**Chapter 1**

# Introduction

India is ranked to be the second largest in agriculture. Agriculture is the backbone of India. Today the Indians are second largest in the production of wheat, rice etc. cropping patters has been changed according to the lifestyle of people. Technology makes people lazy these days and so the farming these days are also developed with new technology. Plant protection is a mandatory thing in agriculture in order to intimate the farmers about their crops, the farmers should login to the website using particular username and password so that he can see the details that include date, time and data when is sent by the Arduino board using Wi-Fi from which they can monitor their lands in a easy way. Dry lands are a big threat to the farmers. To take care of these major and mandatory things few things are being implemented in our paper. In this project we are proposing the system which prevents the spoilage of crops due to heavy rains and sun rays. This is achieved with embedded system design using IOT technology. The actual concept of this project is protecting the crops from heavy rainfall and sun rays by covering the field automatically and also to save the collected rainwater. To achieve this, in this system we use IOT, sensor and soil moisture sensor. We also use renewable energy sources, solar power that is generated from solar panel as the power source to this project. Large number of the population depends on leaf crop either for its cultivation or for the purpose of processing. It is observed that the development in agriculture is sluggish nowadays due to the attack of disease.

## Benefits of improved farming technologies

The benefits of improved farming technologies depend on whether India develops infrastructure such as irrigation network, flood control systems, reliable electricity usage and production capacity. As irrigation becomes the major part to have growth at economic status of our country. In time watering and proper application of bio fertilizer, along with proper monitoring of fields results in the social modernization of Indian agriculture system. This became quite interesting concept which is going to be major tool to reduce the power cost required for atomization of agriculture. This system uses the integration of both wired and wireless techniques and ARM controller to have regular monitoring on the environmental

conditions of farm and provides the necessary precautions to be taken for yield to increase for modern agriculture.

In the world, the economy of many countries is dependent upon agriculture. In spite of economic development agriculture is the backbone of the economy. Agriculture is the main stay of economy. It contributes to the gross domestic product. Agriculture meets food requirements of the people and produces several raw materials for industries. But because of animal interference and fire in agricultural lands, there will be huge loss of crops. Crop will be totally getting destroyed. There will be large amount of loss of farmer. To avoid these financial losses, it is very important to protect agricultural field or farms from animal and fire. To overcome this problem, in our proposed work we shall design a system to prevent the entry of animals into the farm. Our main purpose of project is to develop intruder alert to the farm, to avoid losses due to animals and fire. These intruder alert protect the crop from damaging that indirectly increase yield of the crop. The develop system will not harmful and injurious to animal as well as human beings. Theme of project is to design a intelligent security system for farm protection by using Embedded system.

India is an agricultural land. Farming has unendingly been India's most critical economical sector. While the greater part of India's population is indulged into farming, the farmers still experience numerous issues. Accordingly, interruption of creatures in local locations is being raised step by step which is influencing the human existence, property that makes struggle among human and creatures. Agriculture is the foundation of the economy, nevertheless, would bring about gigantic harvest misfortune due to creature interruption in agricultural land. Elephants and other creatures entering people’s place of residence has bought adverse consequence in different ways, for example, crop annihilation, harm to food stores, water supply, homes and other properties, injury, and human demise. Struggle between human creatures may likewise be a difficult issue where huge amounts of cash are squandered, and life is in danger. Farmers in India have been confronting genuine dangers from natural calamity, bugs and harm by creatures prompting lower yields. Conventional techniques trailed by farmers aren't much viable and it's not achievable to recruit monitors to focus an eye on the yields and prevent nature creatures. Consequently, this zone is to be checked consistently to forestall section of this sort of creatures or the other undesirable

## Agricultural land it works

Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds, and fire etc. This leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field 24 hours and guard it. So here we propose automatic crop protection system from animals and fire. This is a Arduino Uno based system using microcontroller. This system uses a motion sensor to detect wild animals approaching near the field and smoke sensor to detect the fire. In such a case the sensor signals the microcontroller to take action. The microcontroller now sounds an alarm to woo the animals away from the field as well as sends data to the farmer and makes alert, so that farmer may know about the issue and come to the spot in case the animals don’t turn away by the alarm. If there is a smoke, it immediately turns ON the motor. This ensures complete safety of crops from animals and from fire thus protecting the farmer’s loss. This is a Arduino Uno based system using microcontroller. This system uses a motion sensor to detect wild animals approaching near the field and smoke sensor to detect the fire. In such a case the sensor signals the microcontroller to take action.

**Chapter 2**

# Literature Survey

Over 50 years since its independence, India has made immense progress in its agriculture system to increase food grain. Two years of severe drought in 1965 and 1966 convinced India to reform its agricultural policy. Even though agricultural policy was success, irrigation infrastructure was very poor hence Indian farmer innovated with tube-wells, to harvest ground water. New technology were adopted to irrigate the agriculture land in large. The lasting benefits of improved farming technologies depends on whether India develops infrastructure such as irrigation network, flood control systems, reliable electricity usage and production capacity. As irrigation becomes the major part to have growth at economic status of our country. In time watering and proper application of bio fertilizer, along with proper monitoring of fields results in the social modernization of Indian agriculture system.This became quite interesting concept which is going to be major tool to reduce the power cost required for automization of agriculture. This system uses the integration of the both wired and wireless techniques and ARM controller to have regular monitoring on the environmental conditions of farm and also provides the necessary precautions to be taken for yield to increase for modern agriculture.

Balaji Banu [1] designed a wireless sensor networks to observe the conditions of the farming and increasing the crop yield and quality. Sensors are used to monitor different conditions of environment like water level, humidity, temperature etc., The processors ATMEGA8535 and ICS8817 BS, analog to digital conversion and wireless sensor nodes with wireless transceiver module based on Zig bee protocol are used in the designing the system. Database and web application is used to retrieve and store data. In this experiment the sensor node failure and energy efficiency are managed.

Liu Dan [2], Joseph Haule, Kisangiri Michael [3] and Wang Weihong, Cao Shuntian [38] carried out experiments on intelligent agriculture greenhouse monitoring system based on ZigBee technology. The system performs data acquisition, processing, transmission and reception functions. The aim of their experiments is to realize greenhouse environment system, where the of system efficiency to manage the environment area and reduce the money and farming cost and also save energy. IOT technology here is based on the B-S structure and cc2530 used like processing chip to work for wireless sensor node and coordinator. The

gateway has Linux operating system and cortex A8 processor act as core. Overall the design realizes remote intelligent monitoring and control of greenhouse and also replaces the traditional wired technology to wireless, also reduces manpower cost.

Joseph haule [3], Dragoş Mihai Ofrim, Bogdan Alexandru Ofrim and Dragoş Ioan Săcăleanu

[18] have proposed an experiment that explains the use of wsn used in automating irrigation. Irrigation control and rescheduling based on wsn are powerful solutions for optimum water management through automatic communication to know the soil moisture conditions of irrigation design. The process used here is to determine the proper frequency and time of watering are important to ensure the efficient use of water, high quality of crop detection delay throughput and load. Simulation is done for agriculture by OPNET. Another design of wsn is deployed for irrigation system using Zig bee protocol which will impact battery life. There are some drawbacks as wsn is still under development stage with unreliable communication times, fragile, power consumption and communication can be lost in agricultural field. so automate irrigation system and scheduling based on wireless sensor networks are used. WSN uses low power and a low data rate and hence energy efficient technology. All the devices and machines controlled with the help of inputs received via sensors which are mixed with soil. Farmers can analyze whether the system performs in normally or some actions are need to be performed.

A system using sensors that monitor different conditions of environment like water level, humidity, temperature etc., the processor along with IC-S8817BS and wireless transceiver module with zigbee protocol is used.The field condition is sent to the farmer via mobile text messages and email from the experts. With this system Sensor node failure and energy efficiency are managed. Zigbee technology is used which sometimes lack in range of communication [1].

A system is proposed for intelligent agriculture greenhouse monitoring system based on Zigbee technology. The system performs data acquisition, processing, transmission and reception functions. The aim of their experiments is to realize greenhouse environment system, where the ofsystem efficiency to manage the environment area and reduce the money and farming cost and alsosave energy. IOT technology here is based on the BS structure andcc2530 used like processing chip towork for wireless sensor node and coordinator. The gateway has Linux operating system and cortex A8processor act as core. Overall the design realizes remote

intelligent monitoring and control ofgreenhouse and also replaces the traditional wired technology to wireless, also reduces manpower cost[3].

A system is proposed for plant growth which can be monitored using thermal imaging technique.Here the irrigation temperature distribution measurement (ITDM) technique has been implied In real time the thermal images comprising of both low and high temperature ITDM values gives betterirrigation. Thermal imaging can provide temperature value of all pixels in the field when compared to thermometry which only provides an average value.For temperatures which are very close inrange, thermal imaging leads to inaccurate information so that the objects can become indifferenciable [4].

A method to evaluate the use of wireless sensor network used in automating irrigation and data are sent to the web server through wireless communication. The sensors are used to sense the temperature, humidity, moisture for crop monitoring. The irrigation is automated when the sensor reading goes below the threshold values. The farmer is regularly intimated with the field conditions. It also explained that in greenhouses, light intensity control can also be automated in addition to irrigation. Here, the prediction of crop water requirement is not efficient [6].

Kwang-il Hwang and his teammates presented a paper on the designing and implementation of wireless sensor gateway for efficient querying and managing through world wide web [1]. Here paper has presented the architecture of the sensor gateway for web-based management and its implementation details.

Sirisha and her team presented a paper on wireless sensor based remote controlled agriculture monitoring system using zigbee [2]. The system consisted of the soi1 monitoring wireless sensor network and remote data center. The sensor node was developed using JN5121 modu1e and IEEE 802.15.4/ZigBee wire1ess microcontroller.

Sonali and her team published a paper on monitoring wireless sensor network using android based smart phone Application [3]. The proposed work of this project is to use the technologies of centralized computing and android programming for the development of the application.

Prof C. H. Chavan and group presented a paper on wireless monitoring of soil moisture, temperature & humidity using zigbee in agriculture [4]. The proposed hardware of this system

includes 8 bit AVR, Blue tooth module, Temperature, humidity and soil moisture sensors, LCD. The system is low cost & low power consuming so that anybody can afford it.

Prabha and her group members published a paper on real-time atomization of agricultural environment for social modernization of Indian agricultural system using Arm 7 [5]. This system uses the integration of the both wired and wireless techniques and ARM controller to have regular monitoring on the environmental conditions of farm and also provides the necessary precautions to be taken for yield to increase for modern agriculture.

Angel C and her teammate Asha S published a research paper on developing a smart environment in agricultural irrigation technique [6]. The paper focuses on a method for developing a smart environment to monitor the irrigational parameter in the entire field. The system also aims on reducing the energy consumption and the cost of communication.

M.Munnira Sulthana, E.Ramakalaivani and A.V.S. Elavarasi presented a paper on the topic- wireless sensor network for remote monitoring of crop field [7]. This paper presents the design and the implementation of a Wireless Sensor Network that monitors the air temperature, humidity and ambient light intensity in a crop field and from remote places.

M.K.Gayatri and J.Jayasakthi, Providing Smart Agribusiness Solutions to Farmers for Improved Yielding Using IoT"; The distributed computer gadgets that can create a complete processing framework from sensors to devices that notice information on the ground from farming field images and from human entertainers and accurately feed the information into the archives alongside the area as GPS arranges the information into the archive automatic irrigation [1-3],

V.Vidya Devi and G. MeenaKumari, "Constant Modernized Horticulture Robotization and Monitoring System" offers a review on how a robotic water system architecture has been built to increase the use of water for rural yields. In addition, an entry way unit manages data from sensors Wilsori'ELECTRIC FENCE" is a border that uses electric stuns to avoid a cap from intersecting species and persons. The stun voltage can have consequences ranging from distress to death. Today, most electric walls are used for horticultural fencing and various kinds of creature control animals detection [5-6] .For eg, military Handbook of Texas, a project report circulated by the Texas State Historical Affiliation, despite the fact that they are also used to protect high-security territories. 4th August 2011

2 Nikesh Gondchawar proposed a work on IoT based smart agriculture. The paper included the automation of irrigation in the farmland. The works of the farmer like spraying, weeding, etc., will be controlled by a Smart GPS based remotecontrolled robot. It also includes smart control and intelligent decision making.[2] The elements in this system are ZigBee modules, camera, actuators with microcontroller and raspberry pi. The robot is also used for moisture sensing, bird and animal scanning, keeping vigilance, etc., The system mainly focusses on smart irrigation and smart control using the data from the sensor set up in the agricultural field. The smart warehouse does temperature and humidity maintenance and theft detection in the warehous

V Nainwal, et al, [16] Sensors are used to detect the presence of objects in the surveillance area and the information is collected over time to extract the event of interest. The information gathered by the surveillance camera i.e., video or still images could be used for further analysis and detection of the intruding object. This system does not utilize advanced techniques for alerting the owner of that area.

Sneha Nahatkar et al, [1] proposed a home embedded surveillance system which evaluates the development of a low cost security system using small PIR (Pyroelectric Infrared) sensor built around a microcontroller with ultra-low alert power. The system senses the signal generated by PIR sensor detecting the presence of individuals not at thermal equilibrium with the surrounding environment. On detecting the presence of any unauthorized person in any specific time interval, it triggers an alarm & sets up a call to a predefined number through a GSM modem. After the MCU sends the sensor signals to the embedded system, the program starts the Web camera which then captures the images which can be viewed and analysed later.

Puja G, Mohammad Umair Bagali proposed the system. This project is based on surveillance with an animal ward-off system employed in farmlands in order to prevent crop vandalization by wild animals. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID’s, various PIR sensors are deployed in the area to detect any motion and hence turns ON a camera when movement is detected, thereby providing real time monitoring.

Sushanth and G. Sujatha, “IoT Based Smart Horticulture The paper seeks to leverage advanced innovation such as IOT and agribusiness using robotization. The key point for increasing the yield of successful harvests is the management of ecological conditions smart agriculture IOT

[7-9] This paper illustrates the enhancement of a system that can track temperature, dampness, humidity and, indeed, the creation of creatures that can crash agricultural crops..

M. Jaya Prabha, R. Ramprabha, V. Vasu Brindha, C. Asha Beaula t: Crops in farms are many times damaged by animals like buffaloes, cows, goats, birds and wild elephants. This causes major losses for the farmers. Farmers can not stay on the field for 24 hours and protect it. To overcome this problem, an animal detection system has been designed to detect the presence of animals and it offres a warning and divert the animal without any harm. The designed system will continuously check for any animal to enter the field. IR sensors and ultrasonic sensor are used in this project to detect animal movement and to give a signal to the controller. Further the animals are being divereted by generating sound and signals, and this signal is being transmitted to GSM and instantly give farmers warning, so the farmers will be aware of the difficulty and available to the spot just in case the animals do not show off by the alarm. The complete safety of crops was ensured by this system from animals thus protecting the farmer’s loss. Keywords: Animal detection, GSM module, IR sensor, Ultrasonic sensor.

Smart Irrigation And Crop Protection Using Arduino 'Thirrunavukkarasu R R Electronics and Communication Engineering 5Meeradevi T Electronics and Communication Engineering 6Ganesh Prabhu S Electronics and Communication Engineering This paper aims at designing and executing the advanced development in communication system for smart irrigation and crop protection from animals that invaded to farms like cows, goat, elephants,etc. and The famers can’t protect the entire farm by staying in farm for all day. So , the PIR sensor is kept in the field to watch out the animal motions. When it detect any movement recorded in the PIR senor. it starts alarming.

**Chapter 3**

# Problem Statement

Agriculture is the backbone of India. Plant protection is a mandatory thing in agriculture in order to intimate the farmers about their crops but there no system protections. Monitoring agricultural environment for various factors such as temperature and humidity. There is no system like to protect the crops from environmental disasters like heavy rain, heavy sun rays, heavy chemical industries pollutions, and fire detection to the crop. Agriculture has always been the primary and the most important sectors of Indian economy. Farmers are the backbone of one’s country, so it is important for us to make sure he has the access to resources that are essential. Conventional methods like scare crows are used even today in an agricultural field to avoid birds and animals from feeding on growing crops. There are many loopholes in such ideas and so improvising agricultural security has become a major issue these days. Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds, and fire etc. This leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field 24 hours and guard it. So here we propose automatic crop protection system

**Chapter 4**

# Objective Of Project

The main objective of this project is to develop a smart sensor network for an agricultural environment. monitoring agricultural environment for various factors such as temperature and humidity along with significance like to protect the crops from environmental like heavy rain, heavy sun rays, heavy chemical industries pollutions, and fire detection to the crop with help of sensors where former can get update on the field. India is ranked to be the second largest in agriculture aim is to protect the crop and give healthy food to society.

the goal of this project is to make ensure of individual farmer to monitor agriculture without high investment. the hardware is constructed with help natural renewable power sources and thus the major part of our work. naturally powered sources used to give more applicant working.

**Chapter 5**

# Existing System

* There is no proper crop protection.
* No natural renewable energy can be used to crops.
* No indication system for former.
* Proposed system:
* The system to build the information to the former to indicate and monitor.
* The system helps daily updating to the former get all data to analyses and to the task automatically.

**Chapter 6**

# Requirements

**Hardware and software: Hardware Requirements:**

1.Arduino mega 2.Sensors

3. WIFI

4.Solar Panel 5.Battery

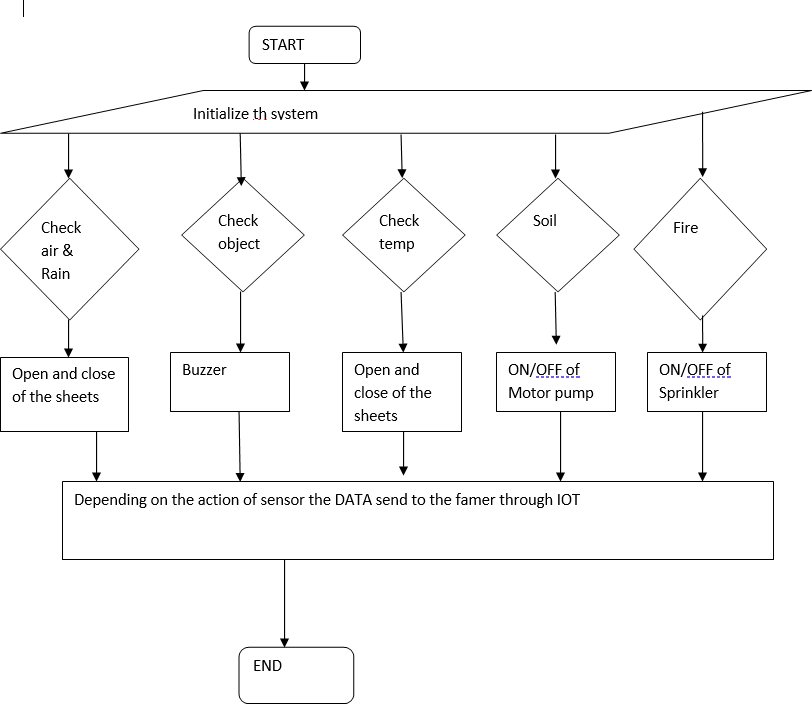
6.DC Motor

**Software Requirements:**

1. Ardunio IDE
2. EMBEDDED C
3. Adafruit IO

**Chapter 7**

# Methodology



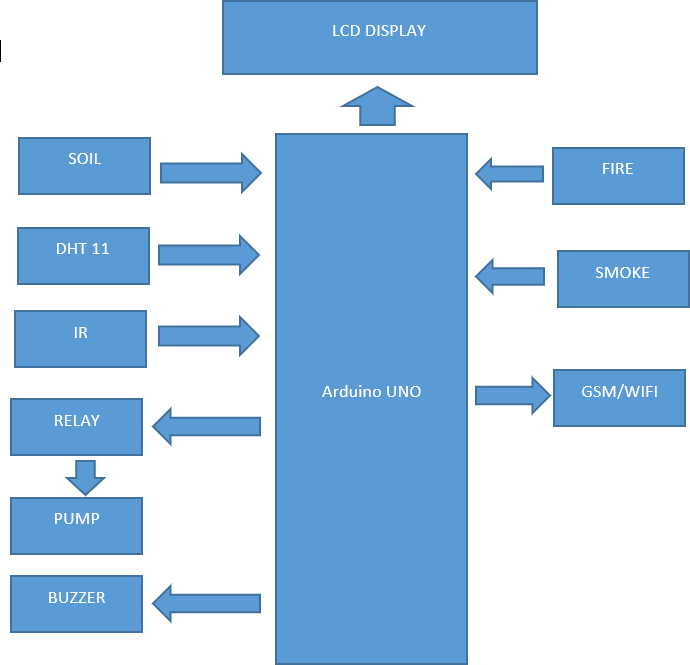
**Fig 7.1 : Flow chat**

This can be used in the agricultural fields.

* Controlling moisture of soil.
* Optimizing the water usage.
* Temperature and Humidity.
* Motor to be ON/OFF.

**Chapter 8**

# Design and Architecture

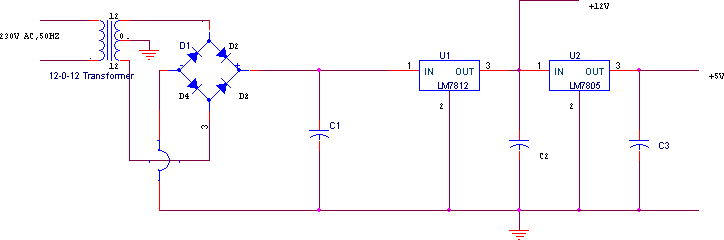


**Fig 8: Block Diagram of Sytem Working**

Firstly the processor checks for the availability of the solar energy, the solar panel is interfaced with the motor, which in turn is connected with the driver The solar panel rotates both in clockwise and anti-clockwise 180º and stops, stores the in a battery The sensor checks for the soil moisture content whose maximum and minimum. When the soil moisture content is less the pump motor will pump the water to the filed The temperature sensor will measure the surrounding temperature of the farm, The rain sensor will sense the heavy rain and closes the panel to protect the crop,All the above information will be informed to the user using IOT technology. Relay is connected to the pump which starts pumping water when the moisture sensor senses the land as dry. Moisture sensor is used for sensing the soil moisture of the crop land to feed them water,we also checking for the IR Sensor,if sensro detected the data is send and buzzer is on to avoid animals to come that area.

1. The system uses the solar power to work all sensors and controller
2. the main aim is to provide crop protection from different types environmental disaster or climate variation,like heavy sun ,heavy rain ,any gas leakage from industries and fire.
3. the sensor is placed to monitor the different caritas like heavy sun,rain etc.
4. the solar panel is used for protecting for high temperature and rain and industrial leakages.
5. the system is developed in such way that the opening and closing system on the crops made, depending on the area the panels are fitted.
6. the power generation is made to revenue or income to the framer as back up or backbone if crop get loss due any caritas .
7. the system is designed to take automatic decision and perform the task.
8. the all data can be sended to framer with help or iot using WIFI, or SMS using GSM.
9. object is placed to detect animals or any inactivity and alarm is buz.
10. fire sensor is to dtetce the fire and alert.
11. as mention different flat from to use and protect the crops with modern technology to make more effective

**8.1. Hardware Working System**



**Fig 8.1.1:Power supply unit**

In every project we need different voltages for different circuits. So we need to construct different power supply of different voltages employing different voltage transformers, rectifier circuits, filter circuits and regulator circuits.

This type of construction requires many components (transformers, capacitors, regulators). So the size of the power supply becomes bulky and costly. To overcome above disadvantages by using regulator IC’S the different voltages (12V, 9V) can be obtained with only one transformer.

The circuit diagram of Dual power supply is shown in the figure 8.1.1 The function of each component of the circuit is explained below. The circuit consists of following stages.

1. Transformer
2. Rectifier
3. Filter
4. Regulator

**Transformer**:

It is an electrical device which transfers the power from one winding to the other winding with isolation. All the electronic gadgets work for less voltage (normally 3V to 12V).So a step down transformer is used, whose function is to step down the AC voltage from 230V to required voltage depending on the need. In this project 12V-0-12V is used. The output of transformer is 12V AC which is connected to the diodes for rectification.

**Rectifier circuit**:

It employs diodes, which converts AC voltage into DC voltage. The output of rectifier circuit is not a pure DC. It also consists of some AC components, which is called ripples. In order to remove these AC components, filter circuits are employed. So the output of rectifier circuit is fed to the filter circuit (capacitor).

**Filter circuit:**

Filter circuit employs electrolytic capacitors in order to remove the AC components. As we know the capacitor does not allow DC components to pass through it because it offers high reactance to the DC component. And offers less reactance to the AC component, so all AC components will be bypasses through the capacitors to ground.

**Regulator:**

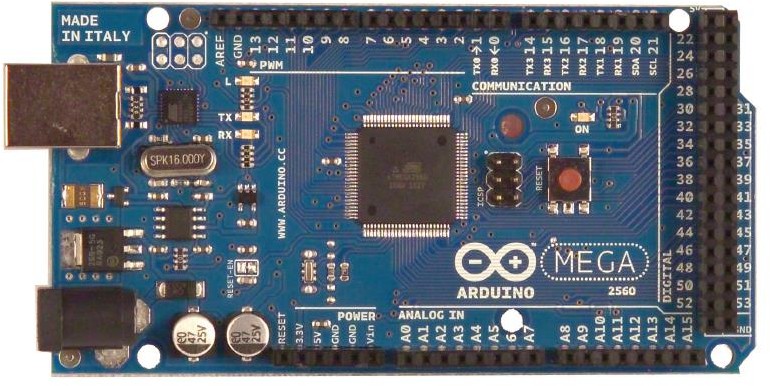
Regulator is an electronic circuit whose function is to keep output always constant though the input is varied. In this project the three terminal IC regulators of 7812 & 7805 is used for providing output DC voltages. E.g. 7809, the number 78 represents the positive regulator IC and 09 represents the output voltage i.e. output is 12V.

* + 1. **Component List:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO.** | **Component Name** | **Specification** | **Value** | **Quantity** |
| 1. | Transformer | Step-down | 12-0-12 | 1 |
|  |  | D1 | IN4001 | 1 |
| 2. | Diode | D2 | IN4001 | 1 |
|  |  | D3 | IN4001 | 1 |
|  |  | D4 | IN4001 | 1 |
|  |  | C1 | 1000uF,25V | 1 |
| 3. | Capacitors | C2 | 0.01uF | 1 |
|  |  | C3 | 0.01uF | 1 |
| 4. | Regulator | U1 | LM7812 | 1 |
|  |  | U2 | LM7805 | 1 |

**Fig 8.1.2: Components list**

* + 1. **Arduino Mega 2560 Datasheet**



**Fig.8.1.3: Arduino Mega 2560 Datasheet**

Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet).It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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 Act to- DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Summary

Microcontroller ATmega2560 Operating Voltage 5V

Input Voltage (recommended) 7-12V Input Voltage (limits) 6-20V

Digital I/O Pins 54 (of which 14 provide PWM output) Analog Input Pins 16

DC Current per I/O Pin 40 mA DC Current for 3.3V Pin 50 mA

Flash Memory 256 KB of which 8 KB used by bootloader SRAM 8 KB

EEPROM 4 KB

Clock Speed 16 MHz

**Power**

The Arduino Mega 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 than7V, 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 Mega2560 differs from all preceding boards in that it does not use the FTDI USB-toserial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

**The power pins are as follows:**

* + - * **VIN.** The input voltage to the Arduino board when it's using an external power source(as opposed 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.
      * **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator,or be supplied by USB or another regulated 5V supply.
      * **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
      * **GND.** Ground pins.

**Memory**

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

**Input and Output**

Each of the 54 digital pins on the Mega 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); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
      * **External Interrupts: 2** (interrupt 0), 3 (interrupt 1), 18 (interrupt 5),19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
      * **PWM:** 0 to 13. Provide 8-bit PWM output with the analogWrite() function.
      * **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
      * **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
      * **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.The Mega2560 has 16 analog inputs, 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 analogReference() function.There are a couple of other pins on the board:
      * **AREF.** Reference voltage for the analog inputs. Used with analogReference().
      * **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

**Communication**

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 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 Mega2560's digital pins.The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. For SPI communication, use the SPI library.

**Programming**

The Arduino Mega can be programmed with the Arduino software (download). For details,see the reference and tutorials.The ATmega2560 on the Arduino Mega 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 for details.

**Automatic (Software) Reset**

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 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 is connected to the reset line of the ATmega2560 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 Mega2560 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 Mega2560. 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 Mega2560 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**

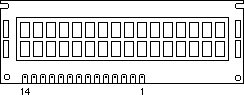
The Arduino Mega2560 has a resettable polyfused that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the usb port, the fuse will automatically break the connection until the short or overload is removed.

**Physical Characteristics and Shield Compatibility**

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively,with the USB connector and power jack extending beyond the former dimension. Three 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. The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins),analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and

Duemilanove / Diecimila. Please note that I2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

* + 1. **Lcd Operation:**



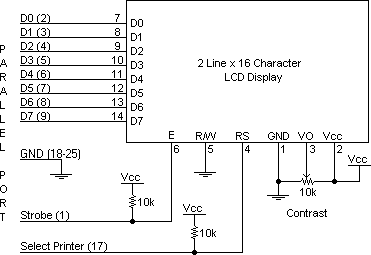
**Fig.8.1.4: Lcd Operation**

In recent years the LCD is finding widespread use replacing LEDs (seven segment LEDs or other multisegment LEDs). This is due to the following reasons:

1. The declining prices of LDCs.
2. The ability to display numbers, characters, and graphics. This is in contrast to LEDs,Which are limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LDC, thereby relieving the CPU of thetask of refreshing the LCD. In contrast, the LED must be refreshed by the CPU (or in some other way) to keep displaying the data.
4. Ease of programming for characters and graphics.

An LCD TV monitor utilizes the sunglasses concept to operate its colored pixels. On the flip side of the LCD screen, there is a huge bright light that shines out in the direction of the observer. On the front side of the display, it includes the millions of pixels, where each pixel can be made up of smaller regions known as sub-pixels. These are colored with different colors like green, blue, and red. Eac

* + 1. **LCD pin descriptions:**



**Fig.8.1.5 LCD pin descriptions**

The LCD discussed in this section has 14 pins. The function of each pin is given in Table 11.1. Figure 8.1.5 shows the pin positions for various sample LCDs.

**VCC, VSS, and VEE:**

While VCC and VSS provide +5V and ground, respectively, VEE is used for controlling LCD contrast. RS, register select. There are two very important registers Inside the LCD. The RS pin is used for their selection as follows. If RS=0, the instruction command code register is selected, allowing the user to send a command such as clear display, cursor at home, etc. IF RS=1 the data register is selected, allowing the user to send data to be displayed on the LCD.

**R/W (read/write):**

R/W input allows the user to write information to the LCD or read information from it. R/W=1 when reading; R/W=0 when writing.

**E(enable):**

The enable pin is used by the LCD to latch information presented to its data pins. When data is supplied to data pins, a high-to-low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins. This pulse must be a minimum of 450 ns wide.

**D0 - D7:**

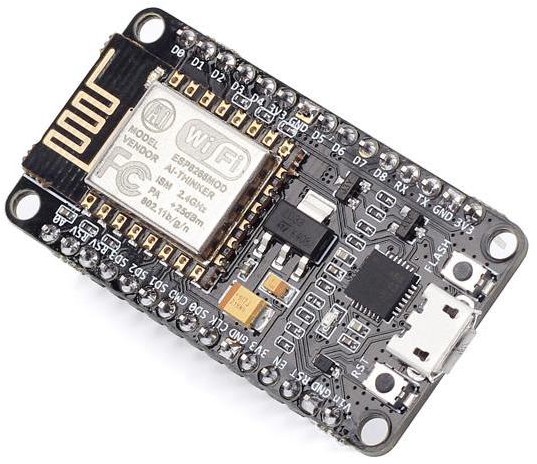
The 8-bit data pins, D0 - D7, are used to send information to the LCD or read the contents of the LCD’s internal registers.To display letters and numbers, we send ASCII codes for the letters A - Z, a - z, and numbers 0 - 9 to these pins while making RS=1.There are also instruction command codes that can be sent to the LCD to clear the display or force the cursor to the home position or blink the cursor. Table 11.2 lists the instruction command codes.We also use RS = 0 to check the busy flag bit to see if the LCD is ready to receive information. The busy flag is D7 and can be read when R/W=1 and RS = 0, as follows: if R/W = 1, RS = 0. When D7 = 1 (busy flag = 1), the LCD is busy taking care of internal operations and will not accept any new information. When D7 = 0, the LCD is ready to receive new information. Note: It is recommended to check the busy flag before writing any data to the LCD.

* + 1. **Pin Descriptions for LCD:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin | Symbol |  | I/O | Description |
| 1 | VSS |  | -- | Ground |
| 2 | VCC |  | -- | +5V power supply |
| 3 | VEE |  | -- | Power supply to control contrast |
| 4 | RS | I | -- | RS=0 to select command register, |
| RS=1 to select data register | | | | |
| 5 | R/W |  | I | R/w-0 for write, R/W=1 for read |
| 6 | E |  | I/O | Enable |
| 7 | DB0 |  | I/O | The 8-bit data bus |
| 8 | DB1 |  | I/O | The 8-bit data bus |
| 9 | DB2 |  | I/O | The 8-bit data bus |
| 10 | DB3 |  | I/O | The 8-bit data bus |
| 11 | DB4 |  | I/O | The 8-bit data bus |
| 12 | DB5 |  | I/O | The 8-bit data bus |
| 13 | DB6 |  | I/O | The 8-bit data bus |
| 14 | DB7 |  | I/O | The 8-bit data bus |

**Fig.8.1.6: Pin Descriptions for LCD**

* + 1. **NodeMCU v2 WIFI MODULE**



**Fig.8.1.7 : NodeMCU v2 WIFI MODULE**

The NodeMcu is an open-source firmware and development kit that helps you to Prototype your IOT product within a few Lua script lines.

**Features:**

Open-source Interactive Programmable Low cost Simple

Smart

WI-FI enabled

**Arduino-like hardware IO**

Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like arduino, but interactively in Lua script.

**Nodejs style network API**

Event-driven API for network applicaitons, which faciliates developers writing code running on a 5mm\*5mm sized MCU in Nodejs style.Greatly speed up your IOT application developing process.

**Specification:**

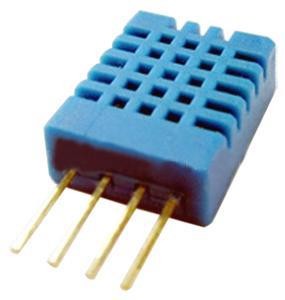
The Development Kit based on ESP8266, integates GPIO, PWM, IIC, 1-Wire and ADC all in one board.Power your developement in the fastest way combinating with NodeMCU Firmware! USB-TTL included, plug&play 10 GPIO, every GPIO can be PWM, I2C, 1-wire

FCC CERTIFIED WI-FI module（Coming soon）PCB antenna

**DHT11 Humidity &**

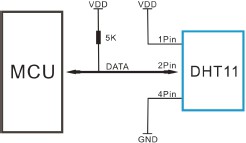
* + 1. **Temperature Sensor**

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness



**Fig.8.1.8 :Temperature Sensor**

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory,which are used by the sensor’s internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users’ request.



**Fig.8.1.9: Power and Pin**

* + 1. **Power and Pin**

DHT11’s power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

**Communication Process: Serial Interface (Single-Wire Two-Way)**

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first.

Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

* + 1. **Soil moisture sensor**

Soil sensors measure the water content in soil. A Soil level probe is made up of multiple Soil sensors.This Water level Sensor can be used to detect the moisture of soil or judge if there is water around the sensor, let the plants in your garden reach out for human help. Insert this module into the soil and then adjust the on-board potentiometer to adjust the sensitivity. The sensor would outputs logic HIGH/LOW when the moisture is higher/lower than the threshold set by the potentiometer. With help of this sensor, it will be realizable to make the plant remind you : Hey, I am thirsty now, please give me some water.

Application

* Botanical gardening.
* Water monitoring.

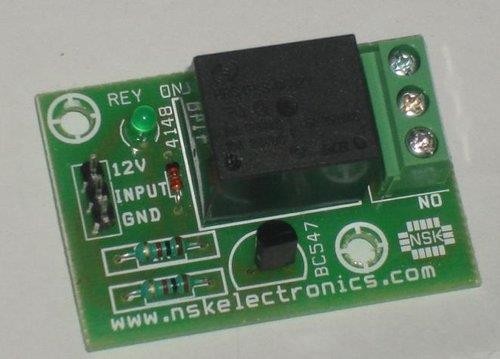
Features

* Digital output, easy to adjust.
* Nickel plating to avoid corrosion.
* Working voltage: 3.3V-5V.
* On-board LM393 chip.
* Dimension of the board: 3.2cm \* 1.4cm.



**Fig.8.1.10: Soil moisture sensor**

* + 1. **Relay:**



**Fig.8.1.11:Relay**

This Board can be used to Control Solonoids,Motors etc.

* Input Logic -5v level from MUC.
* Interfaced with Transister 547.
* Input Pin connected to Burg stick.

**Obstacle sensor**

IR Based Obstacle Detector Adjustable Range with POT Operating Voltage 5v

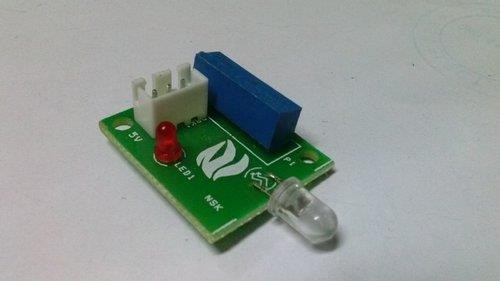
Sensitivity up to – 5cm-Adjustable Logic output -1/0 -5v

Application - Industrial safety devices

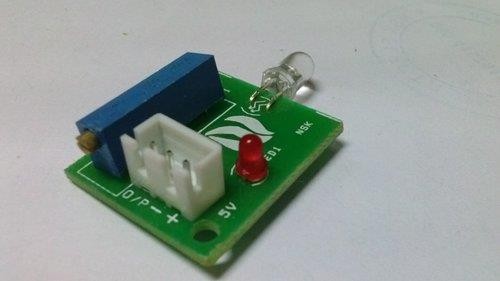
Ranging sensors include sensors that require no physical contact with the object being detected. They allow a robot to see an obstacle without actually having to come into contact with it. This

can prevent possible entanglement, allow for better obstacle avoidance (over touch-feedback methods), and possibly allow software to distinguish between obstacles of different shapes and sizes. There are several methods used to allow a sensor to detect obstacles from a distance. Below are a few common methods ranging in complexity and capability from very basic to very intricate.

* + - 1. **Fire sensor:**



**Fig.8.1.12.1: Fire sensor**



**Fig.8.1.12.2: Fire sensor**

**Description**

The Fire sensor is used to detect fire flames . The module makes use of Fire sensor and comparator to detect fire up to a range of 1 meters. Flame Sensor Working and Its Applications. A sensor which is most sensitive to a normal light is known as a flame sensor. That’s why this sens.

**Feature**

* + - * + Allows your robot to detect flames from upto 1 M away
        + Typical Maximum Range :1 m .
        + Calibration preset for range adjustment.
        + Indicator LED with 3 pin easy interface connector.
        + Input Voltage +5VDC

**Gas Sensor**

MQ135 Gas Sensor module for Air Quality having Digital as well as Analog output. Sensitive material of MQ135 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, The sensors conductivity is more higher along with the gas concentration rising. MQ135 gas sensor has high sensitivity to Ammonia, Sulphide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and suitable for different application. Used for family, Surrounding environment noxious gas detection device, Apply to ammonia, aromatics, sulfur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentration range: 10 to 1000 ppm.

**Specifications of MQ135 Gas Sensor Module:-**

Working voltage:DC 5V

Working Current: 150mA

DOUT: TTL output

AOUT: Analog output

Preheat time:Over 20s

Dimension: 32mm x 22m x 27mm(HIGH 27mm)



**Fig.8.1.13: Gas Sensor**

**8.1.14 Rain**



**Fig.8.1.14: Rain**

Comparator output signal clean wave good driving ability, than 15ma Output: digital switching output (0 and 1) and AO analog voltage output; With fixing bolt holes for easy installation Power adjust sensitivity; Rated voltage and 3.3V-5V Small Board PCB dimensions: 3.2cm x 1.4cm, Power indicator light, the output signal LED indicating lamp.Using wide LM393 voltage Comparator chip is LM393; Output types are AO (Analog o/p voltage) & DO (Digital switching voltage) The length & width of PCB module 3.2cm x 1.4cm; Sensitivity is modifiable through Trimpot ; Red/Green LED lights

Raindrop Sensor is a tool used for sensing rain. It consists of two modules, a rain board that detects the rain and a control module, which compares the analog value, and converts it to a digital value.The raindrop

**Specifications for this item**

|  |  |
| --- | --- |
| Brand Name | OLatus |
| Colour | Multicolour |
| Item Weight | 15.0 grams |
| Manufacturer Series Number | OL-SENSOR-RAIN |
| Model Number | OL-SENSOR-RAIN |
| Number of Items | 1 |
| Part Number | OL-SENSOR-RAIN |

* + 1. **DC MOTOR**



**Fig.8.1.15:DC Motor**

* + 1. **L293D Motor**



**Fig.8.1.16: L293D Motor**

**Features**

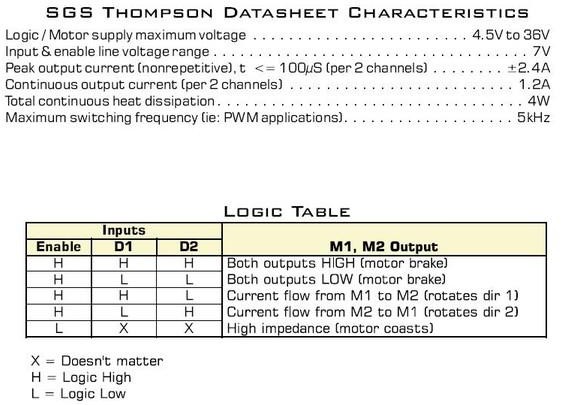
The L293D is a popular motor driver IC that is usable from 6 to12V, at up to 1A total output current. By itself, the IC is somewhat diffcult to wire and use, but the Compact L293D Motor Driver makes it much more convenient to use.

**Board Special Features**

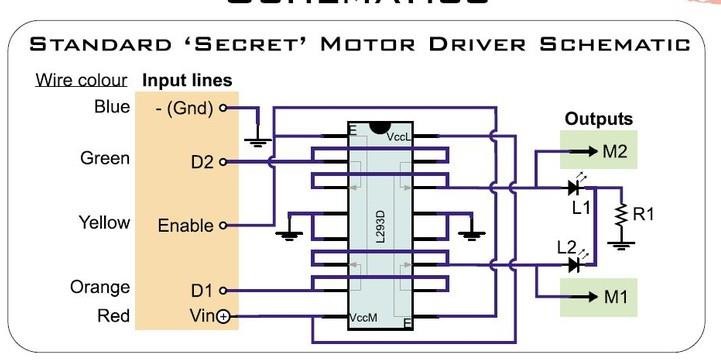
* Four motor direction indicator LEDS
* Schottky EMF-protection diodes
* Socket pin connectors for easy logic interfacing
* Enable pins are user accessible.

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. Dual H- bridge Motor Driver integrated circuit (IC). The l293d can drive small and quiet big motors as well,check.

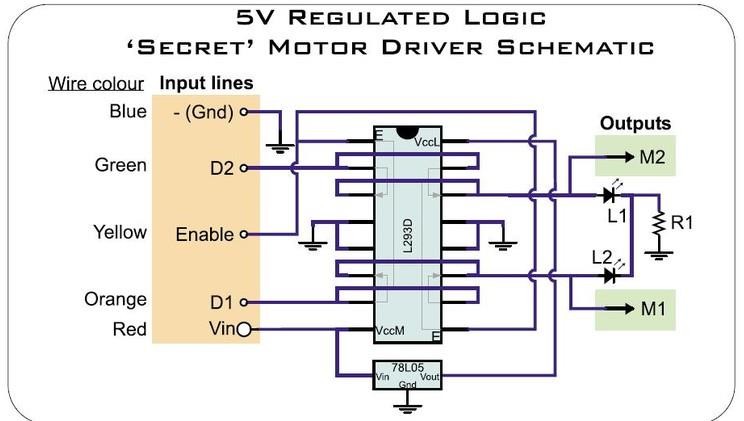
* + - 1. **L298 Features**



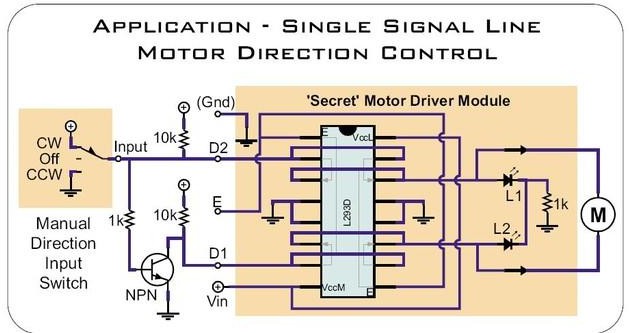
**Fig.8.1.16.1: logic table for L298**



**Fig.8.1.15.2: Standard Secret motor driver L298**



**Fig.8.1.16.2: 5v Regulated logic Secret motor driver L298**



**Fig.8.1.16.3: Signal line Motor Direction control L298**

**8.1.19.WATER PUMP:**



**Fig.8.1.19.Water Pump**

Water Pump The water pump can be defined as a pump which uses the principles like mechanical as well as hydraulic throughout a piping system and to make sufficient force forits future use. They have been approximately in one structure otherwise another because of early civilization.5 Volts water Pump Motor

**BUZZER**

A Buzzer or beeper is an, or piezoelectric. Audio signaling device, which may be mechanical, electro mechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or key

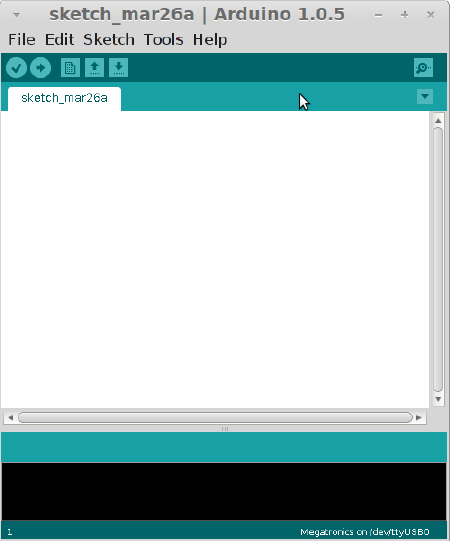


**Fig.8.1.20.Buzzer**

a piezo electric element may be driven with a piezoelectric audio amplifier. sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep. · raindrop sensor is a tool used for sensing rain. it consists of two modules, a rain board that detects the rain and a control module, which compares the analog value, and converts it to a digital value.the raindropa buzzer is a small yet efficient component to add sound features to our project/system. it is very small and compact 2-pin structure hence can be easily used on breadboard, perf board and even on pcbs

**8.2 SOFTWARE**

**Arduino IDE**



**Fig.8.2.1: Arduino IDE 3D-printer**

Arduino is an open-source project, enabling hobbyists to easily take advantage of the powerful Atmega chips. The Arduino IDE is the software where you can write code and upload it to the Atmega chip. The code is then executed on the chip. Most 3D-printer electronics are Arduino- compatible, they use the Atmega chip and enable the user to upload their code using Arduino. This includes Megatronics, Minitronics and RAMPS.

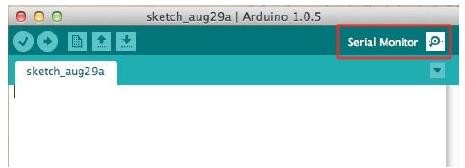
Before you can start using the electronics you need software 'firmware', that translates machine instructions (gcode) into actual movements. There are a few options here, including Marlin and Sprinter and Repetier. The actual firmware is not discussed in this document. You can use Arduino to upload this firmware onto your electronics.

This document will guide you in the steps you need to take. To upload a firmware, you must first open the files using File → Open. Select the.ino file from the directory containing the firmware. Arduino will open several tabswith files.Next step is to select the correct electronics board. From the Tools menu, locate the Board item.

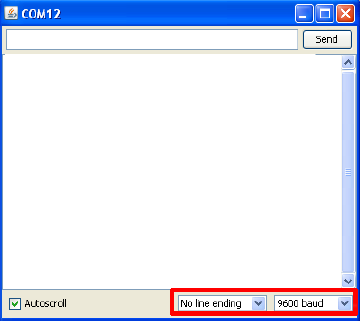
This item should include a few sub items, including Megatronics, Minitronics, Arduino mega 2560 (RAMPS with mega 2560) and Arduino Mega 1280 (RAMPS with mega1280). Select the board that fits your electronics. Also we need to select the serial port the electronics is connected to. In the Tools menu, locate the Serial port item. This should include at least one item if the board is connected and the drivers are installed properly. If there are multiple items here, you need to find out which is the correct one by unplugging the board and checking which port was removed.

Once you have set the board and serial port, you can upload the firmware bypressing File → Upload. Arduino will try to compile the firmware, if any errors occurthe process will stop and you will need to fix the errors before trying again.

Once compilation is complete, the actual upload will start. This may take a minute for alarge sketch.Arduino You can actually 'talk' to the firmware using the Serial monitor. Make sure the correct serial port is selected and locate the Serial monitor button.



**Fig.8.2.2: Arduino IDE 3D-printer This will open a new window:**



**Fig.8.2.3: Arduino IDE Com12**

Make sure the line ending is set to newline and the baud rate corresponds to your firmware (115200 mostly), or you will get jibberish. You can enter a command in the upper box (M105 for example), this will result in response from the firmware. The temperature in this case

**8.3. WEB APPLICATION**

Adafruit IO Overview

Adafruit IO is a system that makes data useful. Our focus is on ease of use, and allowing simple data connections with little programming required. IO includes client libraries that wrap our REST and MQTT APIs. IO is built on Ruby on Rails, and Node.js. Adafruit IO is currently in beta. If you would like to join the beta, head over toi o.adafruit.com to sign up [(http://](http://adafru.it/eZ8).Please)a[dafru.it/eZ8).Please](http://adafru.it/eZ8).Please) keep in mind that Adafruit IO is still in the beta testing stage. There will be bugs! Click Here to Submit a Bug or Feature Request <http://adafru.it/em2The> client libraries are a work in progress. Please check back often for updates.

**Getting Started**

If you haven't already, log into your Adafruit account and then head over to io.adafruit.com [(http://](http://adafru.it/fsU))a[dafru.it/fsU)](http://adafru.it/fsU)) and click onthe 'JOIN THE BETA LIST' button. We are slowly adding new beta users to help test Adafruit IO, and we will eventually openit up to everyone on the list.Once you have been invited to the beta test, open up io.adafruit.com and you will be sent to a welcome dashboard.Check out the following guides to understand the basics of creating a feed and a dashboard:

Adafruit IO Basics: Feeds [(http://](http://adafru.it/ioA))a[dafru.it/ioA)](http://adafru.it/ioA)) Adafruit IO Basics: Dashboards (<http://adafru.it/f5m)>

Also check out the projects page [(http://](http://adafru.it/iQB))a[dafru.it/iQB)](http://adafru.it/iQB)) for a list of projects and examples to help understand the service.Continue on reading this guide to learn about the client libraries that are available to send and receive data with Adafruit IO. In addition you can learn about the protocols that Adafruit IO uses and how to use them with your own client code.

**Client Libraries**

The Adafruit IO client libraries greatly simplify interacting with the server. We have a few libraries built out already: Arduino (<http://adafru.it/iQC)>

**Step #0 - adafruit.io key and feeds**

Before you can go crazy with Internetting your Things, you will need to do a little light config work to get adafruit.io ready for you.To do this we'll introduce three new terms

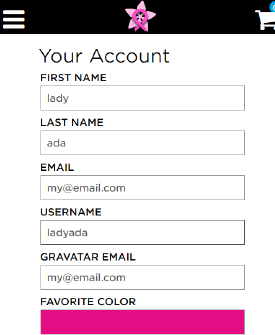
**Account username** - This is the name of your account, which you set when creating your adafruit account.

**Key -** this is a long, unique identifier that you use to authenticate any devices using your account. This is your password! Keep it safe! You get one key per account, but you can, at any time revoke and regenerate your key.

**Feed** - this is basically a set of data that you can read or write from like a sequentialfile. There is some history stored with feeds, with MQTT you cannot access historical data (REST does support it) but you can add data and you can receive the latest added data.

**Where to find your username**

You can find your username by visiting https://accounts.adafruit.com/ (<http://adafru.it/dyy)> and logging in. Your username is right there! you are connecting to adafruit io's mqtt server (a.k.a broker) - you could use another broker and as long as it fits the mqtt 3 or 3.1.1 specs, it ought to work. you are connecting via the internet - wifi, ethernet, and cellular are king here. other transports would need a gateway you are using an arduino or compatible - our code is fairly portable, but it order to keep the examples concrete, we'll be focusing on the arduino library you have already signed up for adafruit io (https://adafru.it/ez8) and logged in

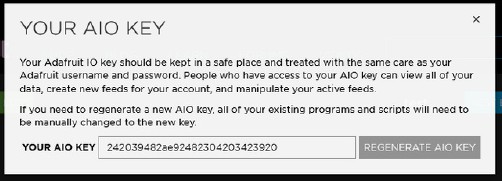


**Fig.8.3.1: Web page login**

Where to find your adafruit.io key Visit https://io.adafruit.com [(http://](http://adafru.it/tCt))a[dafru.it/tCt)](http://adafru.it/tCt)) and look in the top left. You will see a little navbar with a yellow key Click it to see your key. If you regenerate your key, your old key will no longer be valid and l need to update all your projects



**Fig.8.3.2: Web page New block**

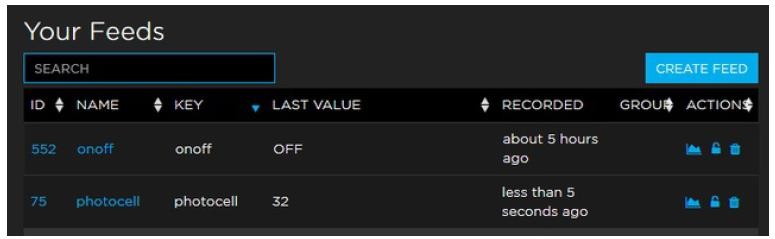


**Fig.8.3.3: Web page Your AIO Key**

**Create your first two feeds**

You can read up on how to do this here https://learn.adafruit.com/adafruit-io-basics-feeds (<http://adafru.it/ioA)> Once you've read that. Go to your feeds page and create two feeds

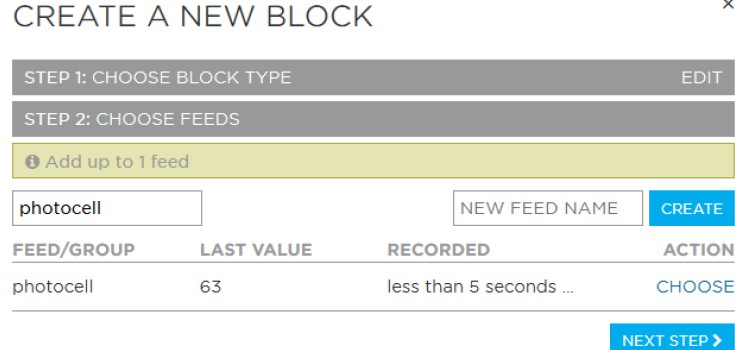
1. photocell - this feed will store light data from your device to adafruit.io
2. onoff - this feed will act as an on/off switch, sending datat o your device from adafruit.io



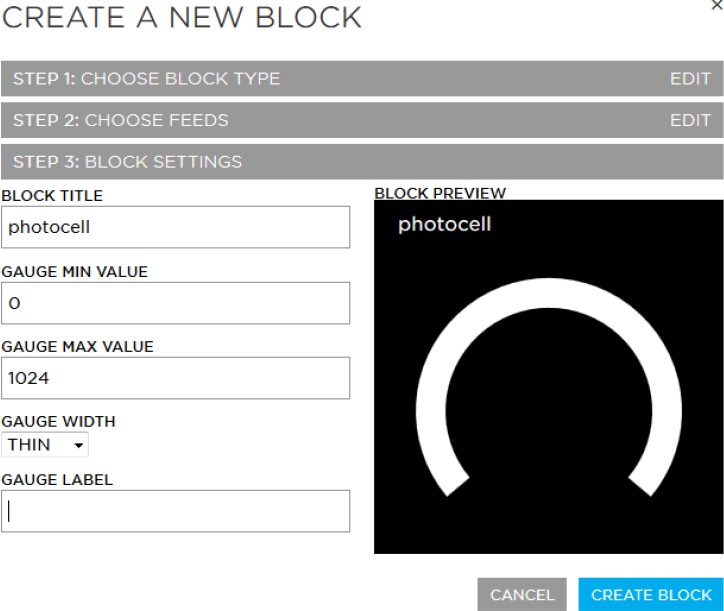
**Fig.8.3.4:Your Feeds**

**Create a dashboard**

Like feeds, this has its very own, excellent tutorial. Read all about it herehttps://learn.adafruit.com/create-an-internet-of-things-dashboard-with-adafruit-dot-io (<http://adafru.it/iWe)> Once you've read it, create a dashboard with a gauge connected to photocell Use a thin type gauge with min value 0 and max value 1024



**Fig.8.3.5: Create a new block Photocell**



**Fig.8.3.6:Create a new block**

**Arduino**

On an Arduino there are two different libraries you can use to access Adafruit IO. One library is based on the REST API, andthe other library is based on the MQTT API. The difference between these libraries is that MQTT keeps a connection to theservice open so it can quickly respond to feed changes. The REST API only connects to the service when a request is made so it's a more appropriate choice for projects that sleep for a period of time (to reduce power usage) and wake up only tosend/receive data. If you aren't sure which library to use, try starting with the Adafruit MQTT library below.

**Adafruit MQTT Client Library**

To use Adafruit IO with the MQTT protocol on an Arduino you can use theA dafruit MQTT Arduino library ([http://adafru.it/fp6).](http://adafru.it/fp6)) This is a general-purpose MQTT library for Arduino that's built to use as few resources as possible so that it can work withplatforms like the Arduino Uno. Unfortunately platforms like the Trinket 3.3V or 5V (based on the ATtiny85) have too littleprogram memory to use the library--stick with a Pro Trinket or better The Adafruit MQTT library currently supports the following networking hardware/platforms:

Adafruit CC3000 (<http://adafru.it/iRa)> Adafruit FONA (<http://adafru.it/dFz)> ESP8266 Arduino [(http://](http://adafru.it/eSH)Generic)a[dafru.it/eSH)Generic](http://adafru.it/eSH)Generic) Arduino Client Interface [(http://](http://adafru.it/fpb))a[dafru.it/fpb)](http://adafru.it/fpb)) (including Ethernet shield and similar network hardware) To install the library you can use the Arduino library

manager [(http://](http://adafru.it/flm))a[dafru.it/flm)](http://adafru.it/flm)) or download the library fromGitHub [(http://](http://adafru.it/fp7))a[dafru.it/fp7)](http://adafru.it/fp7)) and manually install it [(http:](http://adafru.it/dNR))/[/adafru.it/dNR).](http://adafru.it/dNR))

On some platforms the Adafruit MQTT library uses the hardware watchdog to help ensure sketches run reliably. You'll needto install the Adafruit SleepyDog sleep and watchdog library [(http://](http://adafru.it/fp8))a[dafru.it/fp8),](http://adafru.it/fp8)) again using either the Arduino librarymanager or with a manual install.

Finally make sure you have any required libraries for your network hardware installed, such as the CC3000library (<http://adafru.it/cFn)> or FONA library ([http://adafru.it/dDC).](http://adafru.it/dDC))

Once the library is installed open or restart the Arduino IDE and check out the example code included with the library. Theseexamples show the basic usage of the library, like how to connect to Adafruit IO, subscribe to receive changes to a feed, andhow to send values to a feed.

**PubSubClient MQTT Library**

Another popular MQTT library for the Arduino is the PubSubClient MQTT library [(http://](http://adafru.it/e1W))a[dafru.it/e1W)](http://adafru.it/e1W)) and it works great toaccess Adafruit IO. Note that the library only works with networking libraries that support the Arduino Client interface. Thismeans the library will work with the Ethernet shield, CC3000 or even ESP8266 Arduino, but not the FONA platform becauseit does not have a Client interface.

Below is a small example that shows using the PubSubClient library with the CC3000. To use this you will need theA dafruitCC3000 library [(http://](http://adafru.it/cFn))a[dafru.it/cFn)](http://adafru.it/cFn)) and PubSubClient library [(http://](http://adafru.it/e1W))a[dafru.it/e1W)](http://adafru.it/e1W)) installed in Arduino.Note that you'll need to change the following #define configuration lines at the top to fill in your wireless AP connectiondetails and Adafruit IO username, key, and feed name:

#define WLAN\_SSID "... your WiFi SSID..." #define WLAN\_PASS "... your WiFi password..."

#define ADAFRUIT\_USERNAME "... your Adafruit username (see accounts.adafruit.com)..." #define AIO\_KEY "... your Adafruit IO key..."

#define FEED\_PATH ADAFRUIT\_USERNAME "/feeds/feed-name/"

The FEED\_PATH is the path to publish or subscribe to for interacting with a feed. Notice that the path starts with the Adafruit account name and is followed by "/feeds/feed-name" (where "feed-name" is the name of the feed.

**Adafruit IO REST Client Library**

The Adafruit IO Arduino library [(http://](http://adafru.it/fpd))a[dafru.it/fpd)](http://adafru.it/fpd)) is a simple library for sending and receiving the latest value for a feed.

This library uses the send [(http://](http://adafru.it/iRb))a[dafru.it/iRb)](http://adafru.it/iRb)) and last [(http://](http://adafru.it/iRb))a[dafru.it/iRb)](http://adafru.it/iRb)) Adafruit IO REST API calls and takes care of allthe work to use the Adafruit IO REST API.The REST client library supports the following networking platforms/hardware:Adafruit CC3000 [(http://](http://adafru.it/iRa))a[dafru.it/iRa)](http://adafru.it/iRa))

Adafruit FONA (<http://adafru.it/dFz)> ESP8266 Arduino [(http://adafru.it/eSH)](http://adafru.it/eSH))

Generic Arduino Client Interface [(http://](http://adafru.it/fpb))a[dafru.it/fpb)](http://adafru.it/fpb)) (including Ethernet shield and similar network hardware) To install the library you can use the Arduino library manager [(http://](http://adafru.it/flm))a[dafru.it/flm)](http://adafru.it/flm)) or download the library from GitHub [(http://](http://adafru.it/fpd))a[dafru.it/fpd)](http://adafru.it/fpd)) and manually install it ([http://adafru.it/dNR).](http://adafru.it/dNR)) Finally make sure you have any required libraries for your network hardware installed.

**Chapter 9**

# Implementation

In this project we have started implementation on module wise in design part showing some of module are IR sensor,dht11 sensor, moisture sensor and smoke sensor, along with this are the sensor data connecting edge computing. Here we have stated using Arduino all smart device connected on this board here some set action appear into lcd and along with we have star run the code using application.

The Arduino Web Editor allows you to write code and upload sketches to any official Arduino board from your web browser We recommend you use Google Chrome. This IDE (Integrated Development Environment) is part of Arduino Create, an online platform that enables developers to write code, access tutorials, configure boards, and share projects. Designed to provide users with a continuous workflow, Arduino Create connects the dots between each part of a developer's journey from inspiration to implementation. Meaning, you now have ability to manage every aspect of your project right from a single dashboard. The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for new boards.

This IDE lets you write code and save it to the cloud, always backing it up and making it accessible from any device. It automatically recognizes any Arduino and Genuino board connected to your PC, and configures itself accordingly.

An Arduino account is all you need to get started.

SIGN UP TO ARDUINO

Create a new Arduino Account at this link. Complete the registration form, then hit the ‘Create Account ’ button. You will receive an email with a link to activate your account. Select the link and a new page will open with your confirmed account information. Note that you can also use this account to write posts on Arduino’s official Forum, buy products on the online store, add comments on the blog, as well as create tutorials on the Arduino Project Hub

LOG IN THE ARDUINO WEB EDITOR

Once you have successfully registered for an Arduino account, go to create.arduino.cc/editor. After accepting the term and conditions, you should receive an

email with a link to the Forum where you can report any bugs you may encounter and share your feedback.

Pick your Platform: Windows, Mac or Linux

The Arduino Web Editor can run on a variety of Platforms. If you are using Windows, Mac or Linux follow a simple flow to install the Arduino Web Editor plugin, which permits you to upload sketches from the browser onto your boards.

Are you encountering an issue while installing the Plugin? Please open a thread on this Forum page, we will try to troubleshoot your problem and get you up and running!

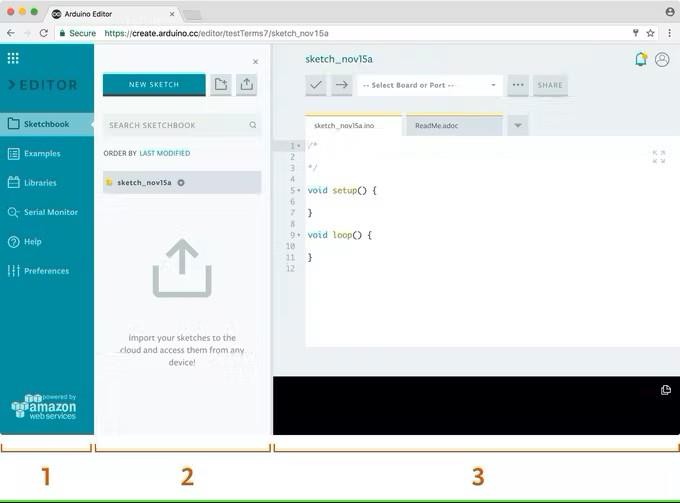
At the end of the plugin installation flow, you will be redirected to the Arduino login page—it’s time to use your credentials! Using the Arduino Web Editor on Chrome OS requires the Arduino Create App to be able to upload sketches to a board. You can install the Create App in the Chrome Web Store.

If you have any questions regarding the use of the Chrome App please write us on this Forum page.

* 1. **ARDUINO WEB EDITOR**

After logging in, you are ready to start using the Arduino Web Editor. The web app is divided into three main columns.

* + - Double check if the Web Editor is displayed the way you like, check the Preferences panel for a few options
    - Connect your Arduino or Genuino board to your computer. Boards and serial ports are auto-discovered and selectable in a single dropdown. Pick the Arduino/Genuino board you want to upload to from the list.
    - Let’s try an example: Choose ‘Examples ’ on the menu on the left, then ‘Basic ’ and ‘Blink ’. The Blink sketch is now displayed in the code area.
    - To upload it to your board, press the ‘Upload ’ button (arrow right) near the dropdown. A ‘BUSY ’ label replaces the buttons during code verifying and uploading. If the upload is successful, the message “Success: Done uploading ” will appear in the bottom output area.



**Figure 9.1: platform write a code workspace**

* + - Once the upload is complete, you should then see on your board the yellow LED with an L next to it start blinking. If this is the case congratulations You have successfully programmed your board to blink its on-board LED!

The Arduino community has written over 700 libraries that you can include in your sketches without having to install a thing. You can browse through all of them in the Library Manager and favorite the ones you like the most. All the libraries available in the Library Manager are automatically detected, so you’ll never need to install them to get your code to compile.

Since all the contributed libraries are up on our servers, if you use them and then you share your sketch with someone, they could just add a copy of your code to their sketchbook and will be able to compile it without having to do anything else.We know that sharing a sketch and its related libraries can be cumbersome, using the Web Editor you can really speed up this process. Try for instance this Thermostat Bot sketch that uses a variety of Contributed libraries.But what if you want to select a specific version of a library Look for it on the Library Manager and star it. It will show up on the the Favorite tab of the Libraries panel. From there you can select a specific version and include it to your sketch. We are adding a metadata

to the sketch about the version number so we can tell the Arduino Builder to pick the one you really want. When building your sketch we will always look at your custom libraries first, then to your favorite ones, and lastly to the Arduino built-in libraries.

If you favorite a library you can also see its related examples, get more information and a link to the GitHub repository, and download it.But what if you want to use your own custom library on the Web IDE Just zip your custom library and click on the 'Import' button on the Library panel.If you want to import multiple custom libraries at once you can do so by creating a single zip file which contains all of them and just import it.

* 1. **EMBEDDED C LANGUAGE**

Embedded C as you can read in above picture is extension to C language. It is a hardware dependent form of C language that depends on the semantics of the hardware for which you are writing your code. It is hardware dependent in nature. The basics concepts of C languages remains the same. What changes is how you will access ports, pins and various peripheral of your board. According to arduino.cc – “Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs – light on a sensor, a finger on a button, or a Twitter message – and turn it into an output – activating a motor, turning on an LED, publishing something online.”

So it is actually a microcontroller board with peripherals and input/output port that was specifically designed for students to create small projects using the set of arduino libraries. This set of arduino library made it a popular microcontroller board and many people started using it for there projects. Even though arduino libraries have enough support but it doesn’t let the program understand what is actually happening behind the picture. Mnay questions remain unsolved because of the use of pre-existing libraries. That’s why decided to start this series of how we will code arduino using Embedded C

* 1. **USER INTERFACE WITH FEEDS**

Now that we’ve got data streaming into feeds on Adafruit IO, and learned of all the cool tricks you can do with your feeds, we can now view and analyze the projects data in real- time. If you want to interface with the feed data, you can always query and fetch data on any computer using the REST or MQTT API. There’s plenty of libraries for a computer programming language to do so.

Ultimately, almost every Internet of Things project will, at some level, need to provide a way for the things to deal with the humans. User interface considerations can range from a mobile app that allows a user to turn lights on and off and adjust the volume on a music player, to output features such as status monitors, analytics, and data visualization dashboards. However, making a custom app with a user interface can be a ton of work. You can create a quick and simple UI using the built in dashboard capability in Adafruit IO. Of course, you should start with making all your feeds and getting them working well.

Once that’s done, you can visit the Dashboard section of the site. Free accounts can have up to 5 Dashboards, and they can have as many elements as you like. One thing to note about elements is that some of them pull from feed data, some push to your feed, and some do both. We’ll explain as we go through them. Note that since all the elements read from the underlying feeds, they can affect each other, which can be great for adding quick visual feedback From your account, click Dashboards and create a new dashboard. A dashboard is just a webpage you can edit. Click Plus to add a block. There are a wide range of great looking, functional blocks.



**Figure 9.3:connected adafruit io**

**Chapter 10**

# Source Code

Source code is the fundamental component of a computer program that is created by a programmer. It can be read and easily understood by a human being. When a programmer types a sequence of C programming language statements into Windows Notepad, for example, and saves the sequence as a text file, the text file is said to contain the source code.

#include <LiquidCrystal.h> #include <SimpleDHT.h> #include <stdio.h>

#include <SoftwareSerial.h> int pinDHT11 = 5; SimpleDHT11 dht11; SoftwareSerial mySerial(2,3);

const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13; LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int IR = 4;

const int MOISTURE = 6; const int PUMP= 7;

const int BUZZER= A2;

#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0

#define Offset 0.00 //deviation compensate #define samplingInterval 20

#define LED 13

#define printInterval 800

#define ArrayLenth 40 //times of collection

int pHArray[ArrayLenth]; //Store the average value of the sensor feedback int pHArrayIndex=0;

int sampletime = 280; int deltatime = 40;

int sleeptime = 9680;

String inputString=""; void setup() {

// put your setup code here, to run once: Serial.begin(9600); mySerial.begin(9600);

lcd.begin(16,2); pinMode(IR , INPUT);

pinMode(BUZZER , OUTPUT);

pinMode(MOISTURE , INPUT); pinMode(PUMP , OUTPUT);

digitalWrite(PUMP, LOW); digitalWrite(BUZZER, LOW);

lcd.clear(); lcd.setCursor(0,0);

lcd.print("IOT BASE AGRI"); lcd.setCursor(0,1); lcd.print("SYSTEM "); delay(3000);

}

void read\_object(void); void read\_soil(void);

void temperaturecheck(void); void read\_ph(void);

void loop() {

// put your main code here, to run repeatedly:

static unsigned long samplingTime = millis(); static unsigned long printTime = millis(); static float pHValue,voltage;

read\_object(); read\_soil(); temperaturecheck();

if(millis()-samplingTime > samplingInterval)

{

pHArray[pHArrayIndex++]=analogRead(SensorPin); if(pHArrayIndex==ArrayLenth)pHArrayIndex=0; voltage = avergearray(pHArray, ArrayLenth)\*5.0/1024; pHValue = 3.5\*voltage+Offset; samplingTime=millis();

}

if(millis() - printTime > printInterval) //Every 800 milliseconds, print a numerical, convert the state of the LED indicator

{

// Serial.print("Voltage:");

// Serial.print(voltage,2); mySerial.print(" pH value: "); mySerial.println(pHValue,2);

lcd.clear(); lcd.setCursor(0, 0); lcd.print("pH:"); lcd.setCursor(3, 0); lcd.print(pHValue);

if(pHValue>=1 && pHValue <=4)

{

lcd.clear(); lcd.setCursor(0, 0);

lcd.print("pH: ACIDIC");delay(2000); mySerial.println("pH:ACIDIC");

}

else if(pHValue>=5 && pHValue <=8)

{

lcd.clear(); lcd.setCursor(0, 0);

lcd.print("pH: NETURAL");delay(2000); mySerial.println("pH:NETURAL");

}

else if(pHValue>=8 && pHValue <=14)

{

lcd.clear(); lcd.setCursor(0, 0); lcd.print("pH: BASIC");

mySerial.println("pH:BASIC"); delay(2000);

}

// digitalWrite(LED,digitalRead(LED)^1); printTime=millis();

}

while (Serial.available())

{

if(Serial.available())

{

inputString=Serial.readString();

}

delay(500); if(inputString.substring(0,1)=="1")

{

inputString="";

mySerial.println("ANIMAL FOUND");

}

if(inputString.substring(0,1)=="2")

{

inputString="";

mySerial.println("ANIMAL FOUND");

}

}

}

double avergearray(int\* arr, int number){ int i;

int max,min; double avg; long amount=0; if(number<=0){

Serial.println("Error number for the array to avraging!/n");

return 0;

}

if(number<5){ //less than 5, calculated directly statistics for(i=0;i<number;i++){

amount+=arr[i];

}

avg = amount/number; return avg;

}else{ if(arr[0]<arr[1]){

min = arr[0];max=arr[1];

}

else{ min=arr[1];max=arr[0];

}

for(i=2;i<number;i++){ if(arr[i]<min){

amount+=min; //arr<min min=arr[i];

}else {

if(arr[i]>max){ amount+=max; //arr>max max=arr[i];

}else{

amount+=arr[i]; //min<=arr<=max

}

}//if

}//for

avg = (double)amount/(number-2);

}//if

return avg; delay(100);

}

void temperaturecheck()

{

int tempcnt = 0; int humycnt = 0;

byte temperature = 0; byte humidity = 0;

int err = SimpleDHTErrSuccess;

if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess)

{

// mySerial.print("Read DHT11 failed, err="); mySerial.println(err);delay(1000);

// mySerial.println("TEMP:"); mySerial.println("29");

// mySerial.println("HUMDITY:");mySerial.println("60"); return;

}

//It will give temperature value mySerial.println("TEMP:"); mySerial.println((int)temperature); lcd.clear();

lcd.setCursor(0,0); lcd.print("TEMP:"); lcd.setCursor(0,1); lcd.print((int)temperature);

//It will give Humidity value delay(3000); mySerial.println("HUMDITY:"); mySerial.println((int)humidity); lcd.clear();

lcd.setCursor(0,0); lcd.print("HUMDITY:"); lcd.setCursor(0,1); lcd.print((int)humidity);

delay(3000); if((int)temperature >35)

{

mySerial.println("HIGH TEMPERATURE "); lcd.clear();

lcd.setCursor(0,0); lcd.print("HIGH "); lcd.setCursor(0,1); lcd.print("TEMPERATURE ");

}

else

{

mySerial.println("NORMAL TEMPERATURE "); lcd.clear();

lcd.setCursor(0,0); lcd.print("NORMAL ");

lcd.setCursor(0,1); lcd.print("TEMPERATURE ")

}

delay(3000);

if((int)humidity >100)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("HIGH "); lcd.setCursor(0,1); lcd.print("HUMIDITY ");

mySerial.println("HIGH HUMIDITY ");

}

else

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("NORMAL ");

lcd.setCursor(0,1); lcd.print("HUMIDITY ");

mySerial.println("NORMAL HUMIDITY ");

}

delay(3000);

}

void read\_object(void)

{

if(digitalRead(IR) == 0)

{

mySerial.println("OBJECT DETECTED"); lcd.clear();

lcd.setCursor(0,0); lcd.print("OBJECT "); lcd.setCursor(0,1); lcd.print("DETECTED"); digitalWrite(BUZZER, HIGH)

}

else

{

mySerial.println("NO OBJECT DETECTED"); lcd.clear();

lcd.setCursor(0,0); lcd.print("NO OBJECT"); lcd.setCursor(0,1); lcd.print("DETECTED");

digitalWrite(BUZZER, LOW);

}

delay(3000);

}

void read\_soil(void)

{

if(digitalRead(MOISTURE) == 1)

{

mySerial.println("LAND IS DRY PUMP ON"); lcd.clear();

lcd.setCursor(0,0);

lcd.print("LAND IS DRY"); lcd.setCursor(0,1); lcd.print("PUMP ON"); digitalWrite(PUMP, HIGH); delay(3000); digitalWrite(PUMP, LOW); delay(1000)

}

else if(digitalRead(MOISTURE) == 0)

{

mySerial.println("LAND IS WET PUMP OFF"); lcd.clear();

lcd.setCursor(0,0); lcd.print("LAND IS WET"); lcd.setCursor(0,1); lcd.print("PUMP OFF");

}

delay(3000);

}

**Chapter11**

# Testing

Internet of Things (IoT) systems continuously collect a large amount of data from heterogeneous "smart objects" through standard died service interfaces. A key challenge is how to use these data and relevant event logs to construct continuously adapted usage profiles and apply them to enhance testing methods, prioritization of tests for the testing of continuous integration of an IoT system. In addition, these usage profiles provide relevance weightings to analyse architecture and behaviour of the system. Based on the analysis, testing methods can predict specific system locations that are susceptible to error, and therefore suggest where expanded runtime monitoring is necessary. Furthermore, IoT aims to connect billions of "smart devices" over the network. Testing even a small IoT system connecting a few dozens of smart devices would require a network of test Virtual Machines (VMs) possibly spreading across the Fog and the Cloud. In this paper we propose a framework for testing of each IoT layer in a separate VM environment, and discuss potential difficulties with optimal VM allocation.

## Iot test smart object

1. **Usability:**
   * We need to make sure the usability of each of the device used here.
   * The medical healthcare tracking device used should be portable enough to be moved into different segments of the medical.
   * The equipment should be smart enough to push not only the notifications but also the error messages, warnings etc.
   * The system should have an option to log all the events to provide clarity to the end users. If it is not capable of doing that, the system should push those as well to a database to store it.
   * The notifications should be shown and handling of the display should be done properly in the devices [computers/mobile devices].
   * Usability in terms of displaying data, processing data, pushing job tasks from the devices should be tested thoroughly.
2. **IoT Security:**
   * IoT Security challenges: IoT is data centric where all the devices/system connected operate based on the data that is available.
   * When it comes to the data flow between devices, there is always a chance that the data can be accessed or read when getting transferred.
   * From a testing standpoint, we need to check if the data is protected/encrypted when getting transferred from one device to the other.
   * Wherever, there is an UI, we need to make sure there is a password protection on it.
3. **Connectivity:**
   * As it is a healthcare solution, connectivity plays a vital role.
   * The system has to be available all the time and should have seamless connectivity with the stakeholders.
   * As per connectivity, two things are very important to test
     + Connectivity, transfer of data, receiving job tasks from the devices should be seamless when the connection is UP and running.
     + The other condition is the connection down scenario. Doesn’t matter how robust is the system and the network, there are chances that the system will go offline. Being a tester, we should test the offline conditions as well. Once the system is not available on the network, there has to be an alert which can prompt the physicians so that they can start to monitor the health conditions manually not depending on the system till it is up. On the other hand, there has to be a mechanism in the system which can store all the data in it during the offline period. Once the system comes online, all that data should get propagated. Data loss should not be there in any condition.
4. **Performance:**
   * When we are talking about a system for a healthcare domain, we need to make sure the system is scalable enough for the whole hospital.
   * When the testing is carried out, it is done for 2-10 patients at a time and the data is propagated to 10-20 devices.
   * When the whole hospital is connected and 180-200 patients are connected to the system, the data that is propagated is much bigger than the tested data.
   * As testers, we need to make sure the system performs the same even though the added data is propagated.
   * We should also test the monitoring utility to display the system usage, power usage, temperature etc.
5. **Compatibility Testing:**
   * Looking at the complex architecture of an IoT system, compatibility testing is a must.
   * Testing items such as, multiple operating system versions, browser types and respective versions, generations of devices, communication modes is necessary for IoT compatibility testing.
6. **Pilot Testing:**
   * As far as the IoT is concerned, Pilot testing is a must.
   * Only testing in Lab makes sure the product/system works fine. But, this may backfire badly when exposed to real-time conditions/steps/scenarios.
   * During pilot testing, the system is exposed to a limited number of users in the real field. They use the application and give feedbacks on the system.
   * These comments come in handy making the application robust enough for the production deployment.
7. **Regulatory Testing:**
   * This being a healthcare system needs to pass through multiple regulatory/compliance checkpoints.
   * Think of a scenario where the product passes through all the testing steps but fails in the final compliance checklist .
   * It is a better practice to get the regulatory requirements in the starting of the development cycle itself. The same should be made part of the testing checklist.
   * By doing that, we make sure the product is certified for the regulatory checklist as well.
8. **Upgrade testing:**
   * IoT is a combination of multiple protocols, devices, operating systems, firmware, hardware, networking layers etc.
   * When an upgrade is performed, be it for the system or for any of the involved items as stated above, thorough regression testing should be carried out/strategy should be adopted so as overcome upgrade related issues.

**The challenges a tester faces in IoT are as follows:**

1. **Hardware-Software Mesh**

IoT is an architecture, which is closely coupled among various hardware and software components. It is not only the software applications that makes the system but also the hardware ones, sensors, communication gateways etc. too play a vital role.

Only functionality testing does not help in completely certifying the system. There is always a dependency on each other in terms of the environment, data transfer etc. So, it becomes a tedious job as compared to testing a generic system [only software/hardware component].

1. **Device Interaction module**

As this is an architecture between different set(s) of hardware and software, it becomes mandatory that they talk to each other in real time/near real time. When they both integrate with each other, things such as security, backward compatibility, upgrade issues becomes a challenge for the testing team.

1. **Real-time data testing**

As we have discussed earlier that a Pilot testing/regulatory testing is mandatory for a system such as this, it also becomes very tough to get such data.

Being in testing team, getting regulatory checkpoints or getting the system deployed in the pilot is very tough. The step becomes even tougher if the system is related to Healthcare as per our example. So, that stays as a big challenge for the testing team.

1. **UI**

The IoT is spread across devices belonging to every platform [iOS, Android, Windows, linux]. Now, testing that out on devices can be done but testing it on all possible devices is almost impossible.

We cannot omit the possibility of the UI being accessed from a device which we don’t possess or simulate. That’s a challenge which is tough to overcome.

1. **Network availability**

Network connection plays a vital role as IoT is all about the data being communicated in faster speeds all the time. IoT architecture has to be tested in all kinds of network connectivity/speeds.

To test this, virtual network simulators are mostly used to vary the network load, connectivity, stability etc. But, real time data/network is always a new scenario and testing team doesn’t know where the bottleneck would develop in the long run.

**Iot Testing Tools**

There are various tools which are used during testing IoT systems

1. **Software:**
   1. **Adafruit io:** This is an Open source application used to monitor the traffic in the interface, source/destination host addresses etc.
   2. **Tcpdump:** This does a similar job that of the Wireshark except, this does not have a GUI. This is a command line based utility which helps the user in displaying the TCP/IP and other packets that are transmitted or received over a network.
2. **Hardware:**
   1. **JTAG Dongle:** This is similar to a debugger in PC applications. This helps in debugging the target platform code and show variable step by step.
   2. **Digital Storage Oscilloscope:** This is used to check various events with time stamps, glitches in power supply, signal integrity check.
   3. **Software Defined Radio:** This is used to emulate receiver and transmitter for a large range of wireless gateways.

To the developing world around us, IoT is a growing market and has a lot of opportunities. The time is not far when IoT becomes essential for the testers to survive in the development world.

The IoT-enabled gadget, smart device application, and communication module play a vital role in studying and evaluating the performance and behavior of various IoT services.

Poor design of IoT-enabled devices and services can hamper the correct functioning of the application and in turn negatively affect the end-user experience.

**Test case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Steps** | **Test Action hardware and**  **software** | **description** | **Result**  **fail** | **Results**  **Pass** |
| Step 1 | Switch on power supply | If its power is their switch on /off automatically  Lcd display | No power | Power on system ready for  successfully |
| Step 2 | We have tested module  wise | Model testing | No module | loaded  successfully |
| Step 3 | IR sensor | Object it their | No object | Light and buzzer on  successfully |
| Step 4 | Dht11 sensor | Temperature and  humidity | No temp  No hum | Shows levels  successfully |
| Step 5 | Soil moisture | Wet or dry | No wet | Pump on successfully |
| Step 6 | Smoke sensor | Fires detected sensing | No fire | Alarm on  successfully |
| Step 7 | Lcd | Display all the function | No | Started |
| Step 8 | Wi- fi | Switch on if the data will send on ui and display serial  data | No data appears | data shown successfully |
| Step 9 | Web page login | Webpage login only one person who have connected  The device | Login incorrect | Login successfully |
| Step 10 | Data visible | Checking device one by one sss | No device active | Data must visible  Successful |
| Step 11 | Switch | Off and on power supply | Shutdown  system | Power on  successfully |

**Table 11.1: Test cases for the project**

**Chapter12**

# Snapshot

* 1. **Project Prototype:**



**Figure.12.1: shows the project prototype with title.**

* 1. **Shows The Fire Alert And Also Turn ON The Motor**



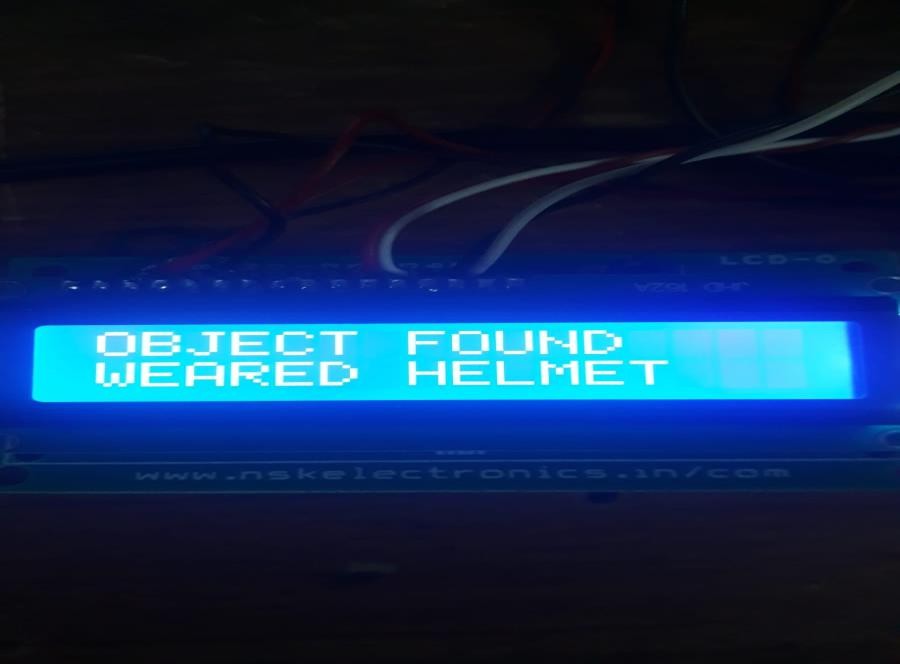
**Figure 12.2: Shows the fire alert and also turn ON the motor**

* 1. **Smoke detected**



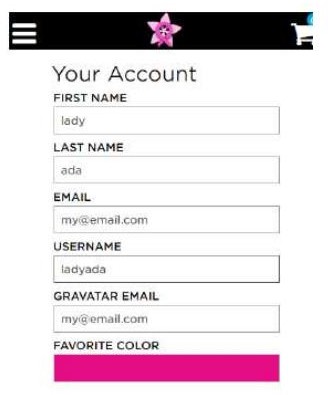
**Figure 12.3: Smoke detected**

* 1. **:Object detected**



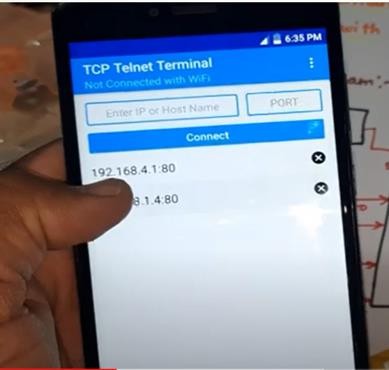
**Figure12.4: Object detected**

* 1. **web page login**



**Figure.12.4: web page login**

* 1. **Connection device**



**Figure.12.6: Connection device**

* 1. **Serial data display in mobile**



**Figure.12.7: Serial data display in mobile**

# Conclusion

This IOT based agriculture monitoring system serves as a reliable and efficient system for monitoring agricultural parameters. The corrective action can be taken and reduce the human power, but it also allows user to see accurate changes in it can be increased. Farmers encounter severe animal threats in rural parts of India. It requires urgent attention as no effective solution has come into use till date for this problem. Hence, this venture conveys an incredible social significance as it intends to resolve this issue. Although the project does not completely do the needful by reducing the problems faced by the farmers, but it ensures farmers to not suffer from the loss of crops by alerting them in prior about the attack of animals.

The problem of crop vandalization by wild animals and fire has become a major social problem in current time. It requires urgent attention as no effective solution exists till date for this problem. Thus this project carries a great social relevance as it aims to address this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achieving better crop yields thus leading to their economic wellbeing.

# Future Enhancement

In the future, there will be very large scope, this project can be made based on Image processing in which wild animal and fire can be detected by cameras and if it comes towards farm then system will be directly activated through wireless networks. Wild animals can also be detected by using wireless networks such as laser wireless sensors and by sensing this laser or sensor’s security system will be activated.

Detection of soil moisture level can be added to the system

* Control of water usage by using temperature level and humidity level
* This project can be extended for cattle monitoring

# Reference

1. Y. Erdem, L. Arin, T. Erdem, S. Polat, M. Deveci, H. Okursoy and H. T. Gültas, Crop water stress index for assessing irrigation scheduling of drip irrigated broccoli (Brassica oleracea L. var. italica), Agriculture. Water Manage., vol. 98, no. 1, Dec. 2010, pp. 148–156.
2. S. L. Davis and M. D. Dukes, Irrigation scheduling performance by evapotranspiration- based controllers, Agriculture. Water Manage., vol. 98, no. 1, Dec. 2010, pp. 19–28.
3. X. Wang, W. Yang, A. Wheaton, N. Cooley and B. Moran, Efficient registration of optical and IR images for automatic plant water stress assessment, Computer. Electron. Agriculture., vol. 74, no. 2, Nov. 2010, pp. 230–237.
4. K. W. Migliaccio, B. Schaffer, J. H. Crane and F. S. Davies, Plant response to evapotranspiration and soil water sensor irrigation scheduling methods for papaya production in south Florida, Agriculture. Water Manage., vol. 97, no. 10, Oct. 2010, pp. 1452–1460.
5. S. A. O’Shaughnessy and S. R. Evett, Canopy temperature based system effectively schedules and controls center pivot irrigation of cotton, Agriculture. Water Manage., vol. 97, no. 9, Apr. 2010, pp. 1310–1316.
6. W. A. Jury and H. J. Vaux, The emerging global water crisis: Managing scarcity and conflict between water users, Adv. Agronomy, vol. 95, Sep. 2007, pp. 1–76.
7. K. S. Nemali and M. W. Van Iersel, An automated system for controlling drought stress and irrigation in potted plants, Sci. Horticult., vol. 110, no. 3, Nov. 2006, pp. 292–297.
8. G. Yuan, Y. Luo, X. Sun and D. Tang, Evaluation of a crop water stress index for detecting water stress in winter wheat in the North China Plain, Agriculture. Water Manage vol. 64, no. 1,Jan. 2004, pp. 29–40. [9] R. G. Allen, L. S. Pereira, D. Raesand M. Smith, -Guidelines for Computing Crop Water Requirements—FAO Irrigation and Drainage , Italy: FAO, 1998, pp.30-55.

[10] S. B. Idso, R. D. Jackson, P. J. Pinter, Jr., R. J. Reginato and J. L. Hatfield, Normalizing the stressdegree-day parameter for environmental variability,” Agriculture. Meteorology vol. 24, Jan. 1981, pp. 45–55.