DEVELOPMENT OF LOW COST SOIL MOISTURE SENSOR AND WIRELESS SYSTEM NETWORK FOR CROP IRRIGATION IN GIS

By

Sibghat Ullah

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in GIS and Remote Sensing

Institute of Geographical Information Systems

School of Civil and Environmental Engineering

National University of Sciences & Technology

Islamabad, Pakistan

(March 2012)

CERTIFICATE

Certified that the contents and form of thesis entitled "**DEVELOPMENT OF LOW COST SOIL MOISTURE SENSOR AND WIRELESS SYSTEM NETWORK FOR CROP IRRIGATION IN GIS"** submitted by Mr. Sibghat Ullah have been found satisfactory for the requirement of the degree.

Supervisor: ____________________

Dr. Javed Iqbal
HoD
IGIS, SCEE, NUST

Member: ________________________

Dr. James M McKinion , Electronic Engineer,
USDA-ARS, USA

Member:________________________

Sir Saad Saleem Bhatti
IGIS, SCEE, NUST

Member: ________________________

Dr. Shareef Ullah
SEECS, NUST

DEDICATION

To My Parents & Siblings without whose moral support and prayers it would not have been possible to complete this research in time

ACKNOWLEDGEMENTS

Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis. I express my sincere thanks to my respected supervisor Prof Dr. Javid Iqbal for his worthy guidance in preparing this report and her kind and valuable supervision on all the practical work performed. I have always looked up to his for guidance in any problem I faced in my work and I consider myself privileged to work under his supervision which always prove to me with cogent vision of exploring things and ideas.

I greatly acknowledge Brig *(*Retired*)* Sarwar Jamal, Associate Principal Institute of Geographical Information Systems (IGIS), NUST for providing a platform to utilize my skills in the research work. I would like to thank all the faculty members, non-teaching staff and my fellow students for the help provided to me at various stages during this research work.

It is pleasure to thanks all my friends, colleagues and family members, particularly my parents for their sincere attitude, support and prayers which inspired me every moment towards successful completion of my tasks and attainment of progress in life.

Sibghat Ullah

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

GPS	Global Positioning System
GIS	Geographical Information System
WSN	Wireless Sensor Network
GNP	Gross Nation Product
ADC	Analog-to-Digital
GND	Ground
RF	Radio Frequency
Tx	Transmitter
Rx	Receiver
GUI	Graphical User Interface
UART	Universal Asynchronous Receiver/Transmitter
SDSS	Spatial Decision Support System
IR	Irrigation Water Requirement
IOT	Irrigation Operating Time
VRI	Variable Rate Irrigation
V	Voltage

ABSTRACT

Pakistan is water scarce country, with limited water resources. About 96% of the total available water is used for agricultural purposes, leaving 2% for domestic and another 2% for industrial use. The agriculture sector contributes about 25% of the Pakistan's GNP. Farmers in Pakistan receive their share of irrigation waters on a rotational basis. To protect their water, the farmers are using more than the optimum quantity of water required for healthy crops. Keeping in view the scarcity of water a project was designed with the objectives to (i) develop a soil moisture sensor and (ii) wireless Sensors Network (WSN) in determining the adequacy of water level for ideal irrigation system for crops in Pakistan. Perfection and precision brings heavy costs to the developers of such technologically advanced systems. The greater the sophistication of the system, the greater is the associated cost. An effort has been made to bring down the cost of this sophistication to the end user enabling not only larger economic output but also embedding different techniques and ideas of soil moisture and temperature sensors for better decision making using GIS, each time it is required and for every situation. The results have less human involvement thereby reducing errors. Also saving on time and cost will increase the use of proposed system for watering the less appropriate lands for better crops yield. GIS based standalone application that will enable agricultural professionals using wireless technology to collect the data from crop field ,transferring it to the computer thereby generating soil moisture and temperature map of the area and designing of variable rate irrigation system by using GIS techniques.

Chapter 1

INTRODUCTION

1.1 BACKGROUND INFORMATION

Pakistan is an agricultural country and agriculture is a vital sector of Pakistan's economy. The sector directly supports three-quarters of the country's population, employs half the labor force**,** and contributes towards a large share of foreign exchange earnings. The key to a much-needed improvement of productivity lies in a more efficient use of resources, principally land and water. Pakistan depends on one of the world's largest irrigation systems to support agricultural production (encyclopedia, 2010). The quality of irrigation system, on which Pakistan depends for about one third of its water supplies, is deteriorating. Water allocation policies continue to pour excess water into areas where it is not needed which only serves to raise the water table further and brings more salt to the once fertile topsoil. The supply of water in our rivers remains irregular during the year. The old methods of irrigation with flood water are still being used by the farmer who wastes about 50 to 60 percent of water. This wastage of water in natural flooding is mainly because of unbalanced water distribution to crops World Bank, (2006).

To overcome this problem, sprinkler form of irrigation was being introduced. The use of water sprinklers has revolutionized the irrigation of fields. Widespread domestic use of sprinklers began in the mid-20th century, and the sprinkler continues to help business and homeowners to keep grass watered, especially during times of drought. Fields could now be watered automatically and sprinklers can be installed in large field. These systems replaced the older methods of irrigation Lee, (2009). Water distribution through sprinkler is similar to natural rainfall. Water is distributed through a system of pipes usually by pumping. It is then sprayed into the air through sprinklers so that it breaks up into small water drops which fall to the ground. The sprinklers system enables a uniform application of water over the fields. It is adaptable to any farmable slope, whether uniform or undulating. Sprinklers are also adaptable to most soils and are hence lately being widely used to enhance productivity.

1.2 GEOGRAPHICAL INFORMATION SYSTEM (GIS) APPLICATION IN AGRICULTURE

Efficiency in the agricultural sector can be augmenting effectively by using Information Technology tools such as remote sensing and Geographical Information System (GIS). It can ensure greater reliability of estimates and forecasting that will help in the process of planning and policymaking. Efforts to improve and harness latest remote sensing and information technology techniques to capture, collate, add value and disseminate data into appropriate destinations are being helpful for managing risk and in accelerating the growth process.

A Geographic Information System (GIS) is a system for capturing, storing, analyzing and managing data and associated attributes which are spatially referenced to the earth. It is computer software that provides data storage, retrieval, and transformation of spatial (field) data. GIS software for digital agriculture will store data, such as soil type, nutrient levels, etc, in layers and assign that information to the particular field location. A fully functional GIS can be used to analyze characteristics between layers to develop application maps or other management options Wallace, (2000).

Spatial analysis, the study of geographic features, and the relationships that exist between them can be applied to many areas of the agriculture industry. By better understanding how features within the landscape interact, decision makers can optimize operational efficiency and improve economic returns. Regardless of scale, whether at the farm field level analyzing crop yield information or across an entire country, GIS is becoming fully integrated and widely accepted system for assisting government agencies to manage programs that support farmers. Preserving the environment, managing assets, constant overproduction with low prices, changing government subsidies, biotechnology, and intense global competitions are reasons agricultural professionals are using geographic information system (GIS) technology Cyber-Swift, (2007).

The development in the field of computer technology has given new direction to handling and using spatial data for assessment, planning and monitoring. Precision Agriculture or Precision Farming offers the potential to automate and simplify the collection and analysis of information by using GIS and other technologies. It is the art and science of using new advanced technologies for optimum profitability, sustainability, protection of land resources fields and to enhance crop production. It provides information to make better management decisions and to identify, analyze and manage variability within fields by doing all practices of crop production in right place at right time and in most appropriate way. In order to gather and use information effectively, it is important for anyone considering precision farming to be familiar with the technological tools by combining the Global Positioning System (GPS) and Geographic Information Systems (GIS).These technologies enable the coupling of real-time data collection with accurate position information, leading to the efficient manipulation and analysis of large amounts of geospatial data Ayday and Safak, (2009).

1.3 GEOGRAPHICAL INFORMATION SYSTEM (GIS): INTEGRATION WITH WIRELESS SENSOR NETWORK (WSN)

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, soil or pollutants. Recent advances in wireless technologies and electronic systems have facilitated the development of small, low-cost and low-power sensor nodes that communicate in short distances. WSN along with GPS technology for each sampling points in field, collects data of high-precision, geo-spatial information and communicates between measuring instrument and portable data analysis devices or computer with wireless network based technology Cordova-Lope et al., (2007). It also analyzes data of experimental field processing and helps in interpolating data of soil moisture content**.** WSN provides huge amount of data.

Inputs and output balance on a farm is essential to its success and profitability. The capacity of GIS to analyze and visualize agriculture related environments and workflows have proven to be very favorable to the people involved in the farming industry. While natural inputs in farming cannot be controlled, they can be better understood and managed with GIS applications such as crop yield estimates, soil amendment analyses, and erosion identification and remediation ESRI, (2009).

WSN provides very large amount of data. With the integration of the power of WSN with GIS, the data can be obtained in less time and the evaluation time can be greatly reduced. Technological advancements in the electronic fields, like the any other application area, tend to prevent these natural disasters. Data collection, sharing, sending and analyzing has been the most popular and important studies for the last decades so as to prevent a possible natural disaster. Data collection here is the most time consuming and expensive part of these types of studies. Therefore, wireless sensors and WSN are the most popular, helping and suitable devices and methodology for these types of studies Bernhardsen, (2002).

The wireless industry realized the capabilities of GIS very soon by using it for business process automation. It helped the engineers in the estimation of coverage at a site of interest by adopting GIS as a radio wave propagation modeling tool. By the help of these tools the engineers can even plan and optimize the existing network, enabling them in analysis of the result of the potential changes. However, the entire implementation of GIS by WSN is still partial. The benefits of geospatial data analysis are being realized, but the incorporation is making progress even faster Lukas, (2007).

Efficient water management is a major concern in many crop systems. WSN have a big potential for represent the inherent soil variability present in fields with more accuracy than the current systems available. Thus, the benefit for the producers is a better decision support system that allows maximizing their productivity while saving water. Also, WSN eliminates difficulties to wire sensor stations across the field and reduces maintenance cost. Since installation of WSN is easier than existing wired solutions, sensors can be more densely deployed to provide local detailed data. Instead than irrigating an entire field in response to broad sensor data, each section could be activated based on local sensors.

Ruiz-Garcia et al (2009) wrote a review of “State of the Art and Current Trends in Wireless Sensor Technologies and Applications in Agriculture and Food Industry”. The Wireless Sensor Technologies (WST) proves very applicable and productive in environmental monitoring, precision agriculture, cold chain control or traceability etc. WSN can operate in a wide variety of environments and is advantageous in cost, size, power, flexibility and distributed intelligence, compared to wired ones and are very reliable and stable. The use of WSN in precision agriculture increase efficiency, productivity and profitability while minimizing unintended impacts on wildlife and the environment. The real time information from the fields will provide a solid base for farmers to adjust strategies at any time instead of take decisions based in some hypothetical average condition, which may not exist anywhere in the reality. Efficient water management being a major concern in many crop systems, WSN have a big potential here to represent the inherent soil variability present in fields with more accuracy. Thus, the benefit for the producers is a better decision support system that maximizes their productivity while saving water. Also, WSN reduces difficulties to wire sensor stations across the field cutting the maintenance cost. Sensors can be more densely deployed to provide local detailed data since the installation is easy.

The main advantages of WSN for monitoring are its longer reading range, the flexibility and different network topologies that can be configured, the variety of sensors that are already Sensors available and low power consumption. One problem the study noted might be that these monitoring systems create huge volumes of data that are difficult to manage, causing a huge increase in the daily volume of data in a corporate information technology system Morais et al, (2009)

1.4 SOIL MOISTURE AND TEMPERATURE TECHNIQUES

Soil Moisture is the water that is held in the pores between soil particles. Surface soil moisture is the water that is in the upper 10 cm of soil. The amount of moisture found in soil varies greatly with the type of soil, climate and the amount of organic material in that soil. It is used for different scientific and technical purposes, and is expressed as a ratio, which can range from 0 (completely dry) to the value of the materials' porosity at saturation. It can be given on a volumetric or mass (gravimetric) basis. Root zone soil moisture is the water that is available to plants, which is generally considered to be in the upper 200 cm of soil. Soil moisture levels determine timing of irrigation (Gardner, 1979). The different techniques to measure soil moisture are as under:

- Remote sensing
- Gravimeter
- Tentiometer
- Electrical Resistance

A temperature sensor measures the temperature of the soil. Soil Temperature has a significant role in determining the rate of plant growth, and its survival. The temperature in the soil changes greatly with depth, the greater the depth the lower the temperature and vice versa. Soil temperature can be measured for areas receiving direct sunlight as well as for areas in shade.

1.5 RATIONALE

Pakistan is water scarce country, with limited water resources. It is one of the world's most arid countries, with an average rainfall of under 240 mm a year. The population and the economy are heavily dependent on an annual influx into the Indus river system (including the Indus, Jhelum, Chenab, Ravi, Beas and Sutlej rivers) of about 180 billion cubic meters of water, that emanates from the neighboring countries and is mostly derived from snow-melt in the Himalayas. About 96% of the total available water is used for agricultural purposes, leaving 2% for domestic and another 2% for industrial use. By far most water is used for irrigated agriculture, emphasizing the particular significance of agriculture in the country. Farmers in Pakistan receive their share of irrigation waters on a rotational basis. To protect the right of share of their water, the farmers are using more than the optimum quantity of water required for healthy crops. Most of the farmers use the flood (surface) irrigation system. Due to lack of modern irrigation techniques and agricultural practices further add to the wastage of irrigation water. Application of water for irrigating the crop field at the right time and in the right amount is mandatory for consistently high crop yields. For real time applications, it is unavoidable to minimize acquisition time frames, and remove intermediate processes that are typically required to get information from the field into a GIS application. Therefore, a study is required for a Wireless System to be installed where time, money and efforts are minimized enabling correct decision, each time it is required and for every situation.

1.6 SCOPE OF STUDY

During the last century, growth of the earth population has created many environmental problems. Today, we are facing many serious environmental problems. One of the critical problems is the climate change. It influences the human beings indirectly and directly. Global warming (a result of the climate change) is causing a decrease in water supplies which in turn affects the agricultural fields. As a result of this water shortage, inadequate and intermittent irrigation pattern is followed which decreases the quality and quantity of the agricultural products. Wireless Sensor Network (WSN) technique is available for agricultural fields and is a very useful and powerful tool for data collection. Sensor nodes are great for deployment in hostile environments or over large geographical areas.

Integration of WSN with the wireless communication technique provide an opportunity for low power consuming, cheap and easily installed techniques to prepare an accurate irrigation scheduling plans. Integration of WSN with GIS creates an easy to understand and visually effective map by using huge amount of data. Application of this technique for agricultural area will control the water wastage; increase the production quality and amount of the plant therefore positively affecting the economy.

This research presents a GIS application that uses soil moisture and temperature to find VRI. The customized GIS application will intend to assist in updating the geographical site status, mapping of the site, develop site specific water applications, and mapping of soil moisture zones assisting them in advanced precision agriculture decision making.

In WSN, a GIS is preferred for site network planning and expansion. The capability to layer information onto the earth's surface, entirely with the non-spatial data, allows experts the distinctive skill to simulate and assess the network from their work stations at the office. This saves important time and reduces the number of visits an agronomist/soil scientist must make to the field. Therefore the use of GIS for storing, managing, analyzing, and visualizing the data becomes a logical choice in WSN due to the large amounts of spatial and non-spatial data associated with the diverse field of WSN.

1.7 OBJECTVES

The objectives of the study were to a) design and develop soil moisture and temperature sensors, b) develop site specific water application to optimize water usage and c) map soil moisture distribution using GIS.

MATERIALS AND METHODS

The study was completed in two main phases i.e. Phase 1: Sensor design and Phase 2: Application development.

2.1 PHASE 1: SENSOR DESIGN

First phase consist of sensor designing and development:

2.1.1 SOIL MOISTURE SENSOR

There are several methods currently used to measure soil moisture content. Some methods and techniques are more precise, expensive and complicated than others. In order to achieve a suitable compromise of complexity, price and accuracy, it is important to analyze the advantages and disadvantages of each technique. All electrical currents are subject to some level of resistance. A conductor is a material which contains movable electric charges. In metallic conductors such as copper or aluminum, the movable charged particles can be very low if the medium is air, ceramic or stone and very high in case of water and soil. Mineral salts contained in water help electrical currents to be conducted through the soil .The moisture content is therefore essential to the level of resistance in that particular soil.

A probe is a device used to measure movement of electron, potentials and random electron currents in plasma. It is mainly consist of one or two small collecting separated electrodes through to which various potentials are applied, examine thoroughly with the corresponding collection currents being measured. Soil moisture sensor has been developed by using two rods of aluminum material held apart at a fixed distance that provide variation in moisture from 100 %(maximum) to 0 %(minimum). The more moisture in the soil the better the conductivity and current can easily flow. When the amount of water changes in the soil, the probe will measure a change in voltage due to change in resistance.

2.1.1.1 Principle Operation of Soil Moisture Sensor

The main principle working of developed sensor is based on "*More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity more poorly (more resistance)*". It means "*sensor functions as a variable resistor*". Movement of water between the soil and the sensor results in changes to the electrical resistance between the probes. Probes are inserted into ground and current is passed through it and electrical resistance can then be converted to soil water potential. As water potential value is in analog form so it must first be converted to digital form using Analog-to-Digital (ADC). In physical world parameters such as temperature, pressure, humidity, and soil moisture are analog signals. To convert these physical quantities into electrical signals we need an analog to digital converter (ADC), which is an electronic device that converts continuous signals (analog) into discrete (digital) representation as shown in fig 2.2. Analog to digital converters are the most widely used devices for data acquisition.

2.1.2 TEMPERATURE SENSOR

Temperature sensor LM35 is used in this research project. It is a three pins integrated-circuit temperature sensors that can be used to measure temperature in Celsius (Centigrade ^{o}C), in which output voltage varies linearly with temperature (^{o}C). It does not require any external source or calibration. The basic equation used to convert output voltage to temperature is:

$$T = Vout * (100\ C^{0}/\mathrm{V}) \qquad 2.1$$

T=Temperature

Vout =Analog Output

V=Input Voltage

This equation shows that temperature will be maximum when Vout / V is equal to 1.

LM 35 consists of three pins, details are as follow:

- Pin 1: Input Voltage(VDD)
- Pin 2: Analogue Voltage Out(VOUT)
- Pin 3: Ground

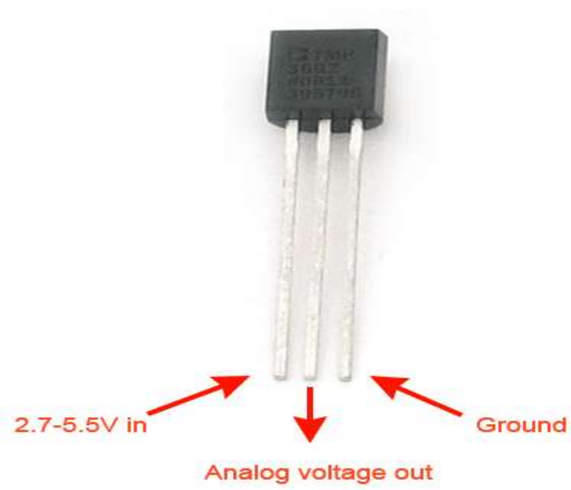


Figure 2.1.Temperature sensor LM35.

Figure 2.2.Getting data from analog world.

2.1.3 MICROCONTROLLER

Microcontroller is a mini computer with CPU, integrated circuits, memory, and programmable input/output peripherals on a small chip. It is mainly used in products that require a degree of control to be exerted by the user. Microcontroller system consists of two main components:

- Hardware
- Software

Microcontroller **Atmel AT89S51** is used in this project due to its cost effectiveness. It is 40 Pin low-powers, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The AT89S51 provides the following standard features: 128 bytes of RAM, 32 I/O lines, two data pointers, two 16-bit timer/counters, two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. The main reason of selecting Atmel AT89S51 for this project is due to its low cost and easily programmable.

Figure 2.2. Atmel AT89S51.

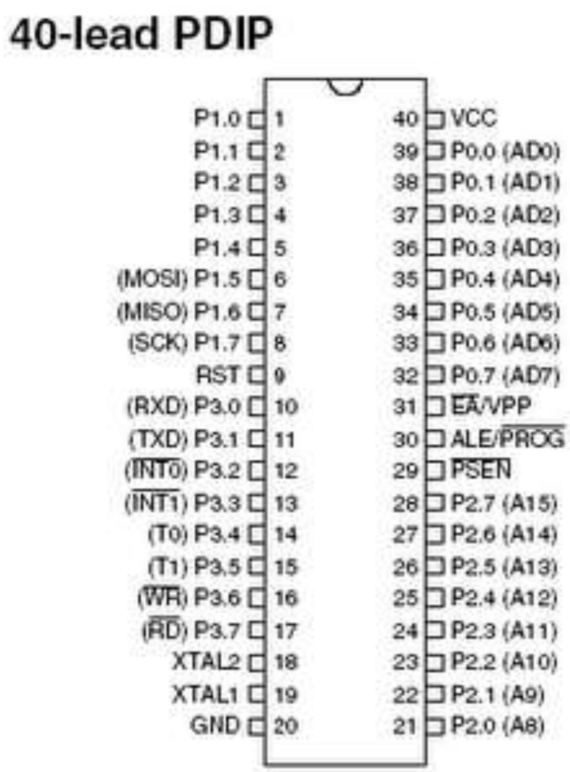


Figure 2. 4. Pin description of atmel AT89S51.

2.1.4 PINS DETAILS

Pins names and functions of microcontroller AT89S51 are as under:

VCC: Supply voltage.

GND: Ground.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

Port 0: 8-bit open drain bidirectional port.

Port 1: 8-bit bidirectional I/O port with internal pull-ups.

Port 2: 8-bit bidirectional I/O port with internal pull-ups.

Port 3: 8-bit bidirectional I/O port.

Port 3 shows the following features:

P3.0: RxD (serial input port)

P3.1: TxD (serial output port)

P3.2: INT0 (external interrupt 0)

P3.3:INT1 (external interrupt 1)

P3.4:T0 (timer 0 external input)

P3.5: T1 (timer 1 external input)

P3.6: WR (external data memory write strobe)

P3.7: RD (external data memory read strobe)

2.1.5 WIRELESS TRANSCEIVER (TRANSMITTER / RECEIVER)

A Wireless Sensor Network (WSN) is a system consists of radio frequency (RF) transceivers, microcontrollers and sensors which are use to communicate and transfer information between different points that are remotely connected within a field. A wireless RF transceiver is a device that consists of transmitter (Tx) and a receiver (Rx) which are combined together and share general circuitry. Radio Frequency (RF) uses radio signals to communicate through larger distances making it suitable for long range. It collects real-time data and transmits it via radio frequency to antennas located throughout the observatory area. The transmitter/receiver (Tx/Rx) pair operates at a frequency of 433 MHz .The transceiver used in this research project is of range 100-150 m in open field as shown in figure 2.5.

2.1.5.1Sensor Node

Sensor nodes are small stand alone embedded devices capable of performing some processes like gathering sensory information and communicating with other connected nodes in the network. The sensor node design consists of temperature and soil moisture sensor**.**

Figure 2. 5.Wireless transceiver of range (100-150) m.

2.1.5.2 Actuator Node

An actuator is a mechanical device use to convert an electrical control signal to a physical action, and constitutes the mechanism by which an agent acts upon the physical environment. It is responsible for actuating a mechanical device (sprinklers) connecting with a solid relay.

2.1.5.3 Sink Node

To collect sensor node data packet wirelessly and transfer it to the connected PC via Serial interface and to send request to actuator packet generated by control/decision. This node acts as a central module and connects each sensor nodes to transmit, process, and retrieve data.

2.1.6 NETWORKING

A star topology is one of the most common network setups where each of the devices and computers on a network connect to a central hub. A major disadvantage of Star topology network is that if the central Master fails, all devices connected to that master will be disconnected. In this project RF master-slave protocol is used, in which system allows communicating several nodes (slave) with the central commanding module known as Master. Master gives the commands and all nodes execute the instructions as shown in fig 2.6. The host is usually master, connected to the computer through serial interfacing, and all the actuators and sensors attached to different nodes are considered as slaves. No communication is allowed between slaves unless requested by the master.

Figure 2.6. Star topology, transceiver connected to computer is master and all other are its slave.

2.1.7 RELAYS

Control of electrical equipment is essential for proper function. A relay is an electrically operated switch that is under the control of another electrical circuit. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions. A relay is able to control an output circuit of higher power than the input circuit. In this application relays are attached to the actuator node for the automation of sprinklers attached to it.

2.2 PHASE 2: APPLICATION DESIGN

Second phase consist of standalone application using Visual Studio:

2.2.1 GRAPHICAL USER INTERFACE (GUI)

One of the tasks is to develop a standalone application which has the capability of retrieving data and storing into a Geo-database by connecting master with serial interfacing to computer. Graphical User Interface (GUI) is the main interface that allows users to interact and control electronic devices and applications. This application has been developed using C# programming language at Microsoft visual studio IDE version 2008 using ArcObjects. Visual studio supports many programming languages like Visual Basic, Visual C++, Visual C# and Visual J#. Since the connection is based on serial port, the Microsoft COM Component in Visual Studio is used to add a serial communication to facility. ArcObjects open development environment of the ArcGIS software 9.3.1 from (Environmental Systems Research Institute, Redlands, CA) was chosen as the development platform for this application. ArcObjects can be integrated with any Microsoft component object model (COM) compliant programming languages to develop customized GIS applications Zeiler, (2001).

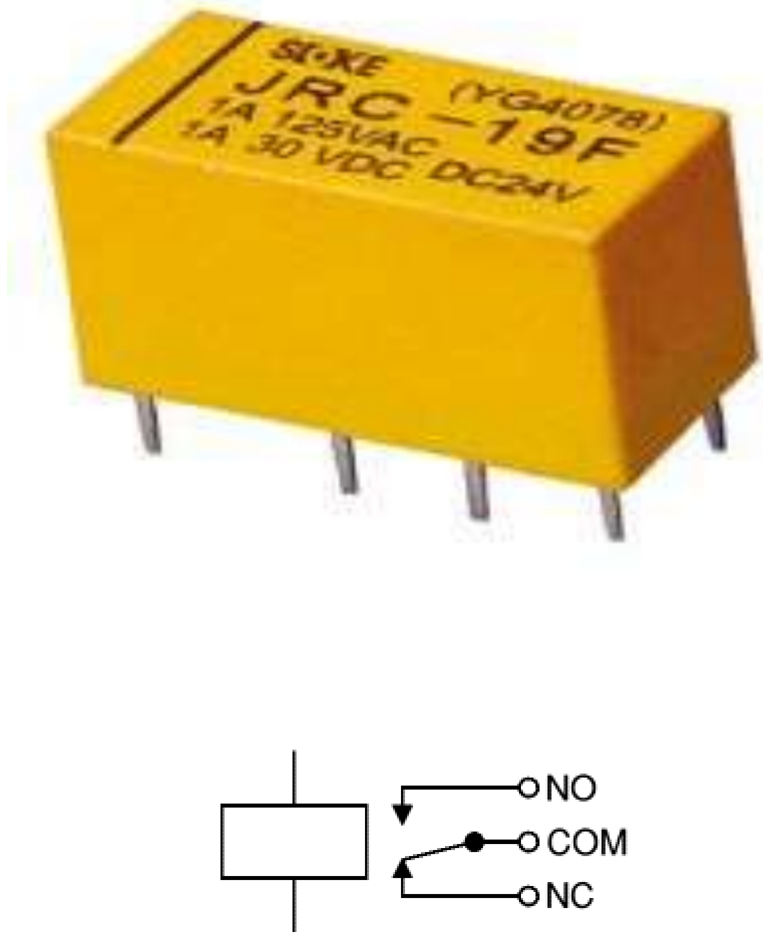


Figure 2. 7. Solid relay for automation.

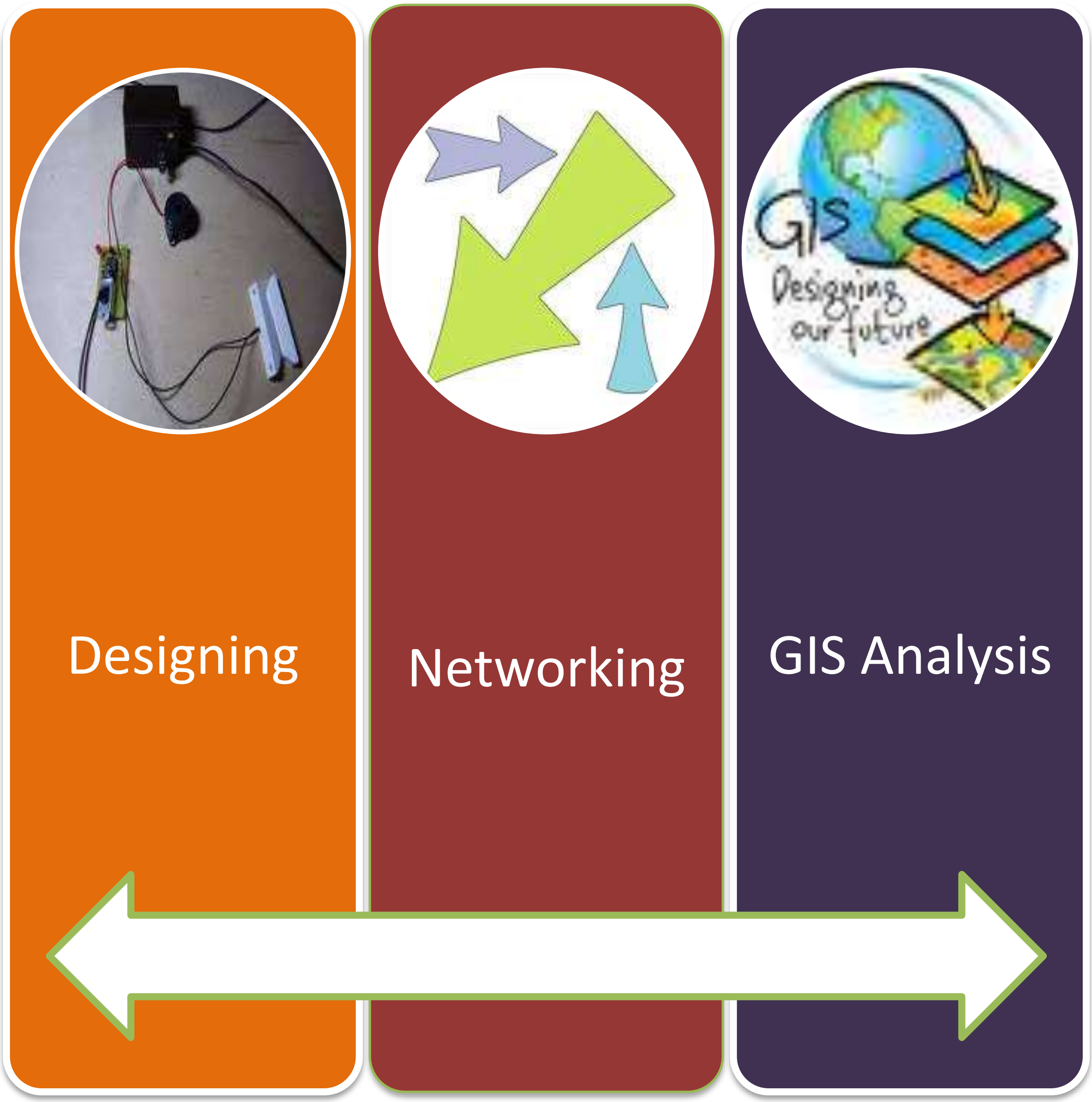


Figure 2. 8. Three Phases of research methodology; designing, networking and gis analysis.

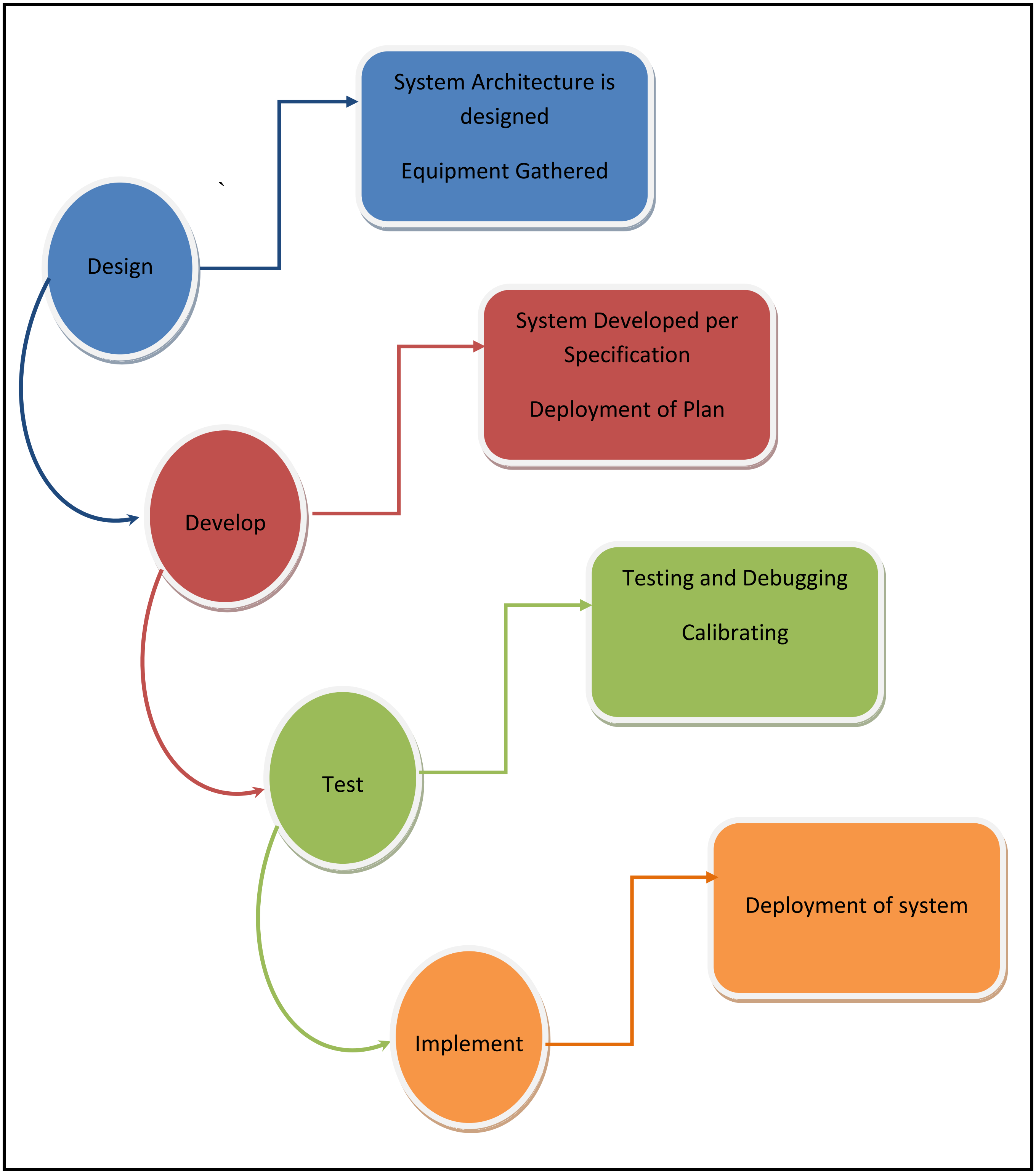


Figure 2.9. System architecture flow control.

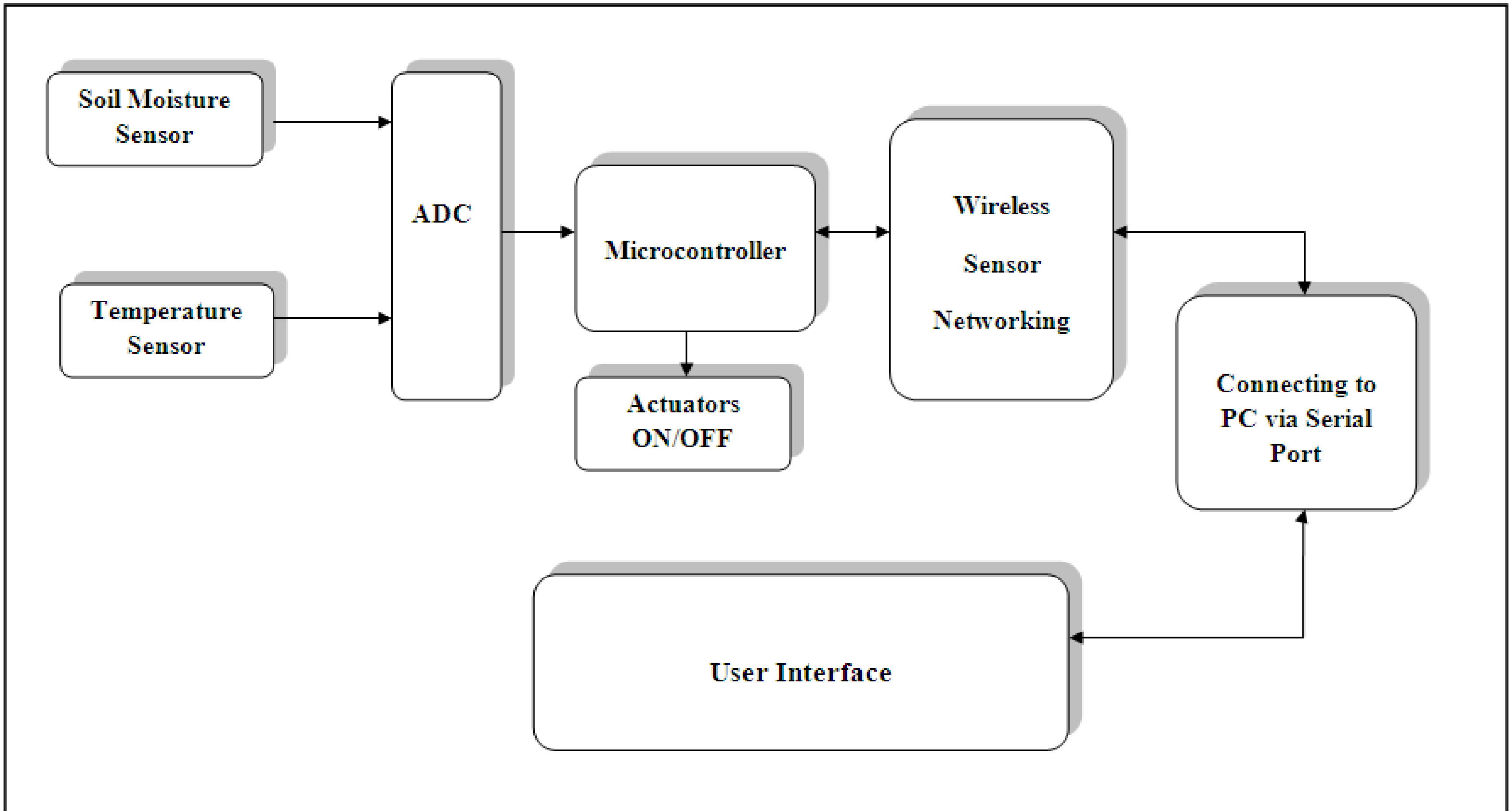


Figure 2. 10. Architectural view of soil moisture and temperature system.

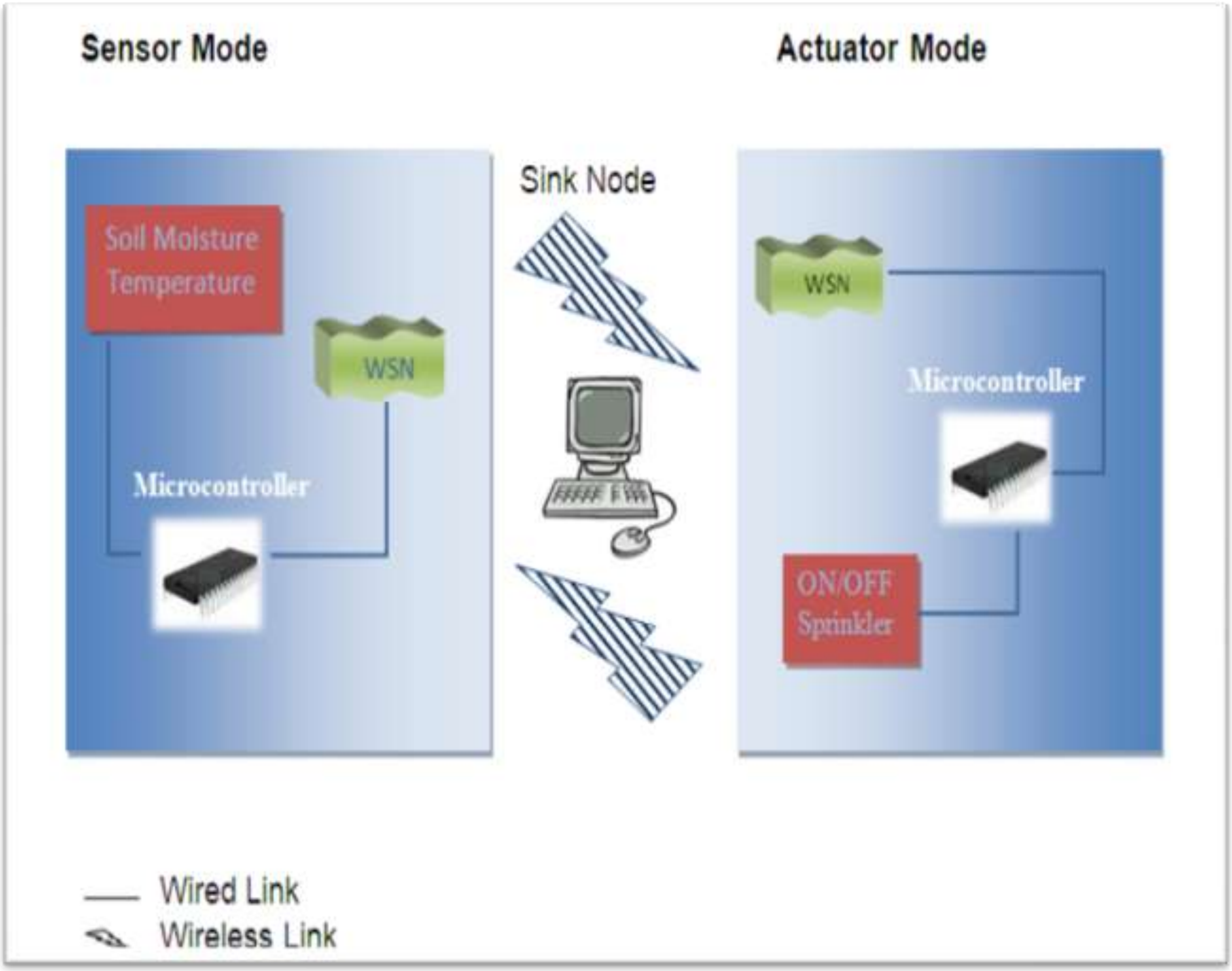


Figure 2. 11. General frame work of application.

2.2.2 COMMUNICATING WITH COMPUTER

Serial port or Universal Asynchronous Receiver/Transmitter (UART) is a serial communication that transfers in or out one bit at a time. The baud rate is the number of times per second a serial communication signal changes states; a state being a voltage level, a frequency, or a frequency phase angle. It is used to configure baud rate generator in the microcontroller to match the baud to 9600. Serial Interface is used to collect sensor node data packet wirelessly and transfer it to the connected PC via Serial interface port and to send request and actuator packet generated by control/decision. Serial port is used in this project is as shown in figure 2.12.The serial ports are accessed through any version of Windows and Visual C# by using a few system application interface (API) functions. When the computer writes a byte to the UART's transmit data register (TDR), the UARTs will start to transmit it on the serial line. The UART's status register contains a flag bit which the computer can read to see if the UART is ready to transmit another byte. Another status register bit says whether the UART has received a byte from the serial line, in which case the computer should read it from the receive data register (RDR). If another byte is received before the previous one is read, the UARTs will signal an "overrun" error via another status bit.

2.2.3 SPATIAL DECISION SUPPORT SYSTEM (SDSS) AND VARIABLE RATE IRRIGATION (VRI)

Spatial Decision Support System (SDSS) is designed to solve ill- or semi-structured problems, where objectives cannot be precisely or fully defined. It has a user interface that is both easy to use and powerful. It enables the user to combine data and models in a flexible manner. It also helps the user to explore the solution space and supports a variety of decision-making styles, and is easily adapted to provide new functions as the needs of the user change. In addition it allows storage of complex structures common in spatial data, include analytical techniques that unique to spatial analysis, and provide output in the form of maps or any other spatial format Malczewski, (1997).

Variable-rate irrigation is an innovative technology that enables a sprinkler irrigation system to optimize irrigation application. Most fields are not uniform due to natural variations in soil type or topography. These individual zones can be managed either by reducing problem areas or by planting multiple crops. VRI technology is used in sprinkler irrigation system to easily apply varying rates of irrigation water based on the individual management zones within a field. VRI is calculated by using following equations:

2.2.3.1 Landscape Evapotranspiration

The water use **(ET_L)** can be calculated for a specific plant by using a reference evapotranspiration rate (ET_O), and applying a de **(K_L)** to convert the reported ETo to ET_L.

$$ET_L = K_L \, x \, ET_O \qquad 2.2$$

Where: ET_L = Evapotranspiration landscape, mm

K_L = Landscape coefficient

ET_O = Reference evapotranspiration for a grass reference crop, mm

2.2.3.2 Irrigation Water Requirement

The irrigation water requirement **(IR)** is determined by dividing the landscape water requirement by the application efficiency (Ae).

$$IR = ET_L / Ae \qquad 2.3$$

Where:

IR = Irrigation water requirement, mm

ET_L = Evapotranspiration landscape, mm

Ae = Application efficiency, %

2.2.3.3 Irrigation Operating Time

The landscape water requirement and the application rate determine the maximum length of time the system should operate for each irrigation event.

$$OT = ET_L \, x60/AR \qquad 2.4$$

Where:

OT = Operating Time, minutes

ETL = Evapotranspiration landscape, mm

AR = Application rate, mm/hr

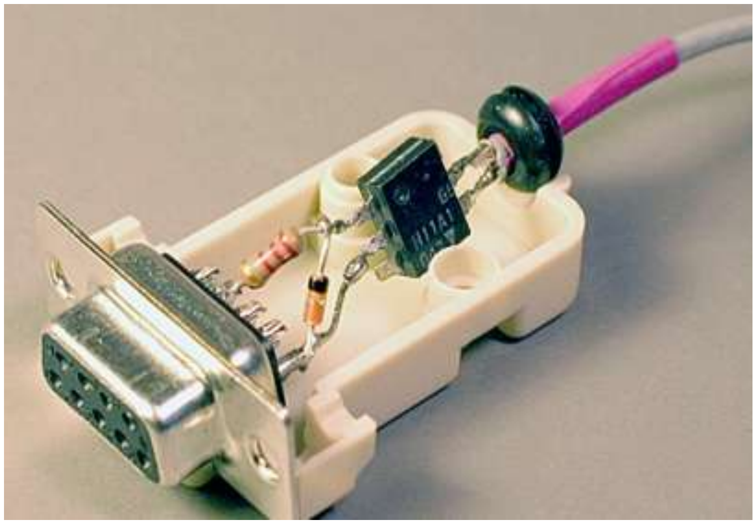
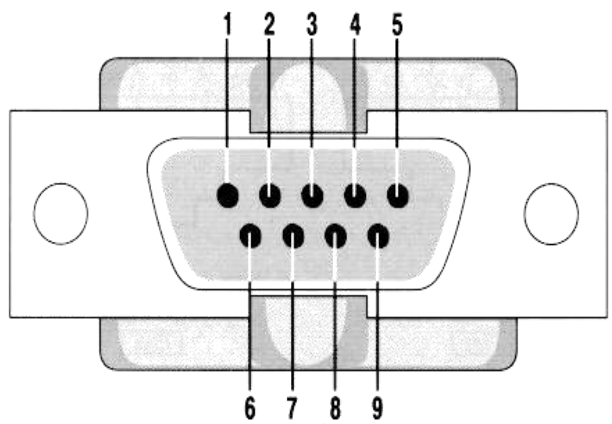


Figure 2.10.Serial port.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 2. 11. Pins description in serial port.

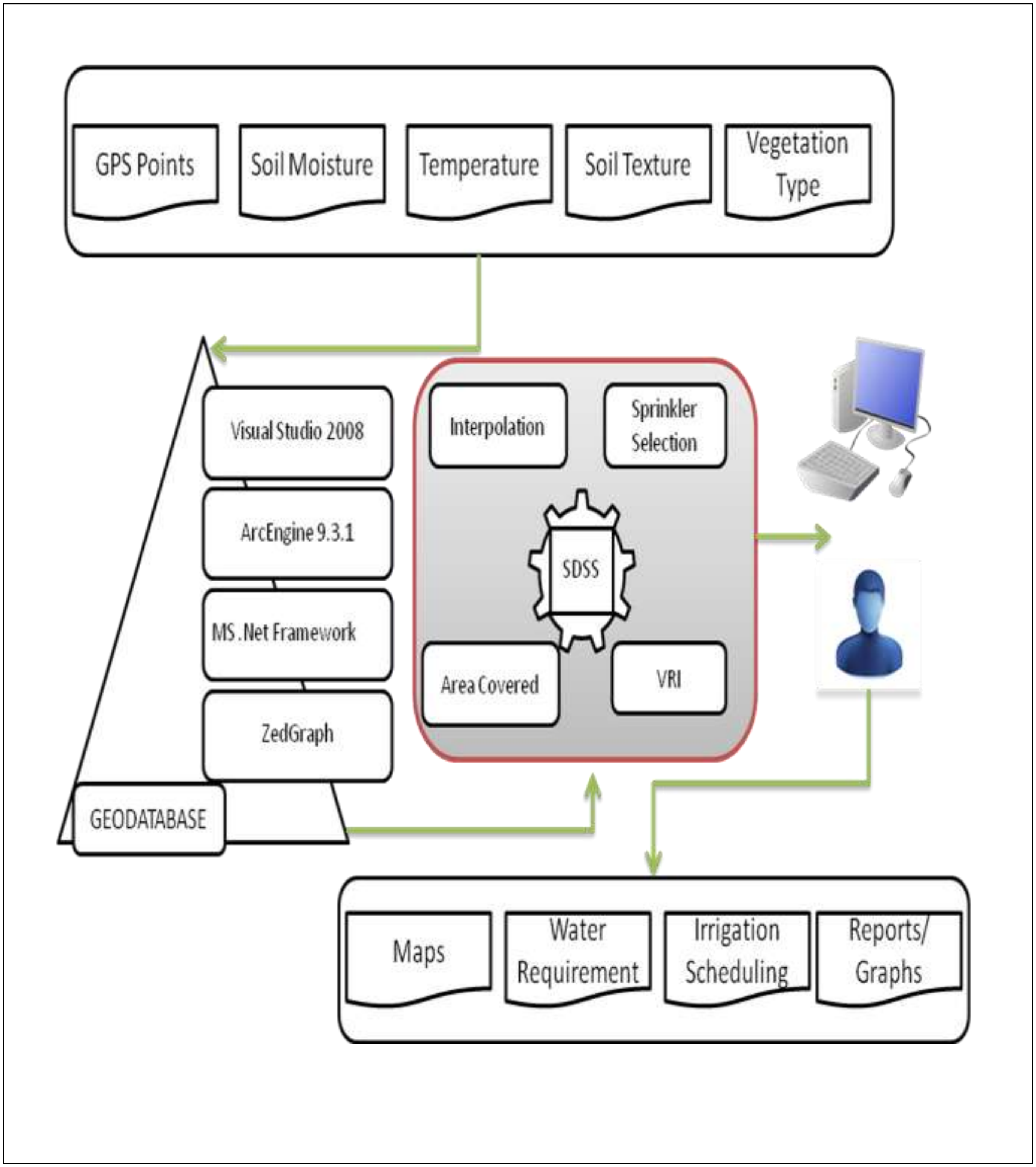


Figure 2.12: Work flow of application development phase.

LEVEL 0: DFD for Wireless Sensor Network

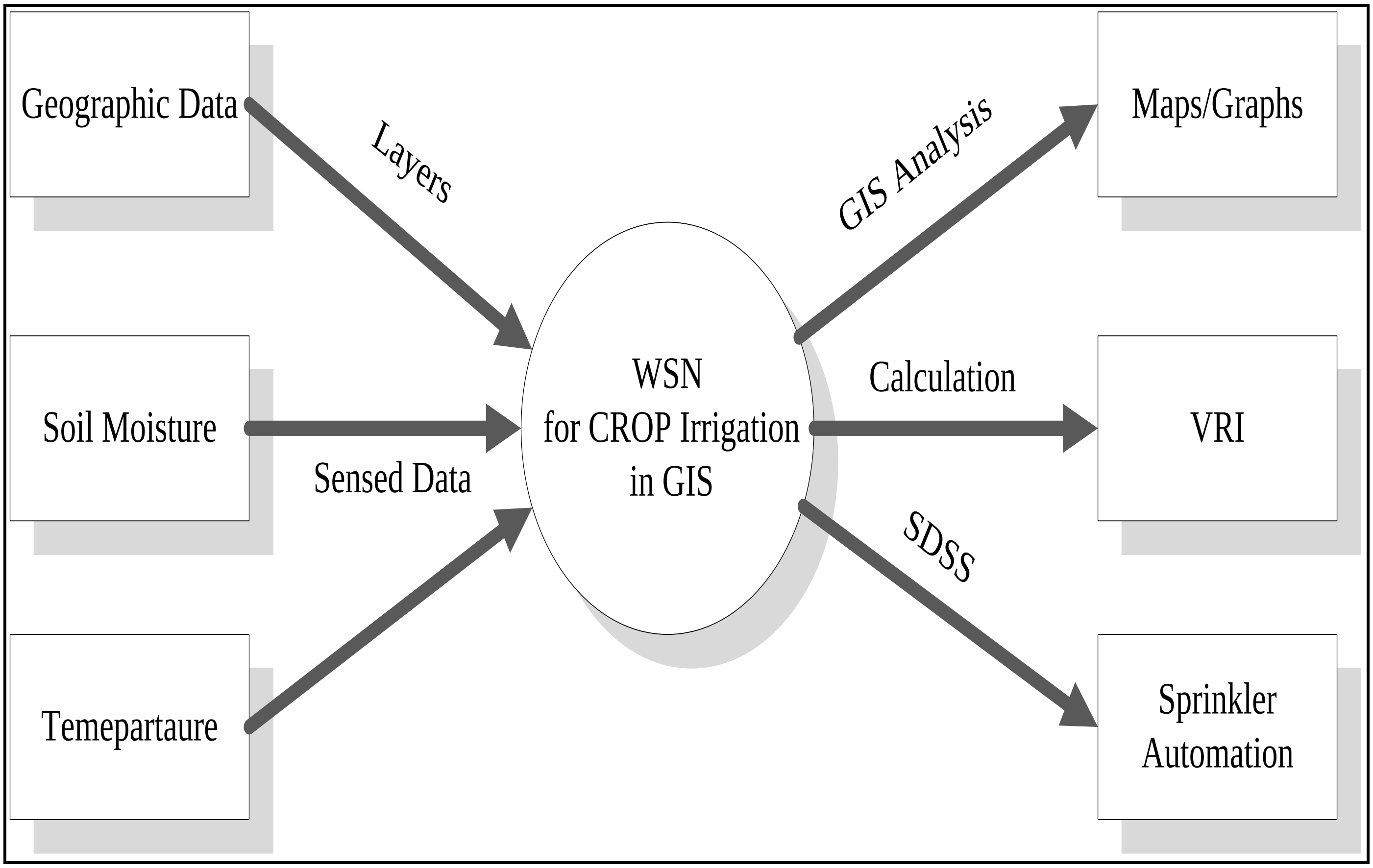


Figure 2. 13. Data flow diagram level 0, general framework of application.

LEVEL 1: Internal Architecture

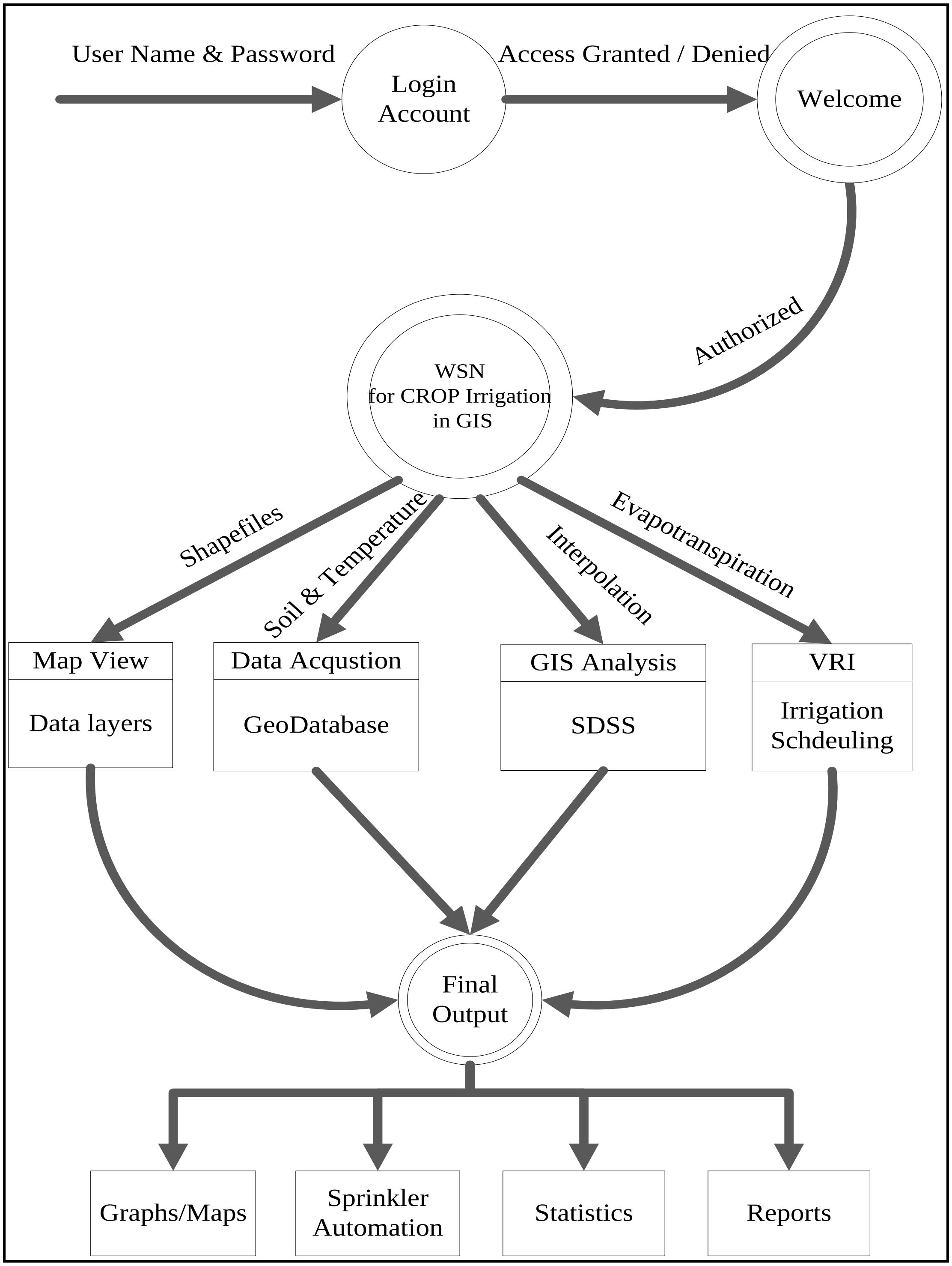


Figure 2. 14. Data flow diagram level 1, architecture of application.

LEVEL 2: DFD for Login Form

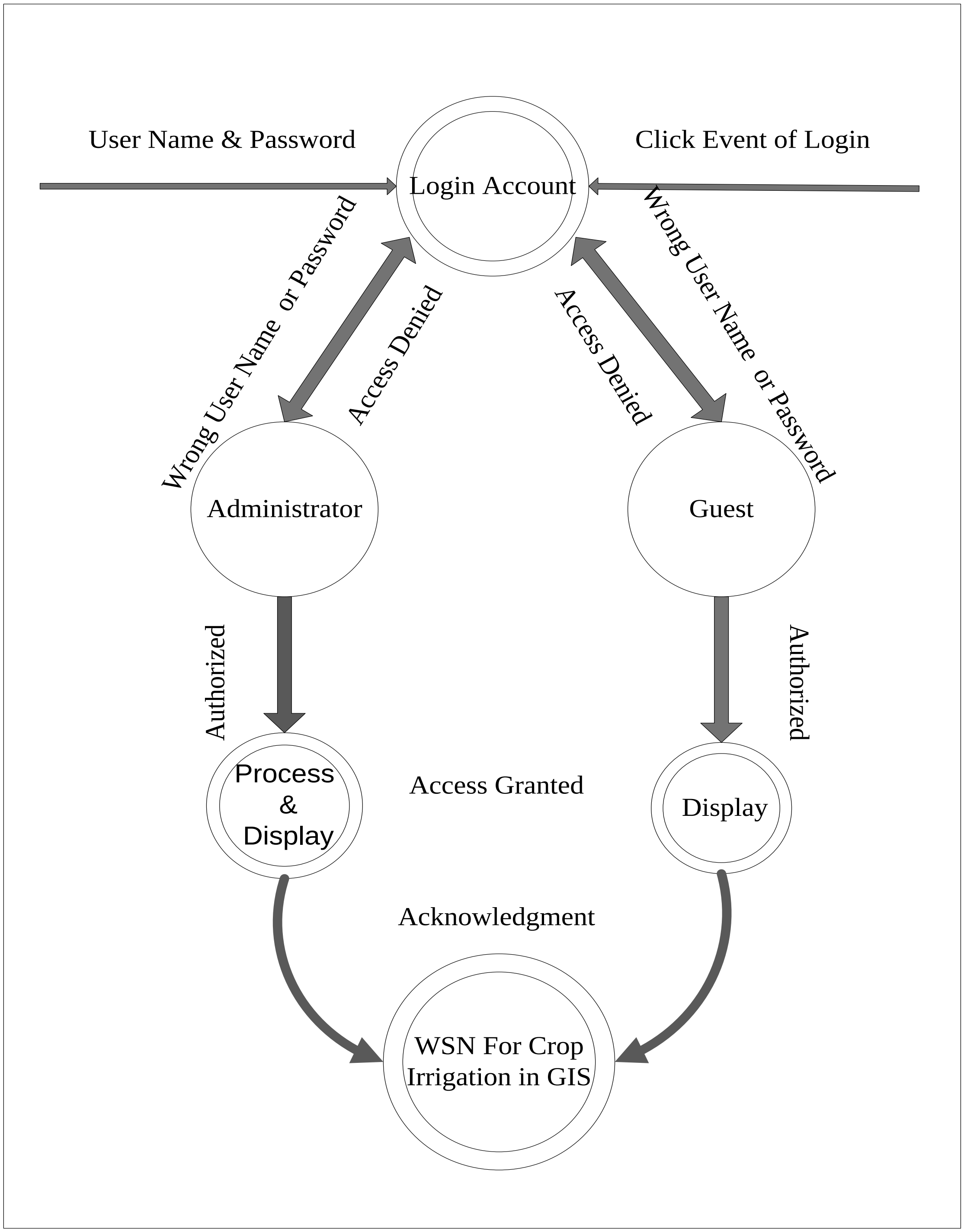


Figure 2. 15. Data flow diagram level 1, login page.

LEVEL 2: DFD for Data Acquisition

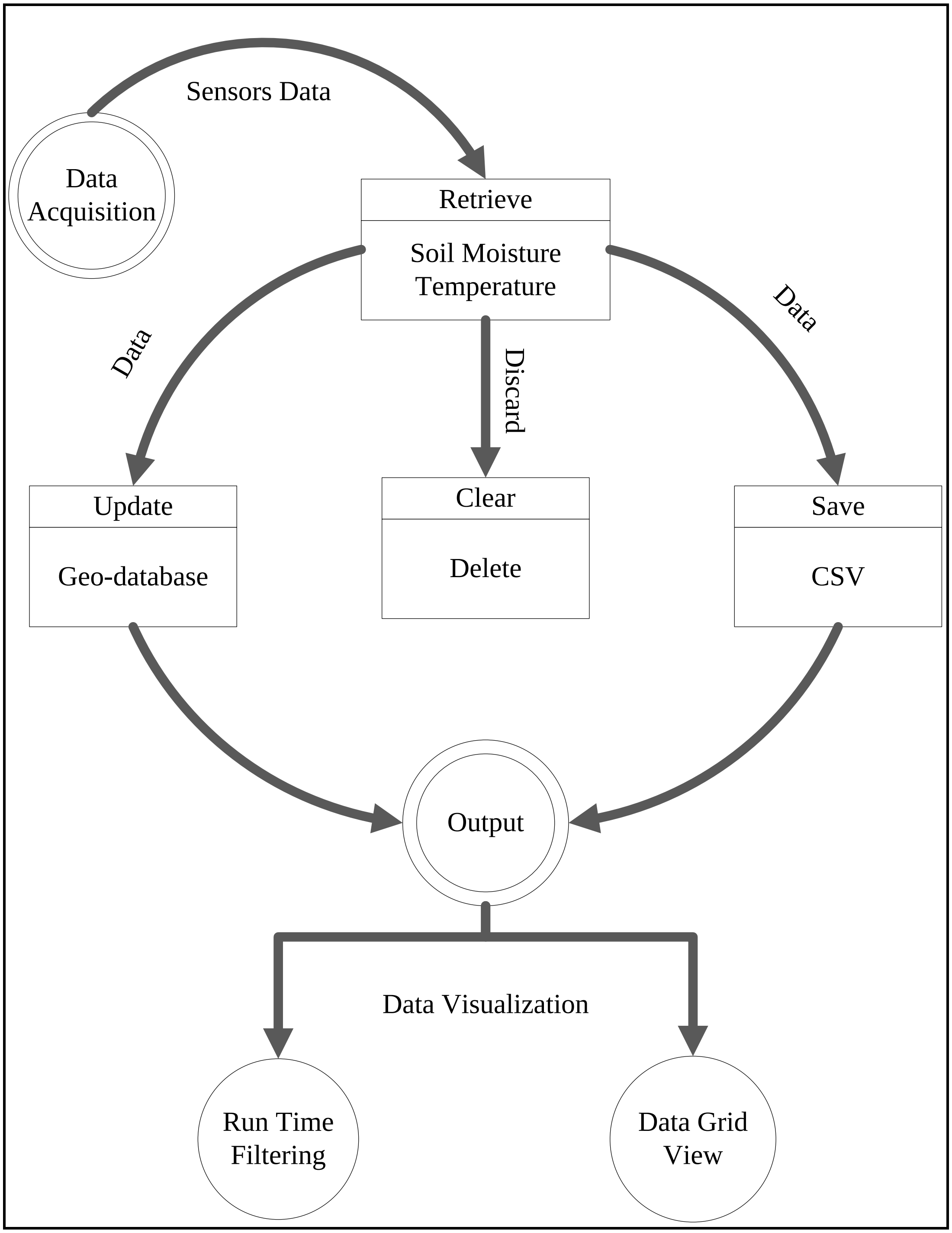


Figure 2. 16. Data Flow diagram level 2, data acquisition tab.

LEVEL 2: DFD for SDSS and VRI

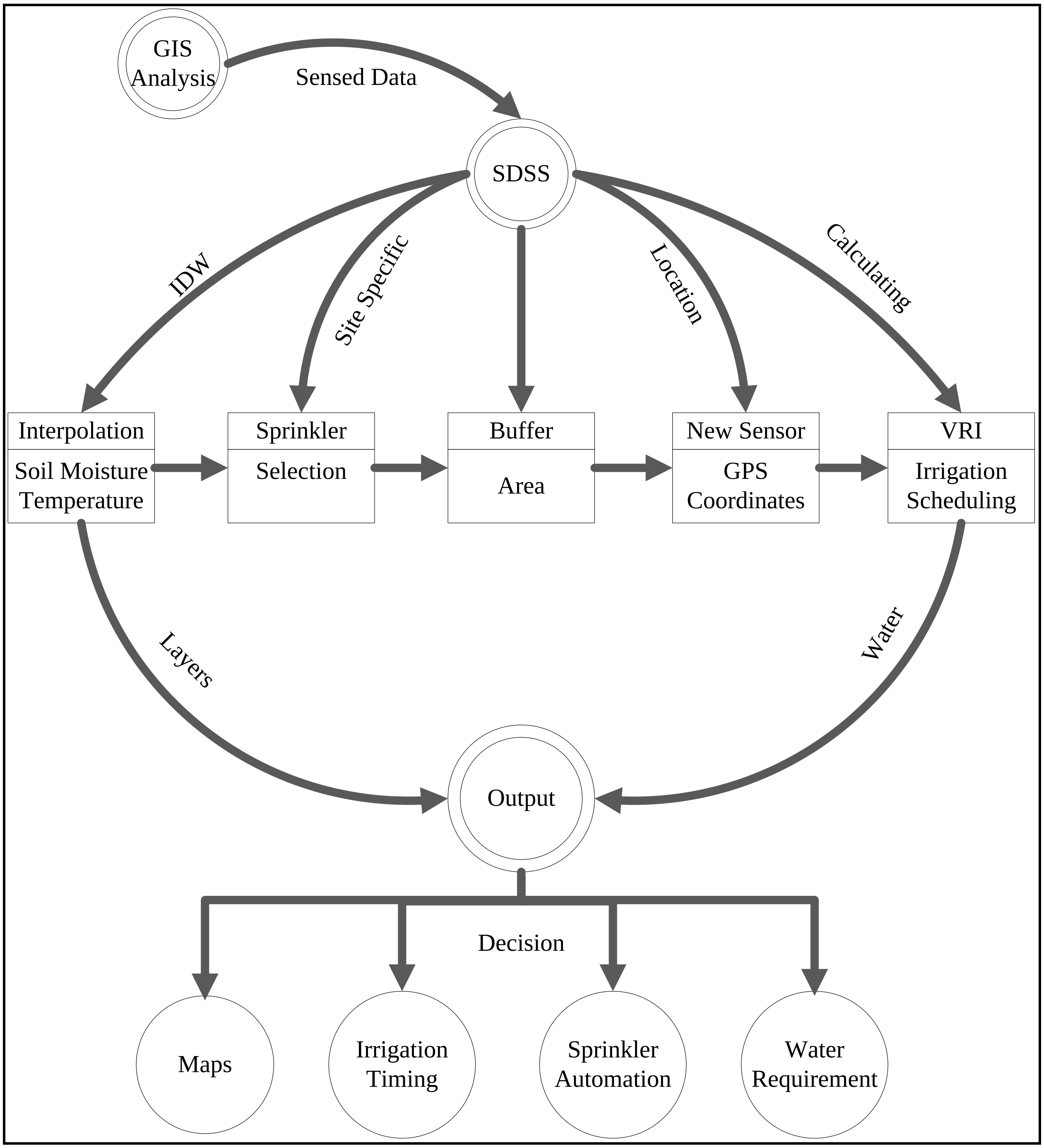


Figure 2. 17. Data Flow diagram level 2, gis analysis tab.

LEVEL 2: DFD for MAP View

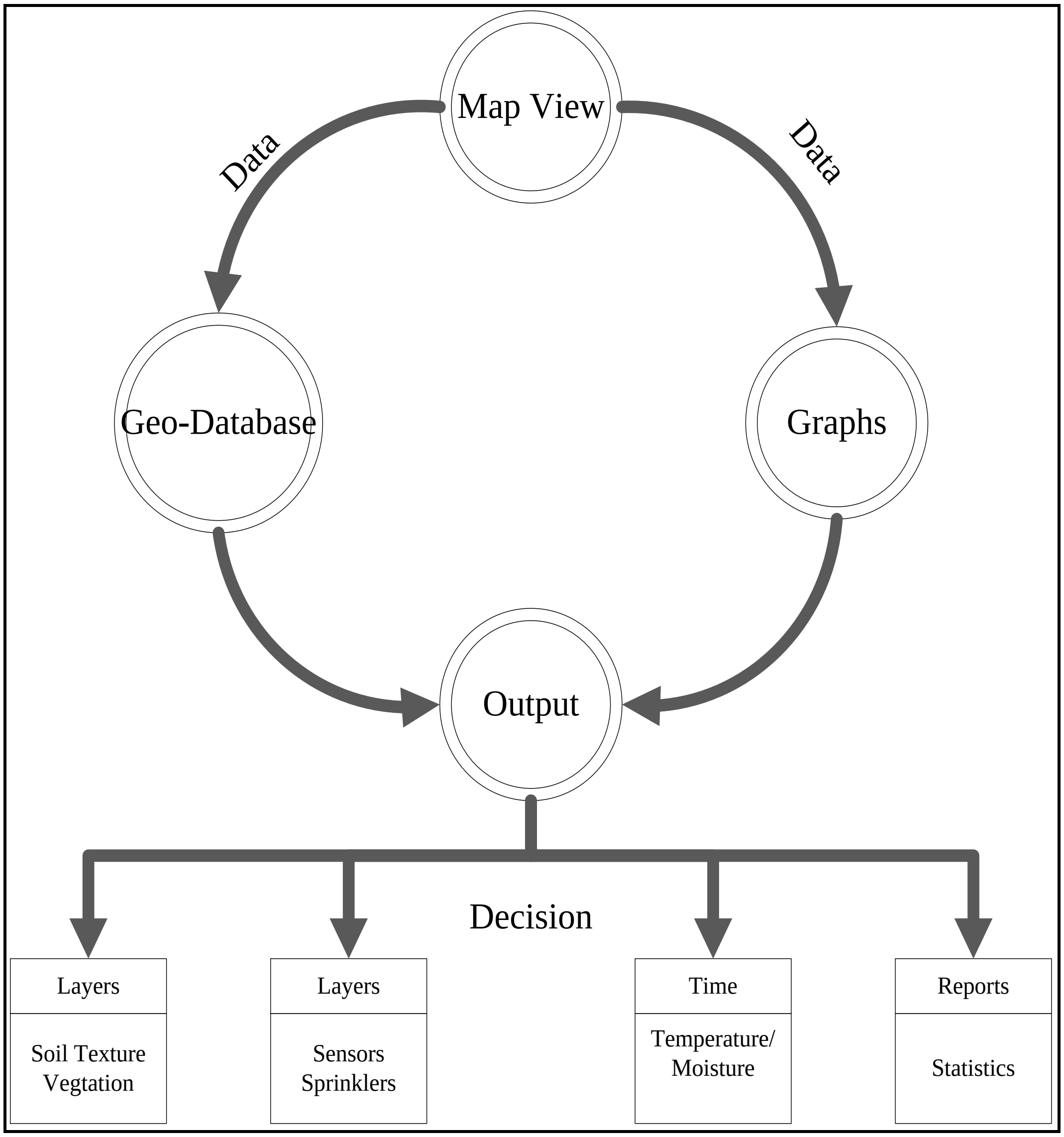


Figure 2. 18. Data flow diagram level 2, map view tab.

RESULTS AND DISCUSSIONS

It has been estimated that the world's irrigation efficiency is less than 40 percent. In order to improve irrigation efficiency, it is important to monitor the soil moisture content to avoid over irrigation. The development and implementation of site-specific farming has been made possible by combining the Global Positioning System (GPS) and geographic information systems (GIS). The GIS-based customized application presented in this research demonstrates how accurate and timely information helps the user in making decisions for precision agriculture using soil moisture mapping and irrigation scheduling. There are inexpensive and easy-to-use methods and techniques that can be developed for use by all farmers. This research shows the ease and usefulness of using customized GIS application and WSN in agriculture sector. Many semi-structured spatial decision problems that are related to agriculture sector can be solved by using GIS techniques.

Ayday and Safak, (2009) developed integrated wireless soil moisture sensor network. In this study, wireless sensor networks were used for measuring of soil moisture content of the soil. The data collected by the sensors is transferred to the computer (in a hop by hop fashion) and soil moisture map of the area was prepared by using GIS techniques. They observed from the prepared map that the soil moisture content is spatially variable.

Rehman et al., (2010) studied the crop irrigation control system using indigenously developed WSAN with decision support system. The software for irrigation control using WSAN was developed in Visual Basic 6. The software provides the following facilities: (i) data acquisition from sensor motes,(ii) data aggregation and storage of sensed data into database, and (iii) decision support system.

Xijun and Li Mei, (2009) developed irrigation area automatic system, high performance embedded micro-controller and low-power technology was used to design the water wireless sensor network. The sensor node gathers the hydrographic information such as water-level, gate position and rainfall. The sink node receives the real-time data; the information center stores and processes those data which are transmitted from the sink node through the GPRS network. The system replaces the wired transmission with the wireless transmission, which reduces the costs in installment and maintenance, and improves the system's reliability and extension. It has better application prospect**.**

The objectives for this research were to investigate and design a low cost soil moisture sensor for crop irrigation using GIS techniques. Spatial Decision Support System (SDSS) was designed for irrigation problems, where objectives cannot be precisely or fully defined. VRI is used due to un-uniform variations in soil type and topography.

Soil moisture sensors, after sensing the data, send it to wireless transceivers from where the data is transmitted to the control / master transceiver. The master transceiver is attached to the computer system via serial port as shown in fig 3.2. The receiving application running on the system acquires the data and saves it to the geo-database. The spatial decision support system reads this data from geo-database and applies various GIS analysis to calculate Variable Rate Irrigation (VRI) system.

3.1 SOIL MOISTURE CONDITIONS AND CALIBRATION

The two aluminum leads act as the sensor probes as shown in figure 3.1. They are immersed into the specimen soil whose moisture content is under test. The soil is examined under three conditions which are as follow:

Case 1: Dry Condition

The probes are placed in the soil under dry conditions and are inserted up to a depth of 10 cm in the soil. The voltage output of in this case ranges from 0 to 1.5V respectively.

Case 2: Optimum Condition

When water is added to the soil, it penetrates through the successive layers of soil. This water increases the moisture content of the soil and leads to an increase in its conductivity which forms a conductive path between the two sensor probes. The voltage output of ADC in the optimum case ranges from 1.7 to 3.6V approximately

Case 3: Excess Water Condition

With the increase in water content beyond the optimum level, the conductivity of the soil increases significantly and a steady conduction path is established between the two sensor probes and the voltage output from the sensor increases no further beyond a certain limit. The maximum possible value for it is not more than 4.7V.

Figure 3.1.Soil moisture sensor probe.

Prototype of Sensors Developed

Figure 3.2.Wireless transceivers (master / slave) and computer.

Field Application

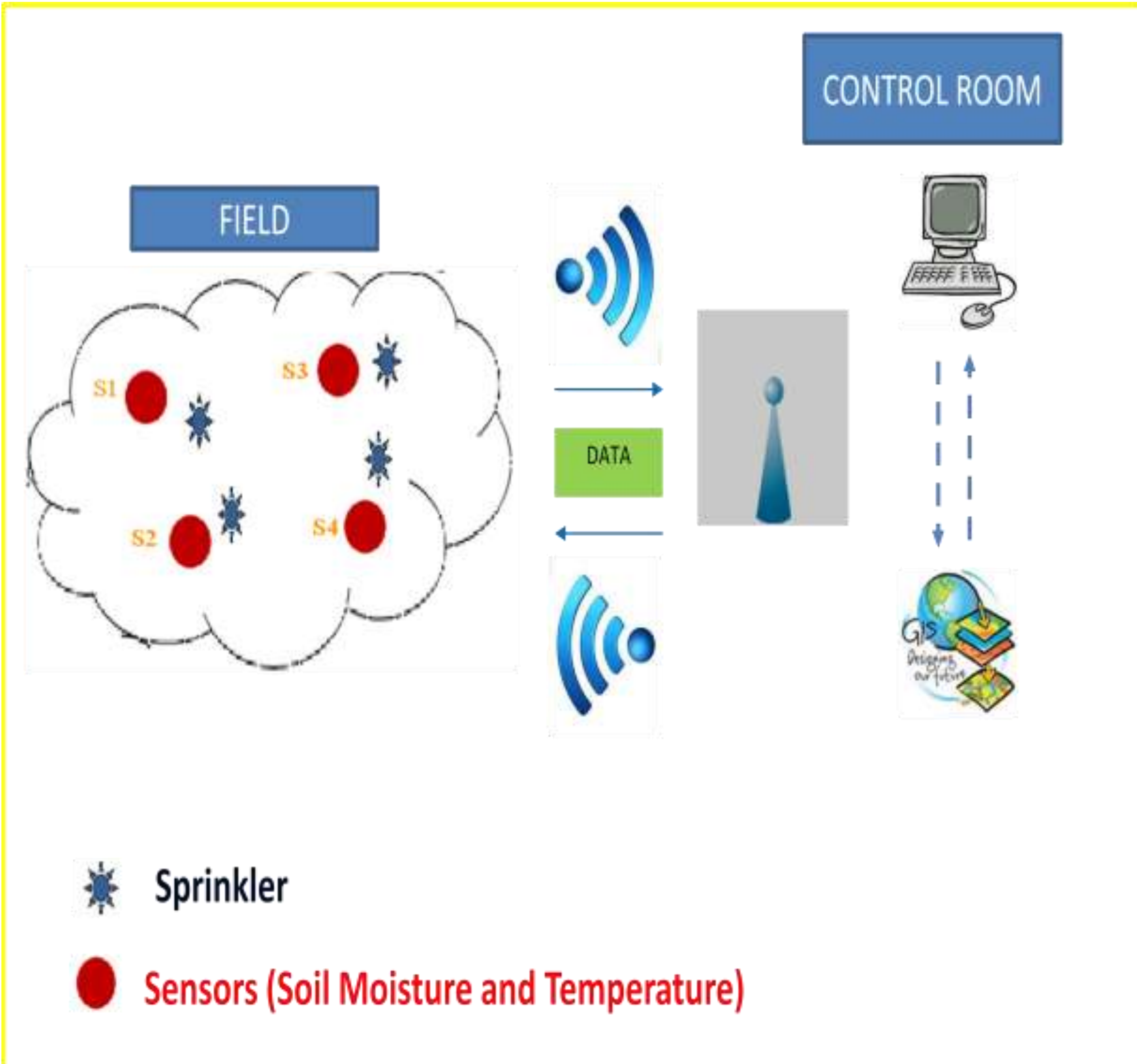


Figure 3.3.Field application showing sensor installed in fields and their communication.

Table 3.1. Soil Moisture sensor readings.

Optimum Range(V)	**Soil Moisture Condition**
0~1.6V	Dry Condition
1.7~3.6V	Optimum Condition
>3.6V	Excess water condition

Table 3.2. Temperature sensor readings.

Optimum Range (V)	**Temperatures Range in (^{o}C)**
0~1.25 V	Below 25 ^{o}C
1.26~2.7 V	25~50 ^{o}C
2.8~3.75 V	50~75 ^{o}C
3.76~5 V	75~100 ^{o}C

Table 3.3.Soil moisture and lab test values.

Sensor soil moisture content (%)				**Lab Test (%)**			
1	**2**	**3**	**Average**	**1**	**2**	**3**	**Average**
11	12	12	11.7	13.15	13.19	13.09	13.14
15	16	17	16	17.8	18.1	17.9	17.93
21	20	21	20.67	18.41	18.02	18.31	18.25
31	29	29	29.67	25.71	25.2	26.01	25.64
42	40	39	40.33	37.39	37.65	37.15	37.40
51	51	55	52.33	47.8	47.5	47.6	47.63
59	62	61	60.67	55.9	55.78	55.89	55.86
75	74	76	75.00	67.23	67.01	67.11	67.12

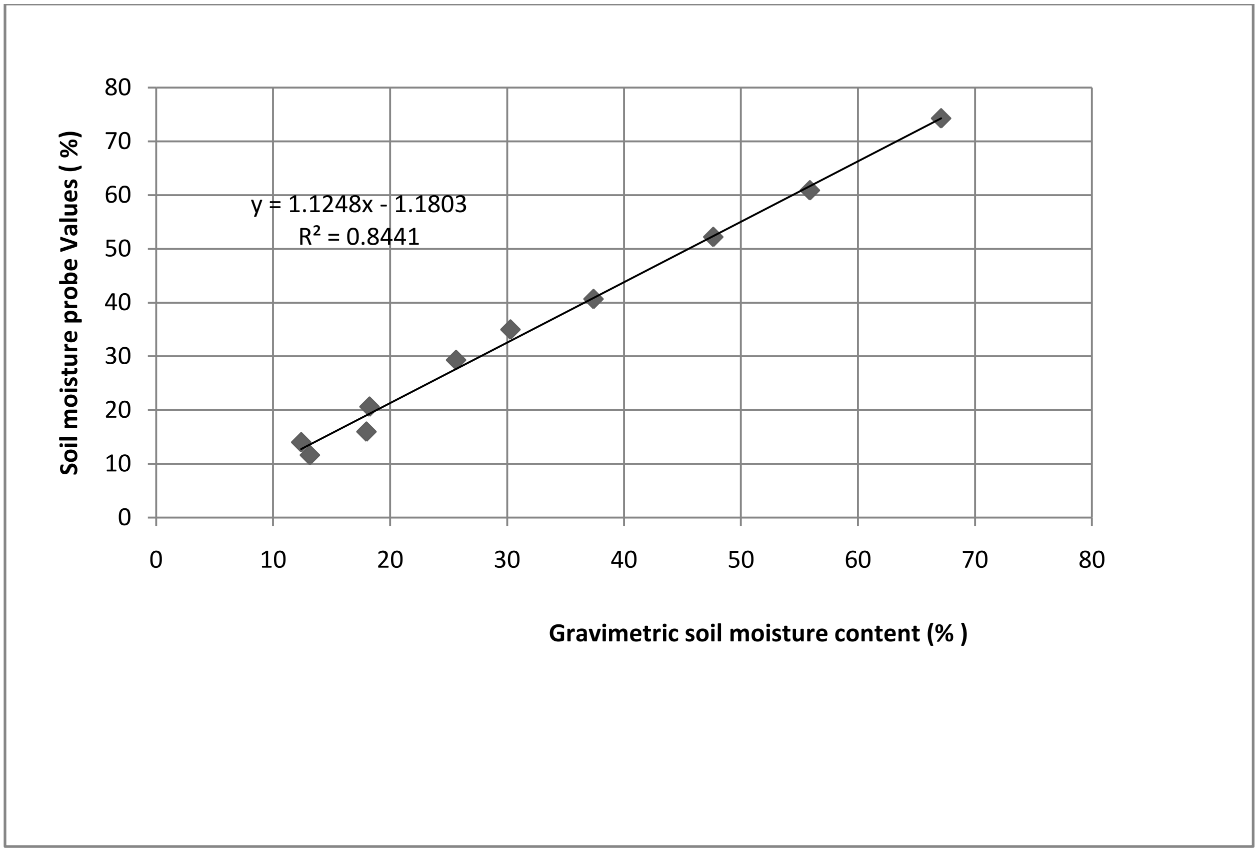


Figure 3.3. Soil moisture calibration equation and graph.

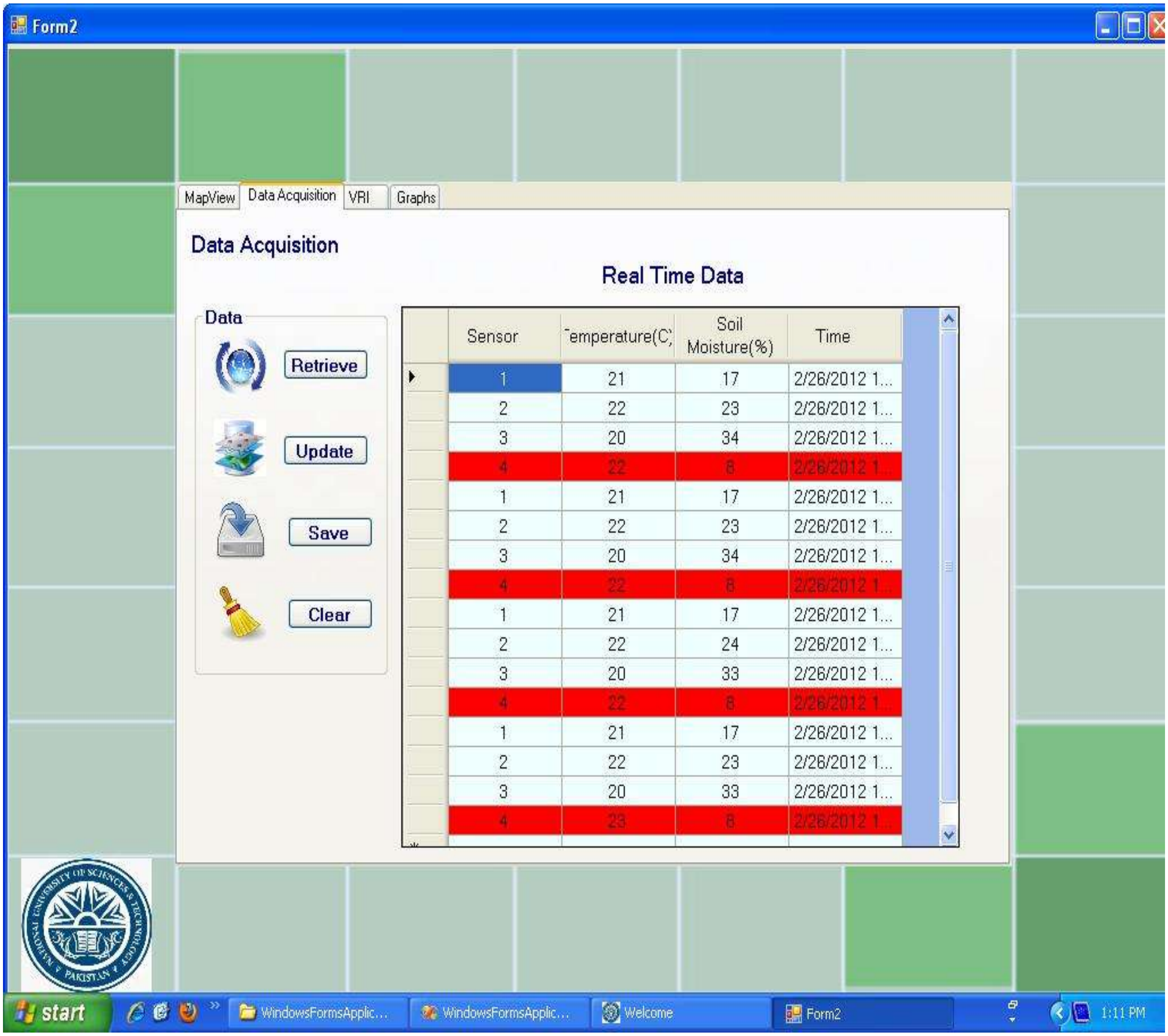


Figure 3.4. Data acquisition phase, showing the run-time filtering and update of geodatabase.

3.2 DATA ACQUISITION PHASE

Data acquisition phase comprises of following functions as shown in fig 3.3.

3.2.1 RETRIEVE FUNCTION

The “Retrieve” function uses a Data Grid View to show all the sensed data retrieved from WSN via a serial port. The Run time filtering layer is located in the receive path just after the serial port receive packet from WSN. Filtering criterion is based on the critical value of soil moisture sensor. Red spot shows all the values which are below critical level i.e. 10%.

3.2.2 UPDATE FUNCTION

The “Update” database function is used to update GeoDatabase. It populates all the values from Data Grid View and insert in different fields comprising of device ID, time, temperature and soil moisture values in attribute table of soil moisture layer. Time is a key factor in retrieving data; user gets information that at what time and date that particular data is received.

3.2.3 SAVE FUNCTION

The “Save” functions save all the Data Grid Value in Comma Separated Value (CSV). These files are particularly used to transport large amounts of tabular data between applications that are not directly connected in lesser time. The files are easily editable using common spreadsheet applications like Microsoft Excel.

3.2.4 CLEAR FUNCTION

The “Clear” function deletes all the values form Data Grid View, Geodatabase and CSV File Format. Once the information is deleted it is deleted forever.

3.3 NEW SENSOR EVENT

The XY event takes the latitude and longitude from the user as the input to create a point at specific coordinates to an active map view. XY event adds a point to Soil Moisture layer and update the Geo-Database as shown in figure 3.5.

3.4 SPATIAL ANALYST EVENT

The Spatial Analysts event uses the IDW method to interpolate sensors points with respect to soil moisture and temperature value. The event performs the interpolation by taking sensors layer and a mask dataset as input, creates a raster interpolation object and adds layer to the active map view as shown in figure 3.6. Inverse distance weighted methods are based on the assumption that the interpolating surface should be influenced most by the nearby points and less by the more distant points. The interpolating surface is a weighted average of the scatter points and the weight assigned to each scatter point diminishes as the distance from the interpolation point to the scatter point increases.

Figure 3.5. New sensor examples, adding a point on the specific location.

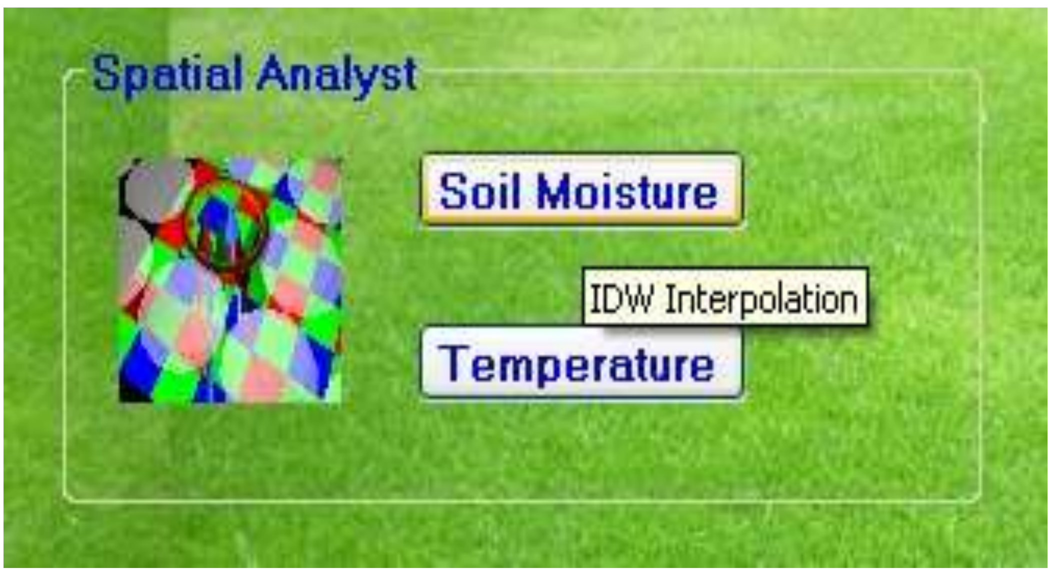


Figure 3.6. Soil moisture and temperature interpolation.

3.5 ANALYSIS TOOL EVENT

Analysis event calculates the total number of area that a particular sprinkler covers the field. The function creates a graphic element to draw the shape of the facility selected by the user in buffer region. The event takes input radius of sprinkler in meters to generate buffers and cumulates all the buffers to find total area. A message box is displayed showing total number of new sprinklers to be installed to cover vegetation area and creates a new layer “Area Covered “to the active map view.

3.6 VARIABLE RATE IRRIGATION EVENT

The event Variable Rate Irrigation System is used by the user to provide input variables for Soil and Vegetation Type, Species, Density and Microclimate Factor along with Sprinkler Type. The System automatically calculates the values of these user inputs to determine the water requirement underground in both inches and millimeter.

The output of this event will notify following things to the user:

1. Water requirement - sufficient or insufficient.
2. The time required to fulfill the water shortage if water is reported insufficient.

Once it is known that water level is insufficient the event Field Calculator is used to determine the area where water is below desired level by calculating following:

1. Temperature
2. Moisture

The result displays the number of sensors where the water level is low. In this example the value shown is 3. This means there are three areas on the field where sprinklers should start working.

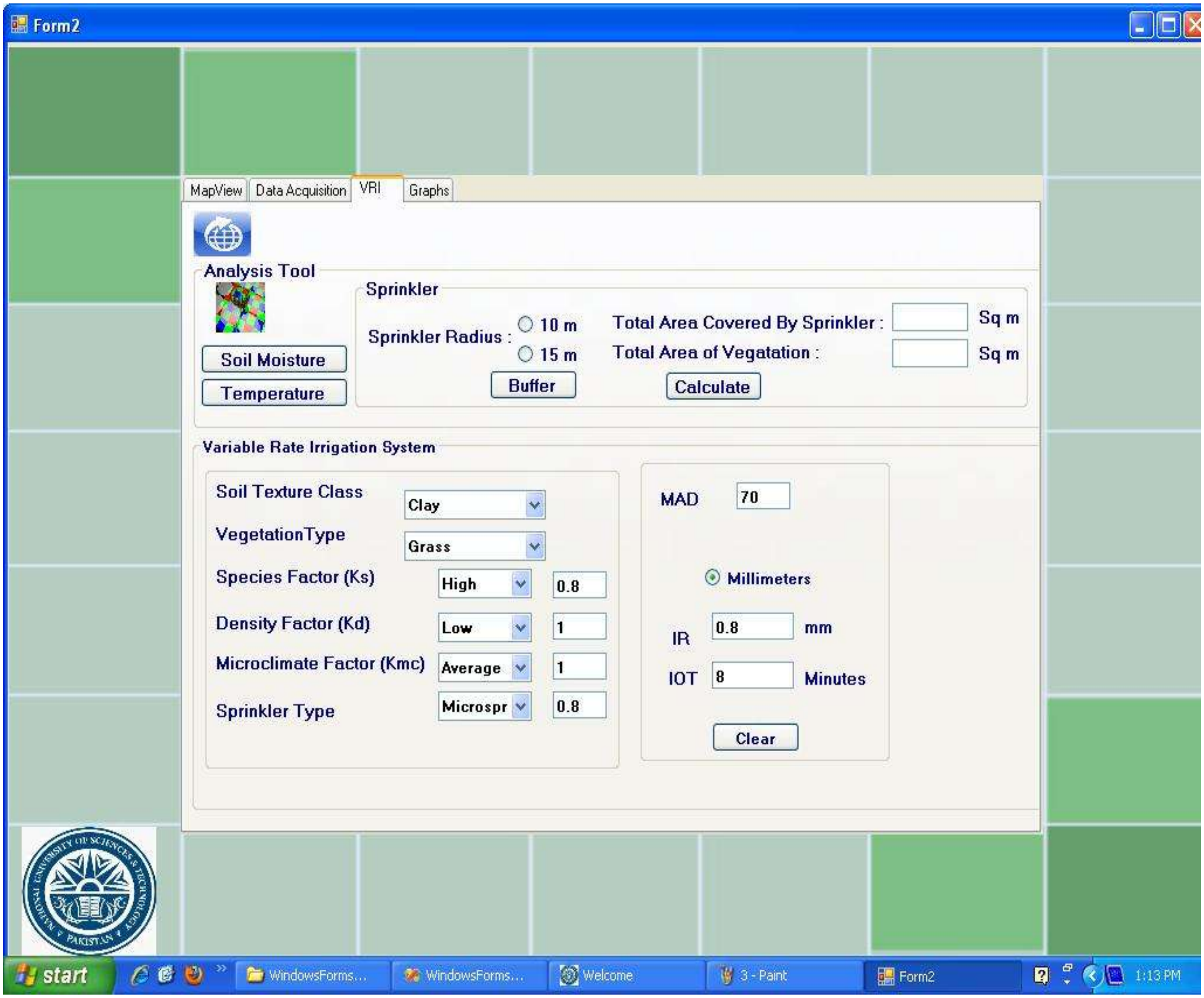


Figure 3.7. VRI, showing the perimeters required to calculate IWR and IOR.

3.7 CALCULATION EVENT

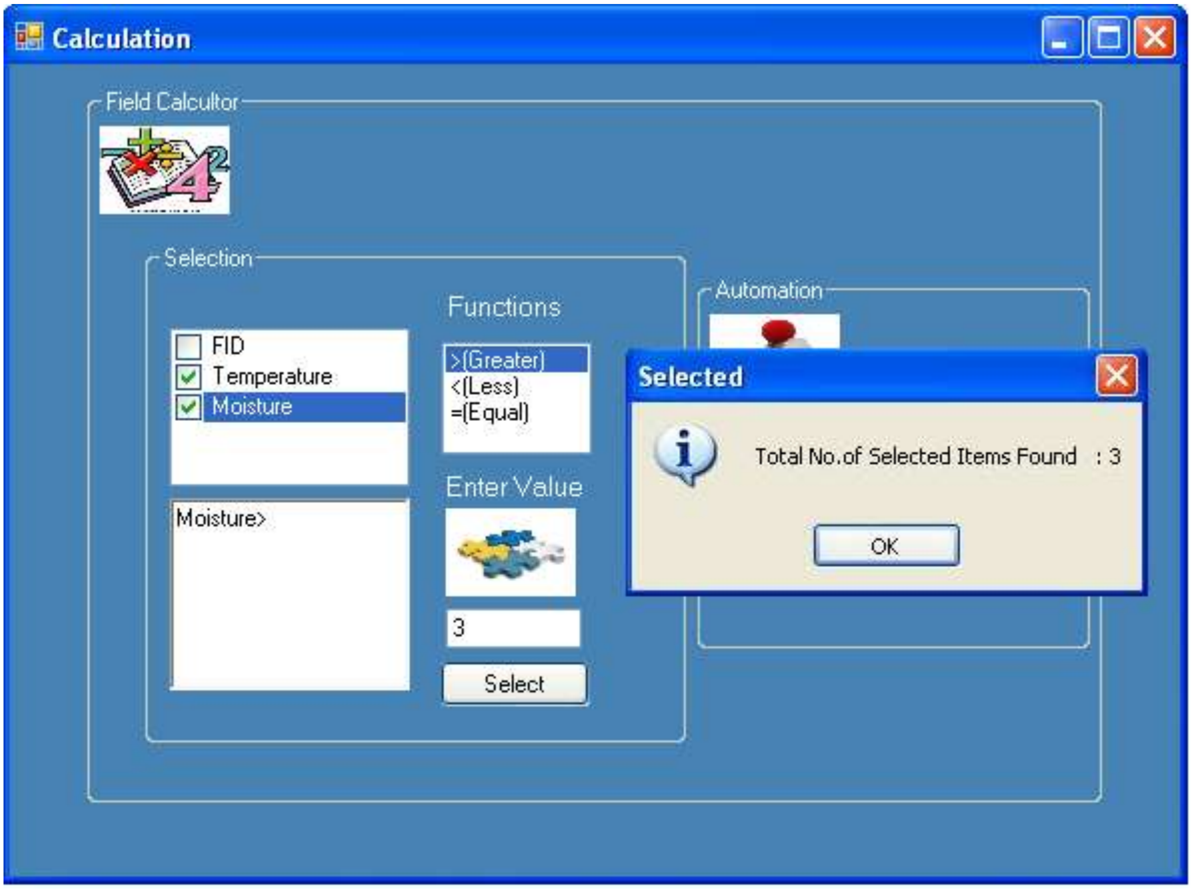


Figure 3.8.Field calculator, showing values found that are greater than 3 on the active map.

3.8 GRAPH EVENT

The event Graph generates two types of graphs for users. They are:

1. Temperature Vs Time

The Graph shows the temperature values of all the sensors per unit time.

2. Soil Moisture Vs Time

The Graph shows the soil moisture values of all the sensors per unit time.

These graphs are used for reports and statistical data by the users for record keeping and decision making using ZEDGRAPH library.

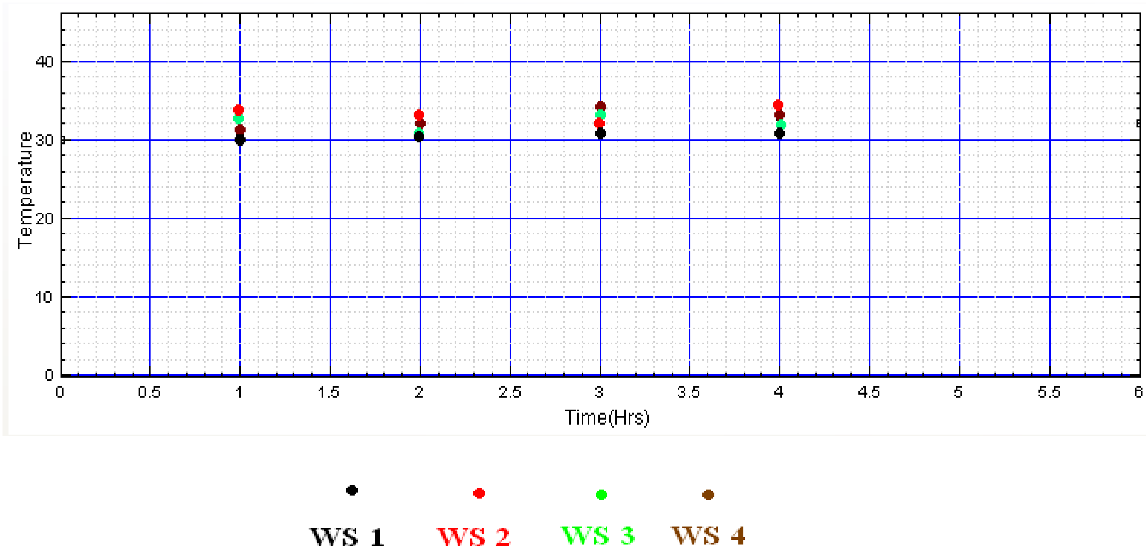


Figure 3.9. Graph showing temperatures of all sensors vs. time.

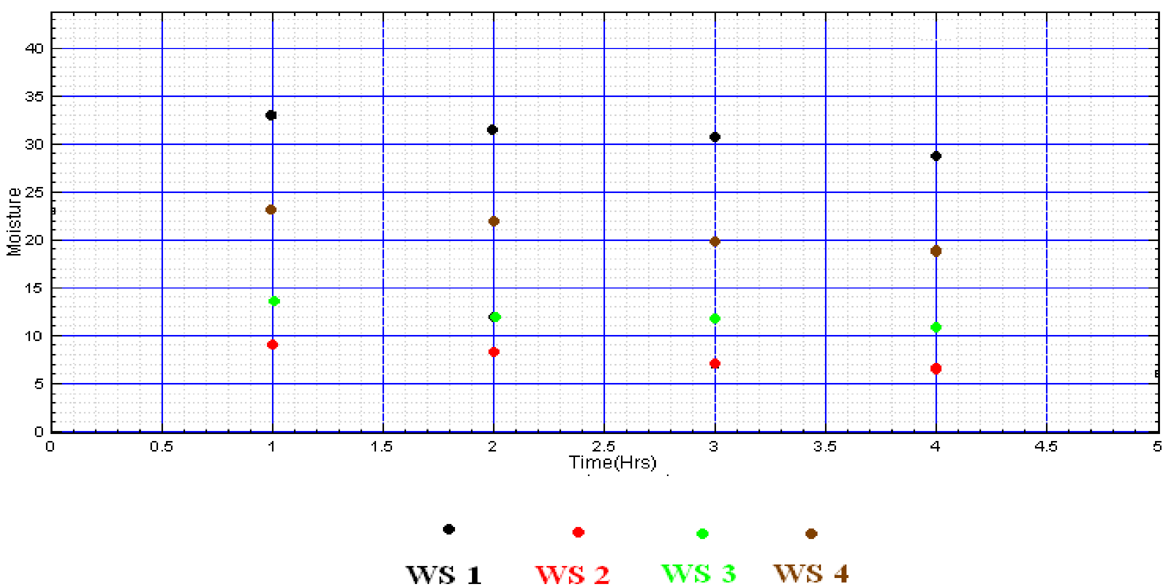


Figure 3.20. Graph showing soil moisture of all sensors vs. time.

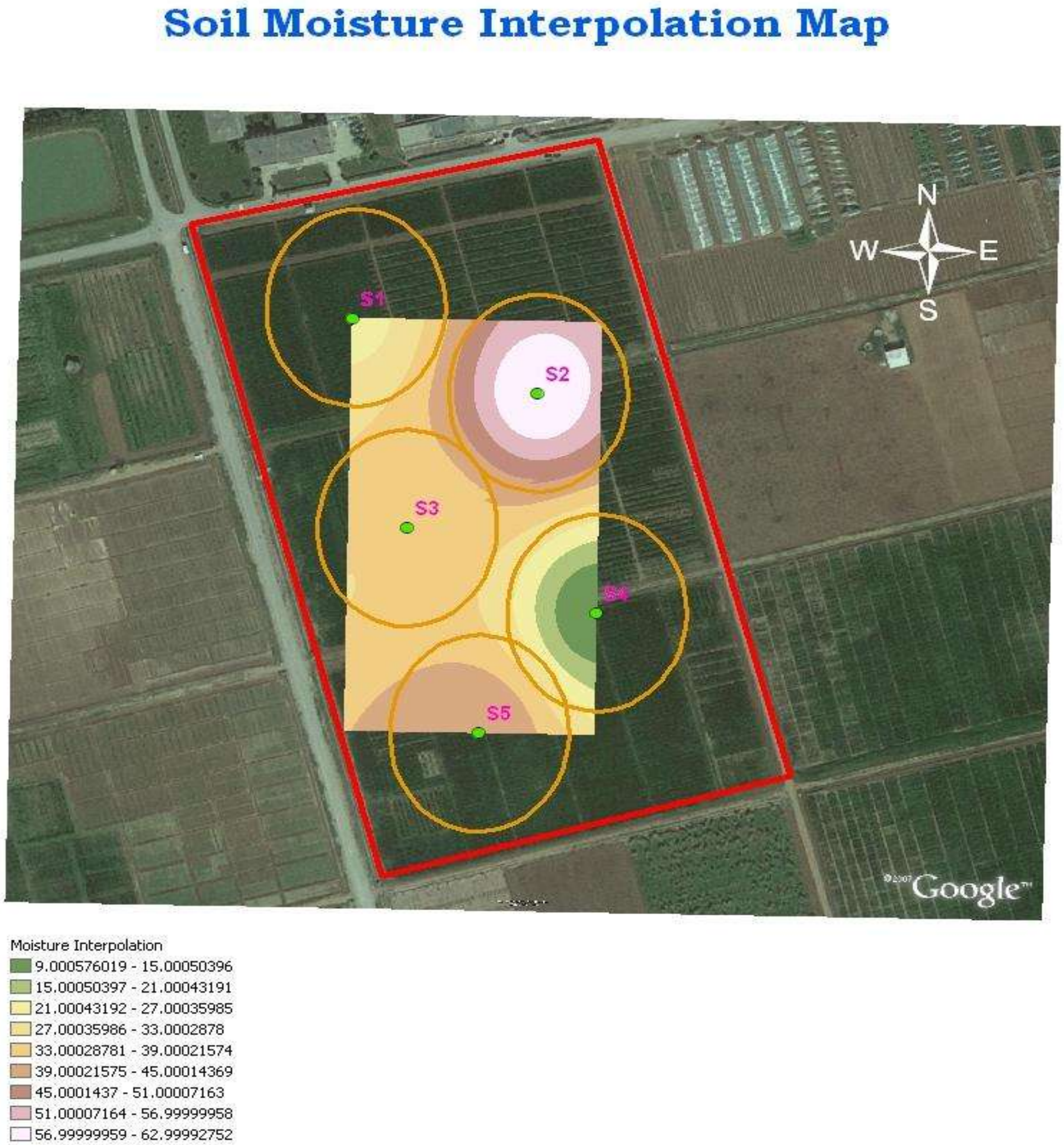


Figure 3.31 .The map shows soil moisture, vegetation area along with sensors' locations and 20m buffers around the sensors showing the area covered.

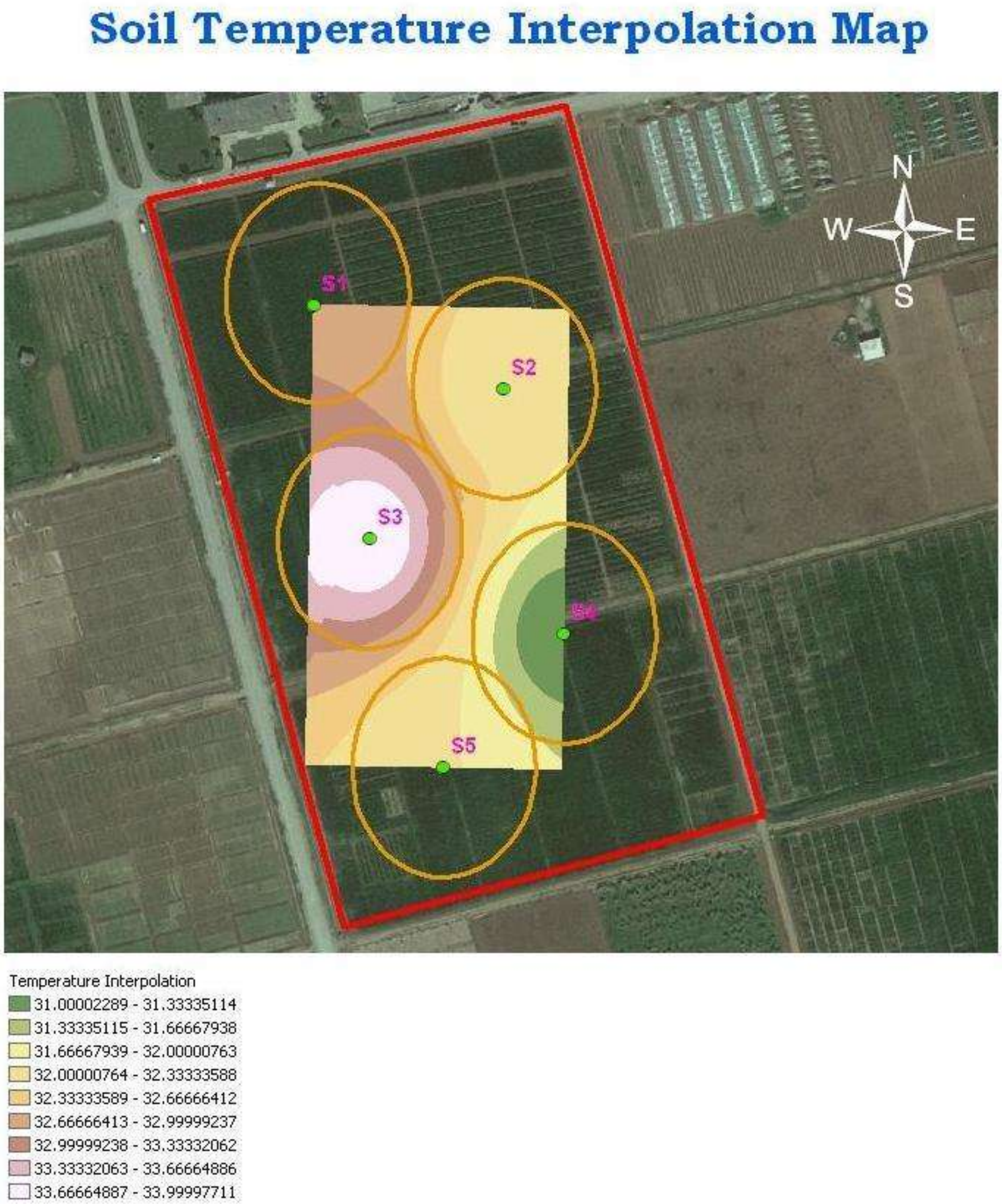


Figure 3.12.The map shows temperature, vegetation area along with sensors' locations and 20m buffers around the sensors showing the area covered.

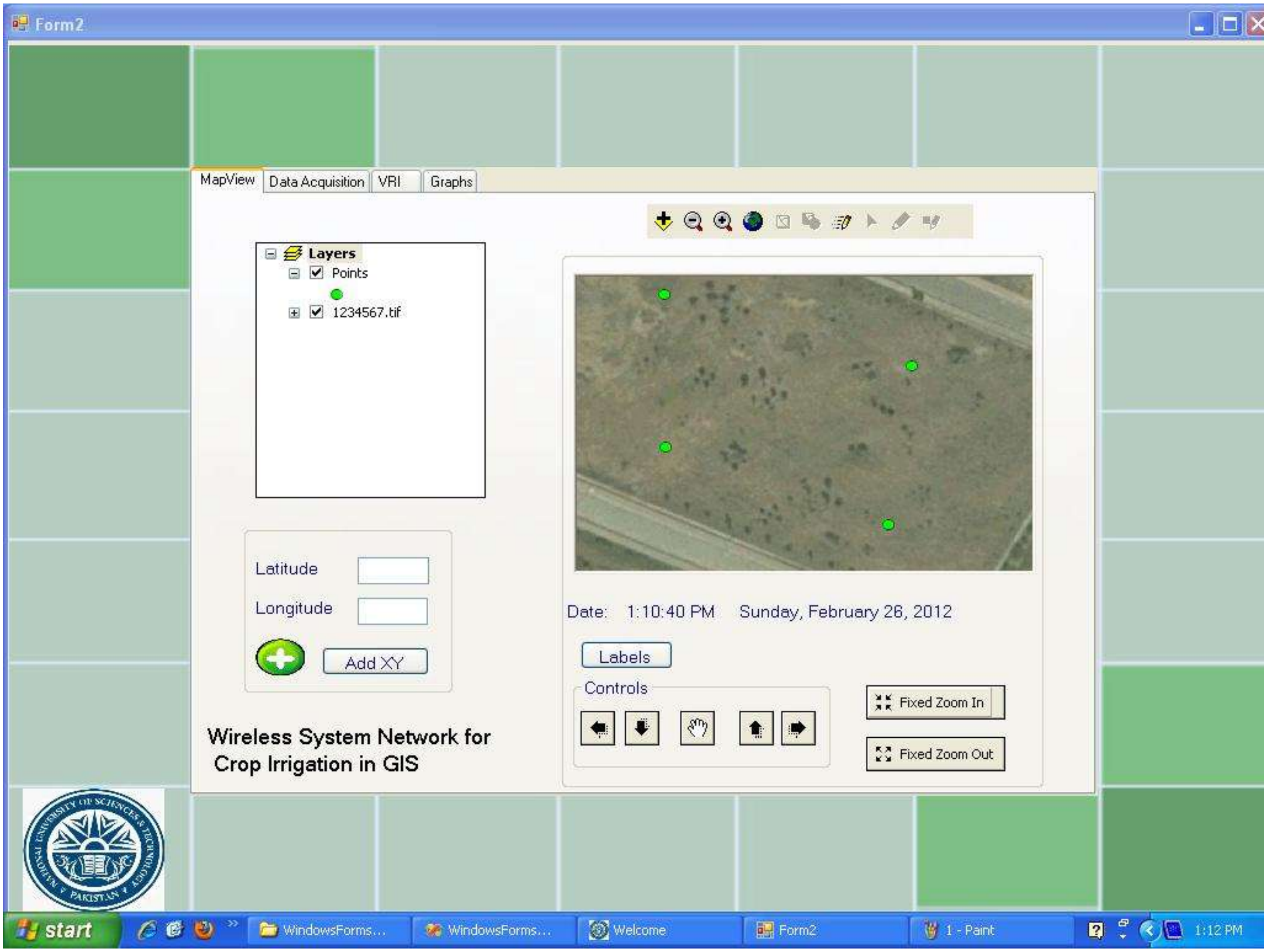


Figure 3.13. Snapshot of standalone GIS application developed.

CONCLUSIONS

Application of water for irrigating the crop field at the right time and in the right amount is mandatory for consistently high crop yields. For real time applications, it is unavoidable to minimize acquisition time frames, and remove intermediate processes that are typically required to get information from the field into a GIS application. Therefore, this research illustrates Wireless System to be installed where time, money and efforts are minimized enabling correct decision, each time it is required and for every situation. WSN is an emerging technology, which in the form presented in this research, has been available for around 5-6 years. Integration of wireless sensor network with geographic information system creates an easy and quick map by using huge amount of run time data. The networks are made up of tiny intelligent sensing devices with cheap and easily installed wireless communication facilities technique, to prepare an accurate schedule plan for irrigation.

An attempt has been made to come up with most cost effective way of irrigation with wireless technology to collect the data from crop field and transfer it to the computer thereby generating soil moisture and temperature map of the area and designing of variable rate irrigation system by using (GIS) techniques. The use of modern techniques of irrigation system which could be tailored according to the requirements of different soils and crops could save water.

RECOMMENDATIONS

Following are the suggestions which can be applied for the betterment of the application:

- To improve accuracy and efficiency of sensor, by using microcontrollers like PIC.
- Solar batteries can be used to reduce the cost of external batteries and for prolonged functioning of the sensors with automatic recharging capability.
- Control and monitor irrigation system over the internet is getting faster and faster using web based GIS applications. The application can be programmed as a web based application so that this could be accessed by the farmers over the internet.
- Graphical simulations at the specific location can be developed, from which farmers can see precisely how much water their plants use.
- The application may be enhanced further by converting it into a pivot system.

5.1 FUTURE RESEARCH OPPORTUNITIES

GSM has been continually enhanced to provide platforms that deliver an increasingly broad range of services as demand grows. GSM service providers are well networked globally. Application of GSM in irrigation can be preferred over WSN, as it covers a vast area and communication is much faster than WSN.

REFERENCES

Clark J.S., Agarwal P., Bell D.M., Flikkema P.G., Gelfand A., Nguyen X.L., Ward E., Yang J. (2011) Inferential ecosystem models, from network data to prediction. Ecological Applications 21:1523-1536.

Coates R.W., Delwiche M.J. (2009) Wireless Mesh Network for Irrigation Control and Sensing. Transactions of the ASABE 52:971-981.

Kruger M., Grosse C. (2009) Event-based Data Acquisition in Wireless Sensor Networks for Structural Health Monitoring. Tm-Technisches Messen 76:568-577.

Gardner, Walter H. (1979). How water moves in the soil, Crops and Soils Magazine. American Society of Agronomy, Inc., Madison, Wisconsin.

Ong J.B., You Z.P., Mills-Beale J., Tan E.L., Pereles B.D., Ong K.G. (2008) A Wireless, Passive Embedded Sensor for Real-Time Monitoring of Water Content in Civil Engineering Materials. IEEE Sensors Journal 8:2053-2058.

CC2420 2.4 GHz IEEE 802.15.4 / Zigbee RF Transceiver; Available online: www.alldatasheet.co.kr (accessed on 6 September 2010).

Zhong, Tang Yue, Ruan Qian, and Cao Jin Zhang.(2009).Development of Integrated Data Platform for Power Grid Companies. Power System Technology, 17: 23-44.

Hasan, M. H., Khan, R. A. (2002). Pakistan Water Sector Strategy. Ministry of Water and Supply Bernhardsen, T. (2002) . Geographic Information System, John Wiley and Sons, Inc., USA

Xijun, Y., Limei, L., Lizhong, X. (2009). The Application of Wireless Sensor Network in the Irrigation Area Automatic System. International Conference on Networks Security, Wireless Communications and Trusted Computing. Dubai

Rehman, A., Shaikh, Z. A., Yousaf, H., Nawaz, F., Kirmani, M., & Kiran, S. (2010). Crop Irrigation Control using Wireless Sensor and Actuator Network (WSAN).

Riquelme J.A.L., Soto F., Suardiaz J., Sanchez P., Iborra A., Vera J.A. (2009) Wireless Sensor Networks for precision horticulture in Southern Spain. Computers and Electronics in Agriculture 68:25-35.

USC Precision Agriculture; Available online: http://www.gpoaccess.gov/uscode/index.html (accessed on 3 December 2010).

Ruiz-Garcia, L., Lunadei, L., Barreiro, P., & Robla, J. I. (2009). A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends. Sensors 51 (1-3), 23

Encyclopedia (2010). Pakistan – Agriculture. http://www.nationsencyclopedia.com/economies/Asia-and-the-Pacific/Pakistan-AGRICULTURE.html

WDR (2006). Equity and Development -World Bank (accessed on 6 August 2006).

Lee, M.-H.; Shin, C.-S.; Jo, Y.-Y.; Yoe, H.(2009). Implementation of Green House Integrated System in Ubiquitous Agricultural Environments. J. KIISE, 27, 21-26.

Wraith, Jon M., David A. Robinson, Scott B. Jones, and Dan S. Long (2005), spatially characterizing apparent electrical conductivity and water content of surface soils with time domain reflectometry, Computer Electron. Agric. 46(1-3), 239

Zeiler, M. (1999). Modeling Our World. The ESRI's Guide to Geodatabase Design. ESRI Press, Redlands, CA.

Boike, Julia, and Kurt Roth (1997). Time domain reflectometry as a field method for measuring water content and soil water electrical conductivity at a continuous permafrost site, Permafrost Periglac Process, 8(4), 359

Pierce, F. J., Elliott, T. V. (2008), The Soil Moisture Sensor. http://gardenbot.org/howTo/soilMoisture/ Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington. Computers and Electronics in Agriculture 61: 32–43.

J. S. Wallace. (2000). Increasing agricultural water use efficiency to meet future food production.Agriculture, Ecosystems & Environment, 82(1-3):105–119

Sudha M.N., Valarmathi M.L., Babu A.S. (2011) Energy efficient data transmission in automatic irrigation system using wireless sensor networks. Computers and Electronics in Agriculture 78:215-221.

Bencini, L., Palma, D. D., Collodi, G., and Manes, G. (2003).Wireless Sensor Networks for on field Agricultural Management Process. Department of Electronics and Telecommunications - University of Florence, Italy

Kim Y., Jabro J.D., Evans R.G. (2011) Wireless lysimeters for real-time online soil water monitoring. Irrigation Science 29:423-430.

Zhou, Y.M.; Yang, X.L.; Guo, X.S.; Zhou, M.G.; Wang, L.R. (2007). A design of greenhouse monitoring & control system based on ZigBee Wireless Sensor

Network. In 2007 International Conference on Wireless Communications, Networking and Mobile Computing Shangai, China; 21-25 September 2007.

Liu, H.; Meng, Z.; Cui, S. A (2007). Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses. In Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing Shangai, China, 21–25 September 2007.

Hamrita, T.K.; Hoffacker, E.C. (2005). Development of a "smart" wireless soil monitoring sensor prototype using RFID technology. Appl. Eng. Agric. 21, 139-143.

Pierce, F.J.; Elliott, T.V. (2005).Regional and on-Farm Wireless Sensor Networks for Agricultural Systems in Eastern Washington. Comput. Electron. Agric. 61, 32-43.

Zhang R.B., Guo J.J., Zhang L., Zhang Y.C., Wang L.H., Wang Q. (2011). A calibration method of detecting soil water content based on the information-sharing in wireless sensor network. Comput. Electron. Agric 76:161-168.

Park, D.-H.; Kang, B.-J.; Cho, K.-R.; Sin, C.-S.; Cho, S.-E.; Park, J.-W.; Yang, W.-M. A Study on Greenhouse Automatic Control System Based on Wireless Sensor Network. Wireless Pers Commun.. 21, 32-43.

Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. (2006.)Wireless sensor networks: a survey. Comput. Network. 38, 393-422.

Technology, C. Website of Crossbow Technology. Available online: http://www.xbow.com. (accessed on 8th December 2010).

Han, W.; Zhang, N.; Zhang, Y.(2008). A two-layer Wireless Sensor Network for remote sediment monitoring. In 2008 ASABE Annual International Meeting, Rhode Island, USA.

Wang, F. (1990). Fuzzy supervised classification of remote sensing images. IEEE Trans. Geosci. Remote Sens. 28:194–20

Malczewski J, (1997). Spatial decision support systems. The NCGIA Core Curriculum in GIScience, Unit 127. http://www.ncgia.ucsb.edu/ giscc/units/u127/S. (accessed on 26^{h} December 2010).

Batty, M. 1993. Using Geographic Information Systems in Urban Planning and Policy-making. In M.M. Fischer and P.

Zhu, A.X. (1999). A personal construct-based knowledge acquisitionprofile observations. J. Soil Sci. 43:193–210. process for natural resource mapping using GIS. Int. J. Geographic Burrough, P.A., P. van Gaans, and R. Hootsmans. Continuous Information Science 13:119–141

Li Z., Wang N., Hong T.S., Franzen A., Li J.N.A. (2011). Closed-loop drip irrigation control using a hybrid wireless sensor and actuator network. Science China-Information Sciences 54:577-588.

Peterson, C. (1991). Precision GPS navigation for improving agricultural point out that qualify of soil information produced using productivity. GPS World 2:38-4

Band, L.E., and I.D. Moore. (1995). Scale: Landscape attributes and Zevenbergen, L.W., and C.R. Thorne. 1987. Quantitative analysis of geographical information systems. Hydrol. Processes 9:401–422

Pardossi A., Incrocci L. (2011) Traditional and New Approaches to Irrigation Scheduling in Vegetable Crops. Horttechnology 21:309-313.

Mohsenimanesh A., Ward S.M. (2007). On-the-move monitoring of soil-tire interaction on soft soil using wireless data acquisition. Transactions of the ASABE 50:1919-1925.

Appendices

Appendix-1.Main project file source code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using ESRI.ArcGIS.GlobeCore;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.Controls;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.esriSystem;
using ESRI.ArcGIS.ADF;
using ESRI.ArcGIS.Framework;
using ESRI.ArcGIS.GeoAnalyst;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.DataSourcesFile;
using ESRI.ArcGIS.EditorExt;
using ESRI.ArcGIS.DataSourcesGDB;
using ESRI.ArcGIS.DataSourcesRaster;
using ESRI.ArcGIS.SpatialAnalyst;
using ESRI.ArcGIS.Editor;
```

```
using System.Drawing.Drawing2D;
using ESRI.ArcGIS.ArcMapUI;
using ESRI.ArcGIS.Controls;
using ESRI.ArcGIS.ToolbarControl;
using ESRI.ArcGIS.SystemUI;
using ESRI.ArcGIS.Display;
using ZedGraph;
using System.Data.OleDb;
using System.Data.Sql;
using System.IO;
using Final;
namespace WindowsFormsApplication1
{
    public partial class Form2 : Form
    {
        public IRow pRow;
        public IMap mymap;
        public ILayer pFLayer;
        public DataTable datatable;
        public IFeatureLayer pFeatureLayer;
        public IFeatureClass pFeatureClass;
        public ICursor pCursor;
         string RX = "";
        List<Final.ReterivedData> reterivedData = new List<Final.ReterivedData>();
        public IField pfield;
        public IFields pfields;
        public IQueryFilter pfilter;
```

```
public ITable pTable;
public IFeatureSelection pFeatureSelection;
double[] xaxis = new double[5] { 1, 2, 3, 4, 5 };
double[] AWSC = new double[3];
double[] IR = new double[3];
double Species_factor;
double Density_factor;
double Microclimate_factor;
double Availble_Water_Storage_Capacity;
double Application_efficency;
int position = 0;
double[] hi = new double[3];
double[] avg = new double[3];
double[] low = new double[3];
IMap pMap;
  DataTable dt = new DataTable();
DataRow row;
 int[] sprinkler_system = new int[1];
bool flag_Area;
bool flag_buffer = false;
bool flag = false;
double area;
double area1;
bool flag_rb;
bool flag_rb1;
double[] Soil_Mositure = new double[] ;
double[] temperature = new double[];
```

```
double[] temperature2 = new double[]

double[] temperature3 = new double[] ;

double[] temperature4 = new double[] ;

public Form2()

{

    InitializeComponent();


    dt.Columns.Add("Sensor");

    dt.Columns.Add("Temperature(C)");

    dt.Columns.Add("Soil Moisture(%)");

    dt.Columns.Add("Time");

  dateTimePicker1.Format = DateTimePickerFormat.Time;

 dateTimePicker1.CustomFormat = " hh mm ss";

    label1.Text = dateTimePicker1.Text;

    label2.Text = dateTimePicker2.Text;

}

public void sensor1(double a, double b, String c)

{

    System.Data.OleDb.OleDbConnection conn = new
    System.Data.OleDb.OleDbConnection();

    conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data
    Source=D:/SIBGHAT_THESIS_GEODATABASE/abc.accdb";

    conn.Open();

    string sql = "INSERT INTO Sensor1 (Temperature, Moisture, Tim)
    VALUES(@Temperature, @Moisture, @Tim)";

    OleDbCommand z = new OleDbCommand();

    z.CommandText = sql;

    z.Connection = conn;
```

```
    OleDbParameter param = new OleDbParameter("@Temperature", a);
    z.Parameters.Add(param);
    OleDbParameter para = new OleDbParameter("@Moisture", b);
    z.Parameters.Add(para);
    OleDbParameter para1 = new OleDbParameter("@Tim", c);
    z.Parameters.Add(para1);
    z.ExecuteNonQuery();
    conn.Close();
}//Database Connection
public void sensor2(double a, double b, string c)
{
     System.Data.OleDb.OleDbConnection conn = new
    System.Data.OleDb.OleDbConnection();
    conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data
    Source=D:/SIBGHAT_THESIS_GEODATABASE/abc.accdb";
    conn.Open();
    string sql = "INSERT INTO Sensor2 (Temperature, Moisture, Tim)
    VALUES(@Temperature, @Moisture, @Tim)";
    OleDbCommand z = new OleDbCommand();
    z.CommandText = sql;
    z.Connection = conn;

    OleDbParameter param = new OleDbParameter("@Temperature", a);
    z.Parameters.Add(param);

    OleDbParameter para = new OleDbParameter("@Moisture", b);
    z.Parameters.Add(para);
    OleDbParameter para1 = new OleDbParameter("@Tim", c);
```

```
            z.Parameters.Add(para1);

            z.ExecuteNonQuery();

            conn.Close();

        }


        public void sensor3(double a, double b, string c)

        {

            System.Data.OleDb.OleDbConnection conn = new
System.Data.OleDb.OleDbConnection();

            conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data
Source=D:/SIBGHAT_THESIS_GEODATABASE/abc.accdb";

            conn.Open();


            string sql = "INSERT INTO Sensor3 (Temperature, Moisture, Tim)
            VALUES(@Temperature, @Moisture, @Tim)";

            OleDbCommand z = new OleDbCommand();

            z.CommandText = sql;

            z.Connection = conn;

            OleDbParameter param = new OleDbParameter("@Temperature", a);

            z.Parameters.Add(param);

            OleDbParameter para = new OleDbParameter("@Moisture", b);

            z.Parameters.Add(para);

            OleDbParameter para1 = new OleDbParameter("@Tim", c);

            z.Parameters.Add(para1);

            z.ExecuteNonQuery();

            conn.Close();

        }

        public void sensor4(double a, double b, string c)
```

```
{
    System.Data.OleDb.OleDbConnection conn = new
    System.Data.OleDb.OleDbConnection();
    conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data
    Source=D:/SIBGHAT_THESIS_GEODATABASE/abc.accdb";
    conn.Open();
    string sql = "INSERT INTO Sensor4 (Temperature, Moisture, Tim)
    VALUES(@Temperature, @Moisture, @Tim)";
    OleDbCommand z = new OleDbCommand();
    z.CommandText = sql;
    z.Connection = conn;
    OleDbParameter param = new OleDbParameter("@Temperature", a);
    z.Parameters.Add(param);
    OleDbParameter para = new OleDbParameter("@Moisture", b);
    z.Parameters.Add(para);
    OleDbParameter para1 = new OleDbParameter("@Tim", c);
    z.Parameters.Add(para1);
    z.ExecuteNonQuery();
    conn.Close();
}
      private ESRI.ArcGIS.Geometry.IPoint CreatePoint(double x, double y)
     {
        ESRI.ArcGIS.Geometry.IPoint p = new ESRI.ArcGIS.Geometry.Point();
       p =
    axMapControl1.ActiveView.ScreenDisplay.DisplayTransformation.ToMapPoint((int)x,
    (int)y);
        return p;
     }
     private void button3_Click_1(object sender, EventArgs e)
```

```
{
   zedGraphControl1.GraphPane.CurveList.Clear();
    GraphPane myPane = zedGraphControl1.GraphPane;
    PointPairList spl1 = new PointPairList(xaxis, yaxis);
 LineItem myCurve1 = myPane.AddCurve("Sensor1", spl1, Color.Brown,
SymbolType.Triangle);
    myCurve1.Line.Width = 3.0F;
    myPane.Title.Text = "Graph";
    myCurve1.Symbol.Fill.RangeMin = 0;
    myCurve1.Symbol.Fill.RangeMax = 20;
    myCurve1.Symbol.Type = SymbolType.Triangle;
    zedGraphControl1.AxisChange();
    myPane.YAxis.MajorGrid.DashOff = 10;
    //  myPane.XAxis.MajorGrid.IsVisible = true;
    //  myPane.YAxis.MajorGrid.IsVisible = true;
    myPane.XAxis.Scale.Min = 0;
    myPane.XAxis.Scale.MajorStep = 1;
    myPane.YAxis.Scale.MajorStep = 0.5;
    myPane.XAxis.Title.Text = "Time (s)";
    myPane.YAxis.Title.Text = "Soil Moisture";
    //zedGraphControl1.Refresh();


    PointPairList spl12 = new PointPairList(xaxis, yaxis1);
   LineItem myCurve12 = myPane.AddCurve("Sensor2", spl12, Color.Blue,
SymbolType.Triangle);
    myCurve12.Line.Width = 3.0F;
    myCurve12.Symbol.Fill.RangeMin = 0;
    myCurve12.Symbol.Fill.RangeMax = 20;
```

```
myCurve12.Symbol.Type = SymbolType.Triangle;
zedGraphControl1.AxisChange();
// myPane.YAxis.MajorGrid.DashOff = 1;
myPane.XAxis.Scale.Min = 0;
myPane.XAxis.Scale.MajorStep = 1;
myPane.YAxis.Scale.MajorStep = 0.5;
//zedGraphControl1.Invalidate();
//zedGraphControl1.Refresh();
PointPairList spl13 = new PointPairList(xaxis, yaxis2);
        LineItem myCurve13 = myPane.AddCurve("Sensor3", spl13, Color.Black,
SymbolType.Triangle);
myCurve13.Line.Width = 3.0F;
myCurve13.Symbol.Fill.RangeMin = 0;
myCurve13.Symbol.Fill.RangeMax = 20;
myCurve13.Symbol.Type = SymbolType.Triangle;
zedGraphControl1.AxisChange();
// myPane.YAxis.MajorGrid.DashOff = 1;
myPane.XAxis.Scale.Min = 0;
myPane.XAxis.Scale.MajorStep = 1;
myPane.YAxis.Scale.MajorStep = 0.5;
zedGraphControl1.Refresh();
PointPairList spl4 = new PointPairList(xaxis, yaxis3);
        LineItem myCurve14 = myPane.AddCurve("Sensor4", spl4, Color.DarkGreen,
SymbolType.Triangle);
myCurve14.Line.Width = 3.0F;
myCurve14.Symbol.Fill.RangeMin = 0;
myCurve14.Symbol.Fill.RangeMax = 20;
myCurve14.Symbol.Type = SymbolType.Triangle;
```

```
    zedGraphControl1.AxisChange();
    // myPane.YAxis.MajorGrid.DashOff = 1;
    myPane.XAxis.Scale.Min = 0;
    myPane.XAxis.Scale.MajorStep = 1;
    myPane.YAxis.Scale.MajorStep = 0.5;
    zedGraphControl1.Refresh();
}
private void button4_Click(object sender, EventArgs e)
{
   zedGraphControl2.GraphPane.CurveList.Clear();
    GraphPane myPane = zedGraphControl2.GraphPane;

    PointPairList spl1 = new PointPairList(xaxis, tempyaxis);
            LineItem myCurve1 = myPane.AddCurve("Sensor1", spl1, Color.Brown,
SymbolType.Triangle);
    myCurve1.Line.Width = 3.0F;
    myPane.Title.Text = "Graph";
    myCurve1.Symbol.Fill.RangeMin = 0;
    myCurve1.Symbol.Fill.RangeMax = 20;
    myCurve1.Symbol.Type = SymbolType.Triangle;
    zedGraphControl2.AxisChange();
    myPane.YAxis.MajorGrid.DashOff = 10;
    //  myPane.XAxis.MajorGrid.IsVisible = true;
    //  myPane.YAxis.MajorGrid.IsVisible = true;
    myPane.XAxis.Scale.Min = 0;
    myPane.XAxis.Scale.MajorStep = 1;
    myPane.YAxis.Scale.MajorStep = 0.5;
    myPane.XAxis.Title.Text = "Time (s)";
```

```
myPane.YAxis.Title.Text = "Temperature (C)";
zedGraphControl2.Refresh();
GraphPane myPane2 = zedGraphControl2.GraphPane;
PointPairList spl12 = new PointPairList(xaxis, tempyaxis1);
        LineItem myCurve12 = myPane.AddCurve("Sensor2", spl12, Color.Blue,
SymbolType.Triangle);
myCurve12.Line.Width = 3.0F;
myPane2.Title.Text = "Graph";
myCurve12.Symbol.Fill.RangeMin = 0;
myCurve12.Symbol.Fill.RangeMax = 20;
myCurve12.Symbol.Type = SymbolType.Triangle;
zedGraphControl2.AxisChange();
myPane.YAxis.MajorGrid.DashOff = 10;
//  myPane.XAxis.MajorGrid.IsVisible = true;
//  myPane.YAxis.MajorGrid.IsVisible = true;
myPane2.XAxis.Scale.Min = 0;
myPane2.XAxis.Scale.MajorStep = 1;
myPane2.YAxis.Scale.MajorStep = 0.5;
myPane2.XAxis.Title.Text = "Time (s)";
myPane2.YAxis.Title.Text = "Temperature (C)";
//zedGraphControl1.Refresh();
GraphPane myPane3 = zedGraphControl2.GraphPane;
PointPairList spl13 = new PointPairList(xaxis, tempyaxis2);
        LineItem myCurve13 = myPane.AddCurve("Sensor3", spl13, Color.Black,
SymbolType.Triangle);
myCurve13.Line.Width = 3.0F;
myPane3.Title.Text = "Graph";
myCurve13.Symbol.Fill.RangeMin = 0;
```

```
myCurve13.Symbol.Fill.RangeMax = 20;
myCurve13.Symbol.Type = SymbolType.Triangle;
zedGraphControl2.AxisChange();
myPane.YAxis.MajorGrid.DashOff = 10;
//  myPane.XAxis.MajorGrid.IsVisible = true;
//  myPane.YAxis.MajorGrid.IsVisible = true;
myPane3.XAxis.Scale.Min = 0;
myPane3.XAxis.Scale.MajorStep = 1;
myPane3.YAxis.Scale.MajorStep = 0.5;
myPane3.XAxis.Title.Text = "Time (s)";
myPane3.YAxis.Title.Text = "Temperature (C)";
//zedGraphControl1.Refresh();
GraphPane myPane4 = zedGraphControl2.GraphPane;
PointPairList spl14 = new PointPairList(xaxis, tempyaxis3);
        LineItem myCurve14 = myPane.AddCurve("Sensor4", spl14,
Color.DarkGreen, SymbolType.Triangle);
myCurve14.Line.Width = 3.0F;
myPane4.Title.Text = "Graph";
myCurve14.Symbol.Fill.RangeMin = 0;
myCurve14.Symbol.Fill.RangeMax = 20;
myCurve14.Symbol.Type = SymbolType.Triangle;
zedGraphControl2.AxisChange();
myPane.YAxis.MajorGrid.DashOff = 10;
//  myPane.XAxis.MajorGrid.IsVisible = true;
//  myPane.YAxis.MajorGrid.IsVisible = true;
myPane4.XAxis.Scale.Min = 0;
myPane4.XAxis.Scale.MajorStep = 1;
myPane4.YAxis.Scale.MajorStep = 0.5;
```

```
    myPane4.XAxis.Title.Text = "Time (s)";
    myPane4.YAxis.Title.Text = "Temperature (C)";
    zedGraphControl2.Refresh();
}
private void button5_Click(object sender, EventArgs e)
{
    axMapControl2.Visible = false;
    axMapControl3.Visible = false;
    string sFieldName;
    IFeatureClassDescriptor pFCDescr;
    IRasterLayer pRasterLayer;
    IGeoDataset pMaskDataset;
    ILayer pLayer;
    ILayer pLayer2;
    IRaster pRaster;

//axMapControl1.AddShapeFile("D:/SIBGHAT_THESIS_GEODATABASE/Thesis
shapefile/", "Points.shp");
    pFLayer = axMapControl1.Map.get_Layer(getLayerID("Points"));
    pFeatureLayer = (IFeatureLayer)pFLayer;
    pFeatureClass = pFeatureLayer.FeatureClass;
    sFieldName ="temp";
    pFCDescr = new FeatureClassDescriptorClass();
    pFCDescr.Create(pFeatureClass, null, sFieldName);
   pRasterLayer = (IRasterLayer)pLayer2;
    pMaskDataset = (IGeoDataset)pRasterLayer.Raster;
  object pMaskDataset = new ObjectClass();
    object pMDataset = 0.000003;// pMaskDataset;
```

```
IInterpolationOp pIntOp;
IRasterAnalysisEnvironment pEnv;
IRasterRadius pRadius;
IRaster pOutRaster;
pIntOp = new RasterInterpolationOpClass();
pEnv = (IRasterAnalysisEnvironment)pIntOp;
pEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvValue, ref pMDataset);
pRadius = new RasterRadiusClass();
object Missing = Type.Missing;
pRadius.SetVariable(12, ref Missing);
        pOutRaster = pIntOp.IDW(pFCDescr as IGeoDataset, 2, pRadius, ref Missing)
as IRaster;

IRasterLayer pOutRasLayer = new RasterLayerClass();
pOutRasLayer.CreateFromRaster(pOutRaster);
pOutRasLayer.Name = "Temperture ";
axMapControl1.AddLayer(pOutRasLayer);

//  IMap pMap = axMapControl1.Map;
// IFeatureLayer pFeatureLayer;
//IFeatureClass pFeatureClass;
//  string sFieldName;
//  IFeatureClassDescriptor pFCDescr;
//  IRasterLayer pRasterLayer;
//  IGeoDataset pMaskDataset;
// ILayer pLayer;
// ILayer pLayer2;
// IRaster pRaster;
```

```
  // axMapControl1.AddShapeFile("D:/Thesis shapefile/Geographic/Corrected",
"Sensors.shp");
  //  pFLayer = axMapControl1.Map.get_Layer(this.getLayerID("Sensors"));
  //  pFeatureLayer = (IFeatureLayer)pFLayer;
  //  pFeatureClass = pFeatureLayer.FeatureClass;



  //  sFieldName = "Moisture";
  //  pFCDescr = new FeatureClassDescriptorClass();
  //  pFCDescr.Create(pFeatureClass, null, sFieldName);
  //  pLayer2=pMap.get_Layer(this.getLayerID("NUST H-12 Islamabad.jpg"));
  //  pRasterLayer=(IRasterLayer)pLayer2;
  //  pMaskDataset = (IGeoDataset)pRasterLayer.Raster;
  //  object pMDataset = .000001;// pMaskDataset;
  //  IInterpolationOp pIntOp;
  //  IRasterAnalysisEnvironment pEnv;
  //  IRasterRadius pRadius;
  //  IRaster pOutRaster;
  //  pIntOp = new RasterInterpolationOpClass();
  //  pEnv = (IRasterAnalysisEnvironment)pIntOp;
  //  pEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvValue, ref pMDataset);
  //  pRadius = new RasterRadiusClass();
  //  object Missing = Type.Missing;
  //  pRadius.SetVariable(12, ref Missing);
            //  pOutRaster = pIntOp.IDW(pFCDescr as IGeoDataset, 2, pRadius, ref
Missing) as IRaster;


  //  IRasterLayer pOutRasLayer = new RasterLayerClass();
```

```
  //  pOutRasLayer.CreateFromRaster(pOutRaster);
  //  pOutRasLayer.Name = "Soil Moisture";
  //  axMapControl1.AddLayer(pOutRasLayer);
  //  tabControl1.SelectedIndex = 0;
}
public int getLayerID(string name)
{
    IMap mymap = axMapControl1.Map;
    ILayer MyLayer;
    for (int i = 0; i < mymap.LayerCount; i++)
    {
        MyLayer = mymap.get_Layer(i);
        if (MyLayer.Name == name)
        {
            return i;
        }
    }
    return -1;
}

public void CreateBufferPolygonAroundPoint(string PointFeatureClassName, string
PolygonFeatureClassName, Double BufferRadius, int fid)
{
    IMap mymap = axMapControl1.Map;
    ILayer myPointLayer = mymap.get_Layer(this.getLayerID("Sprinklers"));
    IFeatureLayer myPointFeatureLayer = (IFeatureLayer)myPointLayer;
    IFeatureClass myPointFeatureClass = myPointFeatureLayer.FeatureClass;
    IFeature myPointFeature = myPointFeatureClass.GetFeature(fid);
```

```
    //IFeature myPointFeature = myPointFeatureClass.
            ITopologicalOperator MyTopologicalOperator =
(ITopologicalOperator)myPointFeature.Shape;
    //A degree of longitude at the equator is 111.2 kilometers.
    double Meters = BufferRadius;
    double Distance_m = Meters / 111200.000000000;
    //IGeometry MyShape = (IGeometry)MyTopologicalOperator.Buffer(Distance_m);
    //IGeometry MyShape = (IGeometry)MyTopologicalOperator.Buffer(10);
    IPolygon myPolygon = MyTopologicalOperator.Buffer(Distance_m) as IPolygon;
    //IGeometry MyShape = (IGeometry)MyTopologicalOperator.
    ILayer mylayer = mymap.get_Layer(this.getLayerID("buffer"));
    IFeatureLayer MyFeatureLayer = (IFeatureLayer)mylayer;
    IFeatureClass MyFeatureClass = MyFeatureLayer.FeatureClass;
    IFeature MyFeature = MyFeatureClass.CreateFeature();
    MyFeature.Shape = myPolygon;
    MyFeature.Store();
    //axMapControl1.AddLayer(mylayer);
    axMapControl1.Refresh();
    tabControl1.SelectedIndex = 0;
}
private void button6_Click(object sender, EventArgs e)
{
    if (radioButton2.Checked == true)
    {
        axMapControl1.Visible = false;
        axMapControl3.Visible = false;
        axMapControl2.Visible = true;
        tabControl1.SelectedIndex = 0;
```

```
            }
            if (radioButton1.Checked == true)
            {
                axMapControl1.Visible = false;
                axMapControl2.Visible = false;
                axMapControl3.Visible = true;
                tabControl1.SelectedIndex = 0;
            }
axMapControl1.AddShapeFile("D:/Thesis shapefile", "Sprinklers.shp");
            if (radioButton1.Checked==true)
            {
              // axMapControl1.AddShapeFile("D:/Thesis shapefile", "buffer.shp");
               for (int w = 0; w < 4; w++)
               {
                   CreateBufferPolygonAroundPoint("Sprinklers", "Buffer", 15, w);
               }
            }
            else if (radioButton2.Checked==true)
            {
              // axMapControl1.AddShapeFile("D:/Thesis shapefile/Geographic", "buffer.shp");
               for (int w = 0; w < 4; w++)
               {

                 //CreateBufferPolygonAroundPoint("Sprinklers", "Buffer", 10, w);

               }
            }
```

```
if (flag_Area == true)
{
   pFLayer = (IFeatureLayer)axMapControl1.get_Layer(this.getLayerID("Sprinklers"));

   pFeatureLayer = (IFeatureLayer)pFLayer;
   pFeatureClass = pFeatureLayer.FeatureClass;

   pTable = (ITable)pFeatureClass;
   pfields = pTable.Fields;
   pFeatureSelection = pFeatureLayer as IFeatureSelection;
   int counter = pfields.FieldCount;
   counter = 3;
   pfield = pfields.get_Field(1);
   //comboBox1.Items.Add(pfield.Name);
   pfilter = new QueryFilterClass();
   pfilter.WhereClause = "";
   pCursor = pTable.Search(pfilter, false);
   //datatable = new DataTable();
   // //datatable.Columns.Add("Soil");
   // //datatable.Columns.Add("Grass");
   // //datatable.Columns.Add("Area");

   //// dataGridView1.DataSource = datatable;
   //int[] temppp = new int[5];
   //for (int count = 0; count < 4; count++)
{
              pRow = pCursor.NextRow();
```

```
comboBox1.Items.Add(pRow.get_Value(2));
comboBox2.Items.Add(prow.get_Value(6));
temp[count] = int.Parse(pRow.get_Value(3).ToString());
 MessageBox.Show(tempppp[count].ToString());
pRow.Delete();
ary[count] = count;
    }

    tabControl1.SelectedIndex = 0;
}
 }

 private void button11_Click(object sender, EventArgs e)
 {
     try
     {
         SaveFileDialog save = new SaveFileDialog();
         save.Filter = "CSV File (*.csv)|*.csv";
         save.ShowDialog();
         File.WriteAllText(save.FileName, ReturnAsCSV());
     }
     catch (Exception exc)
     {
         MessageBox.Show("Please Stop Retrieving Data First", "ERROR",
MessageBoxButtons.OK, MessageBoxIcon.Information);
     }
 }
 public string ReturnAsCSV()
```

```
{
        StringBuilder sb = new StringBuilder();
    sb.AppendLine("Device Number ");//,Moisture,Temprature");
      foreach (DataGridViewRow dgvr in dataGridView1.Rows)
    {
                sb.AppendLine(temperature.ToString() );//+ "," + c.Moisture + "," +
c.Temperature + ",");
    }
    sb.Remove(sb.Length - 1, 1);
    //sb.Replace(",", "", sb.Length - 1, 1)

    return sb.ToString();
}

private void button16_Click(object sender, EventArgs e)
{
    try
    {
        dt.Clear();
    }
    catch (Exception exc)
    {
                MessageBox.Show("Stop Retreiving Data ", "ERROR",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
            }
```

```
private void comboBoxVegitationType_SelectedIndexChanged(object sender,
EventArgs e)

{

    if (comboBoxVegitationType.SelectedItem.ToString() == "Trees")

    {

      hi[0] = x;

       avg[0] y;

       low[0] = z;

    }

    if (comboBoxVegitationType.SelectedItem.ToString() == "Shrubs")

    {

       hi[0] = x;

       avg[0] y;

       low[0] = z;

    }

    if (comboBoxVegitationType.SelectedItem.ToString() == "Ground cover")

    {

       hi[0] = x;

       avg[0] = y;

       low[0] z;


    }

    if (comboBoxVegitationType.SelectedItem.ToString() == "Mixed")

    {

       hi[0] = x;

       avg[0] =y;

       low[0] = z;

    }
```

```
    if (comboBoxVegitationType.SelectedItem.ToString() == "Grass")
    {

      avg[2] = 1.0;
       low[2] = 0.8;
    }
}
private void comboBoxSoilType_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBoxSoilType.SelectedItem.ToString() == "Clay")
    {
        IR[0] = 0.25;
        AWSC[0] = 0.20;
        sprinkler_system[0] = 70;
        textBox7MAD.Text = sprinkler_system[0].ToString();
    }
    if (comboBoxSoilType.SelectedItem.ToString() == "Silt loam")
    {
        IR[0] = 0.35;
        AWSC[0] = 0.21;
        sprinkler_system[0] = 80;
        textBox7MAD.Text = sprinkler_system[0].ToString();
    }
    if (comboBoxSoilType.SelectedItem.ToString() == "Clay loam")
    {
        IR[0] = 0.30;
        AWSC[0] = 0.20;
```

```
    sprinkler_system[0] = 80;
    textBox7MAD.Text = sprinkler_system[0].ToString();
}
if (comboBoxSoilType.SelectedItem.ToString() == "Loam")
{
    sprinkler_system[0] = 100;
    IR[0] = 0.35;
    AWSC[0] = 0.18;
    textBox7MAD.Text = sprinkler_system[0].ToString();
}
if (comboBoxSoilType.SelectedItem.ToString() == "Fine sandy loam")
{
    sprinkler_system[0] = 100;
    IR[0] = 0.40;
    AWSC[0] = 0.14;
    textBox7MAD.Text = sprinkler_system[0].ToString();
}
if (comboBoxSoilType.SelectedItem.ToString() == "Sandy loam")
{
    sprinkler_system[0] = 100;
    IR[0] = 0.45;
    AWSC[0] = 0.12;
    textBox7MAD.Text = sprinkler_system[0].ToString();
}
if (comboBoxSoilType.SelectedItem.ToString() == "Loamy Sand")
{
    sprinkler_system[0] = 100;
```

```
        IR[0] = 0.65;
        AWSC[0] = 0.10;
        textBox7MAD.Text = sprinkler_system[0].ToString();
    }
    if (comboBoxSoilType.SelectedItem.ToString() == "Sand")
    {
        sprinkler_system[0] = 100;
        IR[0] = 0.75;
        AWSC[0] = 0.08;
        textBox7MAD.Text = sprinkler_system[0].ToString();
    }
 }
private void comboBoxwaterusage_SelectedIndexChanged(object sender, EventArgs e)
 {
    if (comboBoxwaterusage.SelectedItem.ToString() == "High")
    {
        Species_factor = hi[0];
        textBoxspeciefactor.Text = hi[0].ToString();
    }
    if (comboBoxwaterusage.SelectedItem.ToString() == "Average")
    {
        textBoxspeciefactor.Text = "";
        Species_factor = avg[0];
        textBoxspeciefactor.Text = avg[0].ToString();
    }
    if (comboBoxwaterusage.SelectedItem.ToString() == "Low")
    {
```

```
        textBoxspeciefactor.Text = "";
        Species_factor = low[0];
        textBoxspeciefactor.Text = low[0].ToString();
    }
}
private void comboBoxVegitationDensity_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBoxVegitationDensity.SelectedItem.ToString() == "High")
    {
        Density_factor = hi[1];
        textBoxDensityFactor.Text = hi[1].ToString();
    }
    if (comboBoxVegitationDensity.SelectedItem.ToString() == "Average")
    {
        textBoxDensityFactor.Text = "";
        Density_factor = avg[1];

        textBoxDensityFactor.Text = avg[1].ToString();
    }
    if (comboBoxVegitationDensity.SelectedItem.ToString() == "Low")
    {
        textBoxDensityFactor.Text = "";
        Density_factor = low[1];
        textBoxDensityFactor.Text = low[1].ToString();
    }
}
```

```
private void comboBoxMicroclimate_SelectedIndexChanged(object sender, EventArgs
e)
{
    if (comboBoxMicroclimate.SelectedItem.ToString() == "High")
    {
        textBoxMicroclimate_factor.Text = "";
        Microclimate_factor = hi[2];
        textBoxMicroclimate_factor.Text = hi[2].ToString();
    }
    if (comboBoxMicroclimate.SelectedItem.ToString() == "Average")
    {
        textBoxMicroclimate_factor.Text = "";
        Microclimate_factor = avg[2];
        textBoxMicroclimate_factor.Text = avg[2].ToString();
    }
    if (comboBoxMicroclimate.SelectedItem.ToString() == "Low")
    {
        textBoxMicroclimate_factor.Text = "";
        Microclimate_factor = low[2];
        textBoxMicroclimate_factor.Text = low[2].ToString();
    }
}

private void comboBox2Application_SelectedIndexChanged(object sender, EventArgs
e)
{
    if (comboBox2Application.SelectedItem.ToString() == "Microsprinkler")
    {
```

```
        Application_efficency = 0.80;


        textBoxApplication_efficency.Text = Application_efficency.ToString();
    }
    if (comboBox2Application.SelectedItem.ToString() == "Rotor Sprinkler")
    {
        Application_efficency = 0.72;
        textBoxApplication_efficency.Text = Application_efficency.ToString();
    }
    if (comboBox2Application.SelectedItem.ToString() == "Spray heads")
    {
        Application_efficency = 0.70;
        textBoxApplication_efficency.Text = Application_efficency.ToString();
    }
}
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    double Ir = Species_factor * Density_factor * Microclimate_factor;
    textBox10.Text = Ir.ToString();
    double Io = Species_factor * Density_factor * Microclimate_factor * 60;
    textBoxMAD.Text = (Io / 6).ToString();
}
private void button8_Click(object sender, EventArgs e)
{
    double x, y;
   //y = double.Parse(textBox2.Text);
   // x = double.Parse(textBox1.Text);
```

```
    x = 210;
    y = 190;
    DisplayPointOnMap(x, y);
    CreatePoint(x, y);
}

private void DisplayPointOnMap(double x, double y)
{
    IPoint pt = CreatePoint(x, y);
    IGraphicsContainer GraphCont = axMapControl1.ActiveView.GraphicsContainer;
    IElement element = null;
    IMarkerElement markerElem = new MarkerElementClass();
    ISimpleMarkerSymbol markerSymbol = new SimpleMarkerSymbolClass();
    IRgbColor color = new RgbColorClass();
    element = (IElement)markerElem;
    element.Geometry = pt;
    color.Red = 0;
    color.Green = 0;
    color.Blue = 0;

    //IMarkerElement markerElement;

    IEnvelope pEnvelope = null;
    markerSymbol.Color = color;
    markerSymbol.Size = 10;
    markerElem.Symbol = markerSymbol as IMarkerSymbol;
    GraphCont.AddElement(element, 0);
```

```
axMapControl1.ActiveView.PartialRefresh(esriViewDrawPhase.esriViewGraphics, null,
pEnvelope);

 }

 private void button2_Click_1(object sender, EventArgs e)

 {

     DateTime time = DateTime.Now;            // Use current time

     //string format = "ddd";

     Random rand = new Random();

     int step = 1;

     double a;

     double b;

     for (int x = 0; x < 16; x++)

     {


         row = dt.NewRow();


         if (step > 4)

         {

             step = 1;

         }

         row["Sensor"] = step.ToString();

         row["Temperature(C)"] = temperature[x].ToString();

         row["Soil Moisture(%)"] = Soil_Mositure[x].ToString();

         row["Time"] = time.ToString();

         a = temperature[x];

         b = Soil_Mositure[x];

         dt.Rows.Add(row);
```

```
if (step == 1)
{
    sensor1(a, b, time.ToString("ddd"));
}
else if (step == 2)
{
    sensor2(a, b, time.ToString("ddd"));
}
if (step == 3)
{
    sensor3(a, b, time.ToString("ddd"));
}
if (step == 4)
{
    sensor4(a, b, time.ToString("ddd"));
}
step++;
//dataGridView1.Refresh();

System.Threading.Thread.Sleep(1000);
Application.DoEvents();
//dataGridView1.Show();
dataGridView1.DataSource = dt;
foreach (DataGridViewRow dgvr in dataGridView1.Rows)
{

    if (dgvr.Cells[2].Value != null && int.Parse(dgvr.Cells[2].Value.ToString()) <
10)
```

```
            {
                dgvr.DefaultCellStyle.BackColor = Color.Red;
            }
        }
    }
}
        public double Areacalculation(string PolygonFeatureClassName, int fid)
{
    double area = 0;
    IMap mymap = axMapControl2.Map;
    ILayer pLayer = mymap.get_Layer(this.getLayerID("Sprinklers_Buffer1"));
    IFeatureLayer pFeatureLayer = (IFeatureLayer)pLayer;
    IFeatureClass myPointFeatureClass = pFeatureLayer.FeatureClass;
    //int FID = int.Parse("2");
    // IFeature myPointFeature = myPointFeatureClass.GetFeature(fid);
    IFeature myPointFeature = myPointFeatureClass.GetFeature(fid);
    //Ifeature pFeature;
    IArea pArea = myPointFeature.Shape as IArea;
    area = area + pArea.Area;
    return area;
}

private void button9_Click(object sender, EventArgs e)
{
    if (radioButton2.Checked == true)
    {
        area = (15594.264 - 1256) / (Math.PI * 10 * 10);
```

```
                //MessageBox.Show(area.ToString());
                MessageBox.Show("Required " + Math.Round(double.Parse((area-13).ToString()))
+ " more Sprinklers", "Sprinklers", MessageBoxButtons.OK, MessageBoxIcon.Information);
                textBox5.Text = (1256).ToString();
                textBox6.Text = 15594.26.ToString();
            }

            if (radioButton1.Checked == true)
            {
                area = (15594.264 - 2824) / (Math.PI * 15 * 15);
                MessageBox.Show("Required " + Math.Round(double.Parse(area.ToString())) + "
more Sprinklers", "Sprinklers", MessageBoxButtons.OK, MessageBoxIcon.Information);
                textBox5.Text = (2824).ToString();
                textBox6.Text = 15594.26.ToString();
            }
        }

        private void textBox5_TextChanged(object sender, EventArgs e)
         private void button12_Click(object sender, EventArgs e)
        {
            axMapControl2.Visible = false;
            axMapControl3.Visible = false;
            string sFieldName;
            IFeatureClassDescriptor pFCDescr;
            IRasterLayer pRasterLayer;
            IGeoDataset pMaskDataset;
            ILayer pLayer;
            ILayer pLayer2;
```

```
IRaster pRaster;

//axMapControl1.AddShapeFile("D:/SIBGHAT_THESIS_GEODATABASE/Thesis shapefile/", "Points.shp");
pFLayer = axMapControl1.Map.get_Layer(getLayerID("Points"));
pFeatureLayer = (IFeatureLayer)pFLayer;
pFeatureClass = pFeatureLayer.FeatureClass;



sFieldName = "moisture";
pFCDescr = new FeatureClassDescriptorClass();
pFCDescr.Create(pFeatureClass, null, sFieldName);

// pLayer2 = pMap.get_Layer(this.getLayerID("123456.tif"));
// pRasterLayer = (IRasterLayer)pLayer2;
// pMaskDataset = (IGeoDataset)pRasterLayer.Raster;
//// object pMaskDataset = new ObjectClass();
object pMDataset = 0.000003;// pMaskDataset;
IInterpolationOp pIntOp;
IRasterAnalysisEnvironment pEnv;
IRasterRadius pRadius;
IRaster pOutRaster;
pIntOp = new RasterInterpolationOpClass();
pEnv = (IRasterAnalysisEnvironment)pIntOp;
pEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvValue, ref pMDataset);

pRadius = new RasterRadiusClass();
object Missing = Type.Missing;
```

```
    pRadius.SetVariable(12, ref Missing);
            pOutRaster = pIntOp.IDW(pFCDescr as IGeoDataset, 2, pRadius, ref Missing)
as IRaster;


    IRasterLayer pOutRasLayer = new RasterLayerClass();
    pOutRasLayer.CreateFromRaster(pOutRaster);
    pOutRasLayer.Name = "Soil_Moisture ";
    axMapControl1.AddLayer(pOutRasLayer);
}


private void button13_Click(object sender, EventArgs e)
{
    pFLayer = (IFeatureLayer)axMapControl1.get_Layer(this.getLayerID("Points"));
    IGeoFeatureLayer pGeoFeatureLayer;
    pGeoFeatureLayer = (IGeoFeatureLayer)pFLayer;
    string strFieldName = "Id";
   // string area = "moisture";
    IAnnotateLayerPropertiesCollection pAnnoLayerPropsColl;
    pAnnoLayerPropsColl = pGeoFeatureLayer.AnnotationProperties;
    pGeoFeatureLayer.DisplayAnnotation = true;
    IAnnotateLayerProperties pAnnoLayerProps;
    IElementCollection iec;
    pAnnoLayerPropsColl.QueryItem(0, out pAnnoLayerProps, out iec, out iec);
    ILabelEngineLayerProperties pLabelEngineLayerProps;
    pLabelEngineLayerProps = (ILabelEngineLayerProperties)pAnnoLayerProps;
  //  pLabelEngineLayerProps.Expression = "[" + area + "]" + "[" + strFieldName + "]";
    pLabelEngineLayerProps.Expression = "[" + strFieldName + "]";
```

```
    axMapControl1.Refresh();
  }


 private void button15_Click(object sender, EventArgs e)
 {
   try
   {
              MessageBox.Show("Succesfully Updated to Geodatabase", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);
   }
   catch (Exception exc)
   {
                MessageBox.Show("Please Stop Retrieving Data First", "ERROR",
MessageBoxButtons.OK, MessageBoxIcon.Information);
   }
 }
   private void button5_Click_1(object sender, EventArgs e)
 {
    axMapControl2.Visible = false;
    axMapControl3.Visible = false;
    axMapControl1.Visible = true;
    string sFieldName;
    IFeatureClassDescriptor pFCDescr;
    IRasterLayer pRasterLayer;
    IGeoDataset pMaskDataset;
    ILayer pLayer;
    ILayer pLayer2;
    IRaster pRaster;
```

```
//axMapControl1.AddShapeFile("D:/SIBGHAT_THESIS_GEODATABASE/Thesis shapefile/", "Points.shp");
pFLayer = axMapControl1.Map.get_Layer(getLayerID("Points"));
pFeatureLayer = (IFeatureLayer)pFLayer;
pFeatureClass = pFeatureLayer.FeatureClass;
sFieldName = "temp";
pFCDescr = new FeatureClassDescriptorClass();
pFCDescr.Create(pFeatureClass, null, sFieldName);
// pLayer2 = pMap.get_Layer(this.getLayerID("123456.tif"));
// pRasterLayer = (IRasterLayer)pLayer2;
// pMaskDataset = (IGeoDataset)pRasterLayer.Raster;
//// object pMaskDataset = new ObjectClass();
object pMDataset = 0.000003;// pMaskDataset;
IInterpolationOp pIntOp;
IRasterAnalysisEnvironment pEnv;
IRasterRadius pRadius;
IRaster pOutRaster;


pIntOp = new RasterInterpolationOpClass();
pEnv = (IRasterAnalysisEnvironment)pIntOp;
pEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvValue, ref pMDataset);
pRadius = new RasterRadiusClass();
object Missing = Type.Missing;
pRadius.SetVariable(12, ref Missing);
    pOutRaster = pIntOp.IDW(pFCDescr as IGeoDataset, 2, pRadius, ref Missing) as IRaster;
```

```
    IRasterLayer pOutRasLayer = new RasterLayerClass();
    pOutRasLayer.CreateFromRaster(pOutRaster);
    pOutRasLayer.Name = "Temperture ";
    axMapControl1.AddLayer(pOutRasLayer);
    pFLayer = (IFeatureLayer)axMapControl1.get_Layer(this.getLayerID("Points"));
    IGeoFeatureLayer pGeoFeatureLayer12;
    pGeoFeatureLayer12 = (IGeoFeatureLayer)pFLayer;
    string strFieldName12 = "temp";
    // string area = "moisture";
    IAnnotateLayerPropertiesCollection pAnnoLayerPropsCol12;
    pAnnoLayerPropsCol12 = pGeoFeatureLayer12.AnnotationProperties;
    pGeoFeatureLayer12.DisplayAnnotation = true;
    IAnnotateLayerProperties pAnnoLayerProps12;
    IElementCollection iec12;
    pAnnoLayerPropsCol12.QueryItem(0, out pAnnoLayerProps12, out iec12, out iec12);
    ILabelEngineLayerProperties pLabelEngineLayerProps12;
    pLabelEngineLayerProps12 = (ILabelEngineLayerProperties)pAnnoLayerProps12;
    //  pLabelEngineLayerProps.Expression = "[" + area + "]" + "[" + strFieldName + "]";
    pLabelEngineLayerProps12.Expression = "[" + strFieldName12 + "]";
    axMapControl1.Refresh();
    tabControl1.SelectedIndex = 0;
}

private void button12_Click_1(object sender, EventArgs e)
{
    axMapControl2.Visible = false;
    axMapControl3.Visible = false;
```

```
axMapControl1.Visible = true;
string sFieldName;
IFeatureClassDescriptor pFCDescr;
IRasterLayer pRasterLayer;
IGeoDataset pMaskDataset;
ILayer pLayer;
ILayer pLayer2;
IRaster pRaster;

//axMapControl1.AddShapeFile("D:/SIBGHAT_THESIS_GEODATABASE/Thesis shapefile/", "Points.shp");
pFLayer = axMapControl1.Map.get_Layer(getLayerID("Points"));
pFeatureLayer = (IFeatureLayer)pFLayer;
pFeatureClass = pFeatureLayer.FeatureClass;
sFieldName = "moisture";
pFCDescr = new FeatureClassDescriptorClass();
pFCDescr.Create(pFeatureClass, null, sFieldName);

// pLayer2 = pMap.get_Layer(this.getLayerID("123456.tif"));
// pRasterLayer = (IRasterLayer)pLayer2;
// pMaskDataset = (IGeoDataset)pRasterLayer.Raster;
//// object pMaskDataset = new ObjectClass();
object pMDataset = 0.000003;// pMaskDataset;
IInterpolationOp pIntOp;
IRasterAnalysisEnvironment pEnv;
IRasterRadius pRadius;
IRaster pOutRaster;
pIntOp = new RasterInterpolationOpClass();
```

```
pEnv = (IRasterAnalysisEnvironment)pIntOp;
pEnv.SetCellSize(esriRasterEnvSettingEnum.esriRasterEnvValue, ref pMDataset);
pRadius = new RasterRadiusClass();
object Missing = Type.Missing;
pRadius.SetVariable(12, ref Missing);
 pOutRaster = pIntOp.IDW(pFCDescr as IGeoDataset, 2, pRadius, ref Missing) as
IRaster;
IRasterLayer pOutRasLayer = new RasterLayerClass();
pOutRasLayer.CreateFromRaster(pOutRaster);
pOutRasLayer.Name = "Soil_Moisture ";
axMapControl1.AddLayer(pOutRasLayer);
 pFLayer = (IFeatureLayer)axMapControl1.get_Layer(this.getLayerID("Points"));
IGeoFeatureLayer pGeoFeatureLayer;
pGeoFeatureLayer = (IGeoFeatureLayer)pFLayer;
string strFieldName = "Id";
// string area = "moisture";
IAnnotateLayerPropertiesCollection pAnnoLayerPropsColl;
pAnnoLayerPropsColl = pGeoFeatureLayer.AnnotationProperties;
pGeoFeatureLayer.DisplayAnnotation = true;
IAnnotateLayerProperties pAnnoLayerProps;
IElementCollection iec;
pAnnoLayerPropsColl.QueryItem(0, out pAnnoLayerProps, out iec, out iec);
ILabelEngineLayerProperties pLabelEngineLayerProps;
pLabelEngineLayerProps = (ILabelEngineLayerProperties)pAnnoLayerProps;
//  pLabelEngineLayerProps.Expression = "[" + area + "]" + "[" + strFieldName + "]";
pLabelEngineLayerProps.Expression = "[" + strFieldName + "]";
axMapControl1.Refresh();
pFLayer = (IFeatureLayer)axMapControl1.get_Layer(this.getLayerID("Points"));
```

```
            IGeoFeatureLayer pGeoFeatureLayer1;
            pGeoFeatureLayer1 = (IGeoFeatureLayer)pFLayer;
            string strFieldName1 = "moisture";
            // string area = "moisture";
            IAnnotateLayerPropertiesCollection pAnnoLayerPropsColl1;
            pAnnoLayerPropsColl1 = pGeoFeatureLayer.AnnotationProperties;
            pGeoFeatureLayer.DisplayAnnotation = true;
            IAnnotateLayerProperties pAnnoLayerProps1;
            IElementCollection iec1;
            pAnnoLayerPropsColl.QueryItem(0, out pAnnoLayerProps1, out iec1, out iec1);
            ILabelEngineLayerProperties pLabelEngineLayerProps1;
            pLabelEngineLayerProps1 = (ILabelEngineLayerProperties)pAnnoLayerProps;
            //  pLabelEngineLayerProps.Expression = "[" + area + "]" + "[" + strFieldName + "]";
            pLabelEngineLayerProps.Expression = "[" + strFieldName1 + "]";
            axMapControl1.Refresh();
            tabControl1.SelectedIndex = 0;
             }//end
        private void button1_Click_1(object sender, EventArgs e)
        {
            zedGraphControl2.GraphPane.CurveList.Clear();
            GraphPane myPane = zedGraphControl2.GraphPane;


            PointPairList spl1 = new PointPairList(xaxis, tempyaxis);
            LineItem myCurve1 = myPane.AddCurve("Sensor1", spl1, Color.Brown,
SymbolType.Triangle);
            myCurve1.Line.Width = 3.0F;
            myPane.Title.Text = "Graph";
            myCurve1.Symbol.Fill.RangeMin = 0;
```

```
myCurve1.Symbol.Fill.RangeMax = 20;
myCurve1.Symbol.Type = SymbolType.Triangle;
zedGraphControl2.AxisChange();
myPane.YAxis.MajorGrid.DashOff = 10;
//  myPane.XAxis.MajorGrid.IsVisible = true;
//  myPane.YAxis.MajorGrid.IsVisible = true;
myPane.XAxis.Scale.Min = 0;
myPane.XAxis.Scale.MajorStep = 1;
myPane.YAxis.Scale.MajorStep = 0.5;
myPane.XAxis.Title.Text = "Time (s)";
myPane.YAxis.Title.Text = "Temperature (C)";
zedGraphControl2.Refresh();
GraphPane myPane2 = zedGraphControl2.GraphPane;


PointPairList spl12 = new PointPairList(xaxis, tempyaxis1);
LineItem myCurve12 = myPane.AddCurve("Sensor2", spl12, Color.Blue,
SymbolType.Triangle);
myCurve12.Line.Width = 3.0F;
myPane2.Title.Text = "Graph";
myCurve12.Symbol.Fill.RangeMin = 0;
myCurve12.Symbol.Fill.RangeMax = 20;
myCurve12.Symbol.Type = SymbolType.Triangle;
zedGraphControl2.AxisChange();
myPane.YAxis.MajorGrid.DashOff = 10;
//  myPane.XAxis.MajorGrid.IsVisible = true;
//  myPane.YAxis.MajorGrid.IsVisible = true;
myPane2.XAxis.Scale.Min = 0;
myPane2.XAxis.Scale.MajorStep = 1;
```

```
myPane2.YAxis.Scale.MajorStep = 0.5;
myPane2.XAxis.Title.Text = "Time (s)";
myPane2.YAxis.Title.Text = "Temperature (C)";
//zedGraphControl1.Refresh();
GraphPane myPane3 = zedGraphControl2.GraphPane;


PointPairList spl13 = new PointPairList(xaxis, tempyaxis2);
        LineItem myCurve13 = myPane.AddCurve("Sensor3", spl13, Color.Black,
SymbolType.Triangle);
myCurve13.Line.Width = 3.0F;
myPane3.Title.Text = "Graph";
myCurve13.Symbol.Fill.RangeMin = 0;
myCurve13.Symbol.Fill.RangeMax = 20;
myCurve13.Symbol.Type = SymbolType.Triangle;
zedGraphControl2.AxisChange();
myPane.YAxis.MajorGrid.DashOff = 10;
//  myPane.XAxis.MajorGrid.IsVisible = true;
//  myPane.YAxis.MajorGrid.IsVisible = true;
myPane3.XAxis.Scale.Min = 0;
myPane3.XAxis.Scale.MajorStep = 1;
myPane3.YAxis.Scale.MajorStep = 0.5;
myPane3.XAxis.Title.Text = "Time (s)";
myPane3.YAxis.Title.Text = "Temperature (C)";
//zedGraphControl1.Refresh();
GraphPane myPane4 = zedGraphControl2.GraphPane;


PointPairList spl14 = new PointPairList(xaxis, tempyaxis3);
```

```
            LineItem myCurve14 = myPane.AddCurve("Sensor4", spl14,
Color.DarkGreen, SymbolType.Triangle);
    myCurve14.Line.Width = 3.0F;
    myPane4.Title.Text = "Graph";
    myCurve14.Symbol.Fill.RangeMin = 0;
    myCurve14.Symbol.Fill.RangeMax = 20;
    myCurve14.Symbol.Type = SymbolType.Triangle;
    zedGraphControl2.AxisChange();
    myPane.YAxis.MajorGrid.DashOff = 10;
    //  myPane.XAxis.MajorGrid.IsVisible = true;
    //  myPane.YAxis.MajorGrid.IsVisible = true;
    myPane4.XAxis.Scale.Min = 0;
    myPane4.XAxis.Scale.MajorStep = 1;
    myPane4.YAxis.Scale.MajorStep = 0.5;
    myPane4.XAxis.Title.Text = "Time (s)";
    myPane4.YAxis.Title.Text = "Temperature (C)";
    zedGraphControl2.Refresh();
 }
 private void button7_Click(object sender, EventArgs e)
 {
    radioButton3.Checked = false;
    textBoxMAD.Text = "";
    textBox10.Text = "";
 }
 private void button10_Click(object sender, EventArgs e)
 {
      System.Threading.Thread.Sleep(50);
    RX = serialPort1.ReadExisting();
```

```
RX = RX.Replace("\r\n", "").Replace("\n", "").Replace("\n\r", "").Trim();
if (RX == "GET")
{
    serialPort1.Write(ary[position].ToString());
     position++;
     if (position >= ary.Length - 1)
        position = 0;


    serialPort1.Write("1");
    serialPort1.Write("1");
}
  else if (RX != "PC DATA\n\r" && RX != "" && RX != "\n\r" && RX != "PC DATA"
&&
    RX.Contains("TEMP") && RX.Contains("MOISTURE"))
{
    ReterivedData tempData = new ReterivedData();


    RX = RX.Substring(RX.IndexOf("TEMP"), RX.Length - RX.IndexOf("TEMP"));
    row = dt.NewRow();
      tempData.DeviceNumber = int.Parse(RX[4].ToString());//This gets the Device #
    from String
    tempData.Temperature = RX.Substring(RX.IndexOf("TEMP" +
    tempData.DeviceNumber + "=") + 6, RX.IndexOf("M"));
    tempData.Moisture = RX.Substring(RX.IndexOf("MOISTURE " +
    tempData.DeviceNumber + "=") + 11, RX.IndexOf("M") + 1);


    if (int.Parse(tempData.Temperature) > 60)
    {
        tempData.Temperature = "60";
```

```
}
if (int.Parse(tempData.Temperature) < 20)
{
    tempData.Temperature = "20";
}
if (int.Parse(tempData.Moisture) < 4)
{
    tempData.Moisture = "0";
}
if (int.Parse(tempData.Moisture) > 80)
{
    tempData.Moisture = "80";
}
//row["Device"] 9= tempData.DeviceNumber;
row["Temperature(C)"] = tempData.Temperature;
row["Soil Moisture(%)"] = tempData.Moisture;
//dt.Columns.Add("Sensor");
DateTime tm = DateTime.Now;
row["Time"] = tm.ToString();
dt.Rows.Add(row);
reterivedData.Add(tempData);
//MessageBox.Show(tempData.Moisture);
//noo.Add(int.Parse(tempData.Temperature));
// doo.Add(int.Parse(tempData.Moisture));
// MessageBox.Show(serial_counter.ToString());
//serial_counter++;
foreach (DataGridViewRow dgvr in dataGridView1.Rows)
```

```
            {
                if (dgvr.Cells[2].Value != null && int.Parse(dgvr.Cells[2].Value.ToString()) <
10)
                {
                    dgvr.DefaultCellStyle.BackColor = Color.Red;
                }
            }
            dataGridView1.Invoke(new EventHandler(delegate
            {
                dataGridView1.DataSource = dt;
                dataGridView1.Refresh();
            }));
        }
    }//end

    private void serialPort1_DataReceived(object sender,
   System.IO.Ports.SerialDataReceivedEventArgs e)
    {
        this.Invoke(new EventHandler(button10_Click));
    }

    private void button17_Click(object sender, EventArgs e)
    {
        serialPort1.Open();
    }

}
```

```
}//end
```