FABRICATION OF A SMART IRRIGATION SYSTEM

DEPARTMENT OF MECHANICAL ENGINEERING

SONARGAON UNIVERSITY (SU)

Submitted By:

Syed Micdad Bin Aswad
Student ID: BME 1602009209

Md. SidurRahman Khan
Student ID: BME1602009207

Md. RazuAhammed
Student ID: BME1403004017

Md. Alamin
Student ID: BME1602009208

Under the Supervision of

Md.Mainol Hasan

Lecturer

Department of Mechanical Engineering

Sonargaon University

Dhaka,Bangladesh

DECLEARATION

We hereby, declare that the work presented in this project is the outcome of the outcome of the investigation and research work performed by us under the supervision of Md. Mainol Hasan Lecturer, Department of Mechanical Engineering, Sonargaon University (SU).We also declare that part of this project and thereof has been or is being elsewhere for the award of any degree.

Md. Mainol Hasan
Lecturer
Department of Mechanical Engineering,
Sonargaon University
Dhaka, Bangladesh

APPROVAL

This is to certify that the project on **“FABRICATION OF SMART IRRIGATION SYSTEM”** (Syed Micdad Bin Aswad Student - ID: BME1602009209 Md. Sidur Rahman Khan - Student ID: 1602009207, Md. RazuAhammed - Student ID: BME1403004017 and Md. Alamin - Student ID : 1602009208) has been carried out under our supervision. The project has been carried out in partial fulfillment of the requirements for the degree Of Bachelor of Science (B.SC) in Mechanical Engineering of year of 2020 and-has been approved as to -its style and contents.

Md. Mainol Hasan
Lecturer
Department of Mechanical Engineering,
Sonargaon University
Dhaka, Bangladesh

ACKNOWLEDGEMENT

The author wants to express gratefulness, regard, gratitude and sincere thanks to supervisor Md. Mainol Hasan Department of Mechanical Engineering, Sonargaon University (SU), for his guidance and valuable counsel in execution and completion of the study without which it would be impossible to carry out the work. Also thanks to Md.MostofaHossain .Associate Professor and Head of the Department of Mechanical Engineering, (SU) for giving permission to use laboratory facilities of the department. Author grateful to the teachers who helped us directly or indirectly which was very necessary to complete the thesis work.

Finally, we are also grateful to the vice chancellor of Sonargaon University (SU) for his overall support to finish the project works.

ABASTRACT

This work is based on a smart system that will be used to irrigate using solar system. The device will automatically irrigate water by giving it water. Our automated system will prevent you from providing water when there is enough water in the land. Automation of farm activities can transform agricultural domain from being manual and static to intelligent and dynamic leading to higher production with lesser human supervision. This project proposes an automated irrigation system which monitors and maintains the desired soil moisture content via automatic watering. Microcontroller ATMEGA328P on arduino Nano platform is used to implement the control unit. The setup uses soil moisture sensors which measure the exact moisture level in soil. This value enables the system to use appropriate quantity of water which avoids over/under irrigation. IOT is used to keep the farmers updated about the status of Blynks Information from the sensors is regularly updated on a mobile SMS using GSM-GPRS SIM800L modem through which a farmer can check whether the water Blynks apps are ON/OFF at any given time.

Table

CHAPTER – 1

CHAPTER - 2

CHAPTER - 3

METHODOLOGY:

CHAPTER – 4

CHAPTER – 5

List of Table

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

Agriculture is the major source of income for the largest population in Bangladesh and is major contributor to Bangladesh economy. Agriculture plays vital role in the development of agricultural country [1]. However, technological involvement and its usability have to be grown still and cultivated for ago sector in Bangladesh. The agriculture must overcome expanding water deficiencies, restricted availability of lands, while meeting the expanding consumption needs of a world population. New innovative The Internet of Things (IoT)applications are addressing these issues and increasing the quality, quantity, sustainability and cost effectiveness of agricultural production. IoT plays an important role in agriculture industry which can feed 9.6 billion people on the Earth by 2050 [2] Agriculture is the backbone of Bangladeshi Economy. In today's world, as we see rapid growth in global population, agriculture becomes more important to meet the needs of the human race. However, agriculture requires irrigation and with every year we have more water consumption than rainfall, it becomes critical for growers to find ways to conserve water while still achieving the highest yield. But in the present era, the farmers have been using irrigation technique through the manual control in which they irrigate the land at the regular interval. According to statistics, agriculture uses 85% of available freshwater resources worldwide, and this percentage will continue to be dominant in water consumption because of population growth and increased food demand. There is an urgent need to create strategies based on science and technology for sustainable use of water, including technical, agronomic, managerial and institutional improvements. Agricultural irrigation based on Internet technology is based on crop water requirement rules. By using Internet technology and sensor network technology we can control water wastage and to maximize the scientific technologies in irrigation methods. Hence it can greatly improve the utilization of water and can increase water productivity. The Internet of Things (IoT) is a technology where in a mobile device can be used to monitor the function of a device. The Internet of Things (IoT) is concerned with interconnecting communicating objects that are installed at different locations that are possibly distant from each other. Internet of Things (IoT) is a type of network technology, which senses the information from different sensors and makes anything to join. The data processed by the central module is converted to meaningful data and relayed to the user. The user can view the data with the help of a handheld device such as a mobile phone or a tablet. Nowadays water scarcity is a big concern for farming. This project helps the farmers to irrigate the farmland in an efficient manner with automated irrigation system based on soil moisture. The proposed system has been designed to overcome the unnecessary water flow into the agricultural lands. Moisture readings continuously monitored by using sensor and send these values to the assigned IP address. Android application continuously collects the data from that assigned IP address. Once the soil moisture values are exceeded the particular limit then the relay, which is connected to the Arduino microcontroller, controls the Automation of irrigation system using IoT motor. The android application is a simple menu driven application, with 2 options. This includes motor status, moisture values. The motor status indicates the current status of the pump. In this paper, soil moisture sensor sensors placed in root zone of plant and transmit data to android application. Threshold value of soil moisture sensor that was programmed into a microcontroller to control water quantity and soil moisture values are displayed on the android application. This paper on "Smart Irrigation System on Sensing Soil Moisture Content" is intended to create an automated irrigation mechanism which turns the pumping motor ON and OFF on detecting the dampness content of the earth. In this paper only soil moisture value is considered but proposed project provided extension to this existed project by adding temperature and humidity values. Remote Monitoring in Agricultural Greenhouse Using Wireless Sensor and Short Message Service (SMS) In this paper they are sending data via SMS but proposed system sends the values to mobile application. This proposed paper is Arduino based remote irrigation system developed for the agricultural plantation, which is placed at the remote location and required water provides for plantation when the humidity of the soil goes below the set-point value. But in this we did not aware about the soil moisture level so to overcome this drawback proposed system included with extra feature soil moisture value and temperature value which displayed on the farmer mobile application [6]. “Irrigation Control System Using Android and GSM for Efficient Use of Water and Power” this system made use of GSM to control the system which may cost more so to overcome that proposed system used Arduino yun board which already consist of in build Wi-Fi module “Microcontroller based Controlled Irrigation System for Plantation” In this paper old generation with lesser memory microcontroller is used to control the system but proposed system made use of Arduino yun board which is user friendly and it helps to dump the programs easily.

1.2 Scope of the Project.

Includes (list of Deliverables)

- Design and implement of automatic solar irrigation system
- Control field pump on a condition of Soil moisture sensor
- Control reserve pump on a condition of ultrasonic sensor
- Design Implementation of IOT monitoring system
- Design & implementation of SOLAR DUST Sensing & automatic Cleaning System
- Design & implementation of GSM notification of pump on/off status.

Dose Not Include

- There is no Fertilizer sensor
- There is no Cloud storage server to save all data
- duration of time for which the water pump is ON
- There is no Graphical representation the graphical representation of the moisture content levels in the soil.
- In such cases farmer may need to stop the system remotely it's not included in the system.
- To capture photo of the field and save the data base which is not include the system
- A device with a GPS can be integrated to provide specific location of the farmer and more accurate weather reports of agriculture of the field, which is not include the system.

Project Completion Criteria

- Working Smart Irrigation system communicate with Android Application and
- GSM based mobile SMS of pump status.
- Material extraction in controlled manner with appropriate setting.
- Cartesian capable of DC Motor for each tank.
- Solar panel with automatic solar panel cleaning system.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review:

Agriculture is the unquestionably the largest livelihood provider in Bangladesh. Concern about the environmental impacts of technological change in agriculture, and studies exploring farmers' perceptions of this issue are nascent. [4] With rising population, there is a need for increased agricultural production. In order to support greater production in farms, the requirement of the amount of fresh water used in irrigation also rises. Currently, agriculture accounts 83% of the total water consumption in Bangladesh. Unplanned use of water inadvertently results in wastage of water. This suggests that there is an urgent need to develop systems that prevent water wastage without imposing pressure on farmers. Over the past 15 years, farmers started using computer and software systems to organize their financial data and keep track of their transactions with third parties and also monitor their crops more effectively. In the Internet era, where information plays a key role in people's lives, agriculture is rapidly becoming a very data intensive industry where farmers need to collect and evaluate a huge amount of information from a diverse number of devices (e.g. sensors, faming machinery etc.) in order to become more efficient in production and communicating appropriate information . With the advent of open source Arduino boards along with cheap moisture sensors, it is viable to create devices that can monitor the soil moisture content and accordingly irrigating the fields or the landscape as an when needed. The proposed system makes use of microcontroller ATMEGA328P on Arduino UNO platform and IOT which enable farmers to remotely monitor the status of sprinklers installed on the farm by knowing the sensor values thereby, making the farmer’s work much easier as they can concentrate on other farm activities. We will irrigate the solar system in this project through automatic when our land does not contain water; we will automatically irrigate water by giving it water. Our automated system will prevent you from providing water when there is enough water in the land. Automation of farm activities can transform agricultural domain from being manual and static to intelligent and dynamic leading to higher production with lesser human supervision. This project proposes an automated irrigation system which monitors and maintains the desired soil moisture content via automatic watering. Microcontroller ATMEGA328P on Arduino Nano platform is used to implement the control unit. Internet of Things (IoT) plays a crucial role in smart agriculture. Smart farming is an emerging concept, because IoT sensors capable of providing information about their agriculture fields.[3] The setup uses soil moisture sensors which measure the exact moisture level in soil. This value enables the system to use appropriate quantity of water which avoids over/under irrigation. IOT is used to keep the farmers updated about the status of Blynks Information from the sensors is regularly updated on a mobile SMS using GSM-GPRS SIM800L modem through which a farmer can check whether the water Blynks apps are ON/OFF at any given time. The Smart irrigation System has wide scope to automate the complete irrigation system. Here we are building a IoT based Irrigation System using ESP8266 Node MCU Module and DHT11 Sensor. It will not only automatically irrigate the water based on the moisture level in the soil but also send the Data to Blynk apps to keep track of the land condition. The System will consist a water pump which will be used to sprinkle water on the land depending upon the land environmental condition such as Moisture, Temperature and Humidity. We previously build similar Automatic Plant Irrigation System which sends alerts on mobile but not on IoT cloud. Apart from this, soil moisture detector circuit can also be helpful in building Smart Irrigation system. Before starting, it is important to note that the different crops require different Soil Moisture, Temperature and Humidity Condition. So, in this tutorial we are using such a crop which will require soil moisture of about 50-55%. So, when the soil loses its moisture to less than 50% then Motor pump will turn on automatically to sprinkle the water

and it will continue to sprinkle the water until the moisture goes up to 55% and after that the pump will be turned off. The sensor data will be sent to Bylinks apps Server in defined interval of time so that it can be monitored from anywhere in the world

CHAPTER 3
METHODOLOGY

3.1 Arduino Nano

Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling
Lights, motors, and other actuators.

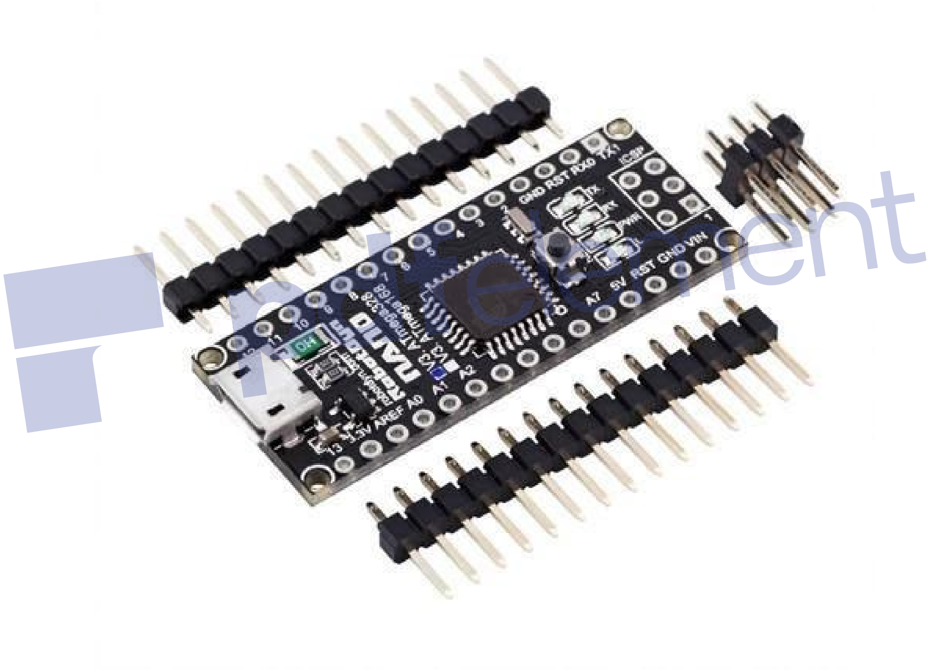


Figure 1: Arduino Nano

Microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, Max MSP).

Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Decimal Demilune has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power.

Nano’s got the breadboard-ability of the Boarding and the Mini+USB with smaller footprint than either, so users have more breadboard space. It’s got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It is two layers. That make it easier to hack and more affordable.

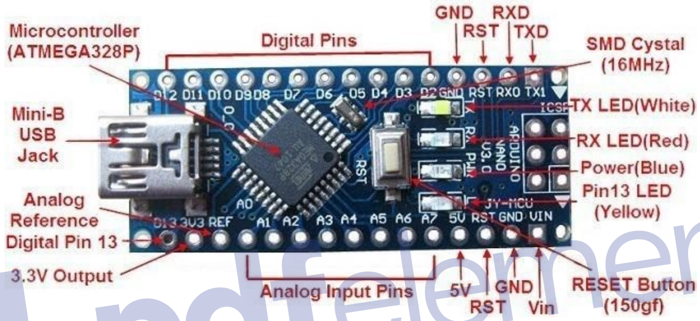


Figure 2: Section of Arduino Nano.

Operating Voltage (logic level):5 V

Input Voltage (recommended):7-12 V

Input Voltage (limits):6-20 V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 8

DC Current per I/O Pin: 40 mA

Flash Memory: 32 KB (of which 2KB used by boot loader)

SRAM: 2 KB

EEPROM: 1 KB

Clock Speed: 16 MHz

Dimensions: 0.70” x 1.70”

Features:

- Automatic reset during program download
- Power OK blue LED
- Green (TX), red (RX) and orange (L) LED
- Auto sensing/switching power input
- Small mini-B USB for programming and serial monitor
- ICSP header for direct program download
- Standard 0.1 spacing DIP (breadboard friendly)
- Manual reset switch

Microcontroller IC ATmega328p:

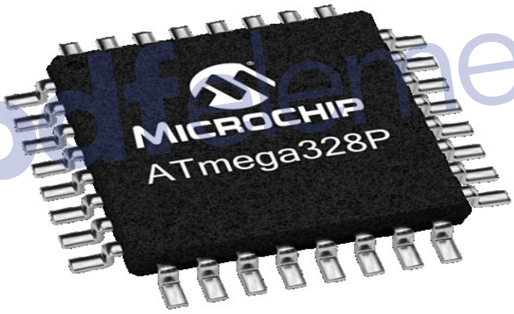


Figure 3: Microcontroller IC AT mega 328p.

The high-performance Microchip Pico Power 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed

3.2 Node MCU

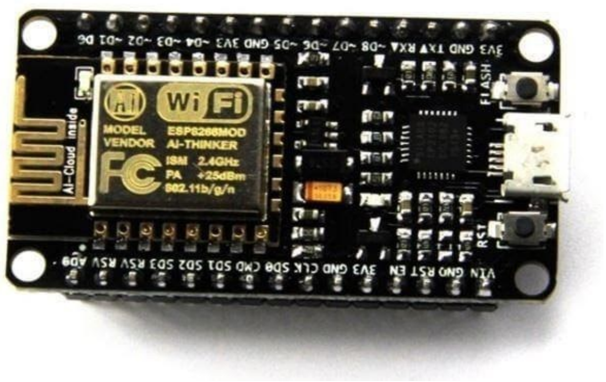


Figure 4: Node MCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espresso Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espresso Non-OS SDK for ESP8266.

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

Features

Wi-Fi Module – ESP-12E module similar to ESP-12 module but with 6 extra GPIOs.

USB – micro USB port for power, programming and debugging

Headers – 2x 2.54mm 15-pin header with access to GPIOs, SPI, UART, ADC, and power pins

Misc. – Reset and Flash buttons

Power – 5V via micro USB port

Dimensions – 49 x 24.5 x 13mm

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of node cu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerbera file of an ESP8266 board, named devkit v0.9.[12] Later that month, Tuan PM ported MQTT client library from Contac to the ESP8266 SoC platform,[and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the NodeMCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.

ESP8266 Arduino Core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFiSoC, popularly called the "ESP8266 Core for the Arduino IDE". This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.

3.3 GSM Module

At the heart of the module is a SIM800L GSM cellular chip from SimCom. The operating voltage of the chip is from **3.4V to 4.4V**, which makes it an ideal candidate for direct LiPo battery supply. This makes it a good choice for embedding into projects without a lot of space.

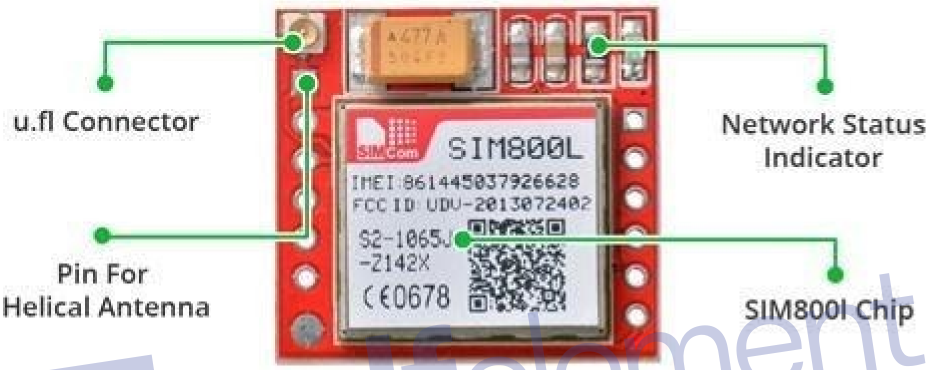
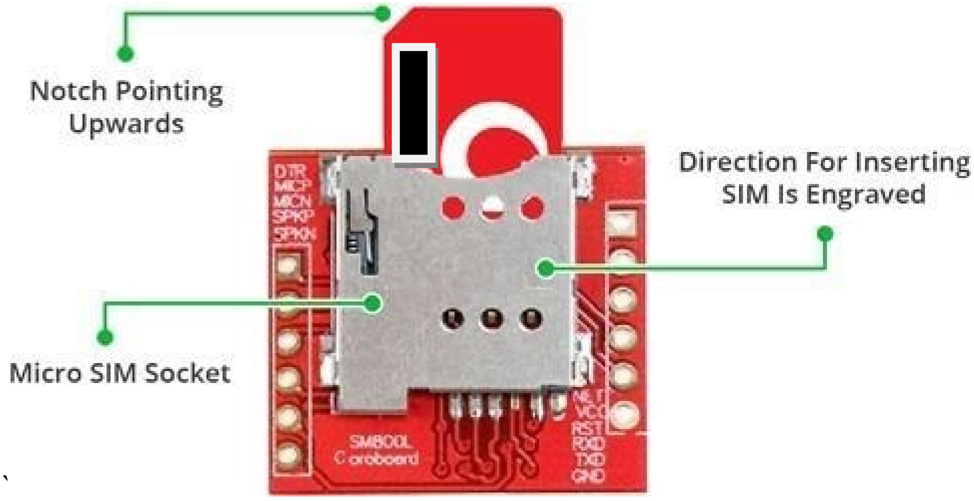


Figure5.GSM Module

All the necessary data pins of SIM800L GSM chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a microcontroller over **UART**. The module supports baud rate from **1200bps** to **115200bps** with Auto-Baud detection.

The module needs an external antenna to connect to a network. The module usually comes with a **Helical Antenna** and solders directly to NET pin on PCB. The board also has a U.FL connector facility in case you want to keep the antenna away from the board.



`

Figure 6:GSM Module

There’s a SIM socket on the back! Any activated, 2G micro SIM card would work perfectly. Correct direction for inserting SIM card is normally engraved on the surface of the SIM socket.

This module measures only 1-inch² but packs a surprising number of features into its little frame. Some of them are listed below:

Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900

- ➢ Connect onto any global GSM network with any 2G SIM
- ➢ Make and receive voice calls using an external 8Ω speaker & electrets microphone
- ➢ Send and receive SMS messages
- ➢ Send and receive GPRS data (TCP/IP, HTTP, etc.)
- ➢ Scan and receive FM radio broadcasts

Transmit Power:

- ➢ Class 4 (2W) for GSM850
- ➢ Class 1 (1W) for DCS1800

Serial-based AT Command Set

FL connectors for cell antennae

Accepts Micro SIM Card

SIM800L GSM Module Pinout

The SIM800L module has total 12 pins that interface it to the outside world. The connections are as follows:

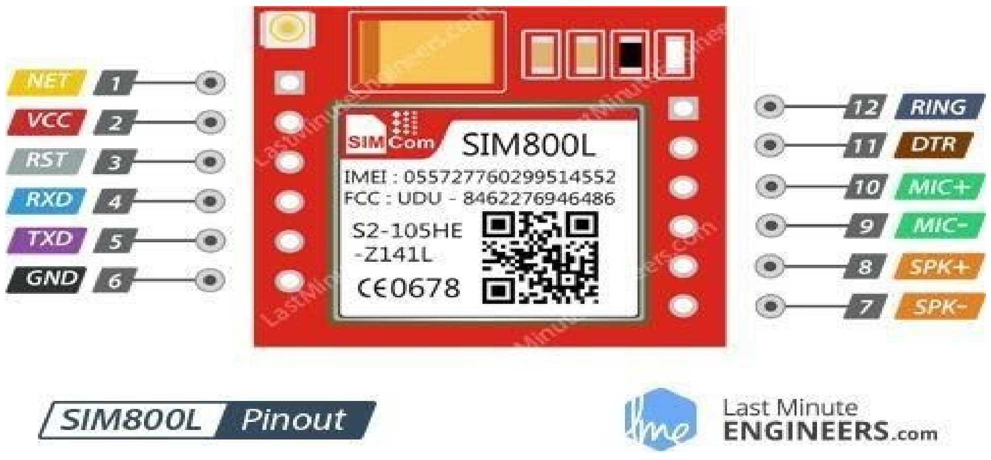


Figure 7:GSM Module Pin out

NET is a pin where you can solder Helical Antenna provided along with the module.

VCC supplies power for the module. This can be anywhere from 3.4V to 4.4 volts. Remember connecting it to 5V pin will likely destroy your module! It doesn't even run on 3.3 V! An external power source like Li-Po battery or DC-DC buck converters rated 3.7V 2A would work.

RST (Reset) is a hard-reset pin. If you absolutely got the module in a bad space, pull this pin low for 100ms to perform a hard reset.

RxD (Receiver) pin is used for serial communication.

TxD (Transmitter) pin is used for serial communication.

GND is the Ground Pin and needs to be connected to GND pin on the Arduino

RING pin acts as a Ring Indicator. It is basically the 'interrupt' out pin from the module. It is by default high and will pulse low for 120ms when a call is received. It can also be configured to pulse when an SMS is received.

DTR pin activates/deactivates sleep mode. Pulling it HIGH will put module in sleep mode, disabling serial communication. Pulling it LOW will wake the module up.

MIC± is a differential microphone input. The two microphone pins can be connected directly to these pins.

SPK± is a differential speaker interface. The two pins of a speaker can be tied directly to these two pins.

3.4 LCD DISPLAY:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs.

The reasons being LCDs are economical; easily programmable, have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.

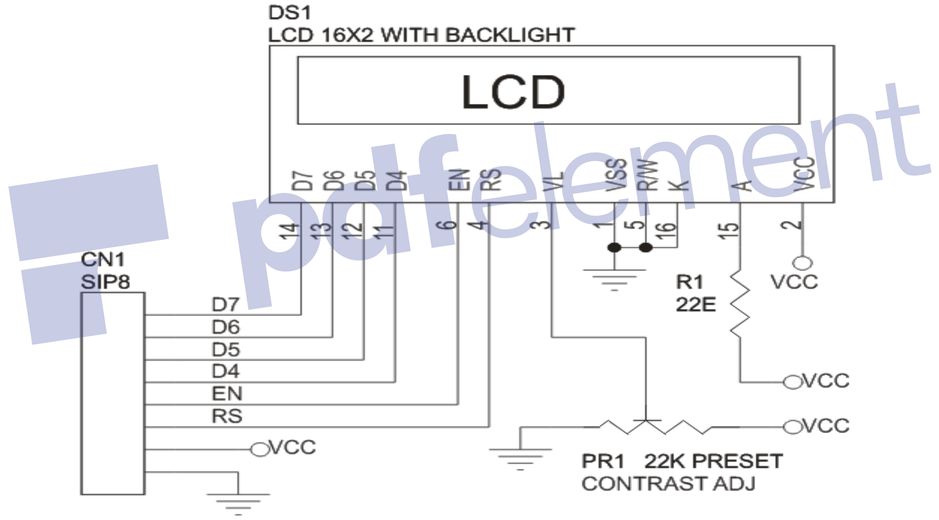


Figure 8: 16*2 LCD Display

Features of LCD Display:

5 x 8 dots with cursor

Built-in controller (KS 0066 or Equivalent) + 5V power supply (Also available for + 3V)

1/16 duty cycle

B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED) N.V. optional for + 3V power supply.

3.5 Moisture sensor:

Soil moisture sensors measure the volumetric water content in soil. Since thedirect gravimetric measurement of free soil moisture requires removing, drying, and weighting of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity.

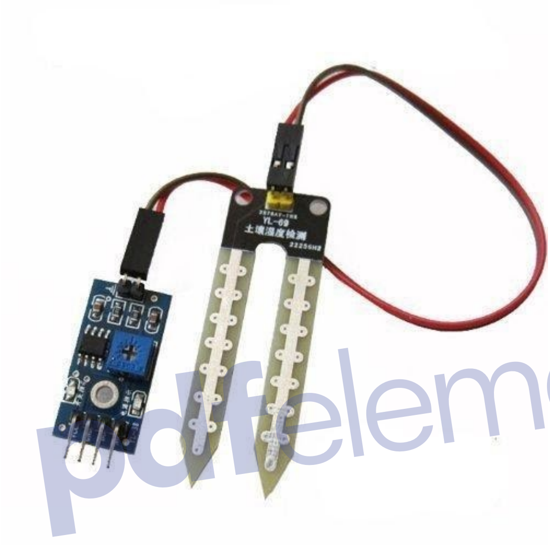


Figure 9: Moisture sensor

Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

3.6 Solar Panel:

These panels are designed with solar cells composed of semiconductor materials. [14] The main function of Solar panels is to convert solar energy into DC electrical energy generally of 12V, which is further used for the rest of the circuit. The number of cells required and their size depends on the rating of the load. The collection of solar cells can produce maximum electricity. But, the solar panel must place exactly at right angles to the sun rays.

Figure 10: Solar Panel

Certain materials can be made to produce electricity when light falls on them; this is called the photovoltaic effect. Solar panels use this effect to convert energy from sunlight into direct current (DC) electrical energy. An inverter unit then changes this into alternating current (AC) for your home's electrical circuits. Any excess energy can be fed back to the electricity grid, for which you may be paid an agreed feed-in tariff, or it could be fed into a battery storage system so you can use the stored power later (at night, for instance).

Solar panels work best when they're north facing, pointed directly at the sun, at an optimal angle and not blocked by trees or shading. The effectiveness of solar panels also depends on where you live and the weather.

Solar power is the conversion of energy from sunlight into electricity, either directly using photovoltaic, indirectly using concentrated solar power, or a combination. Concentrated solar power systems use lenses or mirrors and tracking systems to focus a large area of sunlight into a small beam.

Figure 11: Solar Panel

- Maximum Power (+/-5%): 10W
- Voltage at Pmax (Vmp): 17V
- Current at Pmax (lmp): .58A
- Open Circuit Voltage (Voc): 21.6V
- Wind Survivability: 201kph (125mph)
- Hailstone Survivability: 1" @ 50mph
- Monthly Energy Production: 1.2kWh
- Size: 300x355x18mm (11.8x14x0.7")
- Weight: 1.5kg (3.3lb)

3.7 Battery:

The Battery is an electric device, that is used to store current which is produced from the solar panel and supplied to the corresponding loads. The number of batteries required depends on the load requirement.

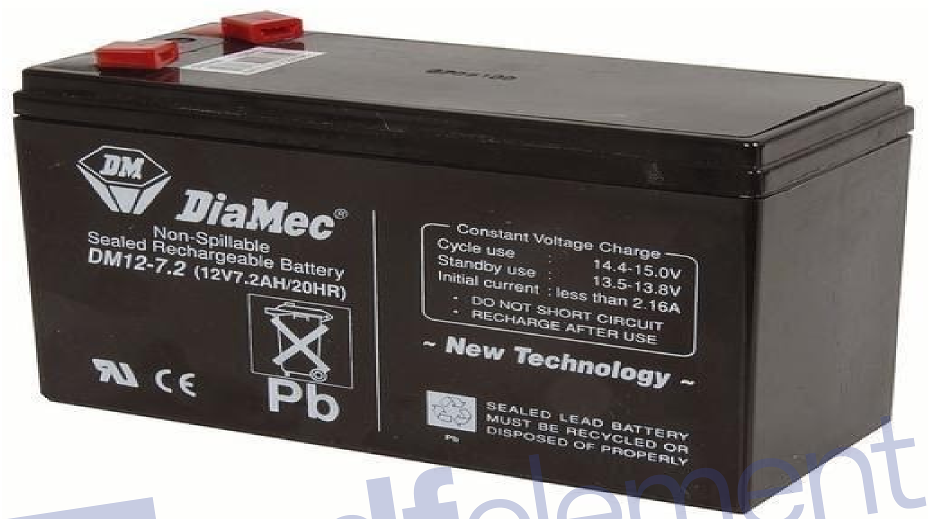


Figure 12: Battery

These are small Sealed Lead Acid batteries that are generally used for semi-portable appliances that need more power or a fair bit of power (more than can be provided by D, C or AA cells) for a longish time. They are typically used for people who go caving, or for lighting for an overnight camping trip where you are travelling on foot. They are not ridiculously heavy or bulky. They are good to drive a lantern on a tinny for that night fishing expedition. Being sealed, you cannot fear that the battery acid (which is actually a GEL) will spill. The amp-hour rating tells you how long the battery will last before it goes flat. For example, a 7.2Ah battery will give 7.2 amps for one hour or 3.6 amps for 2 hours or 1.8 amps for 4 hours, etc. Generally an adequate amount of current for a lantern. They are ideally charged by solar panels.

3.8 Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

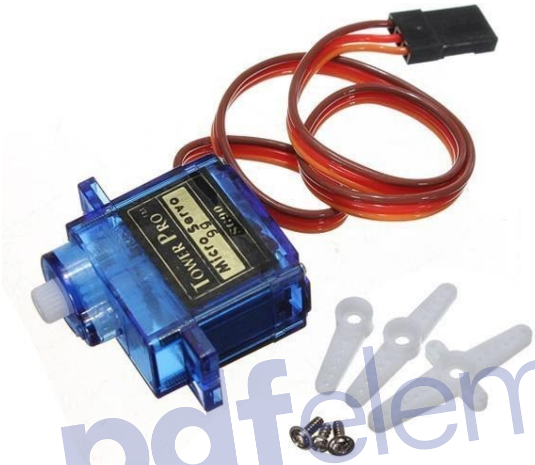


Figure 13: Servo Motor.

A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped). This type of servomotor is not widely used in industrial motion control, but it forms the basis of the simple and cheap servos used for radio-controlled models.

3.9. Dust Sensor:

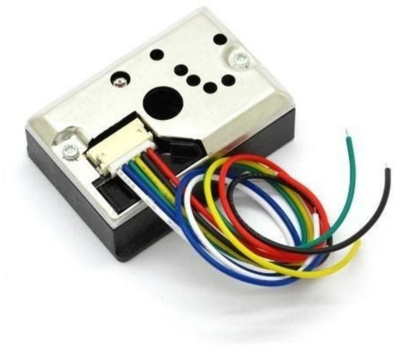


Figure 14: Dust Sensor.

The Sharp Optical Dust Sensor (GP2Y1010AU0F) is especially effective in detecting very fine particles like cigarette smoke, and is commonly used in air purifier systems. An infrared emitting diode and a phototransistor are diagonally arranged into this device, to allow it to detect the reflected light of dust in air. The sensor has a very low current consumption (20mA max, 11mA typical), and can be powered with up to 7VDC. The output of the sensor is an analog voltage proportional to the measured dust density, with a sensitivity of 0.5V/0.1mg/m3.

For your convenience, ITEAD ship this sensor with a 6-pin, 1.5mm pitch mating connector and cable.

Specification

- Supply voltage: 5-7V
- Operating temperature: -10-65 Celsius degree
- Consumption current:20mA max

3.10 DC Pump:

Motor is an electrical machine which converts electrical energy into mechanical energy and pump is mechanical device which converts mechanical energy into work to be done. A pump requires a driving mechanism such as a motor or an engine to operate.

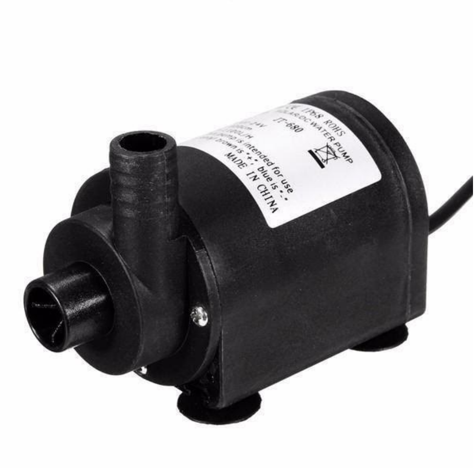


Figure 15: DC 12 Volt Pump Motor.

Specification:

- Material: ABS(Acrylonitrile Butadiene Styrene) + Stainless Steel
- Overall Size: Approx. 80 x 48 x 63mm/3.15 x 1.89 x 2.48"
- Pump Inlet Diameter: 16mm(Outer), 12mm(Inner)
- Pump Outlet Diameter: 12mm(Outer), 6.9mm(Inner)
- Inlet/Outlet: 1/2" male thread
- Voltage: 6-12V DC
- Maximum Rated Current: 1.2A
- Power:16.8W
- Max Flow Rate: 700 L/H
- Max Water Head: 5M
- Max Circulating Water Temperature: 60°C

3.11. Transformer

A transformer is an electrical device used to change the value of an alternating voltage. Transformers are widely used in electrical work. They are encountered daily, in industrial, commercial and domestic situations. They vary in size from miniature units used in electronics to huge units used in power stations. The efficient transmission and distribution of electricity throughout the country would be impossible without the use of power transformers.

Transformers are also used for safety reasons on construction sites when using power tools and in domestic bathroom situations in shaver units. They are used in doorbell operation and also to power electronic equipment, battery chargers, televisions, computers, alarm systems, etc. Transformers vary considerably in construction, size and shape depending on their application.

All transformers rely on the principle of mutual inductance for their operation. Mutual inductance was discussed in detail in Unit 2.1.6 Magnetism, Electromagnetism and Electromagnetic Induction.

Transformer Symbols:

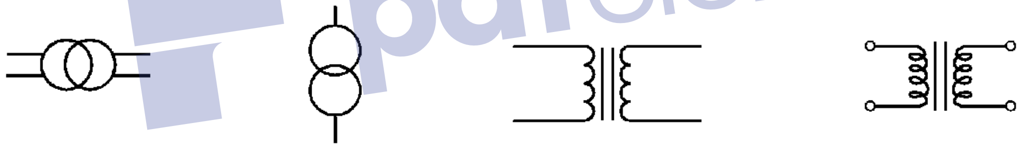


Figure 16: Transformer Symbols

Transformer Construction:

A transformer consists of two coils of wire called windings, which are wound onto a common iron core. The wire used in the two windings, primary and secondary, is coated with an insulating varnish. Both coils are wound onto, but insulated from the iron core. See Figure2.2.8.2:

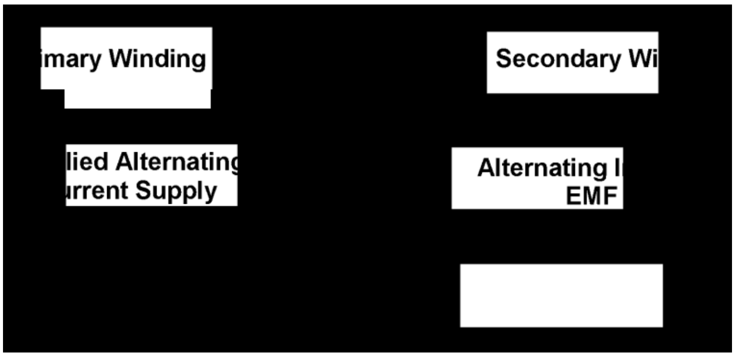


Figure 17: Transformer Construction.

The Transformer Principle:

When a conductor or coil is moved in a stationary magnetic field it cuts the lines of magnetic flux and an EMF is induced in the conductor or coil. This same principle also applies when a conductor is held stationary and the magnetic flux is made to change or vary [13].

Now consider an alternating current applied to a stationary coil. A magnetic field will build up and collapse in the coil, continually rising and falling in harmony with the applied AC current as shown in Figure 2.2.8.3.

Figure 18: Transformer Primary Coil.

If a second coil (coil 2) is placed close to the first coil (coil 1) the alternating magnetic flux in coil 1 links with coil 2. See Figure 4. This results in an EMF being induced in coil 2. This is a process known as mutual induction.

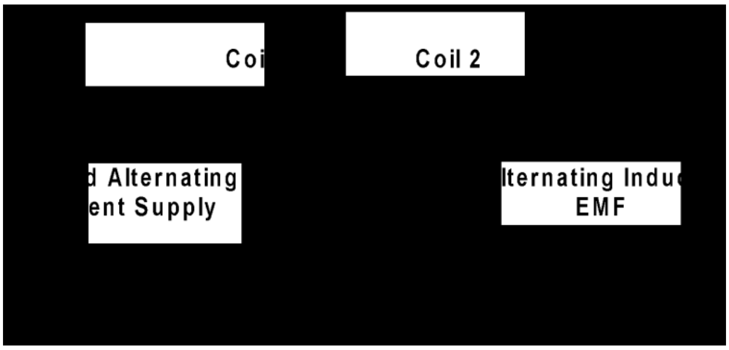


Figure 19: Transformer Primary & secondary Coi

If coil 1 and coil 2 are mounted on an iron core the magnetic flux around both coils will be concentrated. This arrangement of coils and an iron core form the complete device known as a **transformer**. See Figure 20:

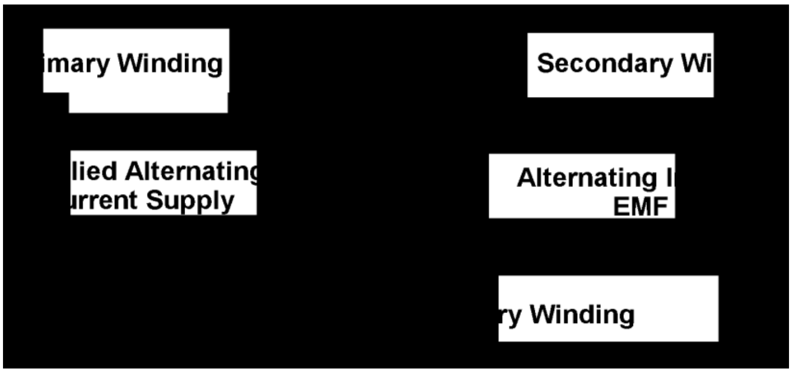


Figure 20: Lines Of Flux set up by primary Winding.[13]

The input coil of a transformer is fed from the AC supply and is called the primary winding. The output coil, to which the load is connected, is called the secondary winding.

It is important to remember that there is no electrical connection between the primary winding and the secondary winding of a transformer. The only common link between the two windings is the magnetic field.

The Full Wave Rectifier:

The first building block in the dc power supply is the full wave rectifier. The purpose of the full wave rectifier (FWR) is to create a rectified ac output from a sinusoidal ac input signal. It does this by using the nonlinear conductivity characteristics of diodes to direct the path of the current.

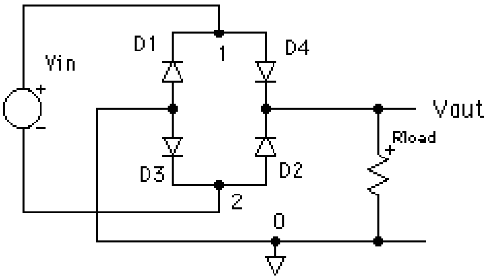


Figure 21: Common four-diode bridge configuration for the FWR[13].

Filtered Full Wave Rectifier

The filtered full wave rectifier is created from the FWR by adding a capacitor across the output.

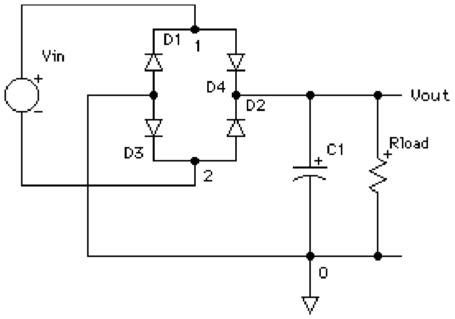


Figure 22: Filtered full wave rectifier

The result of the addition of a capacitor is a smoothing of the FWR output. The output is now a pulsating dc, with a peak to peak variation called ripple. The magnitude of the ripple depends on the input voltage magnitude and frequency, the filter capacitance, and the load resistance.

To describe the source of the voltage ripple, consider the performance of the filtered full wave rectifier above. The input to the rectifier is a sinewave of frequency f. Let Vi be the full wave rectified signal input to the filter stage of the rectifier and Vo be the output. Vi can be approximated as the absolute value of the rectifier input, with frequency 2f.

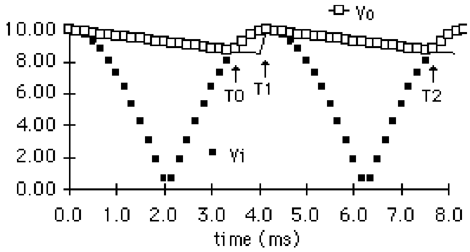


Figure 23: Output (Vi) and input (Vo) of a filtered full wave rectifier

In the time period from T0 to T1, the diode D1 (or D3, depending on the phase of the signal) is forward biased since Vi > VC1 (approximate the forward biased diode as a short circuit). The capacitor C1 charges and the voltage across the load R increases. From T1 to T2, the diodes D1 and D2 are reverse biased (open circuit) because Vcap Vi, and the capacitor discharges through the load R with a time constant of RC seconds.

3.12. Ultrasonic sensor

Ultrasonic sensor

An ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back.

Uses:

Level measuring is used in a wide variety of applications. The substance does not matter - ultrasonic sensors detect almost all materials. Liquids like milk, chemicals, or lacquer, as well as mud or bulk goods are safely controlled.

Working of ultrasonic sensor:

The Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone).

Distance is measured by the formula

Distance = Sound speed x time taken / 2

Different types of ultrasonic sensor with working:

There are four major different types of ultrasonic sensors are currently in used,

- Ultrasonic Proximity Sensors
- Ultrasonic 2 Point Proximity Switches
- Ultrasonic Retro-reflective Sensors
- Ultrasonic Through Beam Sensors

Advantages of ultrasonic sensor:

- The ultrasonic sensor has high frequency, high sensitivity and high penetrating power therefore it can easily detect the external or deep objects.
- These sensors easily interface with microcontroller or any type of controller
- These sensors have greater accuracy then other methods for measuring the thickness and depth of parallel surface.
- These sensors could easily sense the nature, shape and orientation of that specific objects which is within the area of these sensors.
- There sensors are easy to use, not dangerous during operation for nearby objects, person, equipment or material.

Disadvantages of ultrasonic sensor:

- During use of ultrasonic tester for testing it is very important to know the operational manual, in other words it required careful attention for experienced technician.
- During the development inspection procedure of equipment using ultrasonic tester extensive knowledge is required is testing technicians.
- When these sensors are used for inspection purpose then these should be water resistive otherwise, they could be damaged.
- When these sensors are interfaced with microcontroller or any controller then experienced person or programmer is required.

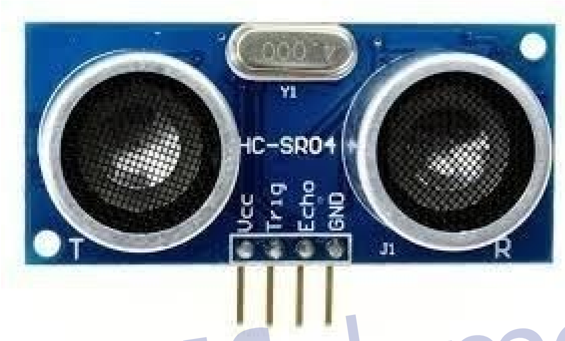


Figure 24: Ultrasonic sensor

3.13. Capacitor

When there is a potential difference across the conductors (e.g., when a capacitor is attached across a battery), an electric field develops across the dielectric, causing positive charge (+Q) to collect on one plate and negative charge (-Q) to collect on the other plate. If a battery has been attached to a capacitor for a sufficient amount of time, no current can flow through the capacitor. However, if an accelerating or alternating voltage is applied across the leads of the capacitor, a displacement current can flow.

Figure 25: Capacitor

An ideal capacitor is characterized by a single constant value for its capacitance. Capacitance is expressed as the ratio of the electric charge (Q) on each conductor to the potential difference (V). The SI unit of capacitance is the farad (F), which is equal to one coulomb per volt (1 C/V). Typical capacitance values range from about 1 pF (10−12 F) to about 1 mF (10−3 F). The capacitance is greater when there is a narrower separation between conductors and when the conductors have a larger surface area. In practice, the dielectric between the plates passes a small amount of leakage current and also has an electric field strength limit, known as the breakdown voltage. The conductors and leads introduce an undesired inductance and resistance. Capacitors are widely used in electronic circuits for blocking direct current while allowing alternating current to pass. In analog filter networks, they smooth the output of power supplies. In resonant circuits they tune radios to particular frequencies. In electric power transmission systems, they stabilize voltage and power flow.

3.14. Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors act to reduce current flow, and, at the sometime, act to lower voltage levels within circuits. Resistors may have fixed resistances or variable resistances, such as those founding thermostats, visitors, trimmers, photo resistors, hamsters and potentiometers. The current through a resistor is in direct proportion to the voltage across the resistor's terminals. This relationship is represented by Ohm's law

Fig. 26: Resistor

Theory of operation

The behavior of an ideal resistor is dictated by the relationship specified by Ohm ‘slaw:

$$V = I.R$$

Ohm's law states that the voltage (V) across a resistor is proportional to the current(I), where the constant of proportionality is the resistance (R).

Equivalently, Ohm's law can be stated:

$$I = V/R$$

This formulation states that the current (I) is proportional to the voltage (V) and inversely proportional to the resistance (R). This is directly used in practical computations. For example, if a 300-ohm resistor is attached across the terminals of a12 volt battery, then a current of12 / 300 = 0.04 amperes flows through that resistor.

3.15. Temperature & Humidity sensor

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermostat to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old. Comes with a 4.7K or 10K resistor.

Advantage of Temperature & Humidity sensor

- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

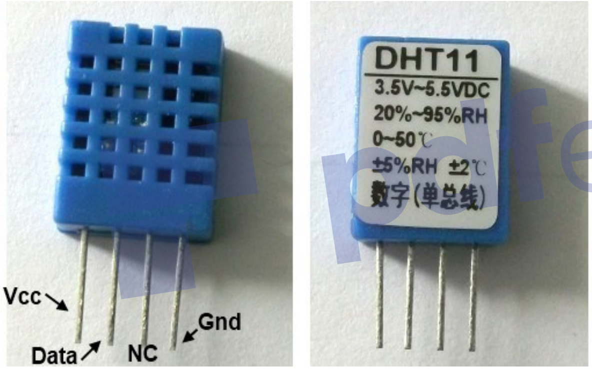


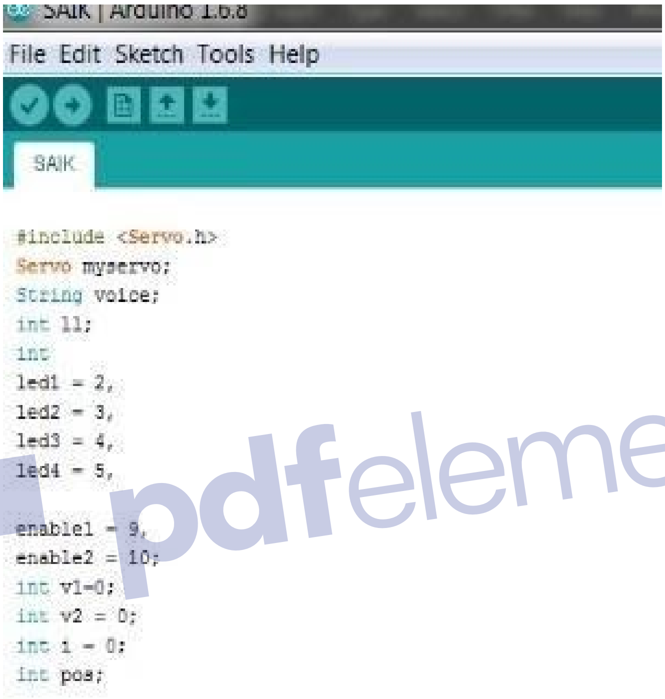
Figure 27: Temperature and Humidity sensor

Specification

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

3.16. Arduino software

The smart microcontroller unit named as Arduino Uno can be programmed with the Arduino software. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Uno from the Tools, Board menu (according to the microcontroller on your board). The IC used named as ATmega328 on the Arduino Uno comes pre burned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer.



```
#include <Servo.h>
Servo myservo;
String voice;
int l1;
int
led1 = 2,
led2 = 3,
led3 = 4,
led4 = 5,

enable1 = 9,
enable2 = 10;
int v1=0;
int v2 = 0;
int i = 0;
int pos;
```

Figure 28: Arduino Software Interface IDE.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

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

The Arduino Uno is one of the latest smart microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers.[15] The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Libc to compile programs, and for uploading the programs it uses avrdude. As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

3.17 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017.Feature led product releases are typically biannual, while maintenance-based service packs are released as required.

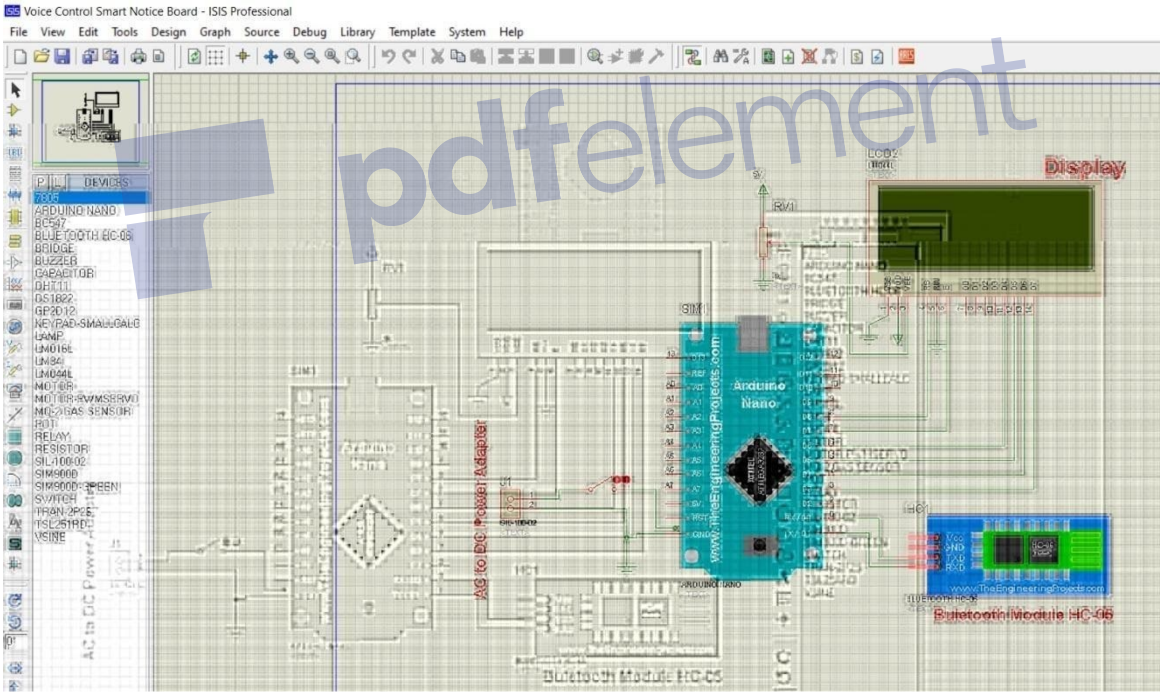


Figure 29: Proteus Software

3.18 Microcontroller Simulation

The microcontroller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool. Support is available for co-simulation of:

- Microchip Technologies PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC33 Microcontrollers.
- Atmel AVR (and Arduino), 8051 and ARM Cortex-M3 Microcontrollers
- NXP 8051, ARM7, ARM Cortex-M0 and ARM Cortex-M3 Microcontrollers.
- Texas Instruments MSP430, PICCOLO DSP and ARM Cortex-M3 Microcontrollers.
- Parallax Basic Stamp, Free scale HC11, 8086 Microcontrollers.

3.19 PCB Design

The PCB Layout module is automatically given connectivity information in the form of a net list from the schematic capture module. It applies this information, together with the user specified design rules and various design automation tools, to assist with error free board design. PCB's of up to 16 copper layers can be produced with design size limited by product configuration.

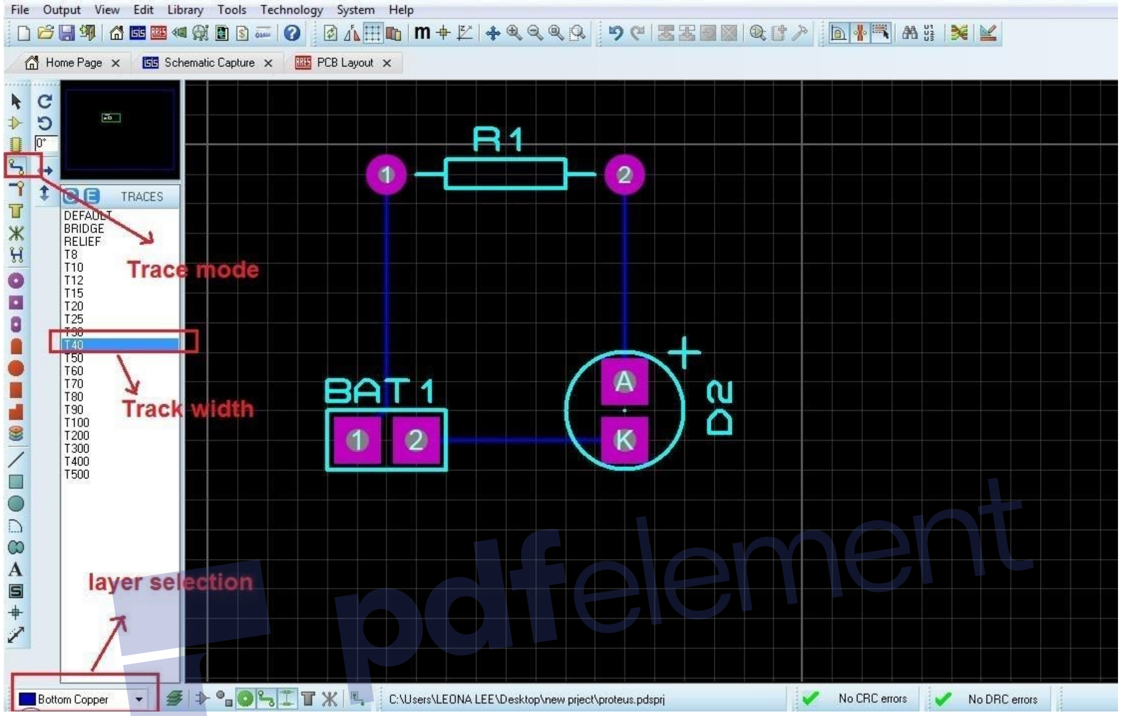


Figure 30: PCB Design

3.21 Working Principle:

The Smart irrigation System has wide scope to automate the complete irrigation system. Here we are building a IoT based Irrigation System using ESP8266 NodeMCU Module and DHT11 Sensor. It will not only automatically irrigate the water based on the moisture level in the soil but also send the Data to Blynk apps to keep track of the land condition. The System will consist a water pump which will be used to sprinkle water on the land depending upon the land environmental condition such as Moisture, Temperature and Humidity. We previously build similar Automatic Plant Irrigation System which sends alerts on mobile but not on IoT cloud. Apart from this, soil moisture detector circuit can also be helpful in building Smart Irrigation system. Before starting, it is important to note that the different crops require different Soil Moisture, Temperature and Humidity Condition. So, in this tutorial we are using such a crop which will require a soil moisture of about 50-55%. So, when the soil loses its moisture to less than 50% then Motor pump will turn on automatically to sprinkle the water and it will continue to sprinkle the water until the moisture goes up to 55% and after that the pump will be turned off. The sensor data will be sent to Bylinks apps Server in defined interval of time so that it can be monitored from anywhere in the world

3.22 Circuit Diagram:

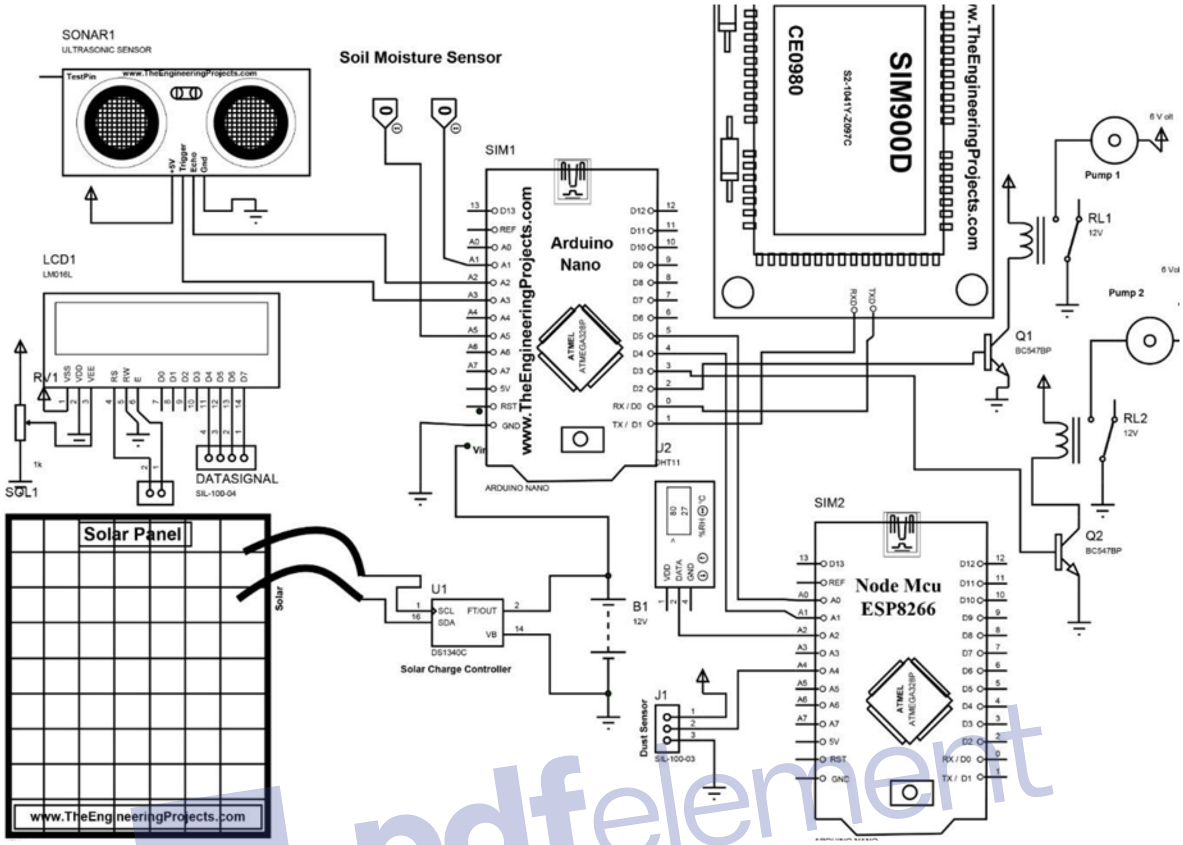


Figure 32: Circuit Diagram of IoT Smart Irrigation System

3.23 Block diagram :

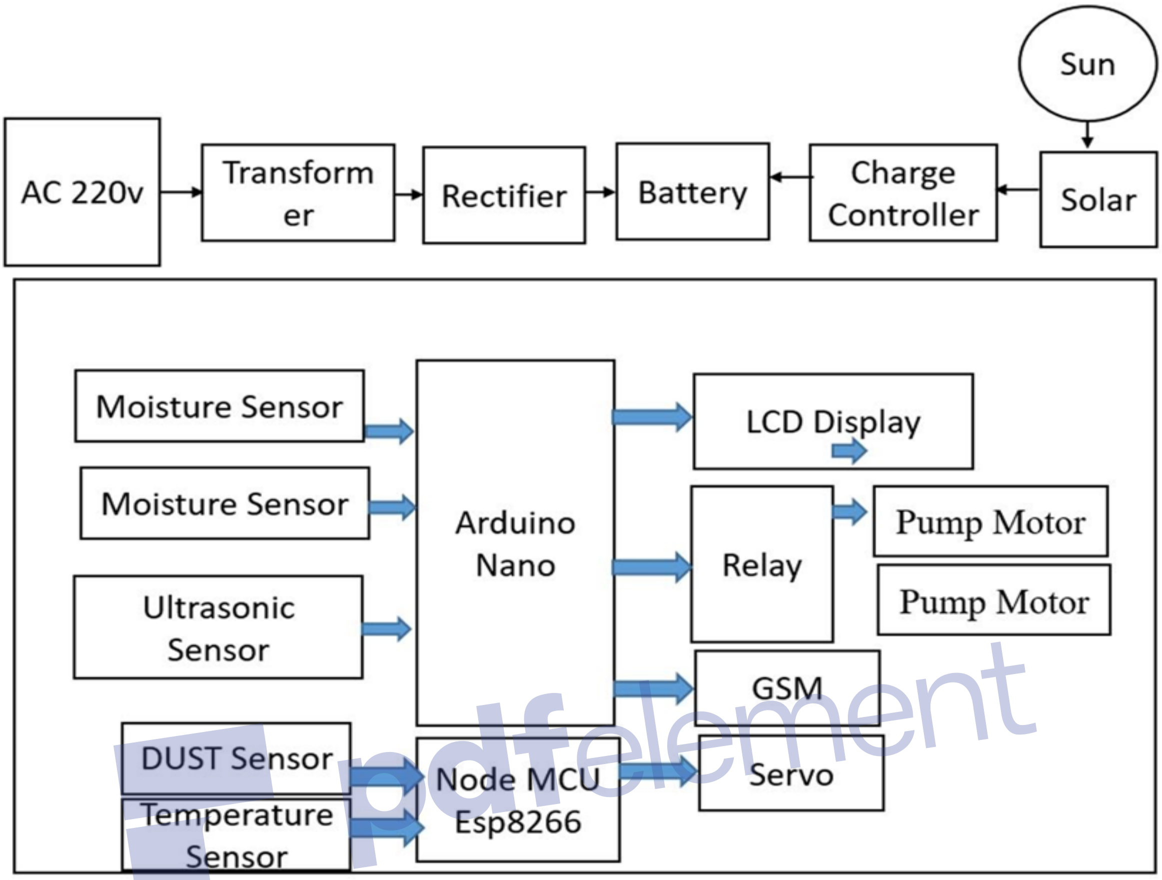


Figure 33: Block Diagram IoT Smart Irrigation System

3.24 Our Prototype Output:

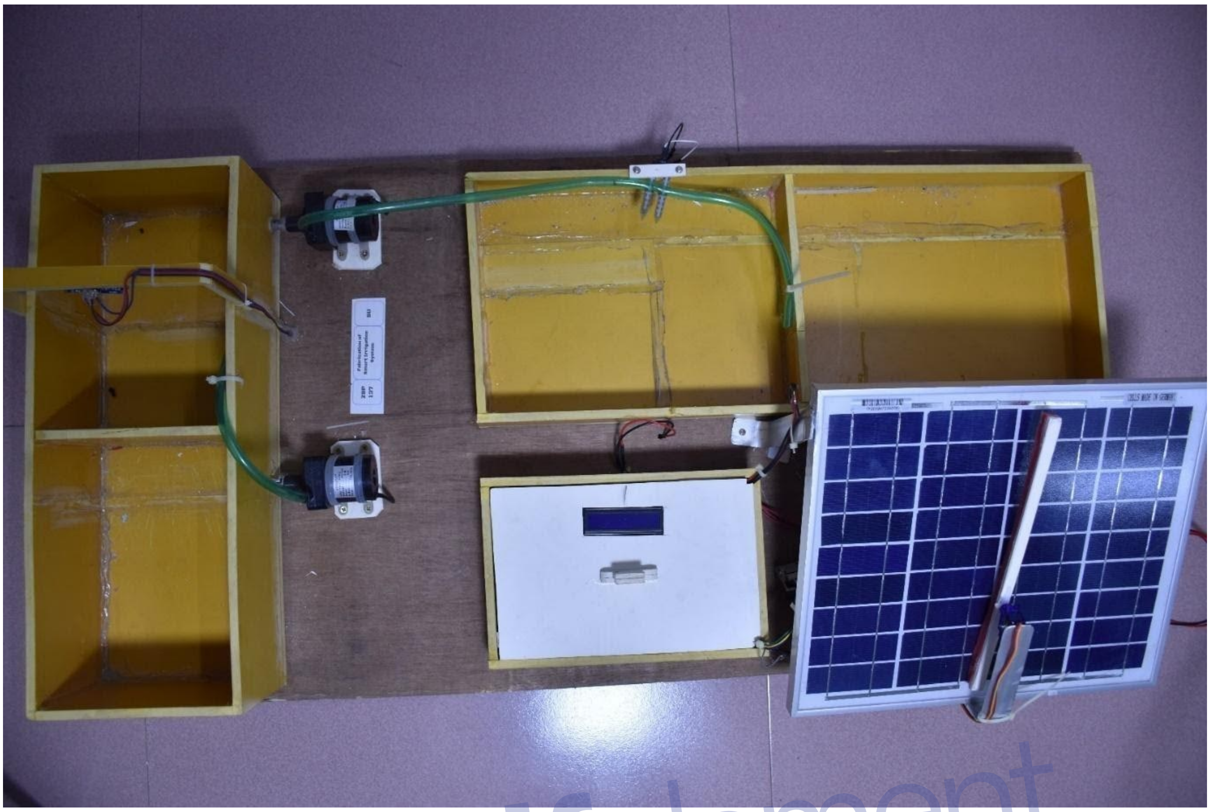


Figure 33: Our phonotype output

CHAPTER 4
RESULT AND DISCUSSION

4.1 Result & Discussion

In this project smart system has been implemented for irrigation system which uses concept of IoT through android application. This project includes various sensors like temperature sensor, ultrasonic, soil moisture sensor, Dust Sensor. Using communication technology message will be sent to IoT.

The soil moisture sensor value depends on the resistance of the soil. Value of the sensor varies 0 being the most wet condition. Thus, here by we conclude that the proposed system removes all the drawbacks of existing system and enhanced with IoT system.

- The Dust sensor detect the dust and automatically clean the solar panel.
- The Ultrasonic sensor check the water level and SWITCH On/Off the motor.
- The smart irrigation system implemented is cost effective for optimizing water resources for agricultural production.

The proposed framework can be utilized to switch on/off the water sprinkler depending on the soil dampness levels in this manner making the prepare easier to utilize. Through this extend it can be concluded that there can be impressive advancement in water system with those of IOT and mechanization

CHAPTER 5

5.1 Conclusion

A system to monitoring moisture levels in the soil was designed and the project provided an opportunity to study the existing systems, along with their features and drawbacks. The proposed system can be used to switch on/off the water sprinkler according to soil moisture levels thereby automating the process of irrigation which is one of the most time-consuming activities in farming. Agriculture is one of the most water-consuming activities. The system uses information from soil moisture sensors to irrigate soil which helps to prevent over irrigation or under irrigation of soil thereby avoiding crop damage. The farm owner can monitor the process online through a website. Through this project it

can be concluded that there can be considerable development in farming with the use of IOT and automation. Thus, the system is a potential solution to the problems faced in the existing manual and cumbersome process of irrigation by enabling efficient utilization of water resources.

5.2. Reference:

[1] IoT based smart agriculture

NikeshGondchawar, RS kawitkar.

[2] IOT based crop-field monitoring and irrigation automation

P Rajalakshmi, S Devi Mahalakshmi.

[3] IoT based monitoring system in smart agriculture.

SR Prathibha, AnupamaHongal, MP Jyothi

[4] Enironmental Impacts of Technological Change in Bangladesh Agriculture: Farmers' Perceptions and Empirical EvidenceSanzidur Rahman, Gopal B. Thapa

[5] Beginning Arduino (Mike McRoberts)

[6] Introduction to Arduino (Alan G. Smith)

[7] Electronic devices and circuit theory (Robert boylestad.

[8] Good friend W. (2015) Validity and reliability; How to Access the Quality of a study.

[9] S. R. Nandurkar, V. R. Thool, R. C. Thool, "Design and Development of Precision Agriculture System Using Wireless Sensor Network", IEEE International Conference on Automation, Control, Energy and Systems (ACES), 2014

[10] Dr. V. VidyaDevi,G. Meena Kumari, "Real- Time Automation and Monitoring System for Modernized Agriculture" ,International Journal of Review and Research in Applied Sciences and Engineering (IJRRASE) Vol-3 No.1. PP 7-12, 2013

[11] Alagupandi, P. et al., 2014, "Smart Irrigation System for Outdoor Environment using Tiny OS," the International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), IEEE.

[12] Lea-Cox, J.D., 2012, "Using Wireless Sensor Networks for Precision Irrigation Scheduling, Problems, Perspectives and Challenges of Agricultural Water Management," In Tech.Dr.N.Prakash, V R.Balaji, M.Sudha, "solar powered automated irrigation system for agriculture", International Journal of Advanced Engineering Technology, Vol. VII,

[13] Transformer Design Principles, Third Edition By

Robert M. Del Vecchio.

Dilipkumar M. Shah.

Pierre T. Feghali.

Bertrand Poulin.

Rajendra Ahuja

[14] Solar Energy: Renewable Energy and the Environment
Majid Ghassemi. Robert foster. Alma Cora.
[15] Arduino + Android Projects for the Evil Genius
2011 SIMON MONK

5.3. Appendix:

5.4 Program Code:

Program is the brain of our project.Arduino is an open-source computer hardwarewhiceworks on the program's instructions.

Code 1:

optical_dust_sensor_coding

```
#include <Servo.h>

#include <Wire.h>

#include <LCD.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27 , 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

//LiquidCrystal_I2C lcd(0x3F , 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

Servo myservo;  // create servo object to control a servo

int pos = 0; // variable to store the servo position

intmeasurePin = A0; //Connect dust sensor to Arduino A0 pin

intledPower = 12;  //Connect 3 led driver pins of dust sensor to Arduino D2

intsamplingTime = 280;

intdeltaTime = 40;

intsleepTime = 9680;

float voMeasured = 0;

float calcVoltage = 0;

float dustDensity = 0;

void setup()

{

 Serial.begin(9600);

 pinMode(ledPower,OUTPUT);

 myservo.attach(7);  // attaches the servo on pin 9 to the servo object

  lcd.begin(16,2);
  lcd.clear();
 }
void loop(){
  digitalWrite(ledPower,LOW); // power on the LED
  delayMicroseconds(samplingTime);
  voMeasured = analogRead(measurePin); // read the dust value
   delayMicroseconds(deltaTime);
  digitalWrite(ledPower,HIGH); // turn the LED off
  delayMicroseconds(sleepTime);
  // 0 - 5V mapped to 0 - 1023 integer values
  // recover voltage
  calcVoltage = voMeasured * (5.0 / 1024.0);
  // linear eqaution taken from http://www.howmuchsnow.com/arduino/airquality/
  // Chris Nafis (c) 2012
  dustDensity = 0.17 * calcVoltage - 0.1;
  Serial.print("Raw Signal Value (0-1023): ");
  Serial.print(voMeasured);

  Serial.print(" - Voltage: ");
  Serial.print(calcVoltage);
  Serial.print(" - Dust Density: ");
  Serial.println(dustDensity); // unit: mg/m3
  if(dustDensity> .30){
  lcd.setCursor(6,0);
 lcd.print("Dust");
 lcd.setCursor(6,1);
 lcd.print(dustDensity);
 for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
 // in steps of 1 degree
   myservo.write(pos);  // tell servo to go to position in variable 'pos'
   delay(15);   // waits 15ms for the servo to reach the position
   }

  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
   myservo.write(pos);   // tell servo to go to position in variable 'pos'
  delay(15);
   }
     delay(1000);
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
  // in steps of 1 degree
   myservo.write(pos);   // tell servo to go to position in variable 'pos'
   delay(15); // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
   myservo.write(pos);  // tell servo to go to position in variable 'pos'
delay(15);
  }
    delay(1000);
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
  // in steps of 1 degree
   myservo.write(pos);   // tell servo to go to position in variable 'pos'
   delay(15);   // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
```

```
    myservo.write(pos);   // tell servo to go to position in variable 'pos'
 delay(15);
  }
      delay(1000);
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
 // in steps of 1 degree
    myservo.write(pos);   // tell servo to go to position in variable 'pos'
    delay(15);  // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);//  tell servo to go to position in variable 'pos'
delay(15);
  }
    delay(1000);
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
 // in steps of 1 degree
    myservo.write(pos);  // tell servo to go to position in variable 'pos'
    delay(15);   // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);  // tell servo to go to position in variable 'pos'
 delay(15);
  }
 delay(1000);
}
if(dustDensity<  0){
  lcd.setCursor(6,0);
lcd.print("Dust");
lcd.setCursor(6,1);
lcd.print(0.00);
}
delay(1000);
}
```

Code 2

Ultrasonic Sensor:

```
#define trigPin 7
#define echoPin 8
int motor1 = 12;
int motor2 = 13;
#define SensorPin A0
#define SensorPin1 A2
float sensorValue = 0;
float sensorValue1 = 0;
#include <SoftwareSerial.h>
SoftwareSerialmySerial (3,4);
char msg;
char call;
String message;
void setup() {
  Serial.begin (9600);
  mySerial.begin(9600);    // Setting the baud rate of GSM Module
 pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  for (inti = 0; i<= 100; i++)
 {
   sensorValue = sensorValue + analogRead(SensorPin);
   delay(1);
 }
 sensorValue = sensorValue/100.0;
 Serial.println(sensorValue);
 delay(30);

 for (inti = 0; i<= 100; i++)
 {
   sensorValue1 = sensorValue1 + analogRead(SensorPin1);
   delay(1);
 }
 sensorValue1 = sensorValue1/100.0;
 Serial.println(sensorValue1);
 delay(30);
  float duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) * 0.0344;
  if (distance >= 400 || distance <= 2){
    Serial.print("Distance = ");
    Serial.println("Out of range");
  }
  else {
    Serial.print("Distance = ");
    Serial.print(distance);
    Serial.println(" cm");
 delay(500);
  }
 if (distance >12 ){
  digitalWrite(12, HIGH);
  }
else {
   digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  }
  if (distance < 12 &sensorValue>700){
  digitalWrite(13, HIGH);
  digitalWrite(12, LOW);
  }
  if ( sensorValue< 700){
  digitalWrite(12, LOW);
  }
```

```
if ( sensorValue1 <850 ){
  digitalWrite(5,HIGH );
 SendMessage("Pump Running Condition ");
 delay(300);
  }
 if ( sensorValue1 > 700){
  digitalWrite(5,LOW );
  }
  if (Serial.available()>0)
   switch(Serial.read())
  {
  }
  }
void SendMessage(String thisIsAString )
{
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);  // Delay of 1000 milli seconds or 1 second
  mySerial.println("AT+CMGS=\"+8801872858607\"\r"); // Replace x with mobile number
  delay(2000);
  mySerial.println(thisIsAString);// The SMS text you want to send
  delay(100);
  mySerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
}
```

Code 3:

Blynk Apps:

```
#define BLYNK_PRINT Serial
```

```
#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "8IRGTvpMvTnwiF64xMtGxM6uefpzCDVe";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "MuktaWifi";

char pass[] = "01947291082AG";

#define DHTPIN 2// What digital pin we're connected to


// Uncomment whatever type you're using!

#define DHTTYPE DHT11   // DHT 11

//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321

//#define DHTTYPE DHT21   // DHT 21, AM2301

DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void sendSensor()

{

  float h = dht.readHumidity();

  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");
```

```
 return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t);
}
void setup()
{
  // Debug console
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  dht.begin();
  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}
void loop()
{
Blynk.run();
  timer.run();
}
```

6.3. List of Component:

Sl.no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)	
1	Arduino Nano	Atmega 328p	2	280	280	
2	Node MCU	ESP8266	1	350	350	
3	Dust Sensor		1	590	590	
4	Pump Motor	12 volt DC	2	300	600	
5	Servo Motor	1kg		120	120	
6	Solar Panel	12 volt 10 wat	1	2000	2000	
7	12 volt step Down Transformer	1A	1	120	120	
8	Bridge Rectifier		1	15	15	
9	LCD Display	16*2	1	280	280	
9	PCB		1	60	60	
10	Ultrasonic Sensor		1	180	180	
11	Moisture Sensor		2	130	260	
12	GSM Module		1	350	350	
13	Buck Boost Converter		2	80	140	
14	Temperature Sensor	DHT11	1	140	140	
13	Others			200	500	
				Total	5925/=	

:

.