**IOT BASED SMART SOIL CULTURE**

**A PROJECT REPORT**

***Submitted by***

**G MANOJ KUMAR 212216106090**

**G SAI KALYAN 212216106085**

**G MANIKANTA 212216106078**

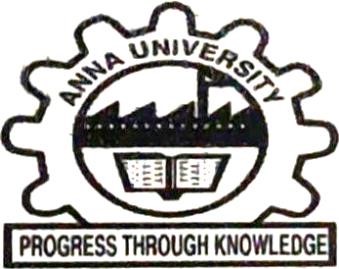
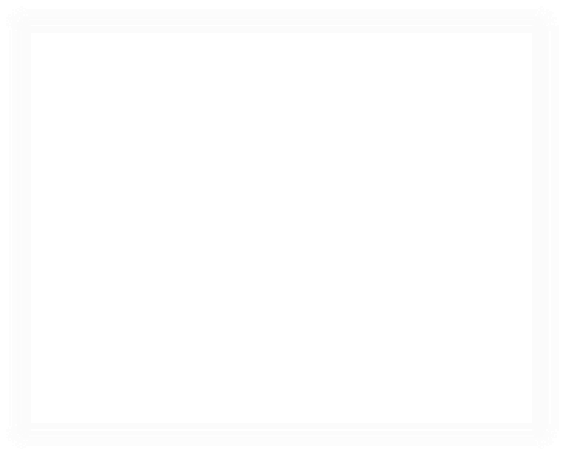
***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

**ELECTRONICS AND COMMUNICATION ENGINEERING**



**SAVEETHA ENGINEERING COLLEGE(AUTONOMOUS)**

**ANNA UNIVERSITY, CHENNAI**

**APRIL 2020**

**SAVEETHA ENGINEERING COLLEGE (AUTONOMOUS)**

**BONAFIDE CERTIFICATE**

Certified that this project report **“IOT BASED SMART SOIL CULTURE”** is the bonafide work of **G MANOJ KUMAR (212216106090), G SAI KALYAN (212216106085),** and **G MANIKANTA (212216106078)** who carried out the project work under my supervision.

##### SIGNATURE SIGNATURE

**Dr. SRIGITHA. S. NATH, M.E., Ph.D.** **J.JAYALAKSHMI,M.E**

|  |  |
| --- | --- |
| Professor | Associate Professor |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Dept of Electronics and  Communication | Dept of Electronics and  Communication |
| Engineering, | Engineering, |
| Saveetha Engineering College, | Saveetha Engineering College, |
| Thandalam, Chennai 602105. | Thandalam, Chennai 602105. |

DATE OF THE VIVA VOCE EXAMINATION: …………………….

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We express our deep sense of gratitude to our honorable and beloved

Founder President ***Dr. N. M. Veeraiyan,*** our President ***Dr. Saveetha Rajesh,*** our Director ***Dr. S.Rajesh*** and other management members for providing the infrastructure needed.

We express our wholehearted gratitude to our principal, ***Dr. R.Ramesh,*** for his whole hearted encouragement in completing this project.

We convey our thanks to ***Dr. Srigitha.S.Nath,*** Professor and Head of the Department of Electronics and Communication Engineering, Saveetha Engineering College, for her kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks and deep sense of gratitude to our Supervisor ***Ms.J.Jayalakshmi,*** Associate Professor, Department of Electronics and Communication Engineering, Saveetha Engineering College and our Project Coordinator, ***Dr .P. Mangayarkarasi,*** Professor, Department of Electronics and Communication Engineering, Saveetha Engineering College, for their valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of the project work and for providing us necessary support and details at the right time and during the progressive reviews.

We owe our thanks to all the members of our ***college, faculty, staff*** and ***technicians*** for their kind and valuable co-operation during the course of the project. We are pleased to acknowledge our sincere thanks to ***our beloved parents, friends*** and ***well-wishers*** who encouraged us to complete this project successfully.

#### ABSTRACT

In the present world, an increase in population occurs due to water scarcity and food scarcity. So to reduce this problem we have to contribute for the growth of agriculture sector. But the problem in this sector is there will be more wastage of water in the form of logging of water while watering the land for irrigation. Therefore, an irrigation system which is automatic has to be developed for supplying water properly and systematically in the fields. In this project an irrigation system for plants is designed that helps supply exact amount of water in the fields without wasting it by sensing the soil moisture content in the soil. By taking this value it decides whether irrigation is needed for the field or not. In this system we are using AtMega328 microcontroller. This microcontroller senses the content of moisture of soil over a time period. If the content of humidity is less than the predefined limit, it sends the information to the farmer so that he can start to provide water to the fields till the threshold is reached**.** Here we are using four types of sensors such as temperature, humidity, moisture and water level sensors. All the sensors are connected to the microcontroller. These sensors are mounted over a robotic vehicle which goes inside the agriculture land and measures all the values and updates in cloud using WI-FI module. The robotic vehicle movement is controlled by using Smartphone.

**CHAPTERS TABLE OF CONTENTS PAGE NUMBER**

**ABSTRACT I**

1. **INTRODUCTION** **1**
   1. Embedded Systems
   2. Applications of Embedded Systems
   3. Types of Embedded System
   4. Embedded Systems Network Applications
2. **SYSTEM ANALYSIS 5**

* 1. Literature Survey

1. **EMBEDDED SYSTEM**  12
   1. Existing System
   2. Proposed System
   3. Project Features
2. **HARDWARE DESCRIPTION 15**

4.1 Arduino Microcontroller

4.2 I/O Ports

4.3 Memory Unit

4.4 Central Processing unit

4.5 Serial Communication

4.6 Timer Unit

4.7 Watchdog

4.8 A/D Converter

4.9 CISC and RISC

4.10 Memory Organization

4.11 I2C Protocol

4.12 Display Unit

4.13 Power Supply

4.14 Temperature Sensor

4.15 Motor Driver

4.16 Humidity Sensor

4.17 Moisture Sensor

4.18 Jump Wire

4.19 Wheels

4.20 Motors

4.21 Bluetooth Device

**5 SOFTWARE DESCRIPTION 46**

5.1 Introduction

5.2 MPLAB IDE

5.3 C18 Comp

**6 CONCLUSION 48**

**7 OUTPUT IMAGE 49**

7.1 Connections for IOT based Smart Agriculture

7.2 Block Diagram

7.3 Simulation for Temperature and Humidity Sensor

7.4 Simulation for Water level Sensor

**8 REFERENCE PAPERS 51**

## LIST OF FIGURES

**FIGURE NO. TITLE PAGE NO.**

1.5 Embedded System Design 4

3.4 Block Diagram 15

4.2 .1 I/O Port 24

4.2.2 Output Drive Values 25

4.3.1 Input Voltage Values 26

4.3.2 Block Diagram of Memory Unit 30

4.4 Representation of Bus 31

4.5 Watch dog timer 32

4.6 A/D Convertor 33

4.7 Program Memory and Stack Memory 35

4.8 Table Register Bank Selection 35

4.11 I2C Bus Terminology 40

4.11.2 I2C Bus Configuration 41

### 

##### CHAPTER 1

##### INTRODUCTION

##### 1.1 OVERVIEW OF THE PROJECT

##### One of the most important problems faced by the world is scarcity of water. We require a system which uses water limitedly without any wastage. Irrigation systems which are smart measure the moisture of plant in order to operate an automatic irrigation system and that reduces the use of excess water. The sensors which are used in this system requires good maintenance for proper performance. Automatic plant irrigation system will concentrate on watering the field regularly by using moisture sensor without any human monitoring. Lot of water can be transpired by healthy plants that results in humidity increase of Green house air. We should avoid relative humidity which is above 80-85% because plant transpiration and incidence of the disease can be increased by it. Condensation on greenhouse structure and on plant surfaces can be prevented by successive heating and sufficient venting. During the warmer summer months, air humidity of greenhouse increases by using cooling system. Humidity control inside the greenhouse can be a challenge during periods with humid and warm conditions outdoor. This system is crop specific and its usage hence is limited. For efficient management of water in crop production it is critical to have a proper scheduling of irrigation, under the conditions of water scarcity particularly. The effects of applied frequency of irrigation, amount of water supplied for irrigation and use of water are particularly important. So our project designs a system which is simple, using an AtMega328 controller which is fixed in Arduino called Arduino Uno for automating the irrigation system and watering the fields with minimum human interventions.

**1.1 EMBEDDED SYSTEM:**

A general definition of embedded systems is: embedded systems are computing systems with tightly coupled hardware and software integration, which are designed to perform a dedicated function. In some cases, embedded systems can function as standalone systems.

One class of embedded processors focuses on size, power consumption, and price. Therefore, some embedded processors are limited in functionality, i.e., a processor is good enough for the class of applications for which it was designed but is likely inadequate for other classes of applications.

Real-time systems are defined as those systems in which the overall correctness of the system depends on both the functional correctness and the timing correctness. The timing correctness is at least as important as the functional correctness.

**1.2 APPLICATION OF EMBEDDED SYSTEM**

In real life we are using so many embedded systems for example

Home application (micro oven, washing machine, security system DVD, Mp3 player etc,)

Air craft, missiles, automotive, nuclear research, personal use (mobile phone, I pod)

**1.3 TYPES OF EMBEDDED SYSTEM**

Embedded System is broadly categorized as

Standalone embedded system

Example: Washing Machine

Networking embedded system

Example: Network Printer

**1.4 EMBEDDED SYSTEM NETWORK APPLICATIONS.**

Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have real-time performance constraints that must be met, for reason such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always separate devices. Most often they are physically built-in to the devices they control.

The software written for embedded systems is often called firmware, and is stored in read-only memory or Flash memory chips rather than a disk drive. It often runs with limited computer hardware resources: small or no keyboard, screen, and little memory.

##### 1.5 Embedded System Design and Development Life Cycle



Figure 1.1: Embedded System Design

**CHAPTER 2**

**2. LITERATURE SURVEY**

# 2.1 Mobile integrated smart irrigation management and monitoring system using IOT [S. Vaishali](https://ieeexplore.ieee.org/author/37086294779);[S. Suraj](https://ieeexplore.ieee.org/author/37086290120);[G. Vignesh](https://ieeexplore.ieee.org/author/37086285395);[S. Dhivya](https://ieeexplore.ieee.org/author/37086292297);[S. Udhayakumar](https://ieeexplore.ieee.org/author/38234895100),December 2007.

Agriculture has been the most important practice from very beginning of the human civilization. Traditional methods that are used for irrigation, such as overhead sprinkler and flood type, is not that much efficient. They results in a lot of wastage of water and can also promote disease such as fungus formation due to over moisture in the soil. Automated irrigation system is essential for conservation of the water and indirectly viability of the farm since it is an important commodity. About 85% of total available water resources across the world are solely used for the irrigation purpose. In upcoming years this demand is likely to increase because of increasing population. To meet this demand we must adopt new techniques which will conserve need of water for irrigation process. In automation system water availability to crop is monitored through sensors and as per need watering is done through the controlled irrigation. The almost infinite capabilities of storage and processing, the rapid elasticity makes cloud computing an attractive solution to the large amount of data generated. The idea is to focus on parameters such as temperature and soil moisture. This is a Mobile Integrated and smart irrigation system using IOT based on application controlled monitoring system. The main objective of this project is to control the water supply and monitor the plants through a Smartphone.

# 2.2 Design of Wireless Sensor Network (WSN) with RF Module for Smart Irrigation System in Large [Radi](https://ieeexplore.ieee.org/author/37086833284);[Murtiningrum](https://ieeexplore.ieee.org/author/37086832301);[Bambang Purwantana](https://ieeexplore.ieee.org/author/37086834491);[Fajar Siti Muzdrikah](https://ieeexplore.ieee.org/author/37086833541);[M. Shohibun Nuha](https://ieeexplore.ieee.org/author/37086830763);[Muhammad Rivai](https://ieeexplore.ieee.org/author/37085345300),April 2017.

Give water to crops that is usually called as irrigation is one of the most important activities in farming system. Because water determine the crops growth and development, it should be available sufficiently inside the planting media. Recently, such control technology for crops irrigation has been developed though its application for such drip irrigation is still limited in small cultivated area. For extensive crop cultivation areas such as plantations, it needs a strategy to apply the technology in large cultivated area. We propose a model that is called as a smart irrigation systems (SIS). In the model, the large cultivated area is divided into several plots, where each plot has an independently automatic irrigation system. In order to be easily managed, each irrigation system in each plot can communicate to others and all information should be collected in a data server that is connected to cloud system. Since that common plantation area has limitation of communication lines, we propose a wireless sensor network (WSN) based on a radio frequency (RF) communication to connect all automatic irrigation systems. This paper focuses on designing a WSN prototype based on a radio frequency (RF) module to develop a smart drip irrigation system for large cultivated land. The WSN is designed from a master node functioned as coordinator and three slave nodes. The master and slave nodes are equipped with an RF module with a 2.4GHz frequency for communication. After construction, the WSN prototype with star topology is then tested for its ability, especially for transmitting data from slave to master and vice versa. The prototype has been tested with varied distance of 50 m, 100 m and 150 m between master and slave in a tree-blocked open space. Based on the data losses, the prototype has a good performance if the distances is less than 100 m.

# 2.3 IoT based smart crop-field monitoring and automation irrigation system

# R. Nageswara Rao;[B. Sridhar](https://ieeexplore.ieee.org/author/37086406699),Feb 2017.

Agriculture plays vital role in the development of agricultural country like India. Issues concerning agriculture have been always hindering the development of the country. The only solution to this problem is smart agriculture by modernizing the current traditional methods of agriculture. Hence the proposed method aims at making agriculture smart using automation and IoT technologies. Internet of Things (IoT) enables various applications crop growth monitoring and selection, irrigation decision support, etc. A Raspberry Pi based automatic irrigation IOT system is proposed to modernization and improves productivity of the crop. main aim of this work to crop development at low quantity water consumption, In order to focus on water available to the plants at the required time, for that purpose most of the farmers waste lot time in the fields. An efficient management of water should be developed and the system circuit complexity to be reduced. The proposed system developed on the information sent from the sensors and estimate the quantity of water needed. A two sensors are used to get the data to the base station the humidity and the temperature of the soil, the humidity, the temperature, and the duration of sunshine per day. The proposed systems based on these values and calculate the water quantity for irrigation is required. The major advantage the system is implementing of Precision Agriculture (PA) with cloud computing, that will optimize the usage of water fertilizers while maximizing the yield of the crops and also will help in analyzing the weather conditions of the field.

# 2.4 Smart irrigation with embedded system [K K Namala](https://ieeexplore.ieee.org/author/37085883229);[Krishna Kanth Prabhu A V](https://ieeexplore.ieee.org/author/37086015733);[Anushree Math](https://ieeexplore.ieee.org/author/37086013839);[Ashwini Kumari](https://ieeexplore.ieee.org/author/37086008859);[Supraja Kulkarni](https://ieeexplore.ieee.org/author/37086011321),July 2016.

This paper proposes intelligent and smart Irrigation system which can be used for controlling the watering or irrigation of flowering plants. It controls the irrigation of plants automatically where the need of human intervention can be reduced. This mainly focused on wastage of water, which is a major concern of modern era. It also aids time saving, cost effectiveness, environmental protection, low maintenance and operating cost and efficient irrigation service. Raspberry Pi (open source) is used in the design of the prototype model in making the system compact and sustainable. The system has sensor which measures the moisture of the soil and switches relay which controls solenoid valve according to the requirement. The model demonstrated gave expected results at the different moisture levels.

# 2.5 Smart home garden irrigation system using Raspberry Pi [S. N. Ishak](https://ieeexplore.ieee.org/author/37085519200);[N. N. N. Abd Malik](https://ieeexplore.ieee.org/author/37302207700);[N. M. Abdul Latiff](https://ieeexplore.ieee.org/author/38527720800);[N. Effiyana Ghazali](https://ieeexplore.ieee.org/author/37400739700);[M. A. Baharudin](https://ieeexplore.ieee.org/author/38260645800),Dec 2018.

Irrigation system is a method of allowing water to drip slowly to the roots of plants, either onto the soil surface or directly onto the root zone, through solenoid valve. However, it is found that the market price of the system is expensive for small area coverage. Thus, this paper proposes a design for smart home garden irrigation system that implements ready-to-use, energy-efficient, and cost effective devices. Raspberry Pi, which is implemented in this system is integrated with multi-sensors such as soil moisture sensors, ultrasonic sensors, and light sensors. This proposed system managed to reduce cost, minimize waste water, and reduce physical human interface. In this paper, the relay is utilized to control the switching of solenoid valve. The system also managed to measure moisture of the soil and control the solenoid valve according to human's requirements. It is conducted with Graphical User Interface (GUI) using Android application to activate watering activity. Email notification is also sent to the home user for alert purposes either for normal or critical operations. An experimental setup has been tested and it is proven that the system can intelligently control and monitor the soil moisture levels in the experiment field.

# 2.6 Smart irrigation: A smart drip irrigation system using cloud, android and data mining [Subhashree Ghosh](https://ieeexplore.ieee.org/author/37086098954);[Sumaiya Sayyed](https://ieeexplore.ieee.org/author/37086094290);[Kanchan Wani](https://ieeexplore.ieee.org/author/37086094278);[Mrunal Mhatre](https://ieeexplore.ieee.org/author/37086099366);[Hyder Ali Hingoliwala](https://ieeexplore.ieee.org/author/37086097199),Feb 2019.

Water is an essential component for the development of plants in agriculture or irrigation. The paper stresses on the need of an externally hosted cloud computing platform to manage the database, android and the isolated server by the users across the country for irrigation. The system proposed in this paper uses information and communication technologies, allowing the user to consider and examine the information obtained by different sensors. Here we are using different sensors like humidity, temperature, moisture, light etc. These sensors give signal to the micro controller. Micro-controller gives the data to the isolated server through a serial communication. According to sensor values graph will be display on PC and Smart phone side and by using this graph user can on or off drip devices. In this we keep threshold value for each sensor. The data is sent and processed on an isolated server, which stores the information from the sensors in a database, allowing further interpretation of data in a simple and flexible way. The intended system may lead to enhance the farming practices, overcoming the water crises and developing an upgraded agricultural system for the country.

# 2.7 Dynamic Model of Soil Moisture for Smart Irrigation Systems [Luisella Balbis](https://ieeexplore.ieee.org/author/37293753500);[Ali Jassim](https://ieeexplore.ieee.org/author/37087028682),Oct 2016.

In geographical areas subject to an arid climate, one of the most precious resources is water, which often get wasted due to inefficient irrigation systems. Automated irrigation systems which relies on closed loop control and feedback from sensors can help to closely match the water supply to the crop demand and avoid waste. In recent years, predictive control strategies were proposed for high tech watering systems which rely heavily on complex models incorporating variables such as crop water demand, soil evapotranspiration, weather conditions etc.. Through these models it is possible to anticipate the water demand of the crop, and give optimal irrigation scheduling with a significant reduction in water consumption. In this paper we developed a model of the soil moisture dynamics based on climatological data of the Kingdom of Bahrain. The model, which is based on the hydraulic balance approach, was validated by comparing its estimated output with real measurements.

# 2.8 Photovoltaic Energy Conversion Smart Irrigation System-Dubai Case Study (Goodbye Overwatering & Waste Energy, Hello Water & Energy Saving) [Shamma Ali](https://ieeexplore.ieee.org/author/37086528167);[Hamda Saif](https://ieeexplore.ieee.org/author/37086525815);[Hasa Rashed](https://ieeexplore.ieee.org/author/37086526670);[Hend AlSharqi](https://ieeexplore.ieee.org/author/37086527348);[Ammar Natsheh](https://ieeexplore.ieee.org/author/37086389919),2016.

Insufficiency of water usage often cited as one fundamentally problems of poor harvest or crops. Therefore, management of water, and sources of water, becomes very important aspect for countries that depend on its agriculture. In order to solve this problem there have been so many exciting innovations introduced in many parts around the world. Among them is a new and smart irrigation system developed to advance agriculture practice, made accessible to anyone who wants to farm. This paper presents the smart irrigation system designed to use energy collected by photovoltaic and a combination of devices for control - such as Arduino Uno, the raspberry pi with a camera and other components. The system is considered as sustainable, efficient and reliable as well as it is easy to access.

# 2.9 IoT based smart irrigation monitoring and controlling system [Shweta B. Saraf](https://ieeexplore.ieee.org/author/37086318638);[Dhanashri H. Gawali](https://ieeexplore.ieee.org/author/37679897700),Sep 2018.

Interconnection of number of devices through internet describes the Internet of things (IoT). Every object is connected with each other through unique identifier so that data can be transferred without human to human interaction. It allows establishing solutions for better management of natural resources. The smart objects embedded with sensors enables interaction with the physical and logical worlds according to the concept of IoT. In this paper proposed system is based on IoT that uses real time input data. Smart farm irrigation system uses android phone for remote monitoring and controlling of drips through wireless sensor network. Zigbee is used for communication between sensor nodes and base station. Real time sensed data handling and demonstration on the server is accomplished using web based java graphical user interface. Wireless monitoring of field irrigation system reduces human intervention and allows remote monitoring and controlling on android phone. Cloud Computing is an attractive solution to the large amount of data generated by the wireless sensor network. This paper proposes and evaluates a cloud-based wireless communication system to monitor and control a set of sensors and actuators to assess the plants water need.

# 2.10 Multithreading design for an embedded irrigation system running on solar power [A. Selmani](https://ieeexplore.ieee.org/author/37085513797);[M. Outanoute](https://ieeexplore.ieee.org/author/37085518932);[M. Amini Alaoui](https://ieeexplore.ieee.org/author/37086386030);[M. El Khayat](https://ieeexplore.ieee.org/author/37086385379);[M. Guerbaoui](https://ieeexplore.ieee.org/author/37085509657);[A. Ed-dahhak](https://ieeexplore.ieee.org/author/37085522756);[A. Lachhab](https://ieeexplore.ieee.org/author/37085446065);[B. Bouchikhi](https://ieeexplore.ieee.org/author/37085451798),Dec 2017.

Technologies of irrigation systems are developed with a focus on intelligent water management, advanced features and remote control of solar photovoltaic watering systems (SPVWSs). The new generation of inexpensive microcontrollers comes with high level of computation capability and provides low-cost solutions. In this direction, this paper presents a new multithreading design and implementation of an embedded irrigation system running on solar power. The design combines the advanced concepts of concurrency programming, namely multithreading and virtual timers, with the embedded computing resources of smart devices. According to the design and test results through an experimental greenhouse, this device provides an accuracy real-time management of all activities of the SPVWS with eventually a low power consumption.

**CHAPTER 3**

**3.1 EXISTING SYSTEM**

**a) Traditional Irrigation system**

Agriculture is one of the major occupations of many people in India. Sugarcane is the major commercial crop in many parts of India. However there are many problems in the areas with enough water as well as areas with scarcity of water. The traditional mechanism used to irrigate the sugarcane field results into more wastage of the water, fertilizers, manpower, electricity etc. Usually, the farmers spend a lot of water, electricity, time and efforts to irrigate their fields. They personally need to visit the field and according to moisture in soil they supply water to farm. They need to switch ON motor and then after irrigating the field they must switch OFF the motor. This whole process is time consuming and farmer can't do any other activities. Moreover, the electricity supply in time is not guaranteed and there are many related problems which are major stumbling blocks in the irrigation process. In a nutshell, traditional irrigation system takes more cost of production along with other side effects leading to conversion into baron land.

**b) Drip irrigation**

Since its introduction in the 1970's drip Irrigation for sugarcane has increasingly gained popularity. Nowadays drip irrigation is known as the most precise, efficient and practical method of delivering water and nutrients to crops . Drip Irrigation systems water individual plants directly at their root zone, eliminating a lot of fungal issues and wasted water. Drip irrigation system allows you to save time by watering large areas of plants all at once. It allows farmers to increase productivity and reduce the use of resources, resulting in significantly higher rate of incentives (ROI), in terms of quality and quantity, compared with any other irrigation method. Growing sugarcane with drip irrigation contributes to significantly higher yield with higher sucrose content; more number of ratoons from each planting cycle; lower water use reduced labor costs (simple to operate); saving in fertilizer etc. However, most people will start out with a less expensive, basic drip irrigation system. Drip irrigation systems can help you water tough areas, like slopes where run off and erosion can happen from other watering methods. Drip irrigation can be set to give these areas a slow penetrating soak, or can be set to deliver water in bursts that can be soaked in before the next burst. Most problems with drip irrigation come from improper installation or not using the right kind of drip irrigation for the site.

**3.2 PROPOSED SYSTEM**

Irrigation is that the artificial manner of watering crops in fields. In the present era, water inadequacy because of over exploitation have resulted the urge of developing a brand new technology that would save water from being wasted and since, agriculture is that the most water intense occupation, thus creating irrigation system sensible would be a better manner of checking water loss. Sensible irrigation system is economical and efficient way of watering fields. It monitors weather, soil conditions, evaporation and plant water use and mechanically adjusts watering schedule. Hence approaching sensible irrigation system has become a primary concern to relinquish farmer a wise tool which would support them in yielding quality crops. Since India is Associate in Nursing agro primarily based country and around sixty-one of the population. India occupies second rank in rice export and as rice crop need vast quantity of water for irrigation purpose, hence sensible irrigation project we tend to use differing kinds of detector to make a farmer up to this point regarding the sector. Sensors used are soil wetness detector , water flow detector and temperature sensor(ds18b20) sort of a detector which might calculate the quantity of water in the sector , a soil wetness detector that can calculate the wetness profile of the sector so as to prevent crops from water work problems and a temperature sensing detector so one will check the temperature of the crops as a result of crops area unit temperature sensitive too and if the smart system aware the farmer before then farmer will use sprinklers so as to cool down temperature of the crops it would save each crop and farmer. Our approach is to form this system accessible from even way distance so farmer have the data and management on the sector 24x7 throughout a year.

Here we are using four types of sensors such as temperature, humidity, moisture and water level sensors. All the sensors are connected to the microcontroller. These sensors are mounted over a robotic vehicle which goes inside the agriculture land and measures all the values and updates in cloud using WI-FI module. The robotic vehicle movement is controlled by using Smartphone.

**3.3 PROJECT FEATURES**

1. Low cost.

2. High speed networking.

3. Low power consumption.

4. Light weight network

5. Broadcast communication

**3.4 BLOCK DIAGRAM**

WIFI MODULE

BLUETOOTH MODULE

TEMPERATURE SENSOR

MOISTURE SENSOR

16x2 LCD DISPLAY

MICROCONTROLLER

MOTOR DRIVER

HUMIDITY SENSOR

BATTERY

MOTOR 1

WATER LEVEL SENSOR

MOTOR 2

**CHAPTER 4**

**HARDWARE DESCRIPTION**

**4.1ARDUINO MICROCONTROLLER:**

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

* 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
* Stronger RESET circuit.
* Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

**Summary**

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

|  |  |  |
| --- | --- | --- |
| Input Voltage (limits) | 6-20V | |
| Digital I/O Pins | 14 | (of which 6 provide PWM output) |
| Analog Input Pins | 6 |  |
| DC Current per I/O Pin | 40 mA | |
| DC Current for 3.3V Pin | 50 Ma | |
| Flash Memory | 32 | KB (ATmega328) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328) | |
| EEPROM | 1 KB (ATmega328) | |
| Clock Speed | 16 | MHz |

**Schematic & Reference Design**

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf

**Note:** The Arduino reference design can use an Atmega8, 168, or 328, Current models use anATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

**Power**

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

1. **VIN.** The input voltage to the Arduino board when it's using an external power source (asopposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
2. **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be suppliedwith power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
3. **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4. **GND.** Ground pins.

**Memory**

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

**Input and Output**

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

* **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pinsare connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
* **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a lowvalue, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
* **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with theanalogWrite()function.

1. **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communicationusing the SPI library.
2. **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, theLED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

* **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using theWire library.There are a couple of other pins on the board:
* **AREF.** Reference voltage for the analog inputs. Used withanalogReference().
* **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button toshields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

**Communication**

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

**Programming**

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

* On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

**Automatic (Software) Reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

**USB Overcurrent Protection**

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

**Physical Characteristics**

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

**4.2.1 I/O PORTS:**

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

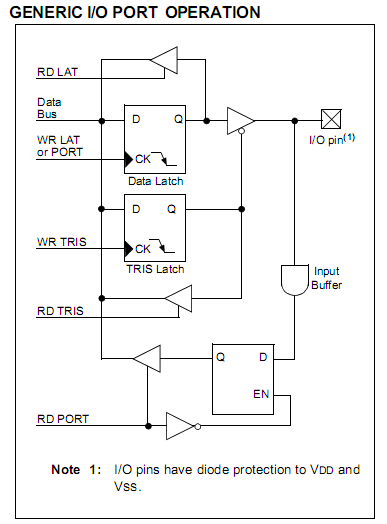


Figure 4.2.1: I/O PORT OPERATION

***Each port has three registers for its operation. These registers are:***

•TRIS register (Data Direction register)

•PORT register (reads the levels on the pins of the device)

•LAT registers (Data Latch)

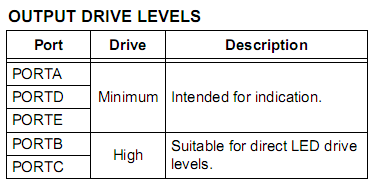
**I/O Port Pin Capabilities:**

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

**4.2.2 PIN OUTPUT DRIVE:**

The output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. PORTB and PORTC are designed to drive higher loads, such as LEDs. All other ports are designed for small loads, typically indication only. Table9-1 sum- marizes the output capabilities.

**“Electrical Characteristics” for more details.**



**4.2.2 OUTPUT DRIVE VALUE**

**4.2.3 INPUT PINS AND VOLTAGE CONSIDERATIONS:**

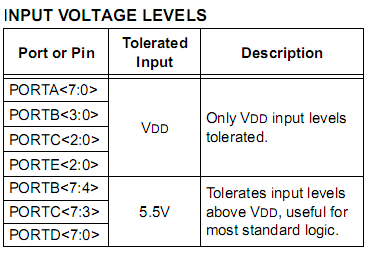
The voltage tolerance of pins used as device inputs is dependent on the pin’s input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V; a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD.

**4.2.4 PORTA, TRISA and LATA Registers:**

PORTA is a 7-bit wide, bidirectional port. It malfunctions as a 5-bit port, depending on the oscillator mode selected. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

The PORTA register reads the status of the pins, whereas writing to it, will write to the port latch. The Data Latch (LATA) register is also memory mapped .Read-modify-write operations on the LATA register read and writes the latched output value for PORTA. The other PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the comparator voltage reference output. The operation of pins, RA<3:0> and RA5, as A/D converter inputs is selected by clearing or setting the control bits in the ADCON1 register (A/D Control Register 1). Pins, RA0 and RA3, may also be used as comparator inputs and by setting the appropriate bits in the CMCON register. To use RA<3:0> as digital inputs, it is also necessary to turn off the comparators.

All PORTA pins have TTL input levels and full CMOS output drivers. The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.



**4.2.5 PORTB, TRISB and LATB Registers:**

PORTB is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin). The Data Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB. Each of the PORTB pins has a weak internal pull-up.

A single control bit can turn on all the pull-ups. This is performed by clearing bit, RBPU (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a POR. Four of the PORTB pins (RB<7:4>) have an interrupt- on-change feature. Only pins configured as inputs can cause this interrupt to occur

The input pins (of RB<7:4>) are compared with the old value latched on the last read of PORTB. The “mismatch” outputs of RB<7:4> are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>). A mismatch condition continues to set flag bit, RBIF.Reading PORTB will end the mismatch condition and allow flag bit, RBIF, to be cleared after one instruction cycle of delay.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature. The RB5 pin is multiplexed with the Timer0 module clock input and one of the comparator outputs to become the RB5/KBI1/SDI1/SDA1/RP8 pin.

**4.2.6 PORTC, TRISC and LATC Registers:**

PORTC is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin). The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC. PORTC is multiplexed with several peripheral functions. The pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. The user should refer to the corresponding peripheral section for additional information. Unlike other PORTC pins, RC4 and RC5 do not have TRISC bits associated with them. As digital ports, they can only function as digital inputs. If an external transceiver is used, RC4 and RC5 always function as inputs from the transceiver. If the on-chip transceiver is used, the data direction is determined by the operation being performed by the module at that time. The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

**4.2.7 PORTD, TRISD and LATD Registers:**

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD. All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output. Each of the PORTD pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is per- formed by setting bit, RDPU (PORTE<7>).

The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a POR. Note that the pull-ups can be used for any set of features, similar to the pull-ups found on PORTB.

**4.2.8 PORTE, TRISE and LATE Registers:**

Depending on the particular PIC18F46J11 family device selected, PORTE is implemented in two different ways. For 44-pin devices, PORTE is a 3-bit wide port. Three pins (RE0/AN5/PMRD, RE1/AN6/PMWR and RE2/ AN7/PMCS) are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. When selected as analog inputs, these pins will read as ‘0’s.

The corresponding Data Direction register is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., put the contents of the output latch on the selected pin).

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

The Data Latch register (LATE) is also memory mapped. Read-modify-write operations on the LATE register read and write the latched output value for PORTE. Each of the PORTE pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by setting bit, REPU (PORTE<6>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a POR. Note that the pull-ups can be used for any set of features, similar to the pull-ups found on PORTB.

Circumstances that we find ourselves in today in the field of microcontrollers had their beginnings in the development of technology of integrated circuits. This development has made it possible to store hundreds of thousands of transistors into one chip. That was a prerequisite for production of microprocessors, and the first computers were made by adding external peripherals such as memory, input-output lines, timers and other. Further increasing of the volume of the package resulted in creation of integrated circuits. These integrated circuits contained both processor and peripherals. That is how the first chip containing a microcomputer, or what would later be known as a microcontroller came about.

**4.3 Memory unit**

Memory is part of the microcontroller whose function is to store data. For a certain input we get the contents of a certain addressed memory location and that's all. Two new concepts are brought to us: addressing and memory location. Memory consists of all memory locations, and addressing is nothing but selecting one of them.

This means that we need to select the desired memory location on one hand, and on the other hand we need to wait for the contents of that location. Besides reading from a memory location, memory must also provide for writing onto it. This is done by supplying an additional line called control line. We will designate this line as R/W (read/write). Control line is used in the following way: if r/w=1, reading is done, and if opposite is true then writing is done on the memory location.

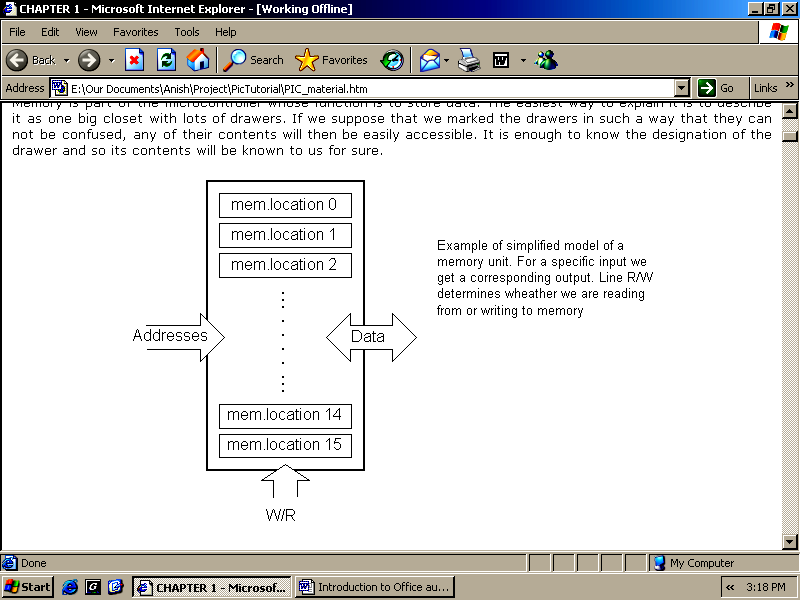


Figure 4.3: Block diagram of memory unit

**4.4 Central Processing Unit**

Let add 3 more memory locations to a specific block that will have a built in capability to multiply, divide, subtract, and move its contents from one memory location onto another. The part we just added in is called "central processing unit" (CPU). Its memory locations are called registers.

Registers are therefore memory locations whose role is to help with performing various mathematical operations or any other operations with data wherever data can be found. Look at the current situation. We have two independent entities (memory and CPU) which are interconnected, and thus any exchange of data is hindered, as well as its functionality.

**Bus:** That "way" is called "bus". Physically, it represents a group of 8, 16, or more wires. There are two types of buses: address and data bus. The first one consists of as many lines as the amount of memory we wish to address and the other one is as wide as data, in our case 8 bits or the connection line. First one serves to transmit address from CPU memory, and the second to connect all blocks inside the microcontroller.

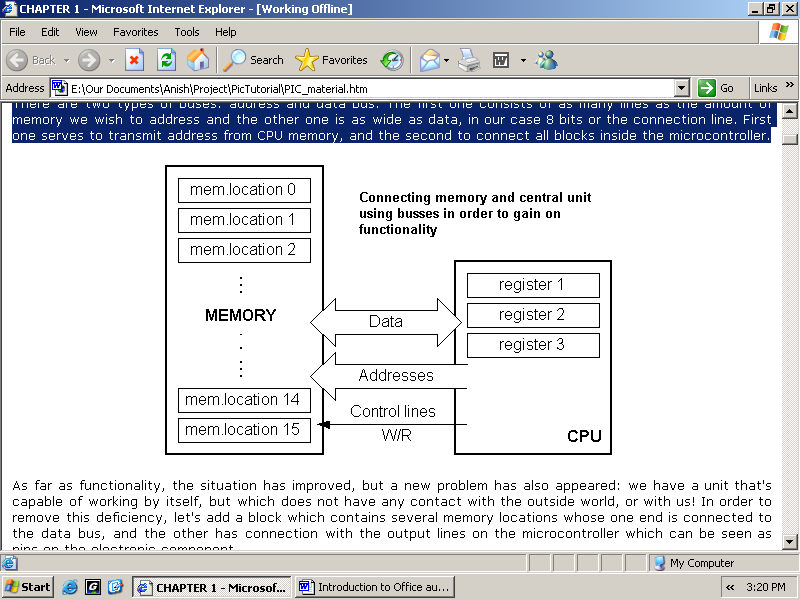


Figure 4.4: Representation of Bus

Those locations we've just added are called "ports". There are several types of ports: input, output or bidirectional ports. When working with ports, first of all it is necessary to choose which port we need to work with, and then to send data to, or take it from the port. When working with it the port acts like a memory location. Something is simply being written into or read from it, and it could be noticed on the pins of the microcontroller.

**4.5 Serial communication**

As we have separate lines for receiving and sending, it is possible to receive and send data (info.) at the same time. So called full-duplex mode block which enables this way of communication is called a serial communication block. Unlike the parallel transmission, data moves here bit by bit, or in a series of bits what defines the term serial communication comes from. After the reception of data we need to read it from the receiving location and store it in memory as opposed to sending where the process is reversed. In order for this to work, we need to set the rules of exchange of data. These rules are called protocol. Data goes from memory through the bus to the sending location, and then to the receiving unit according to the protocol.

**4.6 Timer unit**

The timer block this can give us information about time, duration, protocol etc. The basic unit of the timer is a free-run counter which is in fact a register whose numeric value increments by one in even intervals, so that by taking its value during periods T1 and T2 and on the basis of their difference we can determine how much time has elapsed. This is a very important part of the microcontroller whose understanding requires most of our time.

**4.7 Watchdog:**

One more thing is requiring our attention is a flawless functioning of the microcontroller during its run-time.

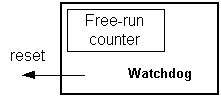


Figure 4.5: Watch dog timer

Suppose that as a result of some interference (which often does occur in industry) our microcontroller stops executing the program, or worse, it starts working incorrectly. Of course, when this happens with a computer, we simply reset it and it will keep working. However, there is no reset button we can push on the microcontroller and thus solve our problem. To overcome this obstacle, we need to introduce one more block called watchdog.

This block is in fact another free-run counter where our program needs to write a zero in every time it executes correctly. In case that program gets "stuck", zero will not be written in, and counter alone will reset the microcontroller upon achieving its maximum value. This will result in executing the program again, and correctly this time around.

**4.8 Analog to Digital Converter**

As the peripheral signals usually are substantially different from the ones that microcontroller can understand (zero and one), they have to be converted into a pattern which can be comprehended by a microcontroller.

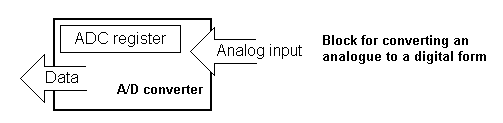


Figure 4.6: Analog to Digital converter

This task is performed by a block for analog to digital conversion or by an ADC. This block is responsible for converting an information about some analog value to a binary number and for follow it through to a CPU block so that CPU block can further process it.

**4.9 CISC and RISC**

Harvard architecture is a newer concept than Von-Neumann's. It rose out of the need to speed up the work of a microcontroller. In Harvard architecture, data bus and address bus are separate. Thus a greater flow of data is possible through the central processing unit, and of course, a greater speed of work. Separating a program from data memory makes it further possible for instructions not to have to be 8-bit words. It is also typical for Harvard architecture to have fewer instructions than von-Neumann's, and to have instructions usually executed in one cycle. Microcontrollers with Harvard architecture are also called "RISC microcontrollers".

RISC stands for Reduced Instruction Set Computer. Microcontrollers with von-Neumann's architecture are called 'CISC microcontrollers'

**4.10 Memory organization**

There are three memory blocks in each of these PICmicro MCUs. The Program Memory and Data Memory have separate buses so that concurrent access can occur.

**4.10.1 Program memory organization**

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound. The reset vector is at 0000h and the interrupt vector is at 0004h.

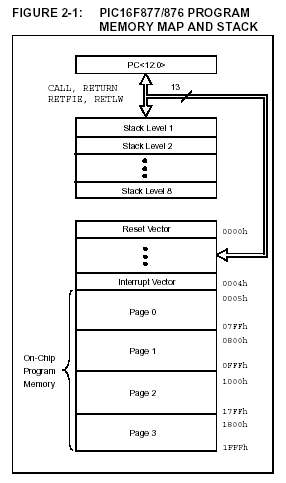


Figure 4.7: Program memory and Stack memory

**4.10.2 Data memory organization**

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1(STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

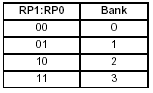


Figure 4.8: Table Register bank selection

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM.

**4.10.3 General purpose register file: The** register file can be accessed either directly or indirectly through the File Select Register FSR.

**4.10.4 Special** f**unction registers**

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. The Special Function Registers can be classified into two sets; core (CPU) and peripheral.

**4.10.5 Status register**

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. The TO and PD bits are not writable, therefore, the result of an instruction with the STATUS register as destination may be different than intended.

**4.11 I²C Protocol**

**I²C** is a multi-master serial computer bus invented by Philips that is used to attach low-speed peripherals to a motherboard, embedded system, or cell phone. The name stands for **Inter-Integrated Circuit** and is pronounced *I-squared-C* and also, incorrectly, *I-two-C*.

I²C uses only two bidirectional open-drain lines, Serial Data (SDA) and Serial Clock (SCL), pulled up with resistors. The I²C reference design has a 7-bit address space with 16 reserved addresses, so a maximum of 112 nodes can communicate on the same bus. The maximum number of nodes is obviously limited by the address space, and also by the total bus capacitance of 400 pF.

To maximize hardware efficiency and circuit simplicity, Philips developed a simple bi-directional 2-wire bus for efficient inter-IC control. This bus is called the Inter IC or I²C -bus. All I²C -bus compatible devices incorporate an on-chip interface which allows them to communicate directly with each other via the I²C -bus. This design concept solves the many interfacing problems encountered when designing digital control circuits. Here are some of the features of the I²C -bus:

* Only two bus lines are required; a serial data line (SDA) and a serial clock line (SCL)
* Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers
* It’s a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer
* Serial, 8-bit oriented, bi-directional data transfers can be made at up to 100 kbit/s in the Standard-mode, up to 400 kbit/s in the Fast-mode, or up to 3.4 Mbit/s in the High-speed mode
* On-chip filtering rejects spikes on the bus data line to preserve data integrity
* The number of ICs that can be connected to the same bus is limited only by a maximum bus capacitance of 400 pF.

For 8-bit oriented digital control applications, such as those requiring microcontrollers, certain design criteria can be established:

* + A complete system usually consists of at least one microcontroller and other peripheral devices such as memories and I/O expanders
  + The cost of connecting the various devices within the system must be minimized
  + A system that performs a control function doesn’t require high-speed data transfer
  + Overall efficiency depends on the devices chosen and the nature of the interconnecting bus structure.

To produce a system to satisfy these criteria, a serial bus structure is needed. Although serial buses don’t have the throughput capability of parallel buses, they do require less wiring and fewer IC connecting pins. However, a bus is not merely an interconnecting wire, it embodies all the formats and procedures for communication within the system.

Devices communicating with each other on a serial bus must have some form of protocol which avoids all possibilities of confusion, data loss and blockage of information. Fast devices must be able to communicate with slow devices. The system must not be dependent on the devices connected to it, otherwise modifications or improvements would be impossible. A procedure has also to be devised to decide which device will be in control of the bus and when. And, if different devices with different clock speeds are connected to the bus, the bus clock source must be defined. All these criteria are involved in the specification of the I²C -bus.

The I²C bus physically consists of 2 active wires and a ground connection. The active wires, called SDA and SCL, are both bi-directional. SDA is the Serial Data line, and SCL is the Serial Clock line.

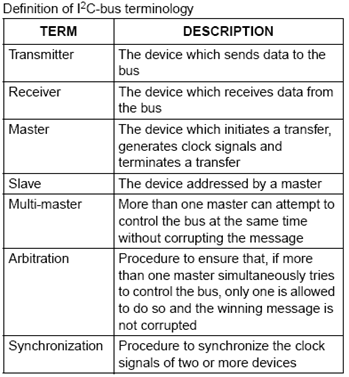
Every device hooked up to the bus has its own unique address, no matter whether it is an MCU, LCD driver, memory, or ASIC. Each of these chips can act as a receiver and/or transmitter, depending on the functionality. Obviously, an LCD driver is only a receiver, while a memory or I/O chip can be both transmitter and receiver.

The I²C bus is a multi-master bus. This means that more than one IC capable of initiating a data transfer can be connected to it. The I²C protocol specification states that the IC that initiates a data transfer on the bus is considered the *Bus Master*. Consequently, at that time, all the other ICs are regarded to be *Bus Slaves*.

I²C -bus compatible ICs allow a system design to rapidly progress directly from a functional block diagram to a prototype. Moreover, since they ‘clip’ directly onto the I²C -bus without any additional external interfacing, they allow a prototype system to be modified or upgraded simply by ‘clipping’ or ‘unclipping’ ICs to or from the bus. Here are some of the features of I²C -bus compatible ICs which are particularly attractive to designers:

* Functional blocks on the block diagram correspond with the actual ICs; designs proceed rapidly from block diagram to final schematic.
* No need to design bus interfaces because the I²C -bus interface is already integrated on-chip.
* Integrated addressing and data-transfer protocol allow systems to be completely software-defined
* The same IC types can often be used in many different applications
* Design-time reduces as designers quickly become familiar with the frequently used functional blocks represented by I²C -bus compatible ICs
* ICs can be added to or removed from a system without affecting any other circuits on the bus
* Fault diagnosis and debugging are simple; malfunctions can be immediately traced
* Software development time can be reduced by assembling a library of reusable software modules.

The I²C -bus supports any IC fabrication process (NMOS, CMOS, bipolar). Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognized by a unique address (whether it’s a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. Obviously an LCD driver is only a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see Table 1). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.



**4.11.1 I²C-BUS TERMINOLOGY**

The I²C -bus is a multi-master bus. This means that more than one device capable of controlling the bus can be connected to it. The possibility of connecting more than one microcontroller to the I²C -bus means that more than one master could try to initiate a data transfer at the same time. To avoid the chaos that might ensue from such an event - an arbitration procedure has been developed. This procedure relies on the wired-AND connection of all I²C interfaces to the I²C -bus. If two or more masters try to put information onto the bus, the first to produce a ‘one’ when the other produces a ‘zero’ will lose the arbitration. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line. Generation of clock signals on the I²C -bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus. Bus clock signals from a master can only be altered when they are stretched by a slow-slave device holding-down the clock line, or by another master when arbitration occurs.

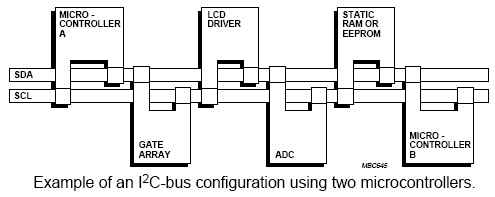


Figure 4.11,2: I²C BUS CONFIGURATION

Both SDA and SCL are bi-directional lines, connected to a positive supply voltage via a current-source or pull-up resistor (see Fig.3). When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function. Data on the I²C -bus can be transferred at rates of up to 100 kbit/s in the Standard-mode, up to 400 kbit/s in the Fast-mode, or up to 3.4 Mbit/s in the High-speed mode. The number of interfaces connected to the bus is solely dependent on the bus capacitance limit of 400 pF.

The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

• Serial clock (SCL) – RC3/SCK/SCL

• Serial data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

REGISTERS

The MSSP module has six registers for I²C operation.

These are:

• MSSP Control Register (SSPCON)

• MSSP Control Register 2 (SSPCON2)

• MSSP Status Register (SSPSTAT)

• Serial Receive/Transmit Buffer Register (SSPBUF)

• MSSP Shift Register (SSPSR) – Not directly accessible

• MSSP Address Register (SSPADD)

SSPCON, SSPCON2 and SSPSTAT are the control and status registers in I²C mode operation. The SSPCON and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write. SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from. SSPADD register holds the slave device address when the SSP is configured in I²C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the baud rate generator reload value.

**4.12. Display unit**:

Cheapest display unit available is LCD. But Pi has an advantage that other display units can be connected to it directly through display port.

**4.13 Power Supply:**

Power Adaptor 12 Volt 1 Amp Charger AC INPUT 100-240V DC OUTPUT 12V 1A Product Description: Professional 12 Volt DC 1 Amp power supply is suitable for powering a wide range of applications including CCTV cameras and wireless routers. Features: 100% Brand New Excellent Quality Short Circuit, Over Voltage & Over Current Protection. Meet CEC Energy Efficiency Level IV. Incredibly Low Fault Rates No Minimum Load. This power supply is a regulated Center Positive power supply and has a 2.1mm x 5.5mm Jack It's plug design is for Indian power socket. So, no plug converter is required. Compact size & light weight. High Reliability. Regulated Stable Voltage. Good quality SMPS Based Adapter Power LED Monitor (LED Glow when in Use) Stabilized Output, low ripple & low interference Single Output Voltage High Efficiency & low energy consumption Input - 100-240 VAC 50/60hZ Category - Switch Mode Power Adaptor (SMPS) Output Type - DC Output - 12Volts 1Amp PLEASE Applications: Powerful 12v 12w 1A max Current Draw. Replaces lower amped adapters 12v 0.5A 1A. 1.5A etc Smart Replacement Gadget Power Supply for LED, SMD, LED Strip, RGB LED Strip Ideal for Routers / Modems / Mobile Phones / Mp3 players / POS Machines etc. Best for Routers, Wifi Routers security/spy camera receiver and some advanced cameras CCTV , Gadgets , Portable Players , Set Top Boxes, best for Toys etc., Charging or any gadgets as per the rating of the device , please study and then buy as this a very technical item only works as per its precise current outputs This power supply is an ideal replacement for a wireless network router such as the Netgear DG834, DG834GT, DG934 etc.. plus a range of many other wireless routers. You will need to check the DC socket size & power rating with the supplier of the router you are using as we cannot confirm it will work.

**4.14 TEMPERATURE SENSOR**

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies

The LM35 datasheet specifies that this ICs are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature.  
The LM35 thus has an advantage over linear temperature sensors calibrated in ? Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centi-grade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4?C at room temperature and ±3⁄4?C over a full −55 to +150?C temperature range.

**4.15 MOTOR DRIVER**

         The L293D motor driver is available for providing User with ease and user friendly interfacing for embedded application. L293D motor driver is mounted on a good quality, single sided non-PTH PCB.  The pins of L293D motor driver IC are connected to connectors for easy access to the driver IC’s pin functions. The L293D is a Dual Full Bridge driver that can drive up to 1Amp per bridge with supply voltage up to 24V. It can drive two DC motors, relays, solenoids, etc. The device is TTL compatible. Two H bridges of L293D can be connected in parallel to increase its current capacity to 2 Amp.

**4.16 HUMIDITY SENSOR**

Measuring the humidity has an important role in the understanding of the environment around you. By knowing the amount of water in the air, you can make decisions about how to best create the atmosphere you desire, or just measure it for future reference. Wine cellars and greenhouses are among applications that measuring humidity can be useful. By using a simple solution such as the SHT21 Breakout Board, your able to quick measure the relative humidity (%RH) and temperature of your environment quickly and easily. The device is easy to communicate with, using I2C simple commands are sent to the chip, which then responds with the measurements. We`ve even added optional I2C pull ups that can be enabled using the PU (pull up) jumper on the board for your convenience and ease during bread boarding.

**4.17 Moisture Sensor**

The humidity sensor is used to measure the water content (moisture) of the soil. This sensor reminds the user to irrigate their plants and also controls the moisture content of the soil. It has been widely used in agriculture, irrigation and the land botanical garden. The ground moisture operating voltage is 5 V, the current required is less than 20 mA, the interface is analog type sensors and operate between 10 and 20℃. The soil moisture sensor uses capacitance to measure the dielectric permittivity of the surrounding soil. In the soil, dielectric permittivity is a function of water content. The sensor creates a voltage proportional to the dielectric permittivity and, therefore, to the water content of the soil. The sensor calculates the average water content over the entire length of the sensor. The soil moisture sensor is used to measure the loss of moisture over time due to evaporation and plants. Monitor soil moisture content to control irrigation in greenhouses and improve bottle biology experiments. The hardware and software needed for the soil moisture sensor is Arduino IDE (Senpinar 2018) humidity sensor software, and the Arduino Uno board. The soil moisture brings connected to the VCC% v of Arduino UNO, GND soil moisture and interconnected sensor Arduino UNO and the last door of the A0 sensor connected to the 0 Arduino analogue board (Baraka et al., 2013). The program will generate the sensor value as output. Take different types of terrain and insert the sensor into the ground. As a result, you will get the value of moisture present in the soil. For demonstration purposes, the user can keep the sensor on the ground. On the serial monitor. Note that the sensor will read the moisture on the floor and show the output.

**4.18 Jump Wire**

A jump cable is used to connect the test plate, the prototype or the internal circuit with other non-joined instruments..

**4.19 Wheels:**

Robotic vehicle has two rear wheels. Universal wheel is fixated in wheelchair base in front both wheels have the same diameter. The two rear wheels are drive using two DC motor connected to each wheel. The instructions are passed through micro-controller to the motor, the wheels rotated in the specific direction.

**4.20 Motors:**

Motors are arguably one of the most important parts of a mobile robotics platform. Excessive use of motors can cause waste of electricity from the on-board batteries. At critical times undersized motors could be short on torque. There are various shape and size of motors Figure shows the 12V DC motor use in wheelchair. The 12VDC motor is connected to the wheels and driver of the microcontroller ATmega328p. One Driver is sufficient to control two motors simultaneously

**4.21 Bluetooth Device:**

Bluetooth device is a main component from which the connection is to be made. It provides security as one device is connected at a time. Frequency is of 2. 4GHz.It is much more cost effective. Used for serial communication between android device and the Robotic vehicle.

**CHAPTER 5**

**SOFTWARE DESCRIPTION**

**5.1 INTRODUCTION**

The software tools which are used for the implementation are listed below, they are

1. MPLAB IDE
2. C18 COMPILER

**5.2 MPLAB IDE**

MPLAB integrated development environment is a comprehensive editor, project manager and design desktop for application of development of embedded design using Microchip PIC MCU and PIC DSC.

MPLAB is a window operating system software program that runs on a PC to develop application for microchip microcontroller and digital signal controller. It is called an integrated development environment or IDE, it provides a single integrated environment to develop code for embedded microcontroller.

Embedded C language is used in MPLAB IDE. Embedded C is a set of language extensions for the C programming language by the C standard committee to address commonality issues that exist between C extensions for different embedded system. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

1. Code speed is governed by the processing power, timing constraints
2. Code size is governed by available program memory and use of programming language.

The Embedded software is associated with each processor which acts as a brain in each embedded systems. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems. Goal of embedded software programming is to get maximum features in minimum space and minimum time. Embedded software needs to include all needed device drivers at manufacturing time and the device drivers are written for the specific hardware.

**5.3 C18 COMPILER**

The MPLAB C18 compiler is a free-standing, optimizing ANSI C compiler for the PIC microcontroller unit. The compiler deviates from the ANSI standard X3.159-1989 only where the standard conflicts with efficient PIC micro MCU support. The compiler is a 32-bit Windows console application and is fully compatible with Microchip’s MPLAB IDE, allowing secure level debugging with the MPLAB ICE in circuit emulator, the MPLAB ICD 2 in circuit debugger or the MPLAB SIM simulator.

**MPLAB C18 has the following features:**

1. ANSI ‘89 compatibility.
2. Integration with the MPLAB IDE for easy-to-use project management and source level debugging.
3. Generation of re-alocatable object modules for enhanced code reuse.
4. Compatibility with object modules generated by the MPASM assembler, allowing complete freedom in mixing assembly and C programming in a single project.
5. Transparent read/write access to external memory.
6. Strong support for inline assembly when total control is absolutely necessary.
7. Efficient code generator engine with multilevel optimization.
8. Extensive library support including PWM, SPITM, I²C TM, UART, USART, string manipulation and math libraries.

**CHAPTER 6**

**CONCLUSION**

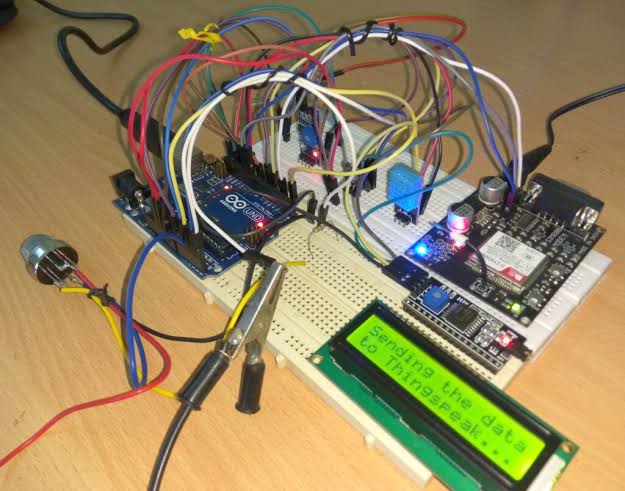
This proposed system is very effective in the field for irrigation, Environmental parameters monitoring and predator detection. The prediction of the crop in the field is also very accurate due to the use of sensors. Due, to the prediction of the crop to be grown the productivity of the crop will be more. This is very useful for the farmers for monitoring the situation of the field without going to the field. This system is very much helpful for the rice crops as it requires much water than the other crops. This system also will be useful for any type of crop. The developed system is of low cost and consumes less power due to the usage of IOT and microcontroller.

**CHAPTER 7**

**OUTPUT IMAGE**

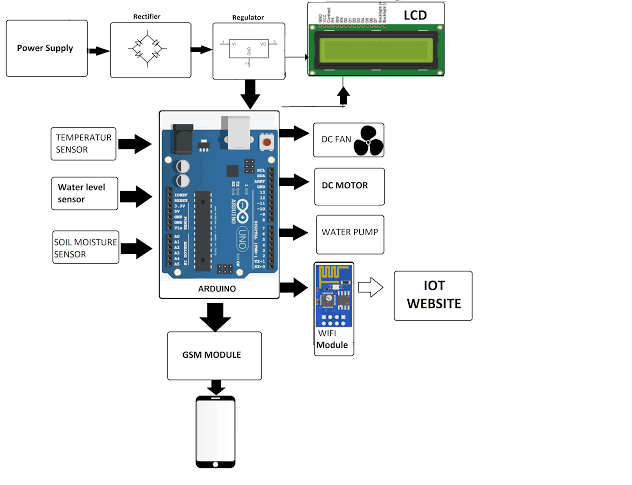
**7.1CONNECTIONS IOT BASED SMART AGRICULTURE**

The basic connections made on the circuit board with the help of arudino and the temperature sensor and LCD

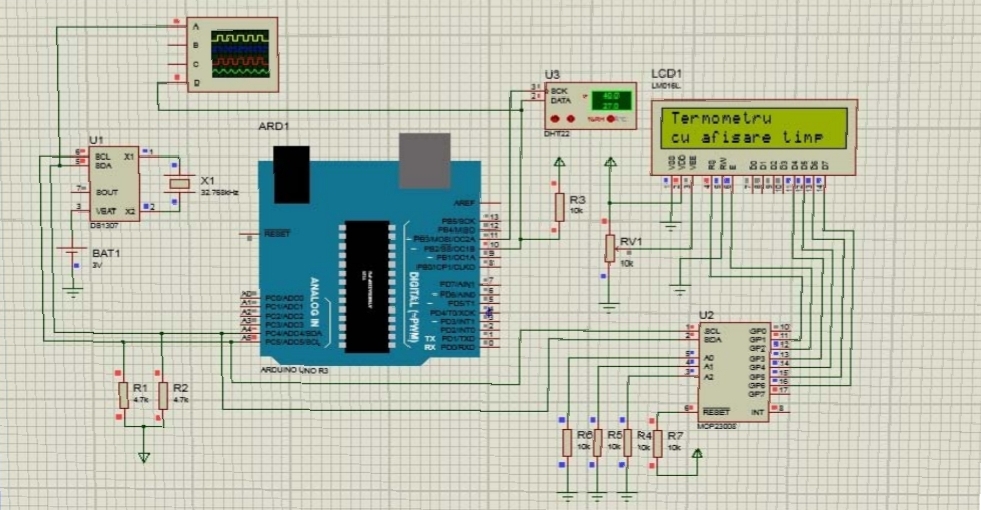
****

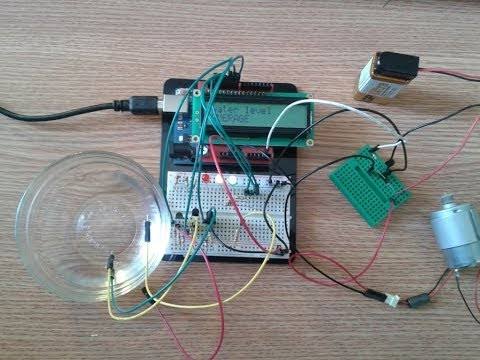
**7.2 BLOCK DIAGRAM**

Circuit connections for the Arudino.



**7.3 SIMULATION FOR TEMPERATURE AND HUMIDITY SENSOR**

****

**7.4 SIMULATION FOR WATER LEVEL SENSOR**

**REFERENCE PAPERS**

1. Benahmedkhelifa, DouliAmel, BouzekriAmel, Chabane Mohamed, Benahmed Tarek, “Smart irrigation using internet of things”, The Fourth International Conference on Future Generation Communication Technologies (FGCT 2015), pp. 26th October 2015.

2. Chandan Kumar sahu, Pramitee Behera,” A Low-Cost Smart Irrigation Control System”, IEEE Sponsored 2nd International Conference On Electronics and communication system (ICECS 2015), pp. 18th June, 2015.

3. Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto- Garibay, and Miguel Ángel Porta-Gándara, “Automated Irrigation System Using a Wireless Sensor Network and GPRS Module”, IEEE Transactions on instrumentation and measurement, pp. 19th august, 2013.

4. Pushkar singh and Sanghamitrasaikia, “Arduino-Based Smart Irrigation Using Water Flow Sensor, Soil Moisture Sensor, Temperature Sensor and ESP8266 WiFi Module”, Humanitarian Technology Conference (R10-HTC), IEEE Region 10, pp. 24th April 2017.

5. Ravi Kishore Kodali and Borade Samar Sarjerao, “A Low-Cost Smart Irrigation System Using MQTT Protocol”, IEEE Region 10 Symposium 19thOctober, 2017.

6. Riadhzaier, Slim zekri, Hemanathajayasuria, Abbas teirab, Nabil hamza and Hamed al-Busaidi, “Design and Implementation of Smart Irrigation System for Groundwater Use at Farm Scale“, 7th International Conference on Modelling, Identification and Control (ICMIC 2015), pp. 18th February, 2016.

7. SuhinthanMaheswararajah, Saman K. Halgamuge,Kithsiri B. Dassanayake, and David Chapman, “Management of Orphaned- Nodes in Wireless Sensor Networks for Smart Irrigation Systems”, IEEE Transactions on Signal processing, Vol.59, No. 10, pp. October, 2011.

8. T.C.Meyer and G.P.Hancker, “Design of smart sprinkler system”, IEEE Sensors Journal, pp. 7th January 2016.

9. Vidya sagar S, Ragav Kumar G, Lino X T Xavier, Sivakumar S, Dr. Ramesh babudurai, “Smart irrigation system with flood avoidance technique”, 2017 3rd International conference on science technology engineering & management (ICONSTEM), pp. 18 January 2018.

10. Joaquin Gutierrez et al., “Smart Phone Irrigation Sensor”, IEEE SensorsJournal.,pp. Sept 2015.

11. Vikhram.B et al.,”Animal Detection system in farm areas”, International Journal of Advanced Research in Computer and Communication Engineering, pp. March, 2017.

12. Prathibha SR, Anupama Hongal, Jyothi MP, “IOT Based Monitoring System in Smart Agriculture”, International conference on Recent Advances in Electronics and Communication Technology, pp. 2017.