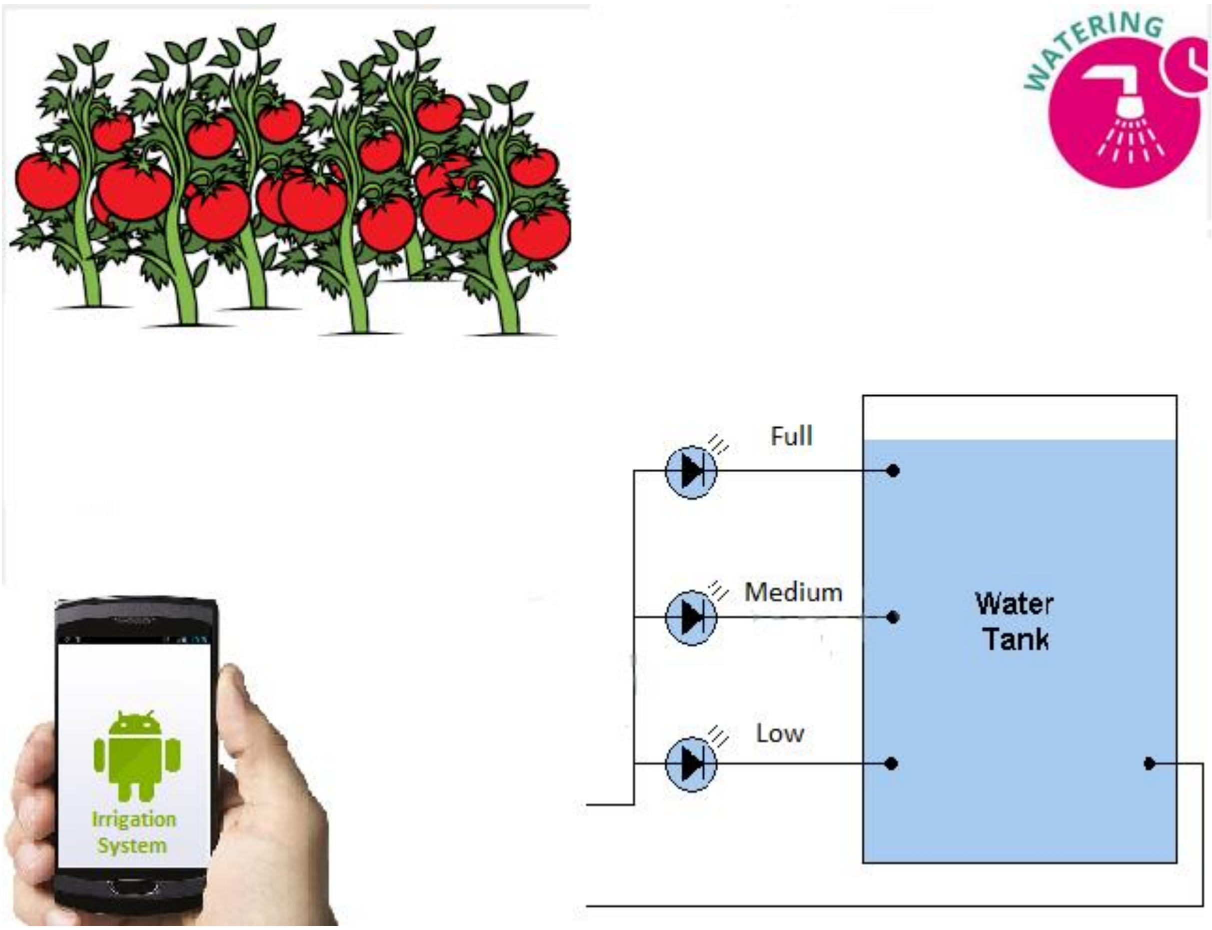

Fig 1.3 Android Based Irrigation System

1.7 Hardware Components

The following hardware components will be required for the project`s objectives to be successfully accomplished.

- Smart Phone.
- Arduino Uno Microcontroller.
- SIM 900 GSM module.
- Soil moisture sensor.
- 2 X Relay switches.
- 2 X Mini water pumps.

REFERENCES

[1] Zim Asset Blue Print

[2] https://www.gardeningknowhow.com

[3] http://rootedinhope.org

[4] https://garden.org

[5] Jifeng Ding, Jiyin Zhao,Biao Ma, College of Electromechanical & Information Engineering,Dalian Nationalities University Dalian, China, "Remote monitoring system of Temperature and humidity based on GSM," in *2nd International Congress onImage and Signal Processing*, 2009

[6] https://www.arduino.cc/en/Guide/HomePage

[7] Viacheslav and Adamchuk. (2011). *McGill University*, Canada: Evaluation of Automated Soil pH Mapping (ASAE Paper No. 99-1100)

[8] B. Boman, S. Smith, B. Tullos (2006): *University of Florida*, Gainesville: Control and Automation in Citrus Microirrigation Systems

[9] Hill. (2005): Automatic Irrigations system

[10] M. F. Leroux. (2005). *McGill University*, Canada: Automated Irrigation system

[11] "Microcontroller to sensor interfacing techniques," BiPOM Electronics INC., [Online]. Available: www.bipom.com.

[12] *ATmega32 Datasheet,* Atmel Corp., 2006.

[13] Design and implementation of a programmable remote controlled and monitored irrigation system Tajedinn abdelgawi Baker , Abdelrasoul jabar kizar alzubaidi*, Bahri University, Sudan,2014*

[14] Theophilus Wellem, Department of Information Systems, Bhudi Setiawan, Department of Informatics,Satya Wacana Christian University Salatiga, Indonesia., "A Microcontroller- based Room Temperature Monitoring System," *International Journal of Computer Applications,* vol. 53, no. 1, 2012.

[15] "Hitachi HD44780 LCD controller," Hitachi, 1998. [Online]. Available: en.wikipedia.org/wiki/Hitachi HD44780 LCD controller.

CHAPTER 2: THEORETICAL ASPECTS

2.1 Introduction

The integration of Telecommunication Systems and irrigation has brought about a lot of efficiency in Agriculture. The advent of technology especially with the coming of smart phones has even made the world in general a better place. The researcher went through previously designed and implemented automated irrigation systems.

2.2 Literature Review

2.2.1 Technologies in Agriculture

Sensors help agriculture by enabling real-time traceability and diagnosis of crop, livestock and farm machine states. Food may benefit directly from genetic tailoring and potentially from producing meat directly in a lab. Automation will help agriculture via large-scale robotic and micro robots to check and maintain crops at the plant level. Engineering involves technologies that extend the reach of agriculture to new means, new places and new areas of the economy. Of particular interest will be synthetic biology, which allows efficiently reprogramming unicellular life to make fuels, by products accessible from organic chemistry and smart devices. Air and soil sensors are fundamental additions to the automated farm, these sensors would enable a real time understanding of current farm, forest or body of water conditions. With crop sensors instead of prescribing field fertilisation before application, high-resolution crop sensors inform application equipment of correct amounts needed. Optical sensors or drones are able to identify crop health across the field. Building on existing geolocation technologies, future swath control could save on seed, minerals, fertilizer and herbicides by reducing overlapping inputs. By pre-computing the shape of the field where the inputs are to be used, and by understanding the relative productivity of different areas of the field, tractors or agbots can procedurally apply inputs at variable rates throughout the field. Agricultural robots (agbots) are used to automate agricultural processes, such as harvesting, fruit picking, ploughing, soil maintenance, weeding, planting and irrigation. Precision agriculture: Farming management based on observing (and responding to) intra-field variations. With satellite imagery and advanced sensors, farmers can optimise returns on inputs while preserving resources at ever larger scales. Further understanding of crop variability, geolocated weather data and precise sensors should allow improved automated decision-making and complementary planting techniques.

Soil and crop sensors

Farming equipment today is being replaced by smart sensors. These have the ability to read different anomalies on plants from their health, water requirements for the crops as well as nitrogen levels within the soil. The sensors trigger application of required inputs based on the given real-time field conditions. When it comes to water usages the sensors optimise water usage avoiding unnecessary yield losses [1]. The Trimble’s GreenSeeker, Topcon’s CropSpec, and Ag Leader’s Opt-Rx are some of the recently innovated optical-sensing technologies used in determining crops` health. Light reflectance on the crop is measured and translated into nitrogen levels. Signal application systems to apply the correct amount of nitrogen the crop needs are then evoked by Electronic controllers. Sensor technology also is available to measure features within the soil, like organic matter content, soil electrical conductivity, even ground elevation. Manufacturers such as Dualem, Veris Technologies and Geonics all make different soil sensor types. Satellite or aerial imaging is also another sensing type. It also referred to as remote sensing. The satellites shoot images of key agricultural areas on certain intervals capturing differences in the crop health. Farmers then apply nutrients based on a prescription from the satellite images.

2.2.2 Control strategies and control systems

The effort that is to be kept in mind for measurement of efficiency and productivity of any system and strategy, management of value and work is to be subdued in accordance of any goal. Put all goals and factors measured in check through inspection or observation. Keeping these facts in mind, we decided to tackle part of the problem by trying to improve the efficiency of input and resource use in systems of output efficiency check.

Control Systems

A control system is a system that controls other systems. A control system is a system of devices or set of devices, that manages, commands, directs or regulates the behaviour of other devices or systems to achieve desire results. Human control has taken a major collapse with science suggestions taking over in the industry to manage and control resources as per a command passed. Automation highly requires control of devices. The main feature of control system is, there should be a clear mathematical relation between input and output of the system. When the relation between input and output of the system can be represented by producing a command then an output is expected to come according to that command.

Two general types of controllers are used to control irrigation systems: open control loop systems and closed control loop systems. Open control loop systems apply a pre-set action, such as is done with simple irrigation timers. Closed control loops receive feedback from sensors, make decisions and apply the results of these decisions to the irrigation system.

A control system in which the control action is totally independent of output of the system then it is called Open loop control system. Manual control system is also an open loop control system which is simple to design and construct [2]. An open loop system also referred to as non-feedback system, is a type of continuous control system in which the output has no influence or effect on the control action of the input signal. In other words, in an open-loop control system the output is neither measured nor fed back for comparison with the input. Therefore, an open-loop system is expected to faithfully follow its input command or set point regardless of the final result. An open loop system has no knowledge of the output condition so cannot self-correct any errors it could make when the pre-set value drifts, even if this results in large deviations from the pre-set value.

Open loop control systems are typically low in cost and readily available from a variety of vendors. They vary in design and complexity and often offer flexibility as to the number of zones and how to schedule work and let the system run accordingly. The knowledge gained in designing an open loop model can be of great value toward suggesting improvement in the system under investigation. By changing system supervision and observing the resulting outputs, valuable insight may be obtained into which variables are most important and how variables interact. A new resource in an Automation system can change output and processing of read information or input. Open loop control systems can be best utilised in automation as commands are passed and embedded to control another system and functionalities that are set. A system that is running on a supervised set up can be altered to implement an automated model that will supervise or control it by processing all the input that was gathered and passed to the control system.

When automation is not well applied? In closed loop systems, the operator develops a general control strategy. Once the general strategy is defined, the control system takes over and makes detailed decisions of when to apply the action need as per the input received. This type of system requires feedback from one or more input devices for instance sensors. Processing decisions are made and actions are carried out based on data from sensors. In this type of system, the feedback and control of the system are done continuously. Closed loop controllers require data acquisition of the problematic parameters as well as system parameters. As these systems contain feedback loop, the closed loop control systems are also called as “Feedback control systems” [3]. By giving the feedback to the input signal, we can accurately control the output of a control system. A closed loop control system can have more than one feedback loop. The state of the system is compared against a specified desired state, and a decision whether or not to initiate an action based on this comparison. Closed loop controllers typically base their processing decisions on sensors or any other input response. The simplest form of a closed loop control system is that it is interrupted by an input value to take action as feed back to that input. By comparing the generated output with the actual condition, the closed loop system maintains and achieves the desired output. If the produced output is deviated from decided actual output, the closed loop control system generate an error signal and the error signal is fed to the input of the signal.

As the closed loop control systems have feedback signal to control the output, these are very accurate and less error prone. Close loop control systems can give feedback to a particular input that has been read and interpret that value of input to check if it is valid. A closed loop system depend on input so as to give back output. They can automatically correct the errors by means of the feedback signal. They provide stability as they do not base on manual check and prompting to process. They anticipate the situation for that data to be processed by a particular system and can give a positive feedback signal or negative feedback signal. The knowledge of closed system can be applied on a problem that will require feedback to be given as there is input coming from other devices and action to be taken as to what is required for that input.

2.2.3 Irrigation systems

2.2.3.1 Irrigation by using simple systems

For many years simple systems have been used to facilitate irrigation, depending on how water is distributed throughout the field. Some common types of irrigation systems include:

- Surface Irrigation, whereby water is distributed over and across land by gravity. No mechanical pump is involved.
- Localise irrigation, whereby water is distributed under low pressure, through a piped network and applied to each plant.
- Drip irrigation, whereby a type of localised irrigation in which water drops of water are delivered at or near the root of plants. In this type of irrigation, evaporation and runoff are minimised.
- Sprinkler irrigation, whereby water is distributed by overhead high pressure sprinklers or guns from a central location in the field or from sprinklers on moving platforms.
- Centre Pivot irrigation, whereby water is distributed by a system of sprinklers that move on wheeled towers in a circular pattern. This system is common in flat areas of Zimbabwe.
- Lateral Move irrigation, whereby water is distributed through a series of pipes, each with a wheel and set of sprinklers, which are rotated either by hand or with a purpose-built mechanism. The sprinklers move a certain distance across the field and then need to have the water hose reconnected for the next distance. The system is less expensive but requires more labour than others.

- Manual irrigation, whereby water is distributed across the land through manual labour and watering cans. This system is very labour intensive.

All the above mentioned systems require constant human intervention to record data manually. There is no remote monitoring involved. Someone has to be on the ground to check if the soil is wet enough, to turn on and off the water pumps. There is no integration with modern technologies in all the above mentioned systems.

2.2.3.2 Automated Irrigation Systems

Marie France Leroux`s automated irrigation system.

The system consists of a water tank, valve, irrigation pipes, moisture sensor and a microcontroller. The microcontroller acquires data from the moisture sensor and when a certain minimum level of moisture is read the microcontroller opens up the valve on the water tank. When a maximum level is reached the microcontroller closes up the valve. Fig 2.1 below is an illustration of the design.

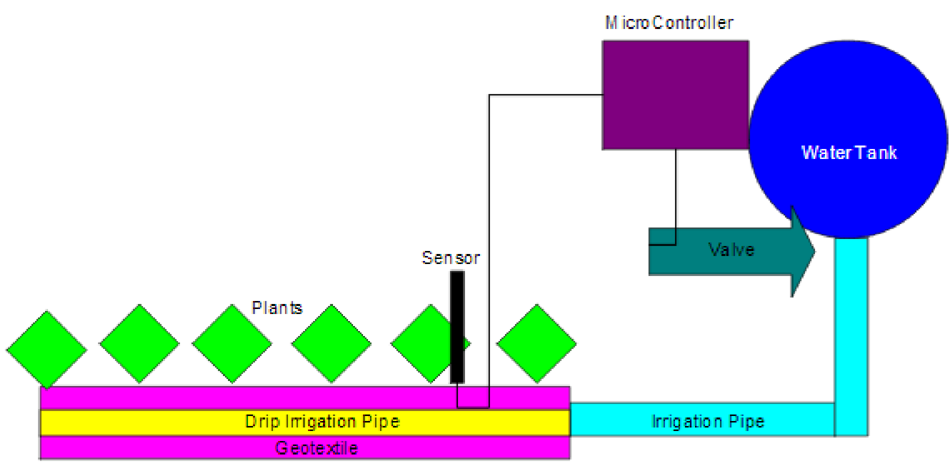

Fig 2.1 Automated Irrigation System [4]

The system proves inefficient in many forms. The system requires someone to monitor whilst on site. If anything goes wrong in one of the system components only a person on site will be able to tell. [4]

The University of Zimbabwe team.

The team comprises of students from the Computer Science Department and led by Lecturer Michael Munyaradzi designed a prototype of a special variability Irrigation System which deploys automated irrigation and saves water usage. The system uses soil humidity, atmospheric and plant sensors placed in varying sections of a field connected to transceivers that relay the data collected to a central node. The node is fitted with microprocessors which compare received data with present thresholds, triggering a valve to open or close water supply. A GSM modem is also fitted to the system to send SMS reports. [5]

Fig 2.2 Variability Irrigation System [5]

The system is efficient and flexible in the sense that a farmer does not necessarily have to be present to monitor the irrigation. It however lacks the modernisation in the sense that instead of texting commands to the system an integration with android applications would be ideal. GUI interfaces, click and go manoeuvring.

Spatial Variability Irrigation System

The system deploys automated irrigation and saves water usage. It was developed three years ago by another team at the University of Zimbabwe. This system utilises the use of sensors that can read the surrounding temperature. The system is placed in varying sections of a field connected to transceivers that relay the data collected to a central node. The node is fitted with microprocessors which compare received data with pre-set thresholds, triggering a valve to open or close water supply. A GSM modem is also fitted to the system to send SMS reports. The team is currently working on an Android based application to allow a user to override the system with manual triggers or set new thresholds remotely.

This nifty setup is not a new solution as it is used in varying forms worldwide but is unique in that it offers a cheaper method of setting up by use of simple radio and GSM communication. It was made as a solution for new farmers through which they can control water loss. Tests conducted indicate that the solution can reduce wastage of water. Shen et, al introduced a GSM-SMS remote measurement and control system for greenhouse based on PC-based database system connected with base station. [6] Base station is developed by using a microcontroller, GSM module and other sensors that gives input into the microcontroller at the base station. The GSM based irrigation system may offer users the flexibility to regulate and control the operations of their irrigation systems with little intervention to reduce runoff from over watering for improvement in crop yield. This enables users to take advantage of the globally deployed GSM networks with its low SMS service cost to use mobile phones and simple SMS commands to manage their irrigation system. It will be possible for users to use SMS to monitor directly the conditions of their farmland and attend to any interrupts that might have occurred.

Fig 2.3 GSM based auto irrigation system [6]

The figure above shows an irrigation system that take action according to commands from a short message service (SMS) reply. That SMS can be transmitted via a GSM module which provides the service. The functionality of the GSM was tested by connecting it to the microcontroller board which was programmed to turn on-and-off an LED using SMS from a mobile phone. The microcontroller is the core of the control system and processes input that comes from the GSM and the command that is passed. GSM commands are based on the results that would have been obtained from other sensors.

PIC 8051 Irrigation system

The project uses an 8051 series microcontroller which is programmed to receive the input signal of varying moisture condition of the soil through the sensing arrangement. This is achieved by using an op-amp as comparator which acts as interface between the sensing arrangement and the microcontroller. Once the controller receives this signal, it generates an output that drives a relay for operating the water pump. An LCD display is also interfaced to the microcontroller to display status of the soil and water pump. The sensing arrangement is made by using two stiff metallic rods inserted into the field at a distance. Connections from the metallic rods are interfaced to the control unit.

The concept in future can be enhanced by integrating GSM technology, such that whenever the water pump switches ON/OFF, an SMS is delivered to the concerned person regarding the status of the pump. We can also control the pump through SMS.

Dynagage Sap Flow System

The Dynagage Flow32-1K Sap Flow system and Dynagage sensors have been servicing research plant scientists throughout the world for over 10 years. The Flow32-1K software makes working with Flow32-1K sap flow system easier than ever before with built-in algorithms for efficient and faster data analysis. New powerful functions include auto-zero and sensor status built into the data logger program. Sap flow data recalculation and automatic charting with an Excel™ Macro link makes the system a superior water relations measurement system. Sap Flow has never been this easy and powerful. Dynagage sap flow sensors are the most accurate and reliable sensors available for measuring plant sap flow. Dynagage is now a key technique in modern water management, hydrology, crop studies, plant water relations, and biomass production.

Sap flow measurements have an almost unlimited number of applications. Sap flow and transpiration rates provide commercial benefits from accurate irrigation schedules, improved irrigation set points and real crop ET coefficients. Sap flow is key data to model annual forest growth rates and conduct environmental remediation projections. After all, who can tell better than the plant how much water is consumed under varying conditions.

EBS - Model 10000 - Irrigation Irrigation Filters

The Amiad EBS Series are automatic filters, with an electric self-cleaning mechanism. The 'EBS' filters range in flow-rates of up to 7200 m^3/h (32000 US gpm), with screens designed ranging from 800-10 micron filtration degree. Inlet/Outlet flanges are available from 8'-36' diameter. The system feature large filtration area, reliable operating mechanism and simple construction make the EBS filter the ideal solution for filtration of high-flow and poor quality water to very fine filtration degrees and automatic flushing according to pressure differential and/or according to time. It can cater for no interruption of downstream flow during flushing, robust and reliable self-cleaning mechanism even on marginal operation conditions and Minimal volume of reject water allows excellent operation during flush mode.

Azud - Automatic Filtration Range

AZUD range of products present multiple combinations to an efficient and economic filtration process, adapting to the different types of water and especial requirements of the agricultural, gardening, golf courses and landscaping areas. AZUD filtration system have been designed to: Protect all the elements of the hydraulic system, avoiding clogging in control elements, protection or water emission. Adequate the quality of water to the required levels. Reduce the frequency and intensity of the maintenance labours. AZUD HELIX AUTOMATIC range is based in a filtering element with a self-cleaning mechanism, which is activated when the direction of the flow is inverted. AZUD HELIX AUTOMATIC incorporates the advantages of Modular and Helix Technologies. High autonomy and effectiveness in the back flushing with low working pressure is a feature of Azud and can be implemented on specific materials compatible with sea water and or saline water.

ZigBee and internet technology.

ZigBee is an open wireless connection standard that can be used in directions communications. The idea was developed for improve irrigation system and reduced cost of irrigation water. Sensors are placed in farm and sense continuously and collect he information. This information stored at centre monitor and also passes to data collection interface and then transmits to the wireless sensor node. [7]

An automatic irrigation system used for irrigate sage crop field as compare to traditional irrigation system using wireless network and GPRS system.[8] The Brutsaert's model used for measure the moisture of agricultural soils by an accurate, on site, real-time method and also derived the speed-moisture curves, the conditions for the actual validity of the curves, and the suitable sound frequency for performing the measurement, for a wide range of agricultural soils in different physical conditions. [9] With all that is mentioned GPRS technique has some disadvantage of slow speed, distance factor and reliability.

2.2.3.4 Computer Based Irrigation Systems

There are also computer based irrigation systems which can be interactive or automatic. Interactive systems are usually built around a microcomputer, either a standard personal computer (PC) or a specially designed unit. The information is transferred into a central unit either directly from sensors in the pipeline or from intermediate units which collect the data from a number of sensors and then process and store them temporarily for further transfer to the central computer. These systems have features that enable the operator to transmit commands back to the various control units of the irrigation system.

2.2.3.5 High tech irrigation systems

These use time clock controllers, or timers as an integral part of an automated irrigation system. A timer starts and stops the irrigation system as of when it was set to trigger the system. [10] The pre-set amount of water can be applied in the field segments by using automatic volume controlled metering valves. With this application irrigation is based on actual dynamic demands of the plant itself, the plant root zone is effectively reflecting all environmental factors acting on the plant. Operating within controlled parameters, the plant itself determines the degree of irrigation required. Various sensors, tensiometers, relative humidity sensors, rain sensors, temperature sensors control the irrigation scheduling. These sensors provide feedback to the controller to control its operation.

2.2.4 Research Project

The researcher`s efforts combine aspects of control strategies, control systems, embedded systems, microcontrollers and the android application in order to solve the problem An automatic irrigation control system has been designed to facilitate the automatic supply of adequate water from a reservoir to field or domestic crops in all agricultural seasons.

Soil moisture content has been detected using acoustic based techniques. The main purpose of this technique is to measure soil moisture in real time. The technique is based on the degree of saturation of soils. This experiment found that the water wastage is reduced with the moisture content following, depending on the kind of soil being monitored by a soil moisture sensor. As a closed loop control system the soil moisture sensor will feed data to the microcontroller, which is the central processing unit and it gives feedback by taking action onto the irrigation pump by turning it on or off. GSM technology is also adapted and is utilised for giving alerts on the system functionality. Any requirement that is requested to be sent to and from the system will be sent through GSM.

2.2.3 Conclusion

Through the chapter we discussed control systems and the analysis, which is the review of when they are most appropriate. There is open loop control systems and also closed loop control systems which are used to control functionalities of other systems, hence the name control systems. We discussed areas where open loop systems and closed loop control systems can be most utilised. Embedded systems and microcontrollers where also explored to a greater extend and their applications in control systems. In this case, automated systems. Other existing systems can be adapted into the research problem.

REFERENCES

[1] Viacheslav and Adamchuk. (2011). *McGill University*, Canada: Evaluation of Automated Soil pH Mapping (ASAE Paper No. 99-1100)

[2] B. Boman, S. Smith, B. Tullos (2006): *University of Florida*, Gainesville: Control and Automation in Citrus Microirrigation Systems

[3] Hill. (2005): Automatic Irrigations system

[4] M. F. Leroux. (2005). *McGill University*, Canada: Automated Irrigation system

[5] [Online]. www.techzim.co.zw

[6] Shen J., Jingling S., Qiuyan H., Wang S., YangYan (2007), *A Remote Measurement and Control System for Greenhouse Based on GSM-SMS*, IEEE 8th International Conference on Electronic Measurement and Instrument.

[7] Feng Z. (2011), *Research on water-saving irrigation automatic control system based on Internet of things Institute of Information Technology*

[8] Joaqun Gutierrez, Juan Francisco Villa-Medina (2013), *Automated Irrigation System Using a Wireless Sensor,* IEEE Transactions on Instrumentation and Measurement Vol. 63, No. 1 January 2014

[9] Jackson T., Cosh M., Bindlish R., Starks P., Bosch D., Seyfried M., Goodrich D, Moran M., Jinyang D. (2010), *Validation of Advanced Microwave Scanning Radiometer Soil Moisture Products*, IEEE

[10] Rajakumar D., Ramah K., Rathika S., Thiyagarajan G. (2008) *Automation in Micro-Irigation. New Delhi: Technology Innovation Management and Entrepreneurship Information Service.*

CHAPTER 3: RESEARCH METHODS AND TECHNIQUES

3.1 Introduction

Research methods show a systematic theoretical analysis of different methods utilised in the project and also the principles associating with the project. This chapter takes a look at the hardware and software requirements, analysis and designs required to come up with the final project model. Our solution of the research problem can be further clarified by modelling all the ideas behind it. Analysis and design of the system can highlight the abstract view of the direction taken in problem solving. A proper analysis of the existing systems has led to a proper design being made of the system. All this is done in order to produce a system which is relevant in solving the problem identified.

3.2 System Design

The hardware components of the system model consist of:

- Moisture Sensor
- Water Tank level detector
- Arduino Uno development board
- GSM (SIM 900) Modem
- 2 X water pumps
- LCD
- Smart Phone

The Software requirements include:

- Proteus Design Suite for simulation
- Arduino IDE for programming via USB
- Android Studio

The success of the project is mostly anchored on the ability to connect together the hardware given above. An in-depth analysis and description of the system prototype is illustrated. The author utilised block diagrams, flowcharts, and a schematic drawings to illustrate the proposed system.

3.3 System`s Block Diagram

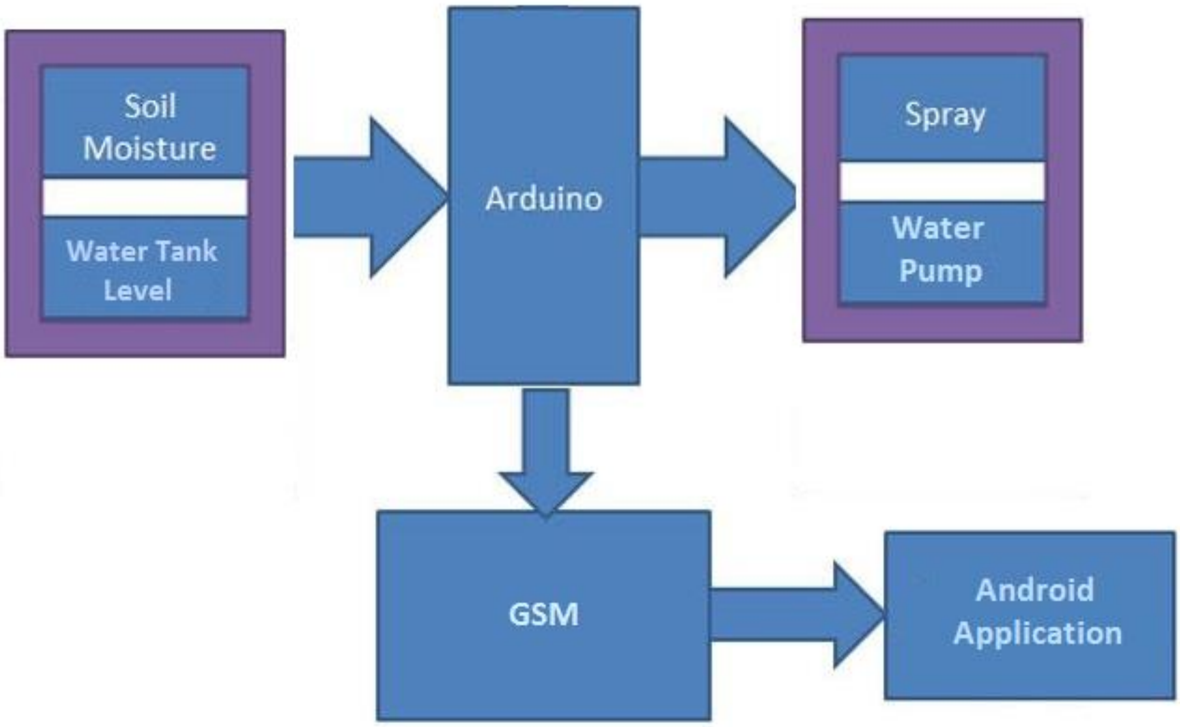

Fig 3.1 System Block Diagram

3.3 Hardware

3.3.1 Arduino Uno

The Arduino Uno controls the Tomatoes Based Android Irrigation System. A SIM900 GSM Modem is plugged into the Arduino board for network connectivity. The SIM900 GSM modem enables the system to transmit SMS messages containing measured soil moisture parameters to the user. The SIM900 GSM enables commands to be passed on to the system through text messages evoked by the android Application. The Arduino gets parameters from the soil moisture sensor in order to trigger the sprinkler. If the soil is wet, the sprinkler is left in the OFF state. If the soil is wet the sprinkler is turned ON. The Arduino also gets parameters from the Water Tank Level Sensor. When the tank gets Empty, the Arduino Turns the Water pump on to fill up the tank. When the water tank is full, the pump is switched OFF.

3.3.2 Soil Moisture Sensor

Soil moisture sensors measure the water content in soil. A soil moisture probe is made up of multiple soil moisture sensors. Detects presence of liquid or moisture between two wire leads and gives active High output. The exposed wire is able to absorb fluid therefore it allows transmission of water vapours into the sensor. These exposed areas are engineered very thinly. Therefore the sensor responds very rapidly to changes in applied moisture, both when being dried that is on process start-up and when called into action if there is moisture ingress into a process. With this sensor the crop field can now interface with the microcontroller, feeding in input into the MCU as there is a unidirectional communication between the two. Soil electrical conductivity is simply measured using two metal conductors spaced apart in the soil except that dissolved salts greatly alter the water conductivity and can confound the measurements. An inexpensive fix is to embed conductors in a porous gypsum block which releases calcium and sulphate ions to swamp the soil background level of ions. Methods for exploiting soil dielectric properties actually measure proxy variables that more or less include a component due to the soil electrical conductivity and are thus inherently sensitive to variations in soil salinity and temperature as well as water. Measurements are also affected by soil bulk density and the proportion of bound and free water determined by the soil type.

Nevertheless, good accuracy and precision can be achieved under specific conditions and some sensor types have become widely adopted for scientific work. In general, conversions from raw sensor readings to volumetric moisture content or water potential using secondary or tertiary methods tend to be sensor or soil specific, affected or precluded at high salinity levels and dependent on temperature.

The soil moisture sensor is pretty straight forward to use. The two large exposed pads function as probes for the sensor, together acting as a variable resistor. The more water that is in the soil means the better the conductivity between the pads will be and will result in a lower resistance.

The Sensor (FC28) is connected in the following manner:

- VCC pin - Arduino’s 5V Output
- GND pin - Arduino’s GND
- DATA pin - Arduino’s Analogue A0

3.3.4 GSM Modem

GSM is a standard developed by the European Telecommunications Standards Institute (ETSI). It was created to describe the protocols for second-generation (2G) digital cellular networks used by mobile phones and is now the default global standard for mobile communications. At present the GSM module is used for remote control activities such as gate control or even temperature control. GSM module consist of GSM modem. A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The modem is the critical part of the component. The module consists of a module powered by a power supply circuit and communication interfaces for a computer. A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities.

The GSM module was connected as follows to interface with Arduino:

- GSM modem TX pin to Arduino’s Rx pin (D0)
- GSM modem RX pin to Arduino’s Tx pin (D1)

Fig 3.2 below shows the connections

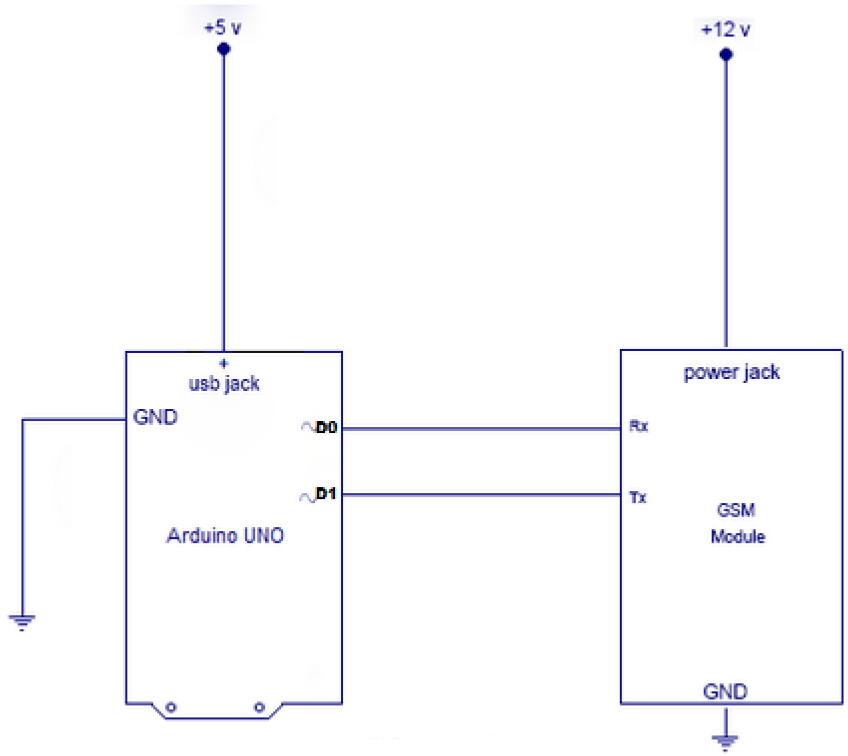

Fig 3.2 GSM SIM 900 Modem and Arduino connection

3.3.5 Liquid Crystal Display Unit

The chosen LCD unit for this System is a 16 * 2 LCD unit because it uses Complimentary Metal-Oxide Semiconductor (CMOS) technology which has low power requirements. It consists of 16 pins; 8 data lines, 3 control lines, 2 power lines, 1 contrast line and 2 pins for back light LED connection. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD such as commands for clearing its screen and setting the cursor position. The data register stores the data, which is ASCII value, to be displayed on the LCD. The connections where done as shown in Fig 3.3 below.

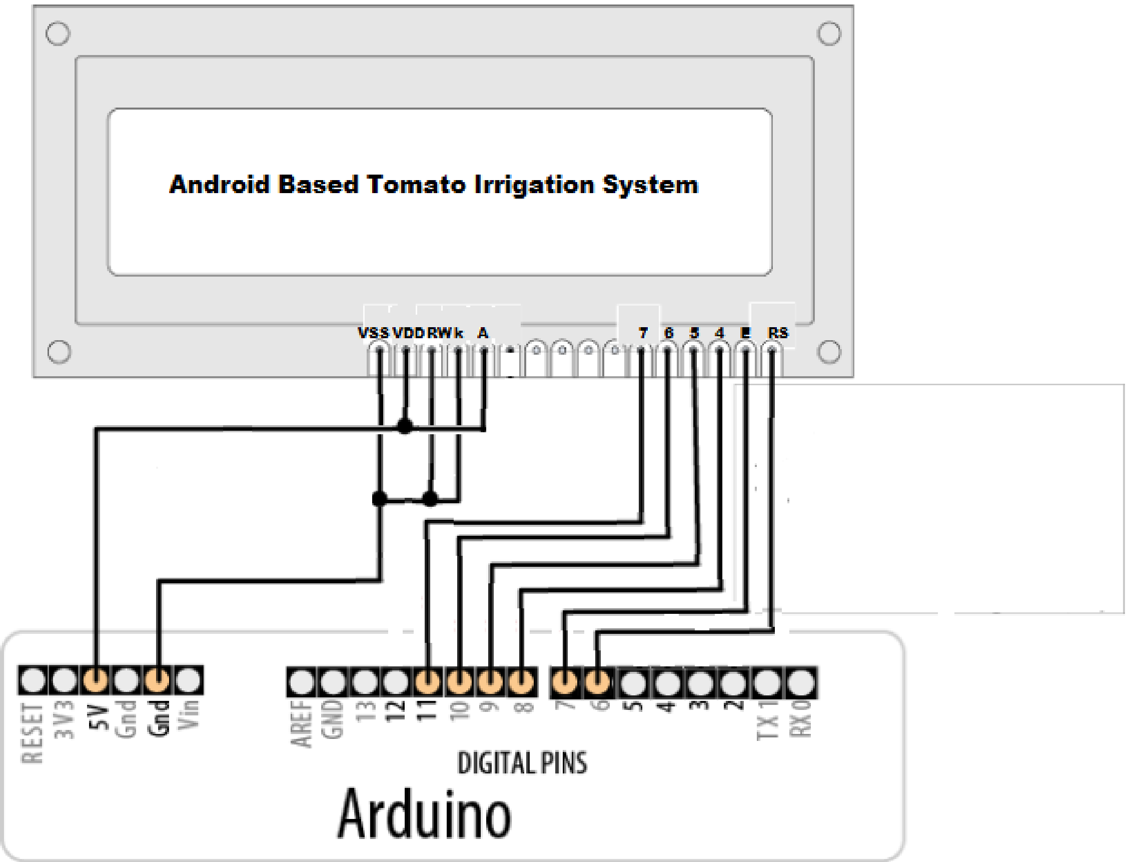

Fig 3.3 Arduino to LCD connections

3.4 Development

3.3.1 Proteus Software

Proteus was used to come up with the circuit diagrams and simulations. Some of the simulations had to be improvised. For example, Water tank levels had to be improvised using buttons, Switching ON water pump and sprinkler using LEDs. Fig 3.4 below illustrates the proteus Design.

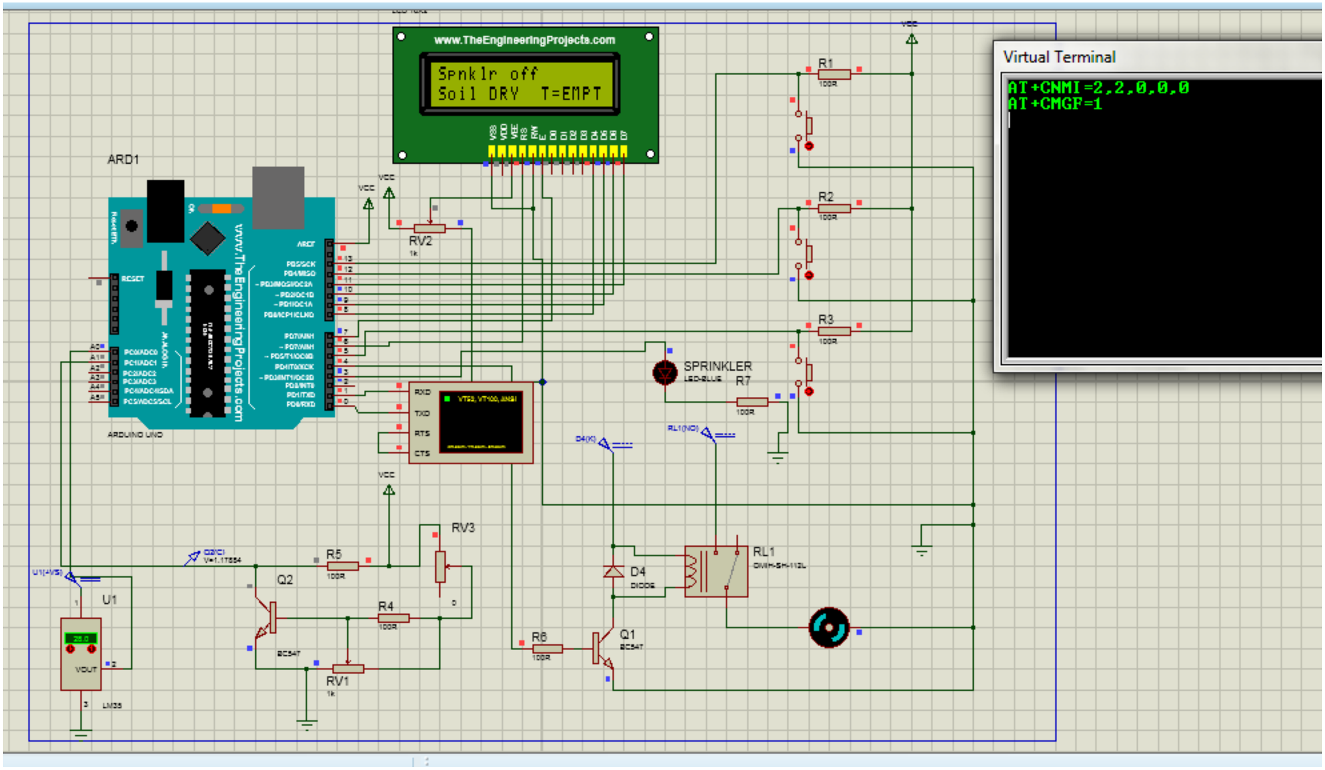

Fig 3.4 Proteus Design and Simulation

3.3.2 Android Studio

Android Studio was used in designing the Android App used to control the system

3.3.3 Bread Board

After Simulation was successful. The components were connected together on the bread board. The board allows for testing of circuit without permanently wiring the componets. It makes is easy for modifications.

3.3.4 Coding

Software development all begins with coding. Assembly language programming was used A typical assembly language program will consist of some assembler directives, sub-routines if they are needed and the main program code.
Assembler directives are a collection of commands that tell the assembler such things as the type of microcontroller being used, its clock speed, etc. They also allow names to be used for memory locations, ports and registers, so making the program more readable.

3.3.5 Programming

After having gone through Assembling and debugging the code was then uploaded into the Arduino Uno.

3.3.6 How the system works

When the soil moisture sensor is injected into the ground the Control system has to be initiated so as to act on the irrigation system (Sprinkler in this case). Our farming ground (field) is one of the actor as we attain input values from it to trigger the irrigation system to act. The moisture sensor works by measuring resistance between its probes. Values were taken and recorded using a multi-meter. These values were when the soil is dry and when the soil is saturated. This gave us a threshold on which to code our program to determine when the soil is dry and when it is wet. When the soil is dry, the system turns ON the irrigation system. When the soil is now wet enough, the system turns OFF the irrigation system. The irrigation system is only turned on provided the tank providing the irrigation water is NOT empty. When the tank is empty, the system turns on the pump to fill up the tank. Once the tank is filled up, the pump is turned OFF. 4 wires were inserted into the tank. One at the very bottom (To complete a circuit), a second one to represent Low Level, a third to represent Half Level and a fourth to represent Full level. The water will complete the circuit by conducting the bottom wire to the other 3 wires. Once a circuit is completed, respective Information is recorded and displayed onto the LCD. When the tank is full the pump stops and when the tank is empty the pump turns on.

The android Application is used to override the whole system. User can issue a command to the system via the Android Application. The Application has a pump button, Status Button and Sprinkler Button. The Application also has Field Tank Level, Soil, Sprinkler and Pump:

- The Pump button is to switch the pump ON/OFF.
- The Sprinkler button is to switch the Sprinkler ON/OFF
- Status Button queries the status of all components and Gives feedback which is displayed in the respective fields mentioned earlier.

The respective buttons on the Application have code embedded in them to send commands as SMS to the GSM module connected on to the Arduino. Fig 3.5 Below shows the system circuit.

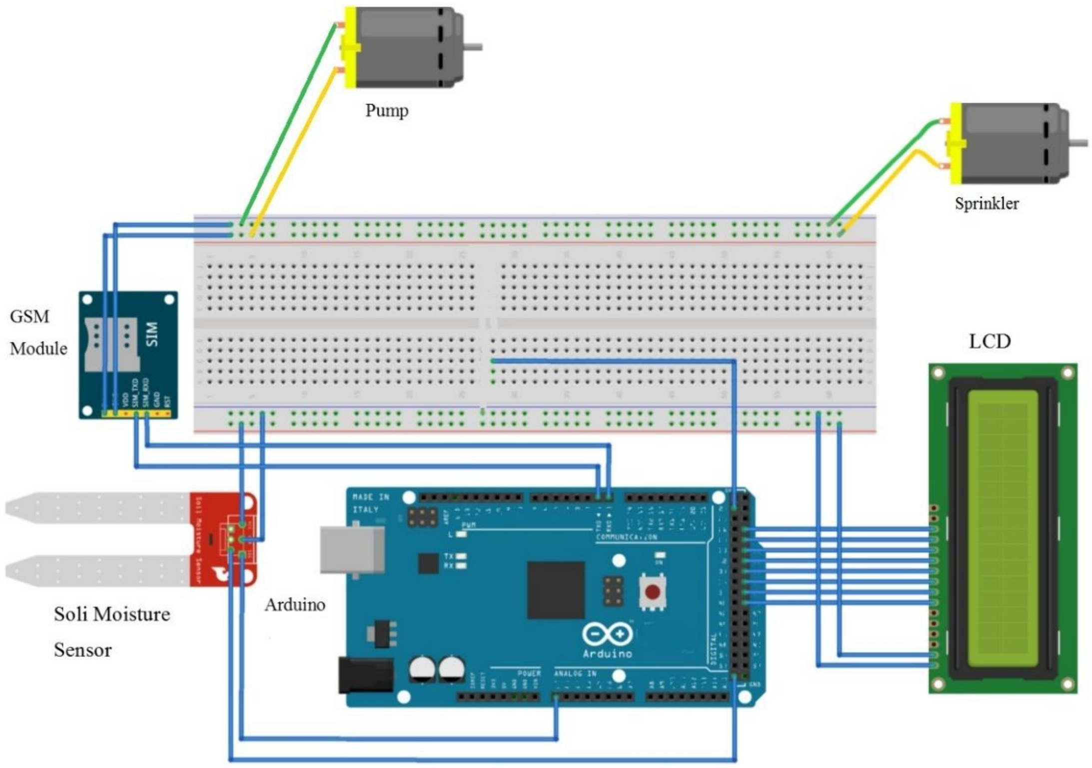

Fig 3.5 System Circuit Diagram

3.3.7 System Flow Chart

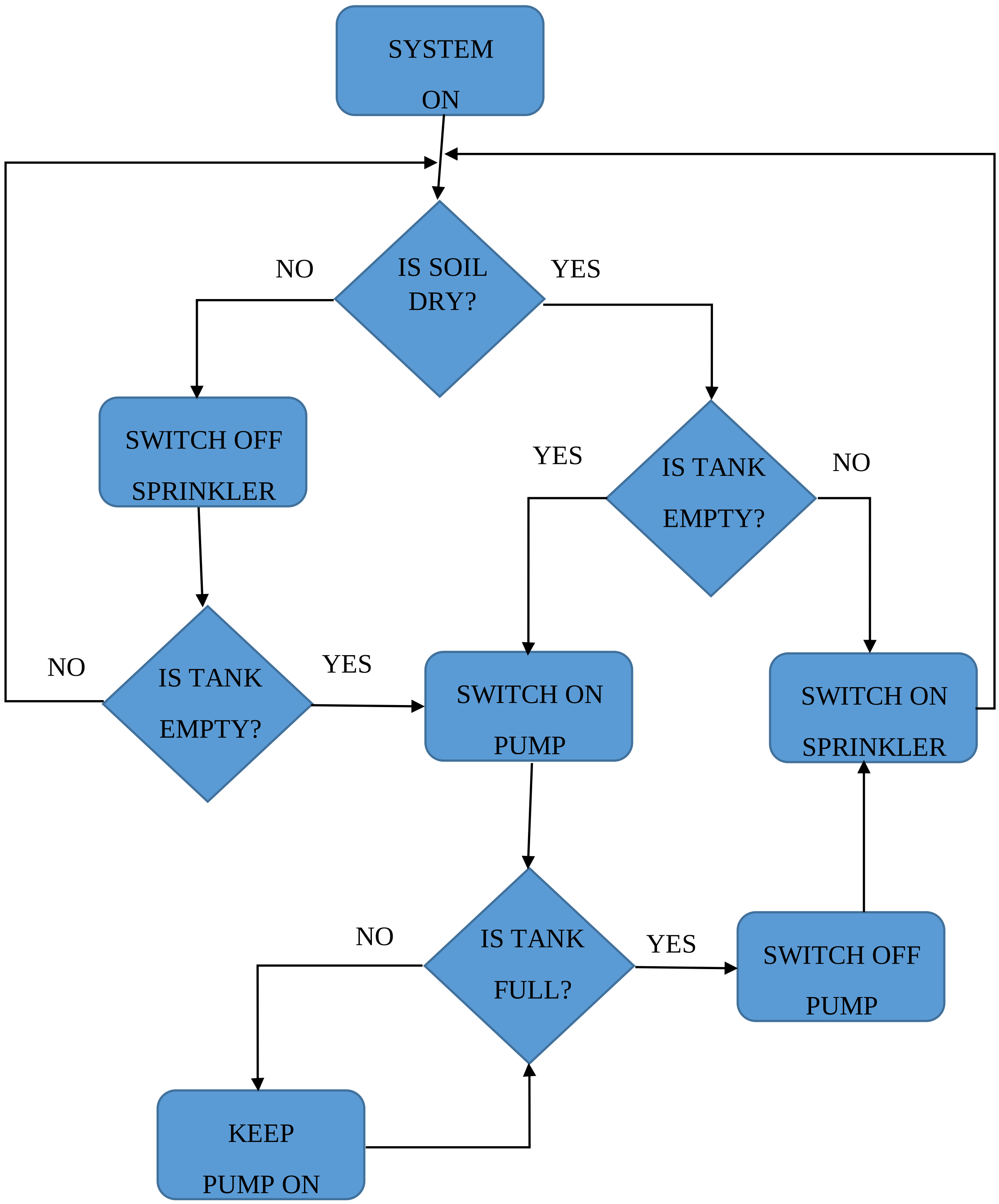

Fig 3.6 System Flow Chart

CHAPTER 4: RESULTS AND ANALYSIS

4.1 Project Layout

Fig 4.1 Project Layout

4.2 Findings

The system managed to start up, giving displaying status of all components on the LCD.

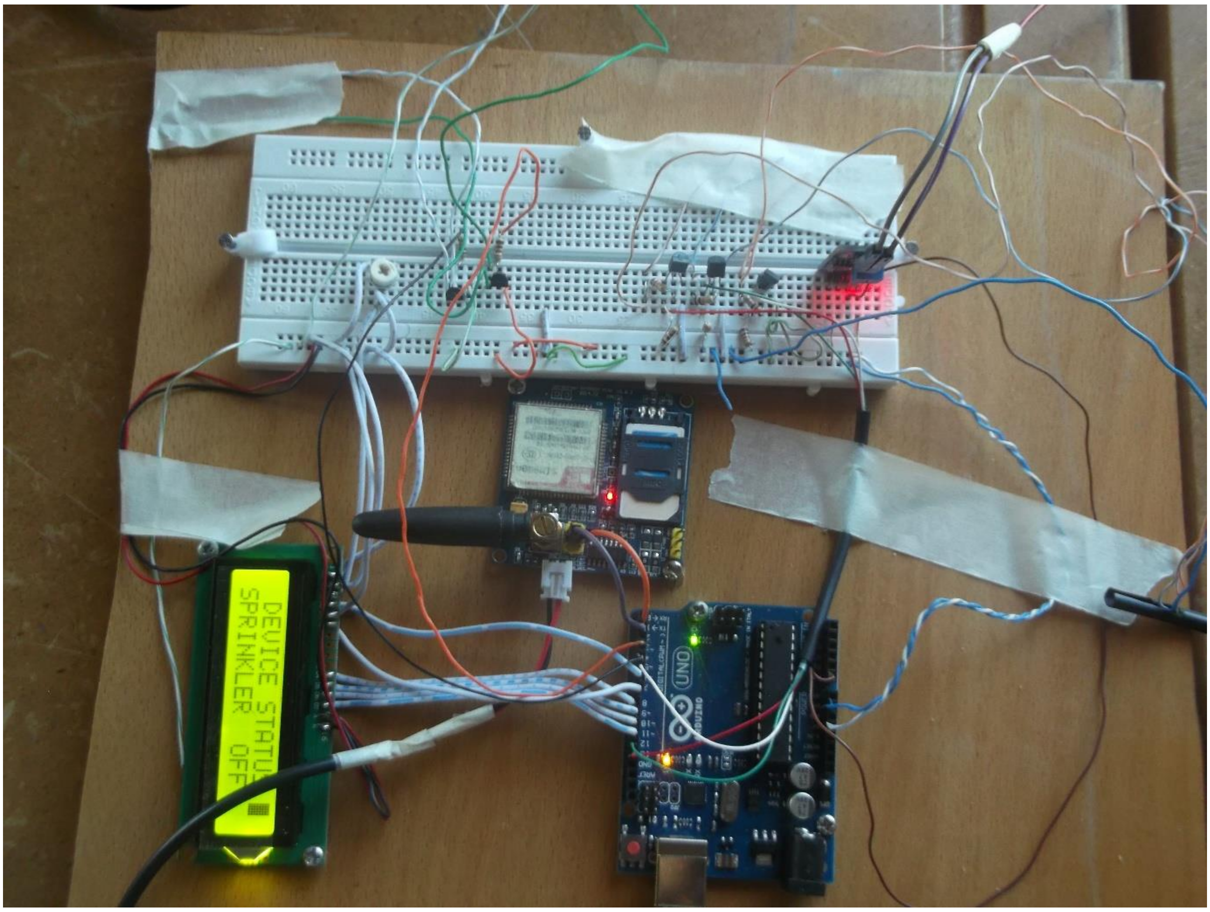

Fig 4.2 Circuit Board

4.2.1 Android Application

Fig 4.3 Phone Screen

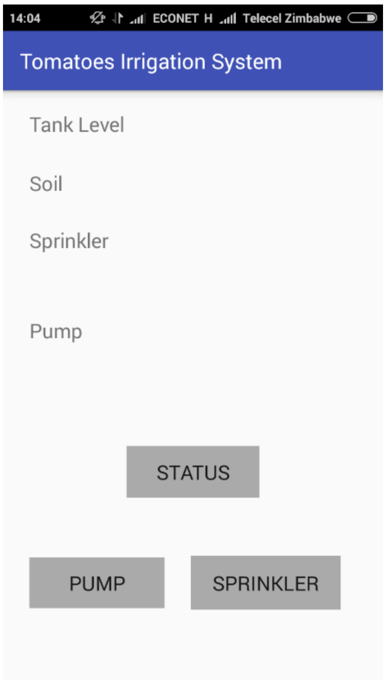

Fig 4.4 Application Window

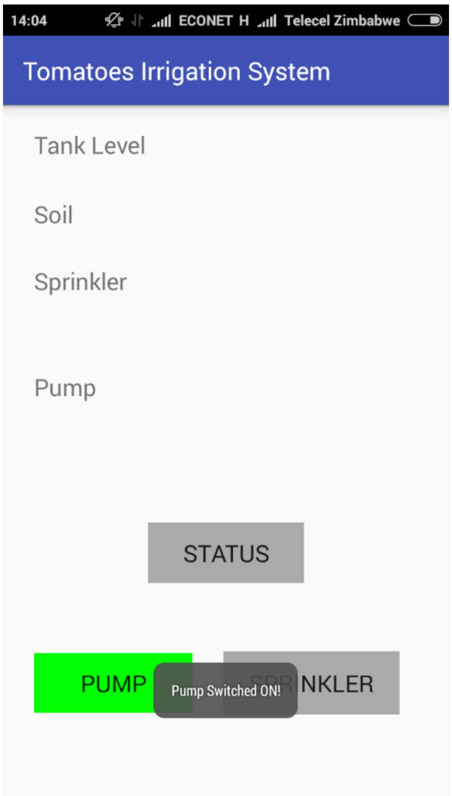

Fig 4.5 Pump Switched ON

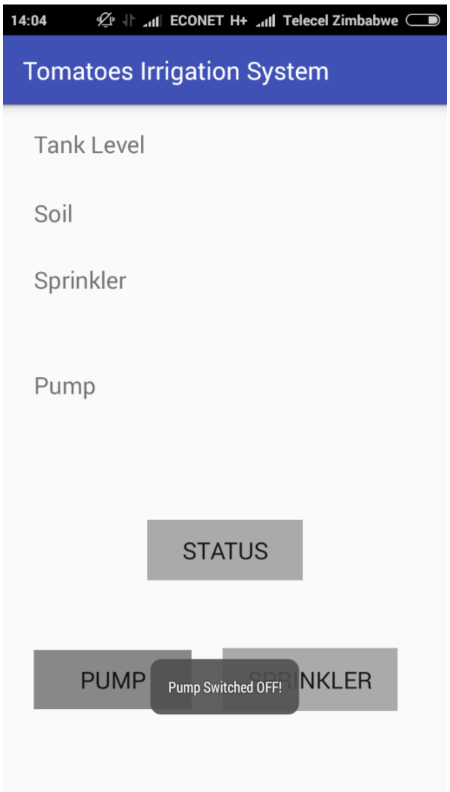

Fig 4.6 Pump Switched Off

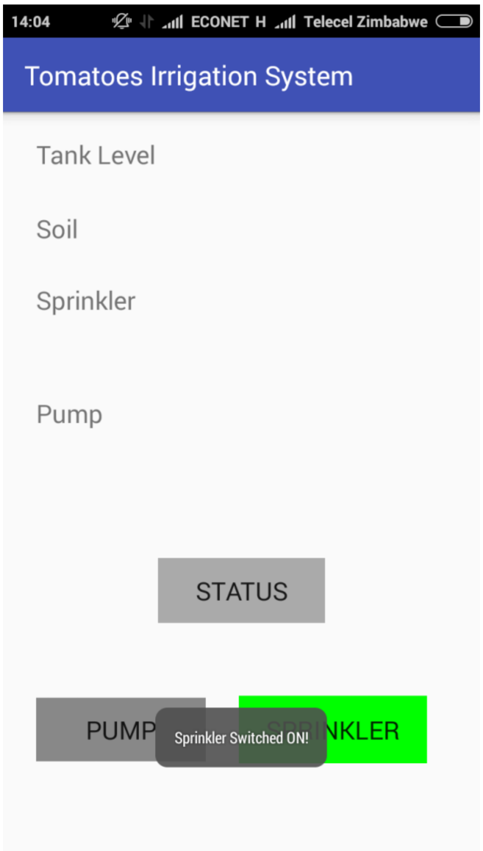

Fig 4.7 Sprinkler Switched ON

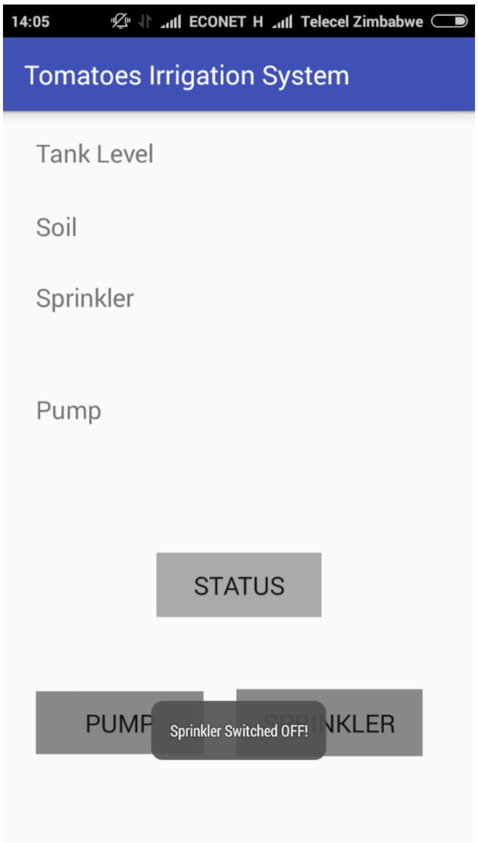

Fig 4.8 Sprinkler Switched OFF

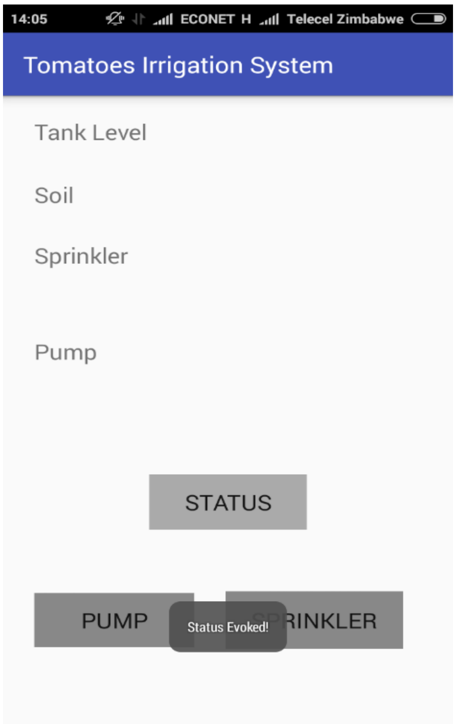

Fig 4.9 Status Evoked

4.2.3 LCD Display

Fig 4.1.0 LCD Display

4.2.4 Water Source and Pump

Fig 4.1.1 Water pump and water source

The container resembles a water source which can be a borehole or a dam. A pump is inserted inside to pump water to the water tank

4.2.5 Water Tank and Sprinkler

Fig 4.1.2 Water tank and sprinkler

The water jug resembles a water tank. The tank has wires inserted inside to measure level of water, from Empty, Low, Half to Full. The pump inside the tank pumps out water to sprinkle the tomatoes.

4.2.6 Soil Moisture Sensor

Fig 4.1.3 Soil Moisture Sensor

4.2.7 Circuit Relays

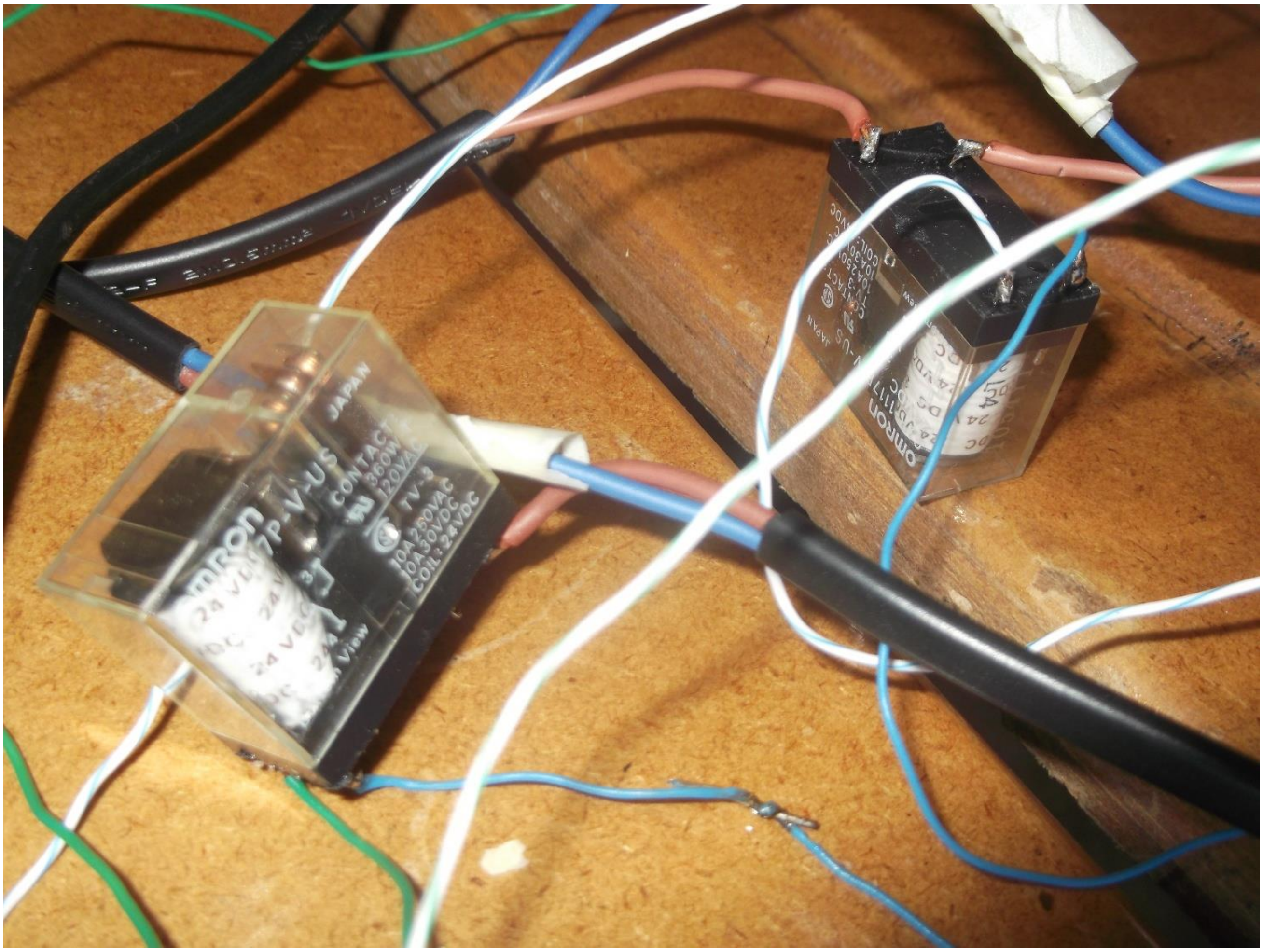

Fig 4.1.4 Circuit Relays

The relays worked in switching ON and OFF of the water pump and the Sprinkler.

4.2.8 GSM Module and Arduino

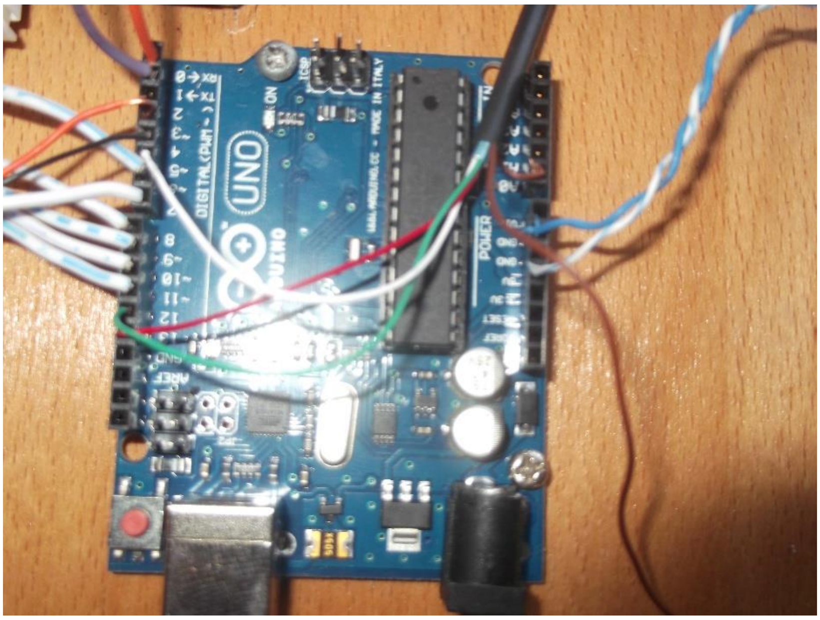

Fig 4.1.5 Arduino Uno

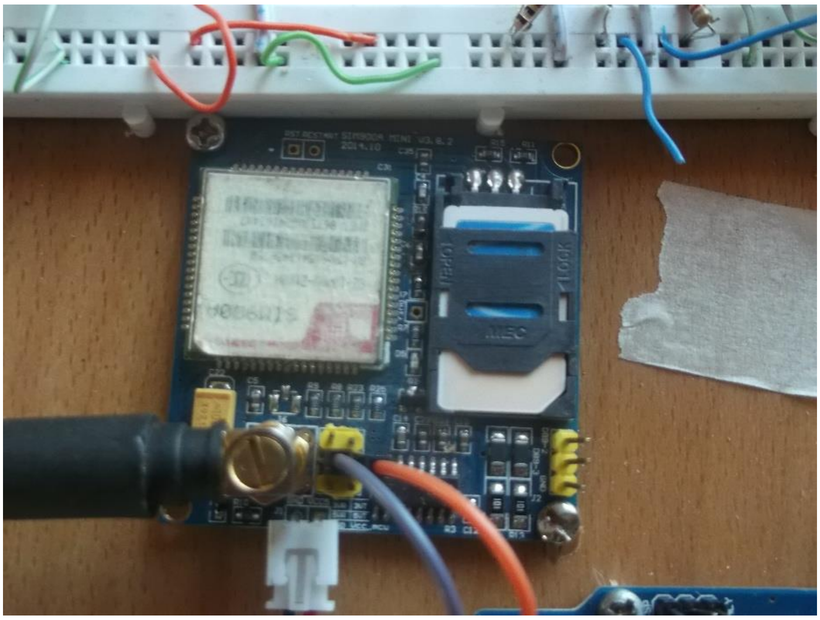

Fig 4.1.6 GSM Module

4.2.3 Analysis of Results

The system performed as expected. User could switch the pump and the sprinkler On and OFF using the Android Application. The Soil Moisture Sensor manager to report when the soil was dry and when it was wet. The tank level wires also manged to indicate status of the water levels. From Empty, Low, Half and Full.

4.2.4 Limitations

Coming up with soil moisture levels proved to be difficult. The figures that were measured on the output from the soil moisture sensor as the soil got wet were varying rapidly. The user had to put a threshold indicating whether the soil is simply dry or wet rather than showing the actual soil moisture percentage levels.

CHAPTER 5: CONCLUSIONS

5.1 Conclusion

The Android Based Tomato Irrigation system was designed and implemented successfully. The project was a success since it met all the specified objectives in the proposal. The system is flexible to incorporate other functions such Temperature sensors, Humidity sensors, plant health sensors and all the other farming monitoring tools.

5.2 Recommendations

Further Improvements can be made to the system:

- Adding Temperature, humidity, amongst other sensors.
- Adding more graphics on the android application, for example as the pump is running, motion icons can be added on to the application. Water tank levels and soil moisture levels can be shown graphically on the android application.
- The system can be further improved to make it accessible over the internet as well and not just sticking to GSM.
- The system can also be further improved by making it flexible to allow the user to enter moisture content levels to be measured and addressed.

APPENDIX

Arduino Code

```
#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

SoftwareSerial gsm(0,1);

LiquidCrystal lcd(6, 7, 8, 9, 10, 11);

//Defining Pins on arduino

#define pumpPin 3

#define tlevel1 13

#define tlevel2 12

#define tlevel3 5

#define moisture A0

#define sprinklerPin 4

//Declairing soil moisture thresholds

const int thrsh = 450;

const int satu = 409;

float moist ;

int i=0;

String incomingString="", device="";

char character,buf[100]="";

boolean pumpStatus = false, sprinklerStatus = false;

long previous_millis=0;


void setup()

{

 Serial.begin(9600);

 lcd.begin(16,2);

 gsm.begin(9600);
```

```
  pinMode(pumpPin,OUTPUT);
  pinMode(sprinklerPin,OUTPUT);
  pinMode(tlevel1,INPUT);
  pinMode(tlevel2,INPUT);
  pinMode(tlevel3,INPUT);
  digitalWrite(pumpPin,LOW);
  digitalWrite(sprinklerPin,LOW);
//LCD Startup display
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" ANDROID BASED ");
  lcd.setCursor(0,1);
  lcd.print(" IRRIGATION SYS ");
  delay(2000);
  Serial.println("AT+CMGF=1");
 }

//Displaying status of irrigation system
void send_Status()
{
gsm.println("AT+CMGF=1");
 gsm.println("AT+CMGS=\"+263783268324\"\r");
//Pump status
gsm.print("PUMP ");
 if(pumpStatus==true)
 gsm.println("ON \n");
 if(pumpStatus==false)
 gsm.println("OFF \n");
//Sprinkler status
gsm.print("SPRINKLER ");
```

```
 if(sprinklerStatus==true)
 gsm.println("ON \n");
 if(sprinklerStatus==false)
 gsm.println("OFF \n");
//Tank level status
gsm.print("TANK LEVEL ");
 if(tlevel1==false)
 gsm.println("FULL \n");
 if(tlevel2==false)
 gsm.println("HALF \n");
 if (tlevel3==false)
 gsm.println("LOW \n");
 if (tlevel3==true)
 gsm.println("EMPTY \n");
//Soil moisture status
gsm.print("SOIL");
float moist = analogRead(moisture);
 if(moist > thrsh)
 gsm.println("DRY\n");
 digitalWrite(sprinklerPin,HIGH);
 if(moist < thrsh)
 gsm.println("WET\n");
  gsm.println((char)26);
 gsm.println("AT+CNMI=2,2,0,0,0");
 }

void loop()
{
top:
 lcd.begin(16,2);
```

```
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(" DEVICE STATUS ");
gsm.println("AT+CNMI=2,2,0,0,0");
 while(1)
 {
incomingString="";
 digitalWrite(pumpPin,pumpStatus);
 digitalWrite(sprinklerPin,sprinklerStatus);
 i=0;
 while(i<4){
 if(millis()-previous_millis > 2000)
 {
 previous_millis = millis();
if(i==0){
 lcd.setCursor(0,1);
 lcd.print("PUMP       ");
 if(pumpStatus==true)
 lcd.print("ON ");
 if(pumpStatus==false)
 lcd.print("OFF ");}
if(i==1){
 lcd.setCursor(0,1);
 lcd.print("SPRINKLER  ");
 if(sprinklerStatus==true)
 lcd.print("ON ");
 if(sprinklerStatus==false)
 lcd.print("OFF ");
 lcd.print("                      ");}
 if(i==2){
```

```
 lcd.setCursor(0,1);
 lcd.print("TANK LEVEL ");
 if(digitalRead(tlevel1)==LOW)
 lcd.print("FULL ");
 else if(digitalRead(tlevel2)==LOW)
 lcd.print("HALF ");
 else if (digitalRead(tlevel3)==LOW)
 lcd.print("LOW    ");
 else if (digitalRead(tlevel3)==HIGH)
 lcd.print("EMPTY   ");}
if(i==3){
 float moist = analogRead(moisture);
 lcd.setCursor(0,1);
 //lcd.print(moist);
 lcd.print("SOIL      ");
 if(moist > thrsh)
 lcd.print(" DRY      ");
 else if(moist < thrsh)
 lcd.print(" WET      ");
  if(moist  < satu) //cut power to pump is soil is saturated
 digitalWrite(pumpPin,LOW);
 }
 i++;
  }
if(Serial.available())
 {
 if(Serial.read()=='*')
 {
 while(!Serial.available()){}
 incomingString += Serial.readStringUntil('#');
```

```
 }

 }

/*When user presses PUMP ON/OFF, Sprinkler ON/OFF on android app. The app sends a text msg to gsm module and evokes and action*/

if(incomingString.equals("PUMP ON")) { lcd.setCursor(0,0); lcd.print(" CMD RECEIVED "); lcd.setCursor(0,1); lcd.print(" PUMP  ON       ");  pumpStatus = true; goto top;}

 if(incomingString.equals("SPRNK ON")) { lcd.setCursor(0,0); lcd.print(" CMD RECEIVED "); lcd.setCursor(0,1); lcd.print(" SPRNKLER ON    ");  sprinklerStatus = true; goto top;}

 if(incomingString.equals("PUMP OFF")) { lcd.setCursor(0,0); lcd.print(" CMD RECEIVED "); lcd.setCursor(0,1); lcd.print(" PUMP OFF     ");  pumpStatus = false; goto top;}

 if(incomingString.equals("SPRNK OFF")) { lcd.setCursor(0,0); lcd.print(" CMD RECEIVED "); lcd.setCursor(0,1); lcd.print(" SPRINKLR   OFF  ");  sprinklerStatus = false; goto top;}

if(incomingString.equals("STATUS")) { lcd.setCursor(0,0); lcd.print(" CMD RECEIVED "); lcd.setCursor(0,1);lcd.print(" SENDING STATUS ");send_Status(); goto top;}

 }

 }

}
```

Android Application Code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bansoft.sendsms"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />

    <uses-permission android:name="android.permission.SEND_SMS" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Tomatoes Irrigation System"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".AndroidIrrigationSys">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

```
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android Application Code – Interface

```
package com.bansoft.sendsms;

import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.telephony.SmsManager;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class AndroidIrrigationSys extends AppCompatActivity {

    Button pumpbtn,sprnkbtn,statusbtn;
    TextView pumptxtv,sprnktxtv,soiltxtv,tleveltxtv;
    String phoneNumber = "+263783268324";
    Integer i=1;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_android_irrigation_sys);

        pumpbtn = (Button) findViewById(R.id.pumpbtn);
        sprnkbtn = (Button) findViewById(R.id.sprnkbtn);
        statusbtn = (Button) findViewById(R.id.statusbtn);

        statusbtn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                //String phoneNo = textPhoneNo.getText().toString();
                String statussms = "*STATUS#";

                try {
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phoneNumber, null, statussms, null,
null);
                    Toast.makeText(getApplicationContext(), "Status Evoked!",
                            Toast.LENGTH_LONG).show();
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(),
                            "NetworkError, please try again later!",
                            Toast.LENGTH_LONG).show();
                    e.printStackTrace();
                }

            }
        });
```

```
        pumpbtn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                if(i==2){

                    pumpbtn.setBackgroundColor(Color.GREEN);

                    String pumpONsms = "*PUMP ON#";


                    try {
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(phoneNumber, null, pumpONsms,
null, null);
                        Toast.makeText(getApplicationContext(), "Pump Switched
ON!",
                                Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(),
                                "NetworkError, please try again later!",
                                Toast.LENGTH_LONG).show();
                        e.printStackTrace();
                    }
                    i=1;

                }

                else if (i==1){

                    pumpbtn.setBackgroundColor(Color.GRAY);

                    String pumpOFFsms = "*PUMP OFF#";


                    try {
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(phoneNumber, null, pumpOFFsms,
null, null);
                        Toast.makeText(getApplicationContext(), "Pump Switched
OFF!",
                                Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(),
                                "NetworkError, please try again later!",
                                Toast.LENGTH_LONG).show();
                        e.printStackTrace();
                    }
                    i=2;


                }



            }
        });

        sprnkbtn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                if(i==2){

                    sprnkbtn.setBackgroundColor(Color.GREEN);
```

```
                    String sprnkONsms = "*SPRNK ON#";


                    try {
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(phoneNumber, null, sprnkONsms,
null, null);

                        Toast.makeText(getApplicationContext(), "Sprinkler Switched
ON!",

                                Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(),
                                "NetworkError, please try again later!",
                                Toast.LENGTH_LONG).show();
                        e.printStackTrace();
                    }
                    i=1;

                }


                else if (i==1){

                    sprnkbtn.setBackgroundColor(Color.GRAY);

                    String sprnkOFFsms = "*SPRNK OFF#";


                    try {
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(phoneNumber, null, sprnkOFFsms,
null, null);

                        Toast.makeText(getApplicationContext(), "Sprinkler Switched
OFF!",

                                Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(),
                                "NetworkError, please try again later!",
                                Toast.LENGTH_LONG).show();
                        e.printStackTrace();
                    }
                    i=2;

                }

            }
        });
    }
}
```