DESIGN AND IMPLEMENTATION OF AUTONOMOUS GREENHOUSE MONITORING AND CONTROL SYSTEM

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

M. Madhuri (318126512031) N. L. V Madhav (318126512036)

A. M. Zafeer (318126512001) Md. Waheed (318126512032)

Under the guidance of
Mr. N. Ram Kumar
B.E, M.E, (Ph.D)
Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)
2021-2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"DESIGN AND IMPLEMENTATION OF AUTONOMOUS GREENHOUSE MONITORING AND CONTOL SYSTEM"** submitted by **M. Madhuri (318126512031), N. L. V Madhav (318126512036), A. M. Zafeer (318126512001), Md. Waheed (318126512032)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide

Mr. N Ram Kumar
B.E, M.E, (Ph.D)
(Assistant Professor)
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department

Dr. V. Rajyalakshmi
Professor & HOD
Department of E.C.E
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa-531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr. N Ram Kumar**, B.E, M.E, (Ph.D.), Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

M. Madhuri (318126512031),
N. L. V. Madhav (318126512036),
A. M. Zafeer (318126512001),
Md. Waheed (318126512032).

ABSTRACT

The field of IOT is diversifying its wings in various domains like medical, industrial, transportation, agriculture, education, mining and many more fields for delivering social and economic benefits for emerging and developing economy. The advancement in integrated on chip computers like Arduino the technology is reaching the ground level with its application in agriculture and aquaculture. A green house is where plants such as flowers and vegetables are grown. Greenhouse warmups during the day as sun's rays penetrate through it, which heats up the plant, soil and structure. Green houses help to protect crops from many diseases, particularly those that are soil borne and splash onto plants in the rain. Numerous farmers fail to get good profits from the greenhouse crops, the reason being that they can't manage two essential factors, which determines plant growth as well as productivity. Greenhouses help to protect crops from many diseases, particularly those that are soil borne and splash onto plants in the rain. Numerous farmers fail to get good profits from the greenhouse crops as they fail to manage two essential factors, which determines plant growth as well as productivity. Green house temperature should not go below a certain degree, High humidity can result to crop transpiration, condensation of water on various greenhouse surfaces, and water evaporation from the humid soil.

To overcome such challenges, this greenhouse monitoring and control system comes to rescue. This project demonstrates the design and implementation of a various sensors for greenhouse environment monitoring and controlling. **The team intends to design an autonomous Greenhouse monitoring system using Arduino microcontroller.**

CONTENTS Page No.

CERTIFICATE i

ACKNOWLEDGEMENT ii

ABSTRACT iii

LIST OF FIGURES vii

1.INTRODUCTION TO IOT 01

1.1INTRODUCTION 02

1.2DEFINITION AND EXAMPLE OF IOT 04

1.2.1 DEFINITION BY ITU-T 04

1.2.2 ELEMENTS OF IOT 04

1.3 APPLICATIONS OF IOT 05

2.PROJECT DESCRIPTION 07

2.1 INTRODUCTION 08

2.2 OBJECTIVE 08

2.3 BLOCK DIAGRAM AND ITS OPERATION 09

2.4 HARDWARE COMPONENTS USED 10

2.5 SOFTWARE USED 10

3. ARDUINO UNO 11

3.1 WHAT IS ARDUINO 12

3.2 SPECIFICATIONS 13

3.3 WHY ARDUINO 13

3.4 SPECIAL MICROCONTROLLER FEATURES 14

3.5 TYPES OF ARDUINO BOARDS 14

3.6 ARDUINO UNO 15

3.7 ARDUINO FUNTIONAL BLOCKS 15

3.7.1 MAIN C 16

3.7.2 POWER 17

3.7.3 MEMORY 18

3.7.4 PIN DESCRIPTION 18

3.7.5 COMMUNICATION 19

3.7.6 RESET BUTTON 20

3.7.7 POWER LED INDICATOR 20

3.7.8. TX/RX LED 20

3.7.9 VOLTAGE REGULATOR 20

3.8 WARNINGS 21

3.9 APPLICATIONS 21

4.HARDWARE COMPONENTS 22

4.1 ARDUINO UNO 23

4.2 NODE MCU 25

4.3 SENSORS 26

4.3.1 LDR 26

4.3.2 RAIN SENSOR 27

4.3.3 SOIL MOISTURE 29

4.3.4 HUMIDITY & TEMPERATURE 31

4.3.5 WATER LEVEL INDICATOR 32

4.4 SERVO MOTOR 33

4.5 SUBMERGABLE PUMP 34

4.6 GENERAL RELAY 34

4.7 EXHAUST 35

4.8 LEDS / HPS LAMPS 36

4.9 LCD DISPLAY 37

5.ARDUINO PROGRAMMING 39

5.1 STRUCTURE 40

5.1.1 SET UP 40

5.1.2 LOOP 40

5.2 VARIABLES 42

5.2.1 VARIABLE SCOPE 43

5.3 FUNCTIONS 44

5.3.1 DIGITAL I/O 44

5.3.2 ANALOG I/O 45

5.3.3 OTHER FUNCTIONS 45

5.4 LIBRARIES 46

6.CIRCUIT DIAGRAM AND OPERATION 47

6.1 CIRCUIT DIAGRAM 48

6.2 PROCEDURE 49

7.ARDUINO IDE SOFTWARE 51

7.1 INTRODUCTION 52

7.2 WRITING SKETCHES 52

7.3 STEPS TO DUMP PROGRAM INTO ARDUINO UNO BOARD 52

7.4 ADVANTAGES 54

8.BLYNK APPLICATION 55

9.RESULTS 58

10.CONCLUSION & FUTURE SCOPE 60

REFERENCES 61

LIST OF FIGURES

Fig No.	Name of figure	Page No.
Figure 1.1 :	IoT Service Model for Smart Space Management	02
Figure 1.2:	An Example of IoT	04
Figure 1.3:	Elements of IoT	05
Figure 2.1:	Block Diagram of Greenhouse Monitoring System	09
Figure 3.1:	Arduino Uno Board	15
Figure 3.2:	Atmega 328P Pin Diagram	18
Figure 4.1:	Arduino Uno	23
Figure 4.2:	Contents on Arduino Board	23
Figure 4.3:	Architecture of Node MCU	25
Figure 4.4:	GPIO Pins on Node MCU(ESP8266)	26
Figure 4.5:	LDR Sensor	26
Figure 4.6:	Rain Sensor	28
Figure 4.7:	Soil-moisture sensor	30
Figure 4.8:	Water level Indicator Sensor	32
Figure 4.9:	Servo motor	33
Figure 4.10:	General Relay	34
Figure 4.11:	Exhaust Fan	35
Figure 4.12:	LCD Display	37
Figure 6.1:	Architecture of the Model	48
Figure 7.1:	Arduing Software HO	52
Figure 7.2:	Arduino Board selection Tool options	53
Figure 7.3:	Port selection	53
Figure 7.4:	Representation of Controllable options in Arduino Software	54
Figure 8.1:	Blynk app for Android	56
Figure 8.2:	Working of Blynk app	57
Figure 9.1:	Blynk Web Interface	59

CHAPTER – 1

INTRODUCTION TO IOT

1.1 INTRODUCTION

The Internet of Things (IoT) is an important topic in technology industry, policy, and engineering circles and has become headline news in both the specialty press and the popular media. This technology is embodied in a wide spectrum of networked products, systems, and sensors, which take advantage of advancements in computing power, electronics miniaturization, and network interconnections to offer new capabilities not previously possible. The basic idea of IoT is the pervasive presence around us of a variety of things or objects such as sensors, actuators, radio frequency identification tags, mobile phones, etc. which through unique addressing schemes, can interact with each other and cooperate with others to reach common goals.

With the rapid development of science and technology, people will be automatically and collaboratively served by the smart devices (like watches, mobiles, computers), smart transportation (like electric cars, bikes) and smart environments (like home automated devices). As one of the enablers of smart world, Internet of Things targets to connect various objects with unique addresses, to enable them to interact with each other and with the word.

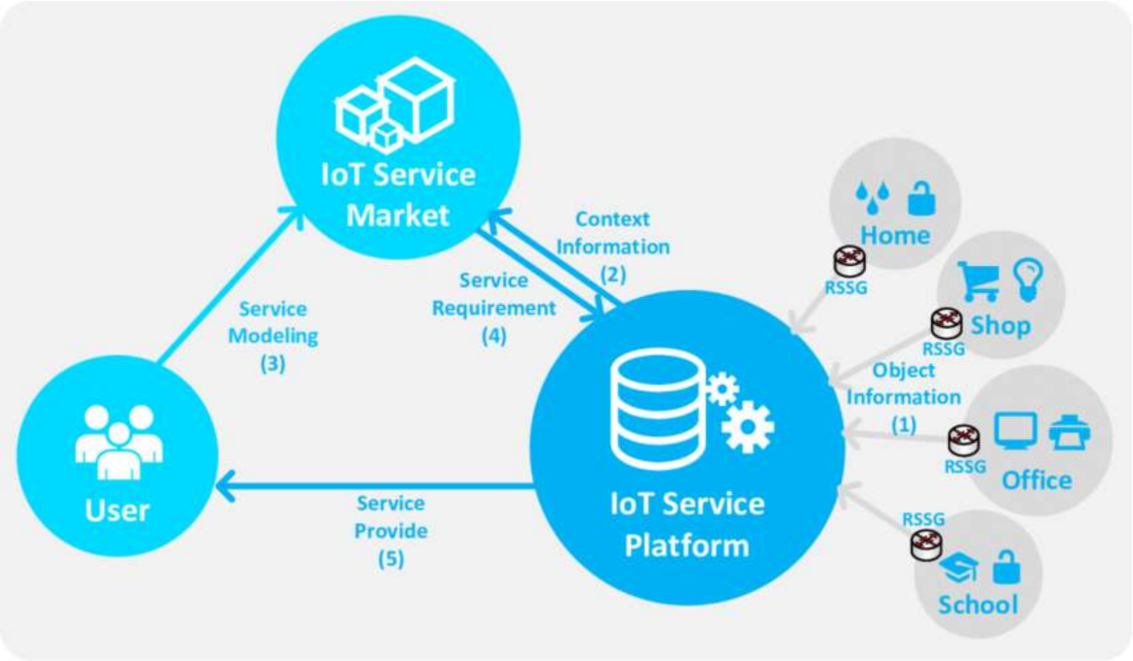


Figure 1.1 : IoT Service Model for Smart Space Management

The Internet of Things is an emerging topic of technical, social, and economic significance. Consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way we work, live, and play. Projections for the impact of IoT on the Internet and economy are impressive, with some anticipating as many as 100 billion connected IoT devices and a global economic impact of more than $11 trillion by 2025.

What is the Internet of Things?

The term "Internet of Things" (IoT) was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors.12 Ashton coined the term to illustrate the power of connecting Radio-Frequency Identification (RFID) tags13 used in corporate supply chains to the Internet to count and track goods without the need for human intervention. Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items. While the term "Internet of Things" is relatively new, the concept of combining computers and networks to monitor and control devices has been around for decades.

If the idea of connecting objects to each other and to the Internet is not new, it is reasonable to ask, "Why is the Internet of Things a newly popular topic today?" From a broad perspective, the confluence of several technology and market trends20 is making it possible to interconnect more and smaller devices cheaply and easily:

- **Ubiquitous Connectivity**—Low–cost, high–speed, pervasive network connectivity, especially through licensed and unlicensed wireless services and technology, makes almost everything "connectable''.
- **Advances in Data Analytics**— New algorithms and rapid increases in computing power, data storage, and cloud services enable the aggregation, correlation, and analysis of vast quantities of data, these large and dynamic datasets provide new opportunities for extracting information and knowledge.
- **Rise of Cloud Computing**– Cloud computing, which leverages remote, networked computing resources to process, manage, and store data, allows small and distributed devices to interact with powerful back-end analytic and control capabilities.

1.2 DEFINITION AND EXAMPLE OF IOT

IoT Definition: The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention. There is, however, no single, universal definition.

1.2.1 DEFINITION BY ITU-T

"In a broad perspective, the IoT can be perceived as a vision with technological and societal implications. From the perspective of technical standardization, IoT can be viewed as a global infrastructure for the information society, enabling advanced services by interconnecting things based on, existing and evolving, interoperable data capture, processing, and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, while maintaining the required privacy."

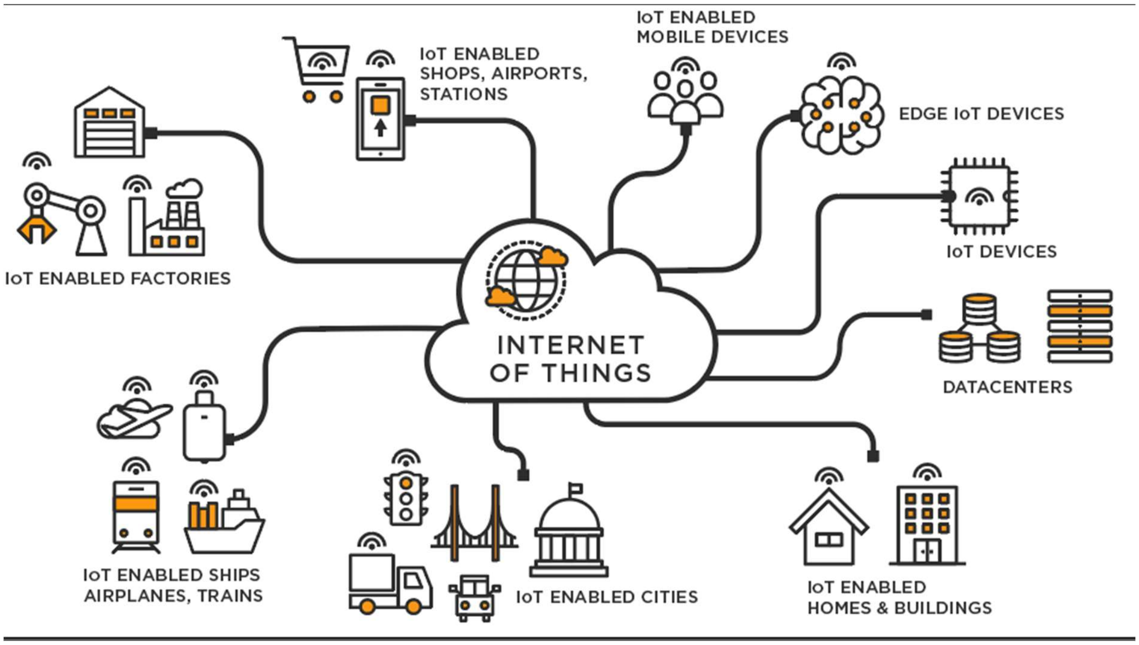

IoT ENABLED MOBILE DEVICES
IoT ENABLED SHOPS, AIRPORTS, STATIONS
EDGE IoT DEVICES
IoT DEVICES
IoT ENABLED FACTORIES
INTERNET OF THINGS
DATACENTERS
IoT ENABLED SHIPS AIRPLANES, TRAINS
IoT ENABLED CITIES
IoT ENABLED HOMES & BUILDINGS

Figure 1.2: An Example of IoT

1.2.2 ELEMENTS OF IOT

There are six elements in IoT. They are identification, sensing, communication technologies, computation, services and sematic.

- Identification plays a crucial role in naming and matching services with their demand. Ex: Electronic Product Code (EPC)

- Sensing is for collecting various data from related objects and sending it to a database. Ex: temperature sensors
- Communication technologies connect heterogeneous objects together to offer special services. Ex: Wi-Fi, Bluetooth
- Computation, the hardware processing units, and software applications perform this task.
- The services in IoT are identity related services, information aggregation services, collaborative aware services, and ubiquitous services.
- Semantic means the ability to extract knowledge intelligently so as to provide the required services.

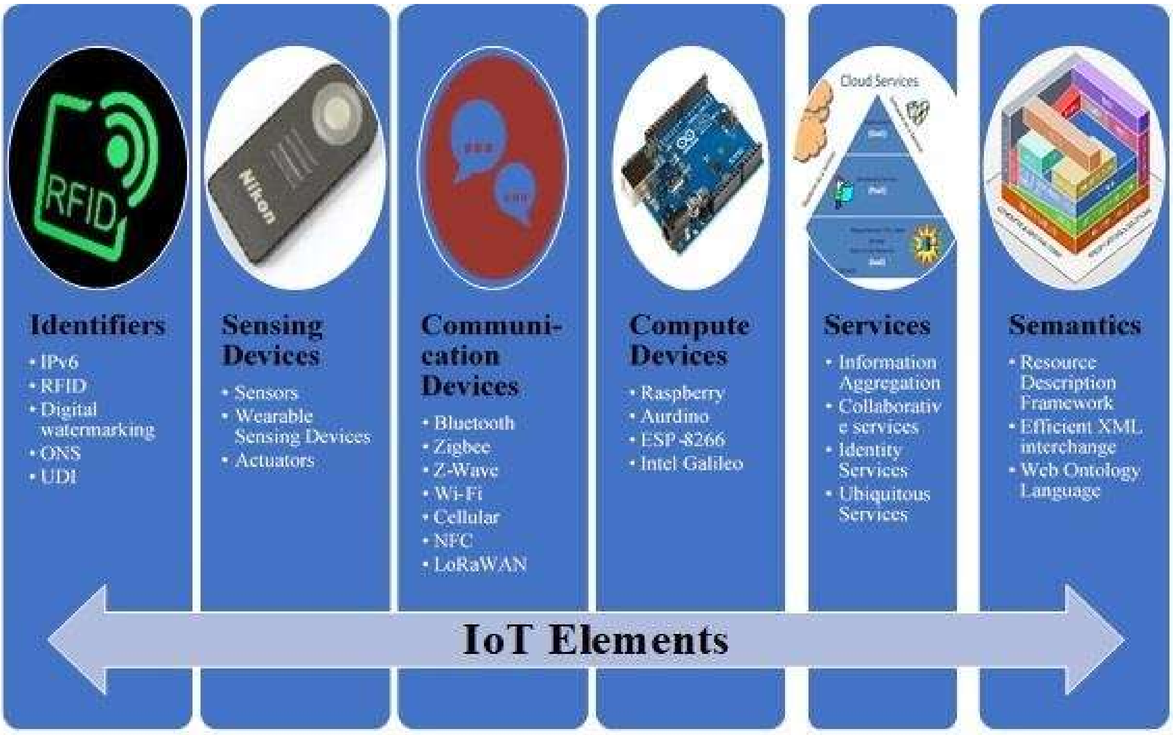


Figure 1.3: Elements of IoT

1.3 APPLICATIONS OF IOT

There are a lot of applications for IoT. We list some application scenarios as follows.

1. **Smart Home:** Personal lifestyle at home is enhanced, by ,making it more convenient and easier to monitor and operate home appliances and systems remotely.
2. **Industrial Automation:** With the minimum involvement, robotic devices are computerized to finish manufacturing tasks. In addition, the productivity is improved.

3. **Smart Healthcare:** Performance of healthcare applications is improved, by embedding sensors and actuators in patients and their medicine for monitoring and tracking patients. For example, by gathering and analysing patients body data with sensors and further delivering analysed data to processing centre.
4. **Smart Grid:** Power suppliers are assisted to control and manage, resources so that power can be offered proportionally to the population growth. Therefore, the energy consumption of houses and buildings could be enhanced. For example, the meters are connected to the network of energy providers, then the energy providers could enhance their services, by collecting, analysing, controlling, monitoring, and managing energy consumption.
5. **Smart City:** Quality of life in the city is ameliorated, by making it more convenient and easier for the residents to obtain information of interest. For instance, according to people's needs, various interconnected systems intelligently offer he desirable services (e.g., transportation, utilities, health, etc) to people.

CHAPTER – 2

PROJECT DESCRIPTION

2.1 INTRODUCTION

At present farmers are depending on manual testing for knowing the different parameters which are essential for a greenhouse like soil-moisture, water control, temperature, and the intensity of light. This will consume time and inaccurate because these parameters may alter with time. To overcome this problem, technology should be brought to agriculture which increases the productivity and minimize the losses by constant monitoring of these parameters by using suitable sensors. This proposed work uses an integrated Arduino UNO in connection with Node MCU which makes it unique. For good quality and productivity of plants the greenhouse is to be monitored and certain parameters are to be kept under control. Various sensors are mounted to continuously monitor and analyze the crop yield.

The Arduino UNO is capable of many functions. Having fourteen digital input-output pins and six analog input pins it can read data and users are able to use that data in many ways. This helps us to implement a web application that hosts and displays the data that is read by Arduino and view the status of the controlled sensors which are connected to Arduino. This is achieved through a combination of the Arduino, Node MCU and the programming using software Arduino IDE to upload codes to Arduino microcontroller.

2.2 OBJECTIVE

The main objective of this project is to design and implement an autonomous greenhouse monitoring and control system using IoT that is capable of monitoring and automating most of the actions needed to maintain greenhouse conditions at check through a manageable web application to which data is fed wirelessly through Wi-Fi using Node MCU. The system has a great flexibility by using Wi-Fi technology to interconnect sensors and other hardware to server is designed by using Arduino Uno, Node MCU, and relay board.

2.3 BLOCK DIAGRAM AND ITS OPERATION

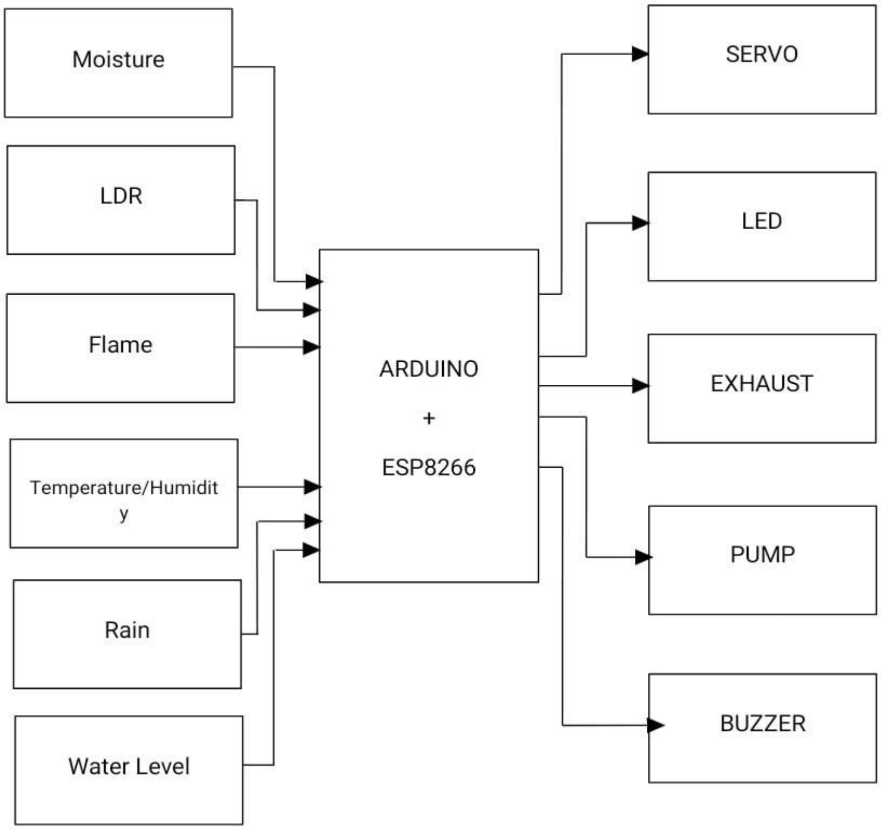


Figure 2.1: Block Diagram of Greenhouse Monitoring System

- The Arduino UNO microcontroller along the ESP8266 enables wireless connectivity and data transfer between the device and the web application.
- Arduino runs the code to get data from the sensors connected in the model. The values are compared against their threshold values pre-defined in the code.
- All the values received from sensors are calibrated.
- After every comparison if the value isn't within the defined threshold, then a signal is sent to the following sensor to act and work to bring that parameter within its threshold value.
- These sensors are connected to the output components which respond to the data received from the Arduino microcontroller.

- We check the parameters like soil-moisture, temperature/humidity, water-level in the reserve tank, intensity of environmental lighting and also if its raining so as to use rainwater instead to water from the tank.
- The digital pin output HIGH/LOW which drives the replay ON/OFF there by corresponding output components will be ON/OFF.
- We use Node MCU(ESP8266) to send all the data to a web application called Blynk which allows us to monitor the greenhouse continuously.
- The data can also be viewed on the mobile application of Blynk called Blynk IoT which timely updates the data and the active components working in real time. The details and levels can also be viewed on the LCD display attached to the device.

2.4 HARDWARE COMPONENTS

- Arduino Uno Microcontroller
- Node MCU(ESP8266)
- Submergible pump
- Servo motor
- General relay
- Sensors
    - Soil-moisture sensor
    - Temperature/Humidity sensor
    - LDR sensor
    - Rain sensor
    - Water-level sensor
    - Fire hazard sensor
- LED lights
- Exhaust fan

2.5 SOFTWARE USED

Arduino IDE (Integrated Development Environment) for coding.

Blynk IoT web application to host and view sensor data in real time.

CHAPTER – 3

ARDUINO UNO

3.1 WHAT IS ARDUINO

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write, and upload computer code to the physical board.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with motors, GPS units, cameras, the internet, and even smart-phone or TV.

It works with sensors like:

- Push buttons touch pads, tilt switches.
- Variable resistors (e.g., volume knob / sliders)
- Photoresistors (sensing light levels) Thermistors (temperature)
- Ultrasound (proximity range finder)

It works with Actuators:

- Lights, LEDS
- Motors
- Speakers
- Displays (LCD)

3.2 SPECIFICATIONS

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

3.3 WHY ARDUINO

- It is an open-source project, software/hardware is extremely accessible and very flexible to be customized and extended.
- It is flexible, offers a variety of digital and analog inputs, SPI (serial interface), digital and PWM outputs.
- It is easy to use, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers.
- It is inexpensive board and comes with free authoring software.
- It has its own MAC address to identify in network.

3.4 SPECIAL MICROCONTROLLER FEATURES

Power-on Reset and Programmable Brown-out Detection

Internal Calibrated Oscillator

External and Internal Interrupt Sources

Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby I/O and Packages 23 Programmable I/O Lines

28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

Operating Voltage: 1.8 - 5.5V

Temperature Range: 40C to 85C

Speed Grade: 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5.V, 0 - 20MHz @ 4.5 5.5V Power Consumption at 1MHz, 1.8V, 25 C

Active Mode: 0.2mA Power-down Mode: 0.1µA

Power-save Mode: 0.75µA (Including 32kHz RTC)

3.5 TYPES OF ARDUINO BOARDS

There have been many revisions of the USB Arduino. Some of them are:

- Arduino Uno
- Arduino Leonard
- Arduino Lilypad
- Arduino Mega
- Arduino Nano
- Arduino Mini
- Arduino Mini Pro
- Arduino BT

3.6 ARDUINO UNO

The Arduino Uno is one of the more popular boards in the Arduino family and a great choice for beginners. It is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP (in circuit serial programming) header and a reset button. It contains everything needed to support the microcontroller.

3.7 ARDUINO FUNTIONAL BLOCKS

1. USB port
2. Power input socket
3. GND pins
4. 5V output pin
5. 3.3V output pin
6. Analog inputs
7. Digital I/O pins
8. PWM outputs
9. AREF pin
10. RESET button
11. Power indication LED
12. Tx and Rx indication LEDS
13. Micro Controller (AT Mega 328)
14. Voltage Regulator

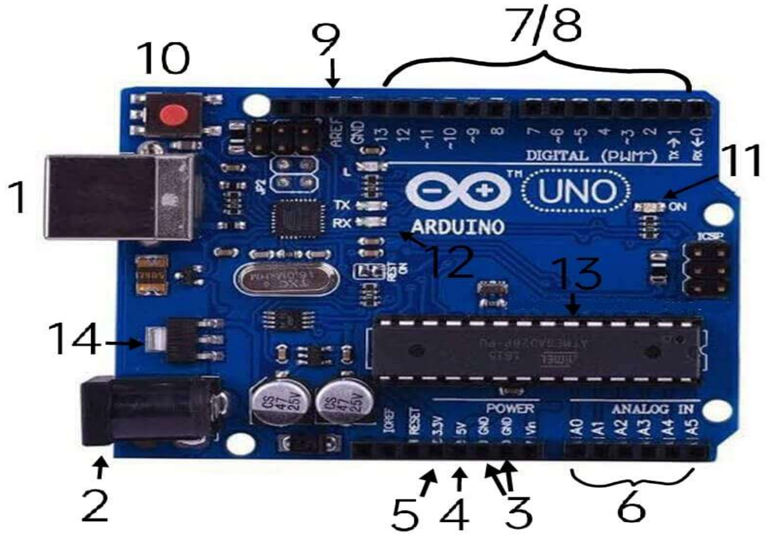


Figure 3.1: Arduino Uno Board

3.7.1 MAIN IC

The black square with all the metal legs is an IC. or Integrated Circuit. Think of it as the brain of our Arduino. The main IC on the Arduino is slightly different from board type to board type but is usually from the AT mega line of ICs from the ATMEL Company. In Arduino UNO AT mega 328 Micro controller is used.

Peripheral Features

- High-performance, Low-power Atmel AVR 8-bit Microcontroller
- Advanced RISC Architecture
    - 131 Powerful Instructions - Most Single-clock Cycle Execution
    - 32 x 8 General Purpose Working Registers
    - Fully Static Operation
    - Up to 16 MIPS Throughput at 16MHz
    - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
    - 32Kbytes of In-System Self-programmable Flash program memory
    - 1024Bytes EEPROM
    - 2Kbytes Internal SRAM
    - Write/Erase Cycles: 10,000 Flash/1,00,000 EEPROM
- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and
- Capture Mode
- Real Time Counter with Separate Oscillator
- Six PWM Channels
- 8-channel 10-bit ADC in TQFP and QFN/MLF package
- 6-channel 10-bit ADC in PDIP Package
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

3.7.2 POWER

The Uno board 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 from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin: The input voltage to the Uno 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.
- SV: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator and can damage your board. We don't advise it.
- 3V3: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND: Ground pins.
- IOREF. This pin on the Uno board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

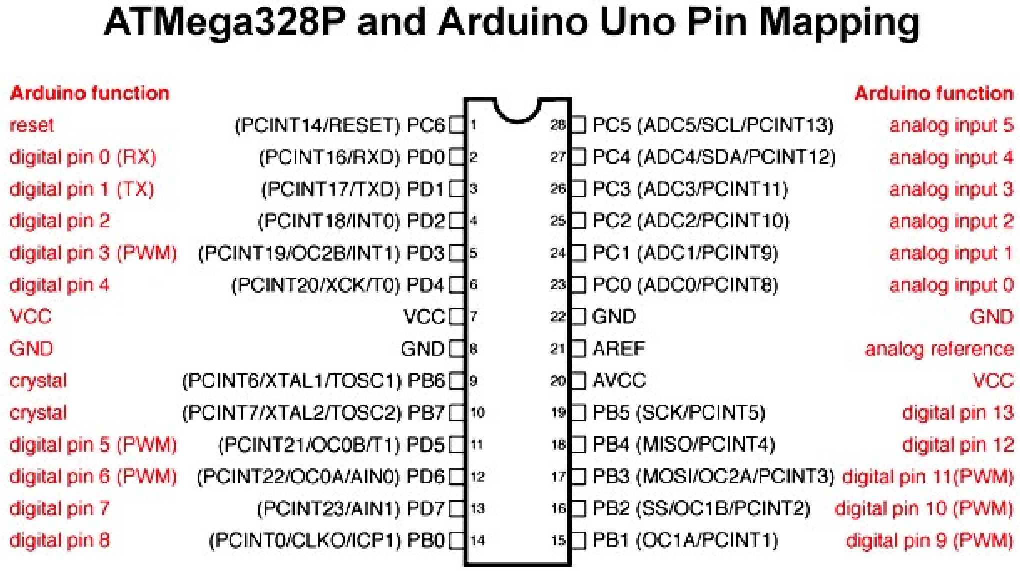
3.7.3 MEMORY

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

3.7.4 PIN DESCRIPTION

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

Digital pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega 168 pins 17, 18 & 19). Avoid low impedance loads on these pins when using the ICSP header.



Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Figure 3.2: Atmega 328P Pin Diagram

In addition, some pins have specialized functions:

- **Pins (2-3):** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. These are digital pins 0 and 1 of Arduino board.
- **Pins (4-5):** These are digital pins 2 and 3 of the Arduino board. These are external interrupts. 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.
- **Pins 5, 11, 12, 15, 16, 17:** These are PWM pins, provides 8-bit PWM output with the analogWrite() function. These are 3, 5, 6, 9, 10 and 11 pins of the Arduino board.
- **Pins (16-19):** These pins support SP1 communication using the SPI library. These are 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) pins of the Arduino board.
- **LED 13:** There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **Pins (27-28):** These are A4 or SDA pin and A5 or SCL pin of the Arduino board-support TWI communication using the Wire library.

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

There are a couple of other pins on the board:

Pin 1: 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.

Pin 21: AREF. Reference voltage for the analog inputs used with analogReference().

3.7.5 COMMUNICATION

The Uno has several facilities for communicating with a computer, another Uno board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. The Arduino Software (IDE) 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 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 serial communication on any of the Uno's digital pins. The ATmega328 also supports 12C (TWI) and SPI communication. For SP1 communication, use the SPI library.

3.7.6 RESET BUTTON

Just like the original Nintendo, the Arduino has a reset button (10). Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

3.7.7 POWER LED INDICATOR

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' (11). This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

3.7.8 TX/RX LED

TX is short for transmit and RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear - once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDS. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

3.7.9 VOLTAGE REGULATOR

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says, it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

3.8 WARNINGS

The Uno has a resettable polyfuse that protects your computer's USB ports from shorts and over current. 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.

3.9 APPLICATIONS

Arduino was basically designed to make the process of using electronics in multidisciplinary projects more accessible. It is intended for artists, designers 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. Because of these features, Arduino finds extensive application in various fields. Arduino projects can be stand-alone, or they can communicate with software running on a computer.

CHAPTER – 4

HARDWARE COMPONENTS

4.1. ARDUINO UNO

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.

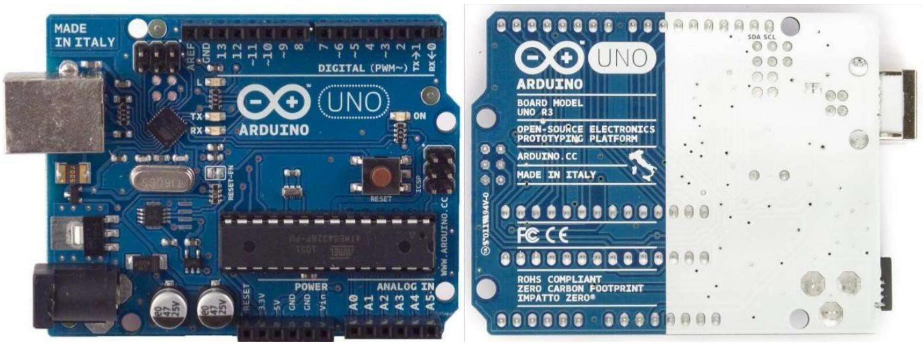


Figure 4.1: Arduino Uno

Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g., Flash, Processing, Max MSP.) The boardscan be assembled by hand or purchased preassembled; the open-source IDE can bedownloaded for free

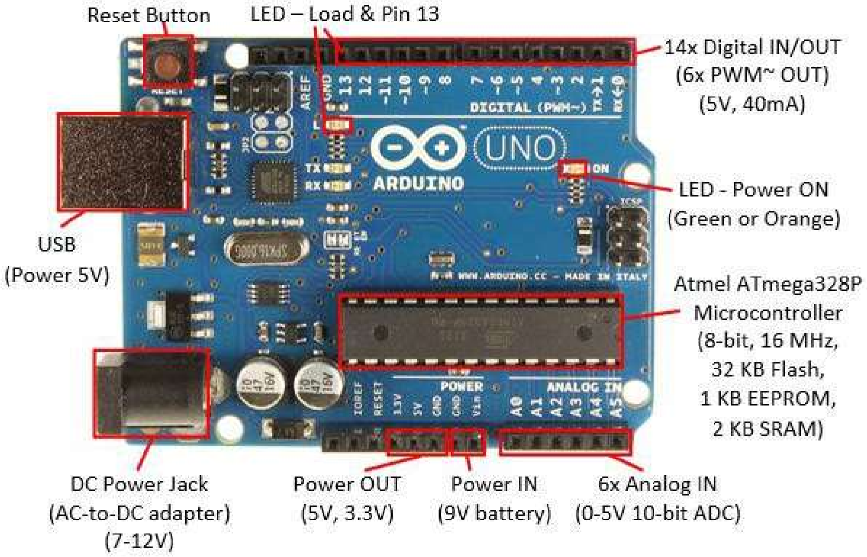


Figure 4.2: Contents on Arduino Board

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All these tools take the messy details of microcontroller programming and wrap it upin an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interestedamateurs over other systems.

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than Rs.600/-
- **Cross-platform** - The Arduino software runs on Windows, MacintoshOSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino program environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.
- **Open source and extensible software**- The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for themodules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

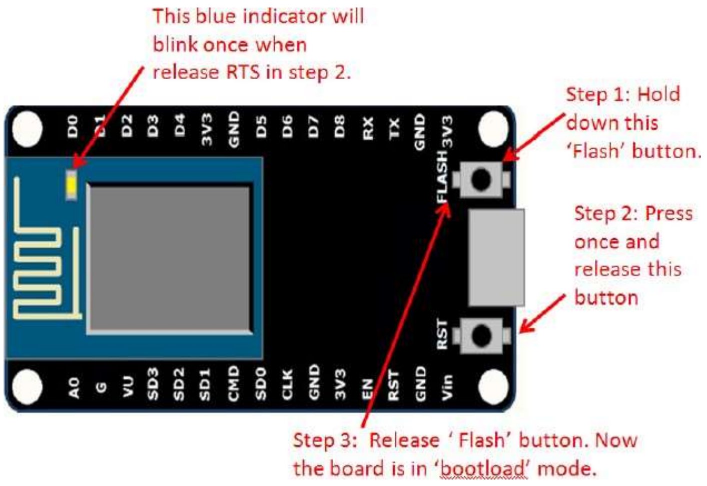
4.2. NODE MCU

Node MCU V3 is an open-source firmware and development kit that plays a vitalrole in designing an IoT product using a few script lines.

Multiple GPIO pins on the board allow us to connect the board with other peripherals and can generate PWM, I2C, SPI, and UART serial communications.

The interface of the module is mainly divided into two parts including both Firmware and Hardware where former runs on the ESP8266 Wi-Fi SoC and later is based on the ESP-12 module.

The firmware is based on Lua – A scripting language that is easy to learn, giving simple programming environment layered with a fast-scripting language that connects you with a well-known developer community.



This blue indicator will blink once when release RTS in step 2.

Step 1: Hold down this 'Flash' button.

Step 2: Press once and release this button

Step 3: Release ' Flash' button. Now the board is in 'bootload' mode.

Figure 4.3: Architecture of Node MCU

And open-source firmware gives you the flexibility to edit, modify and rebuilt the existing module and keep changing the entire interface until you succeedin optimizing the module as per your requirements USB to UART converter is added on the module that helps in converting USBdata to UART data which mainly understands the language of serial communication.

Instead of the regular USB port, Micro-USB port is included in the module that connects it with the computer for dual purposes: programming and powering upthe board.

The board incorporates status LED that blinks and turns off immediately, giving you the status of the module if it is running properly when connected with the computer.

The ability of module to establish a flawless Wi-Fi connection between two channels makes it an ideal choice for incorporating it with other embedded devices like Raspberry Pi.

Node MCU V3 Pinout

Node MCU V3 comes with several GPIO Pins. Following figure shows the Pinout of the board.

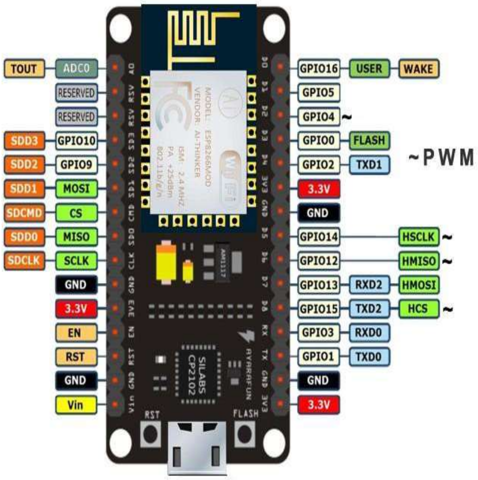


Figure 4.4: GPIO Pins on Node MCU(ESP8266)

There is a candid difference between Vin and VU where former is the regulated voltage that may stand somewhere between 7 to 12 V while later isthe power voltage for USB that must be kept around 5 V.

4.3. SENSORS

4.3.1 LDR SENSOR

The light dependent resistor (LDR) is a sensor whose resistance decreases when light impinges on it. This kind of sensor is commonly used in light sensor circuits in open areas, to control streetlamps for example. Another possible use is in spectroscopic apparatus. In this kind of apparatus, continuous light or pulsed light can be used. Continuous light is used in common spectroscopic apparatus. The use of lock-in amplifiers made the use of pulsed light in spectroscopy easier, as is commonly used in photoacoustic spectroscopy. LDR's are made of semiconductors as light sensitive materials, on an isolating base.

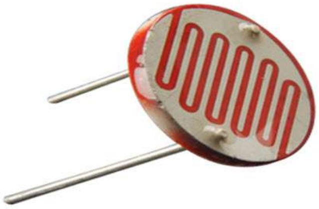


Figure 4.5: LDR Sensor

The most common semiconductors used in this system are cadmium sulphide, lead sulphide, germanium, silicon, and gallium arsenide.

A less known light sensor is the electret microphone. As the electret membrane functions as an absorbing black body, and as the electret microphone case has an air chamber that can be used as photoacoustic chamber, the electret microphone can be used as a detector of pulsed light. This type of microphone can be used to obtain the transmission spectrum of any transparent material. The aim of this communication is to study the response of LDR to pulsed light and the analysis of the spectral curves obtained with a LDR and an electret microphone as light sensors in an optical spectroscopy device.

- **Features:**
  1. Variable resistance
  2. Stable and long lifetime
  3. Fast response and high sensitivity
- **Specifications:**
  1. Maximum power dissipation is 200mW
  2. The maximum voltage at 0 lux is 200V
  3. The peak wavelength is 600nm
  4. Minimum resistance at 10lux is 1.8kΩ
  5. Maximum resistance at 10lux is 4.5kΩ
  6. Typical resistance at100lux is 0.7kΩ
  7. Dark resistance after 1 sec is 0.03MΩ
- **Applications:**
  1. Alarm clocks
  2. Burglar alarm circuits
  3. Light intensity meters

4.3.2 RAIN SENSOR

Nowadays, conserving water as well as its proper usage is essential in everyone’s life. Here is a sensor namely rain sensor which is used to detect the rain and generate an alarm. So, we can conserve water to use it later for different purposes. A rain sensor is one kind of switching device which is used to detect the rainfall. It works like a switch and the working principle of this sensor is, whenever there is rain, the switch will be normally closed.

The rain sensor module/board is shown below. Basically, this board includes nickel coated lines and it works on the resistance principle. This sensor module permits to gauge moisture through analog output pins & it gives a digital output while moisture threshold surpasses.

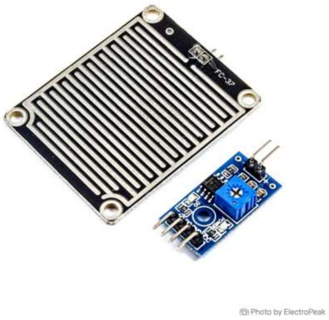


Photo by ElectroPeak

Figure 4.6: Rain Sensor

This module is similar to the LM393 IC because it includes the electronic module as well as PCB. Here PCB is used to collect the raindrops. When the rain falls on the board, then it creates a parallel resistance path to calculate through the operational amplifier.

This sensor is a resistive dipole and based on the moisture only it shows the resistance. For example, it shows more resistance when it is dry and shows less resistance when it is wet.

- **Pin Configuration:**

The pin configuration of this sensor is shown below. This sensor includes four pins which include the following.

1. Pin1 (VCC): It is a 5V DC pin
2. Pin2 (GND): it is a GND (ground) pin
3. Pin3 (DO): It is a low/ high output pin
4. Pin4 (AO): It is an analog output pin

- **Features:**

1. Working voltage 5V.
2. Output format: Digital switching output (0 and 1), and analog voltage output AO.
3. Potentiometer adjusts the sensitivity.
4. Uses a wide voltage LM393 comparator.

- **Specifications:**

1. Operating voltage ranges from 3.3 to 5V
2. The operating current is 15 mA
3. The sensing pad size is 5cm x 4 cm with a nickel plate on one face.
4. Comparator chip is LM393
5. Output types are AO (Analog o/p voltage) & DO (Digital switching voltage)
6. The length & width of PCB module 3.2cm x 1.4cm
7. Sensitivity is modifiable through Trimpot
8. Red/Green LED lights indicators for Power & Output

- **Applications:**

1. Irrigation field
2. Home automation
3. Communication
4. Automobiles

4.3.3 SOIL MOSITURE SENSOR

The soil moisture sensor is one kind of sensor used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content.
The relation among the calculated property as well as moisture of soil should be adjusted & may change based on ecological factors like temperature, type of soil, otherwise electric conductivity. The microwave emission which is reflected can be influenced by the moisture of soil as well as mainly used in agriculture and remote sensing within hydrology.

These sensors normally used to check volumetric water content, and another group of sensors calculates a new property of moisture within soils named water potential. Generally, these sensors are named as soil water potential sensors which include gypsum blocks and tensiometer.

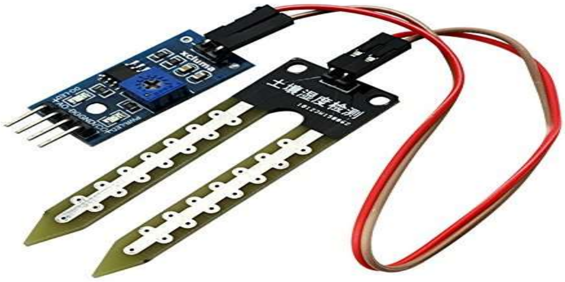


Figure 4.7: Soil-moisture sensor

Soil Moisture Sensor Pin Configuration

1. The FC-28 soil moisture sensor includes 4-pins
2. VCC pin is used for power
3. A0 pin is an analog output
4. D0 pin is a digital output
5. GND pin is a Ground

- **Features:**

1. LM393 based design
2. Easy to use with Microcontrollers or even with normal Digital/Analog IC
3. Small, cheap, and easily available

- **Specifications:**

1. Operating Voltage: 3.3V to 5V DC
2. Operating Current: 15mA
3. Output Digital - 0V to 5V, Adjustable trigger level from preset
4. Output Analog - 0V to 5V based on infrared radiation from fire flame falling on the sensor
5. LEDs indicating output and power

- **Applications:**

1. Gardening
2. Irrigation Systems
3. Used in Controlled Environments

4.3.4 HUMIDITY & TEMPERATURE SENSOR

DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output. DHT11 can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results. DHT11 is a low-cost humidity and temperature sensor which provides high reliability and long-term stability.

It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and outputs a digital signal on the data pin (no analog input pins needed). It’s very simple to use, and libraries and sample codes are available for Arduino and Raspberry Pi.

This module makes is easy to connect the DHT11 sensor to an Arduino or microcontroller as includes the pull up resistor required to use the sensor. Only three connections are required to be made to use the sensor - Vcc, Gnd and Output.

- **Features:**

1. Calibrated digital signal
2. Outstanding long-term stability
3. Extra components not needed

- **Specifications:**

1. Model: WSDCGQ11LM
2. Dimensions: 36 x 36 x 9 mm (1.42 x 1.42 x 0.35 in.)
3. Wireless Protocol: Zigbee
4. Battery: CR2032
5. Temperature Range and Precision: -20° C - +50° C, +0.3° C
6. Humidity Range and Precision: 0 - 100% RH (no condensation), ±3%
7. Atmospheric Pressure Range and Precision: 30 kPa -110 kPa, ±0.12 kPa

- **Applications:**

1. Humidifiers
2. Humidor
3. Cloth dryers
4. Microwave ovens

4.3.5 WATER LEVEL INDICATOR SENSOR

A Water Level Indicator may be defined as a system by which we can get the information of any water reservoir. Water level indicator system is quite useful to reduce the wastage of water from any reservoir, while filling such reservoir. Water is most essential thing on earth .Safe drinking water is essential to human and other life forms even though it provides no calories or organic nutrients.

After assembling the system, what remains is to observe its operation and efficiency.

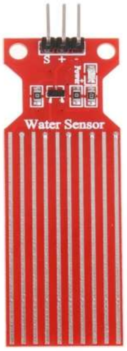


Figure 4.8: Water level Indicator Sensor

This can be done by breaking down the activity of the controller from the detection of water to the working of the pump. We go over the responses obtained when water reaches the sensors and the logic employed behind it. We also try to justify how a system as simple as ours can compete with those available commercially.

- **Features:**

1. Save Electricity & wastage of water.
2. It is a four-level water indicator.
3. Automatic & manual motor control.

- **Specifications:**

1. Main supply: 220 V - 250 V AC / 1 Phase,
2. 50-60 Hz Operating Voltage: 12 V DC. 1 amp. Max Load
3. : 10/25 Amp (1/2 Hp to 20 Hp Water Pump)Switch Angle: +/- 50°C

- **Applications:**

1. Can be used in water tanks to control water levels.
2. Automatically turn ON/OFF pumps.
3. Can be used in factories, commercial complexes, apartments, home,
4. Fuel tank level gauging.
5. Oil tank level control.

4.4. SERVO MOTOR

There are some special types of application of electrical motor where rotation of the motor is required for just a certain angle not continuously for long period of time. For these applications, some special types of motor are required with some special arrangement which makes the motor to rotate a certain angle for a given electrical input (signal). This is normally a simple motor which is controlled for specific angular rotation with the help of additional servomechanism (a typical closed loop feedback control system). Servo motor is a special type of motor which is automatically operated up to certain limit for a given command with help of error-sensing feedback to correct the performance.

Figure 4.9: Servo motor

- **Working Principle of Servo Motor**

A servo motor is basically a DC motor (in some special cases it is AC motor) along with some other special purpose components that make a DC motor a servo. In a servo unit, you will find a small DC motor, a potentiometer, gear arrangement and an intelligent circuitry. The intelligent circuitry along with the potentiometer makes the servo to rotate according to our wishes. As we know, a small DC motor will rotate with high speed, but the torque generated by its rotation will not be enough to move even a light load. This is where the gear system inside a servomechanism comes into the picture. The gear mechanism will take high input speed of the motor (fast) and at the output, we will get an output speed which is slower than original input speed but more practical and widely applicable.

4.5 SUBMERGIBLE PUMP

A Submergible water pump can drain water from a basement or shallow flooded areas, drain and fill a swimming pool or dam.

It can also be utilized in the irrigation needed for agriculture. Water pumps are employed for getting rid of excess water to reduce the downtime from large rain events

The main advantage of this water pumps it's their operating current, because it is so low they can be powered directly by the Arduino 5V outputs, making a very simple set-up that will make it easier and faster for prototyping of your automated watering system.

This Submergible water pump pumps the water from the tank to the plants when there is a decrease in moisture levels in the soil.

4.6 GENERAL RELAY

Relays are the switches which aim at closing and opening the circuits electronically as well as electromechanically. It controls the opening and closing of the circuit contacts of an electronic circuit. When the relay contact is open (NO), the relay isn't energized with the open contact. However, if it is closed (NC), the relay isn't energize given the closed contact. However, when energy (electricity or charge) is supplied, the states are prone to change.

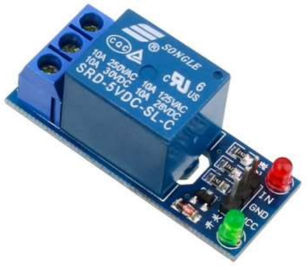


Figure 4.10: General Relay

Relays are normally used in the control panels, manufacturing and building automation to control the power along with switching the smaller current values in a control circuit. However, the supply of amplifying effect can help control the large amperes and voltages because if low voltage is applied to the relay coil, a large voltage can be switched by the contacts. If preventive relays are being used, it can detect overcurrent, overload, undercurrent, and reverse current to ensure the protection of electronic equipment. Last but not the least; it is used to heat the elements, switch on audible alarms, switch the starting coils, and pilots the lights.

DC Relays : Relays designed to operate with a DC input.

AC Relays: Relays designed to operate with an AC input.

Polarized Relays : DC relays that change status depending on the polarity of the control input current.

4.7 EXHAUST

One critical element of operations for greenhouse management is climate control, which includes providing optimum lighting, heating, ventilation, insulation, and greenhouse irrigation. Without optimal climate control, crops will generate much lower yields, decreasing the greenhouse's profitability. When plants receive optimal climate control, high yields are guaranteed.

Appropriate ventilation and cooling is key to achieving maximum greenhouse climate control. The primary purpose of ventilation is to control the heat and humidity inside the greenhouse so that plants can thrive with little risk from insects and disease. Adding exhaust fans and circulation fans can provide optimal airflow to keep commercial greenhouses from overheating. Exhaust fans pull hot air out of the greenhouse so that fresh air enters and cools the greenhouse, greenhouse fan and pad cooling system will greatly help to cool the greenhouse in the summer.

Figure 4.11: Exhaust Fan

The direct current fans, or DC fans, are powered with a potential of fixed value such as the voltage of a battery. Typical voltage values for DC fans are, 5V, 12V, 24V and 48V.

In contrast, the alternating current fans, or AC fans, are powered with a changing voltage of positive and of equal negative value. In general, this changing voltage has sinusoidal shape. Worldwide, the usual value of this sinusoidal voltage may vary in size and in frequency, such as 100VAC, 120VAC, 200VAC, 220VAC, 230VAC or 240VAC, and with frequency (cycles per second) of 50Hz or 60Hz.

In the past, big AC fans were typically less expensive compared to big DC fans. Today however, their price difference is negligible due to their payback advantage. We will try to point out differences between the above fan types, to help you choose and purchase the correct fan type for your application.

4.8 LEDS / HPS LAMPS

The three most common lamp types for high-intensity lighting are high pressure sodium (HPS) lamps, metal halide (MH) lamps**,** and LEDs. HPS lamps are the most common lamp type used in greenhouses today due to their high energy efficiency, moderately long bulb life and relatively small fixture size. However, HPS lamps have a moderately high installation cost, and their bulb life is strongly reduced by the number of on and off cycles. Metal halide lamps are less common but are especially used by garden centre retailers who use their retail space for production early in the season. They emit a whiter light and have a moderate bulb life. Similar to the HPS lamps, they also have a moderately high installation cost, and the bulb life is reduced with the number of starts.

High-intensity LEDs are slowly taking more market share in horticultural lighting. There continues to be large improvement in the technology and light intensity of the lamps. However, they have a relatively high investment cost, often emit directional light, and may have larger fixtures that will shade the crops below. LED fixtures vary widely with manufacturers as to their light pattern (directionality), intensity, relatability, durability, and energy efficiency. For example, some LEDs may emit more directional light, which requires a greater number of lamps per acre, while others actually have energy efficiencies lower than HPS lamps. There may be some situations in which the high investment costs of the LEDs would be advantageous: when they are used in locations with high electricity costs, rebates, low energy availability, or in year-round production facilities.

4.9 LCD DISPLAY

The LCD (Liquid Crystal Display) is a type of display that uses the liquid crystals for its operation.

The LCD display has a 16-pin interface.

The structure of the LCD is shown below:

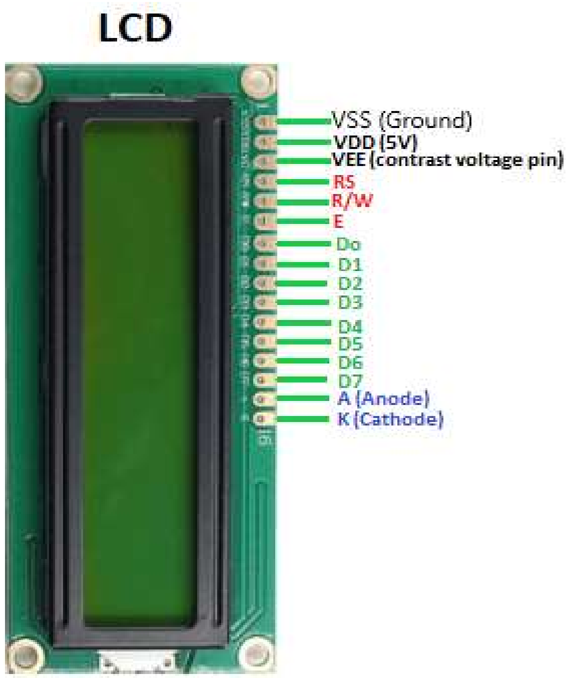


Figure 4.12: LCD Display

The Liquid Crystal Display has a parallel interface. It means that the microcontroller operates several pins at once to control the LCD display.

The 16-pins present on the LCD display are discussed below:

- **RS**

The **Register Select (RS)** pin controls the memory of the LCD in which we write the data. We can select either the **data register** or the **instruction registers**. The LCD looks for the upcoming instruction, which is present in the instruction register.

- **R/W**

The Read/Write pin selects the reading or writing mode.

- **E**

The **Enable** (E) mode is used to enable the writing to the registers. It sends the data to the data pins when the mode is HIGH.

- **D0 to D7**

These are eight data pins numbered as D0, D1, D3, D3, D4, D5, D6, and D7. We can set the state of the data pin either HIGH or LOW.

Pin 1 of the LCD is the **Ground** pin, and pin 2 is the **Vcc** or the voltage source pin.

The pin 3 of the LCD is the **VEE** or the **contrast pin**. For example, we can connect the potentiometer's output to the VEE and can adjust the contrast of the LCD.

The A and K pins are also called as **Backlight pins** (Bklt+ and Bklt-).

CHAPTER – 5

ARDUINO PROGRAMMING

ARDUINO PROGRAMMING

5.1 Structure:

The basic structure of the Arduino programming language is fairly simple and runs in at least two parts. These two required parts, or functions, enclose blocks of statements.

```
Void setup ( )
 {
 Statements;
}
Void loop ( )
 {
 Statements;
 }
```

Where setup ( ) is the preparation, loop ( ) is the execution. Both functions are required for the program to work. The setup function should follow the declaration of any variables at the very beginning of the program.it is the first function to run in the program, is run only once, and is used to set pinMode or initialize serial communication. The loop function follows next and includes the code to be executed continuously- reading inputs, triggering outputs, etc.

5.1.1 Setup ( ) :

The setup ( ) function is called once when your program starts. Use it to initialize pin modes, or begin serial. It must be included in a program even if there are no statements to run.

```
Void setup ( )
{
 pinMode ( pin, OUTPUT ) ; // sets the 'pin' as output
}
```

5.1.2 Loop ( ) :

After calling the setup ( ) function, the loop( ) function does precisely what its name suggests, and lops consecutively , allowing the program to change, respond , and control the Arduino board.

```
Void loop ( )

 { Digitalwrite (pin , HIGH ) ;// turns 'pin' on
Delay (1000); // pauses for one second

 digitalWrite (pin, LOW); // turns ‘pin’ off

delay (1000); // pauses for one second

}
```

Functions :

A function is a block of code that has a name and a block of statement that are executed when the function is called. The functions void setup( ) and void loop( ) have already been discussed and other built-in functions will be discussed. Custom functions can be written t perform repetitive tasks and reduce clutter in a program. Functions are declared by first declaring the function type. This is the type of value to be returned by the function such as int for an integer type function. If no value is to be returned the function type would be void. After type, declare the name given to the function and in parenthesis any parameters being passed to the function.

```
Type functionName

 (Parameters)

{

Statements:

}
```

The following integer type function delay Val ( ) is used to set delay value in a program by reading the value of potentiometer. It first declares a local variable v, sets v to the value of the potentiometer which gives a number between 0-1023, then divides that value by 4 for a final value between 0-255, and finally returns that value back to the main program.

```
IntdelayVal ( )

{

Int v ;

// create temporary variable ‘v’
V = analogRead ( pot ) ;
// read potentiometer value
V = 4;
Return v;
return final value
}
```

5.2 VARIABLES

A variable is a way of naming and storing a numerical value for later use by the program. As their namesake suggests, variable are numbers that can be continually changed as opposed to constants whose value never changes. A variable needs to be declared and optionally assigned to the value needing to be stored. The following code declares a variable called input variable and then assigns it the value obtained on analog input pin 2:

```
Int input Variable = 0 ;
// declares a variable and
// assigns value of 0
 InputVariable =analogRead ( 2 ) ; // set variable to value of
// analog pin 2
```

Input variable is the variable itself. The first line declares that it will contain n int, short for integer. The second line sets the variable to the value at analog pin 2. This makes the value of pin 2 accessible elsewhere in the code. Once a variable has been assigned, or re-assigned, you can test its value to see if it meets certain conditions, or you can use its value directly. As an example, to illustrate three useful operations with variables, the following code test whether the input Variable is less than 100, if true it assigns the value 100 to input Variable , and then sets a delay based on input Variable which is now a minimum of 100.

```
If (input Variable <100)
// tests variable if less than 100
{
```

```
Input Variable =100;

// if true assigns value of 100

}

Delay (inputVariable);

// uses variable as delay
```

5.2.1Variable scope:

A variable can be declared at the beginning of the program before void setup ( ), locally inside of function, and sometime within a statement block such as for loops. Where the variable I declared determines the variable scope, or the ability of certain parts of a program to make use of the variable.

A global variable is one that one can be seen and used by every function and statement in a program. This variable is declared at the beginning of the program before the setup ( ) function.

A local variable is one that is defined inside a function or as part of a for loop.it is only visible and can only be used inside the function in which it was declared.

Ensuring that only one function has access to its variables simplifies the program and reduces the potential for programming errors. The following example shows how to declare a few different types of variables and demonstrates each variable's visibility.

```
int value;

Void setup ( )

{

// no setup needed

}

Void loop ( )

{

For (inti=0;i<20 )

{
```

```
i++;

}

Float f;

}
```

5.3 FUNCTIONS

Segmenting code into functions allows a programmer to create modular pieces of code that perform a defined task and then return to the area of code from which the function was "called". The typical case for creating a function is when one needs to perform the same action multiple times in a program.

5.3.1 DIGITAL I/O

pinMode(pin, mode) :pinMode is used in void setup() to configure a specified pin to behave either as an INPUT or OUTPUT.

Example: pinMode(pin, OUTPUT); // set 'pin' to output mode Arduino digital pins default to inputs, so they do not need to be explicitly declared as inputs with pinMode(). Pin configured as INPUT are said to be in a high impedance state and pins configured as OUTPUT are said to be in a low impedance state. Short circuits on Arduino pins and excessive current can damage or destroy the output pin or damage the entire Atmega chip.

digitalWrite(pin, value) :

Outputs either high or low at(turn off or on) a specified digital pin. The pin can be specified as either a variable or constant (0-13).

Example: digitalWrite(pin, HIGH); //sets 'pin' to high

digitalRead(pin)

It reads the value from a specified digital pin with the result either HIGH or LOW. The pin can be specified as either a variable or constant (0-13).

Example: Value = digitalWrite(pin);

5.3.2 ANALOG I/O

analogRead(pin) : It reads the value from a specified analog pin with a 10-bit resolution. This function only works on the analog in pins (0-5). The resulting integer values range from 0 to 1023. Analog pins unlike digital ones do not need to be first declared as INPUT nor OUTPUT.

Example: Value = analogRead(pin);

// sets 'value' equal to 'pin'.

analogWrite(pin, value): It writes a pseudo analog value using hardware enabled pulse width modulation (PWM) to an output pin marked PWM. This is a hardware function, the pin will generate steady wave after a call to analogWrite in the background until the next call to analogWrite (or a call to digitalRead or digitalWrite on the same pin).a value of 0 generates a steady 0 volts output at the specified pin, a value of 255 generates a steady 5 volts output at the specified pin.

5.3.3 OTHER FUNCTIONS

TIME FUNCTIONS

delay (ms): It pauses the program for the amount of time as specified in milliseconds, where 1000 equals 1 second.

Example: delay(1000);   // waits for one second

millis() : Returns the number of milliseconds since the Arduino board began running the current program as an unsigned long value. This number will overflow (reset back to zero), after approximately 9 hours.

Example: value millis(): // sets 'value' equal to millis()

5.4 LIBRARIES

//sends the value of 'analogValue'

The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data.

Libraries are a collection of code that makes it easy for you to connect sensor, display, module, etc. There are several library functions like standard library functions, With the Arduino Ethernet Shield, this library allows an Arduino board to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones. The library supports up to four concurrent connections.

CHAPTER – 6

CIRCUIT DIAGRAM AND OPERATION

6.1 SYSTEM ARCHITECTURE

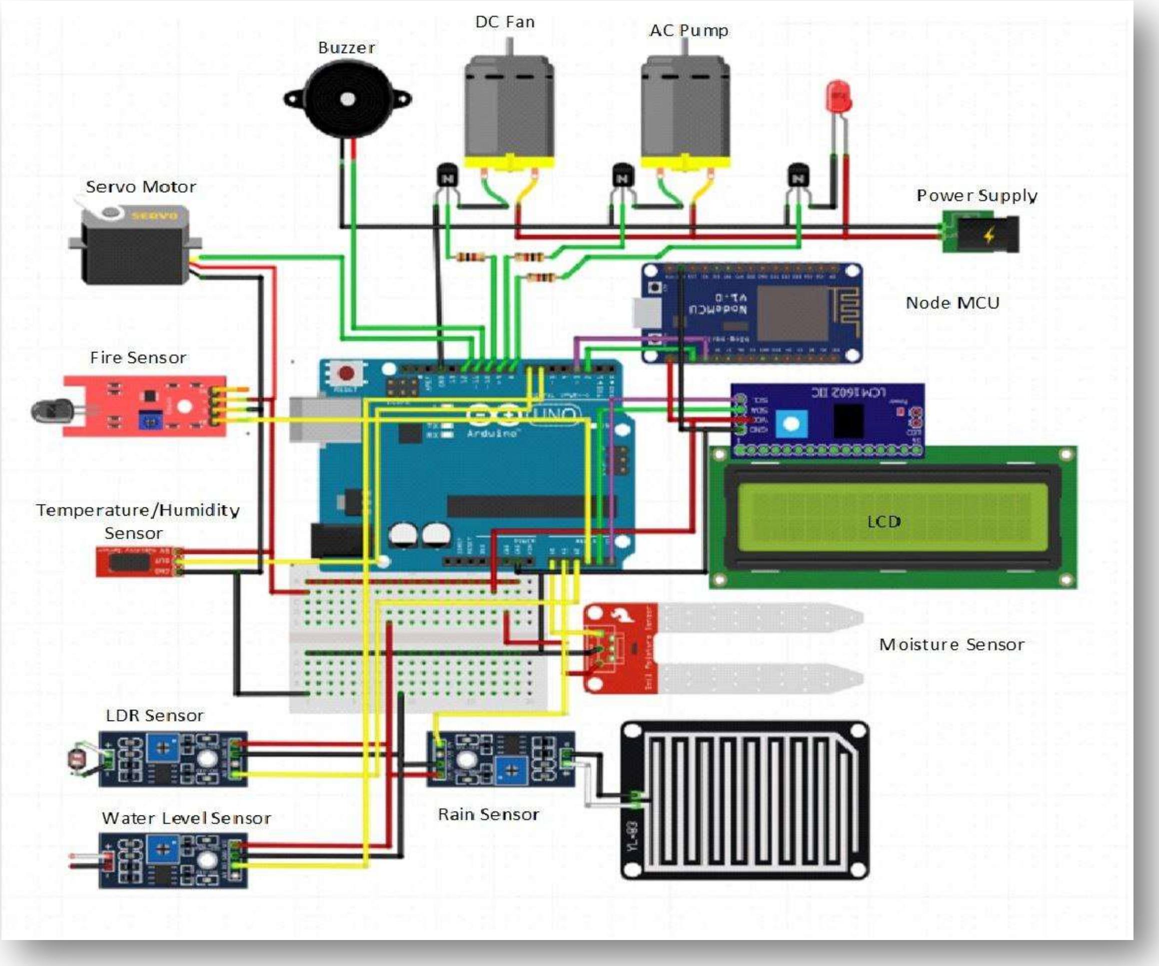


Figure 6.1: Architecture of the Model

The implemented greenhouse system consists of four modules, power module, sensor module, microcontroller module and output module. The monitoring system consists of the following sensors, LDR sensor, rain sensor, temperature/humidity sensor, soil moisture sensor, water level indicator and fire hazard sensor. We also implemented a parameter controlling section which consists of exhaust fan, LEDs, servo motor and submergible pump.

1. POWER MODULE: The power module we opted is regular power supply, in which we used an Adapter which supplies 12V power to the circuit. The 12V power supply is step-downed using two 5V transformers. Now this power is supplied to Arduino UNO, Node MCU, sensors used and the external equipment like servomotor, relay, DC motor and AC pump.

2. SENSOR MODULE: The sensor module comprises of several sensors such as soil-moisture, temperature & humidity, LDR, rain, water level and fire hazard sensors. These sensors are mounted on Arduino UNO and are used for sensing the soil and climatic changes from time to time.

3. MICROCONTROLLER MODULE: It is considered as heart of this architecture. Arduino UNO is used to compute the code. The Program for collecting the sensor data is written in embedded C language and sends that data to the server. The server-side program continuously monitors the sensors for stable conditions. If the values deviate from the threshold range a feasible solution was send to the output module.

4. OUTPUT MODULE: The data which is being collected from the sensors is received by the Arduino and then the output components react to the environmental changes to maintain the favorable conditions required in the greenhouse.

6.2 PROCEDURE

All the components are initialized by supplying the required power of +12V which is stepped down into two 5V power supply using step-down transformers. Six sensors, DHT11 sensor, LDR sensor, Soil moisture sensor rain sensor, water level indicator and fire hazard sensor are operated using a 5V power supply. The exhaust fan, servo motor, and LEDs are operated using another 5V supply which is connected to the Arduino. In this model have implemented a General Relay which receives the complete +12V supply and acts like an on/off switch to the output components (like submergible pump). An Ethernet is used to send environmental parameters to online web application (Blynk). All environmental parameters are stored in the database. So, the user can monitor and control parameters through android mobile application (Blynk IoT app). The user must log in the application by using username and password. DHT11 sensor is used to measure the temperature and humidity. High temperature leads to wilting of leaves and affect the ability of crops to produce effective yield. High humidity affects plants transpiration and photosynthesis process. When plants absorb water and minerals from the soil through their roots, transpiration assists these nutrients to the leaves. If the humidity is too high, the process slows down, preventing the absorption of nutrients. Without the release of moisture, plants lose their ability to cool themselves. An overheated greenhouse causes the soil to dry out quickly and use more water, to reduce the heat and to control humidity an exhaust fan is used to blow out the warm air.

A soil moisture sensor is used to monitor the moisture level of soil so that if there is a deflection in the level of threshold limit it signals the Arduino to automatically activate the pump to supply water to the plants. When the moisture in soil comes to the normal range, the device turns off the water pump timely. LDR sensor is used to measure the intensity of light around the surrounding environment of the greenhouse. As the light quantity decreases the photosynthetic process decreases. When light intensity is lower than a defined level, the system activates the LEDs in order to provide sufficient light to the greenhouse. The LEDs are turned off when there is sufficient sunlight, and the threshold values are satisfied. This device has been optimized to make use of rainwater when available and when the soil moisture levels aren't meeting the required thresholds in rainy seasons. Instead of using the water from the reserve tank it efficiently uses the rainwater to fulfil the moisture content in the soil. The rainwater reaches the plants as the roof is retracted using a servo motor whenever the rain sensor signals the servo motor. The water level sensor works in coordination with other sensors and the pump to indicate the level of water present in the reserve tank. If the water is less an alert is displayed on the web application to refill the tank and then according to the requirements the pump is activated based on its readings.

A General Relay is acts like an on/off switch to the submergible pump. In addition to these sensors this device has been equipped with a safety measure sensor i.e., the fire hazard sensor in order to detect any accidental fire outbreaks. The buzzer is connected to the Fire hazard sensor in order to notify us of any fire breakouts or short circuits inside the greenhouse. If the fire hazard sensor detects any sudden raise in temperature or if wavelength of the light source greater than 760nm then it sends a signal to the buzzer and an alarming sound is activated.

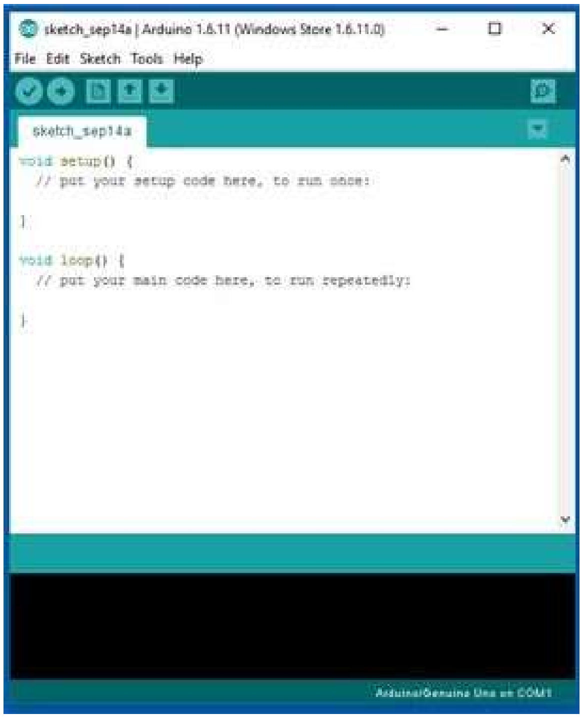
The following environmental parameters which are continuously monitored and maintained are displayed using an LCD screen inside a greenhouse and also the data from the sensors is transmitted by the Node MCU to the online web application (Blynk) and also the user can view these parameters on the mobile application (Blynk IoT). The device provides an ease to timely view data and conditions inside the greenhouse wirelessly.

CHAPTER – 7

ARDUINO IDE SOFTWARE

7.1 INTRODUCTION

The Andune Integrated Development Environment or Arduing Software HO text editor for writing code, a message area, a fext console, a trother with button for con functions and a series of menus. It connects to the Arduino hardware to programs and communicate with them.


sketch_sep14a | Arduino 1.6.11 (Windows Store 1.6.11.0)
File Edit Sketch Tools Help
sketch_sep14a
Arduino/Genuino Uno on COM1

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 7.1: Arduing Software HO

7.2WRITING SKETCHES:

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in text editor and the editor has features for cutting, pasting, searching, replacing text. The message area gives feedback while saving and uploading and also displays errors. The console displays text output by Arduino Software (IDE), including complete error messages and other information. The toolbar buttons allow to verify and upload programs, create, open, and save sketches, and open the serial monitor.

7.3 STEPS TO DUMP THE PROGRAM INTO THE ARDUINO UNO BOARD

1. Select the Arduino UNO board from tools>board>Arduino UNO.

2. Select the serial port to communicate tools>port>serial.

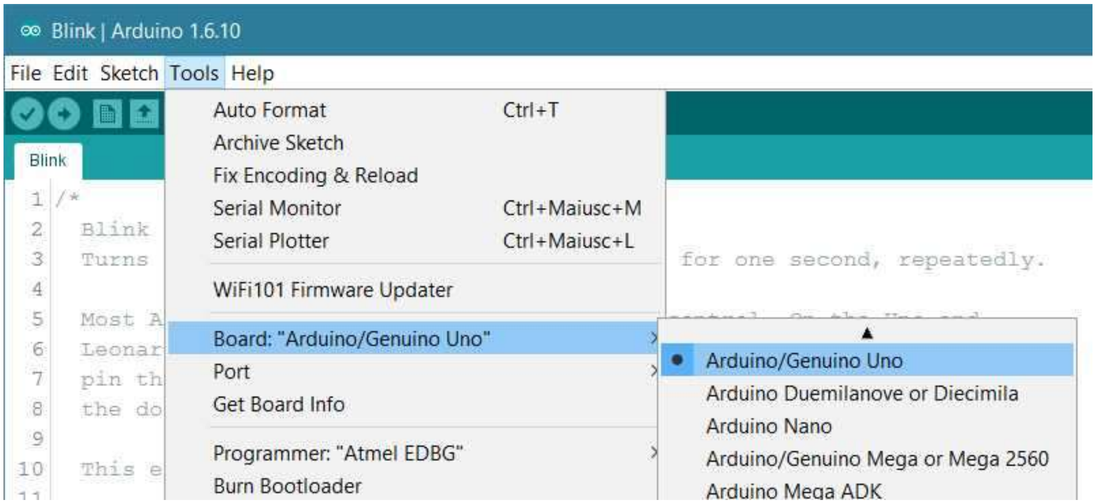
3. The sketches(code) in your sketchbook can be opened from the File>sketchbook or create a new file and write the program.

4.Checks your code for errors compiling it using verify.

5. Save your program/sketch using save.

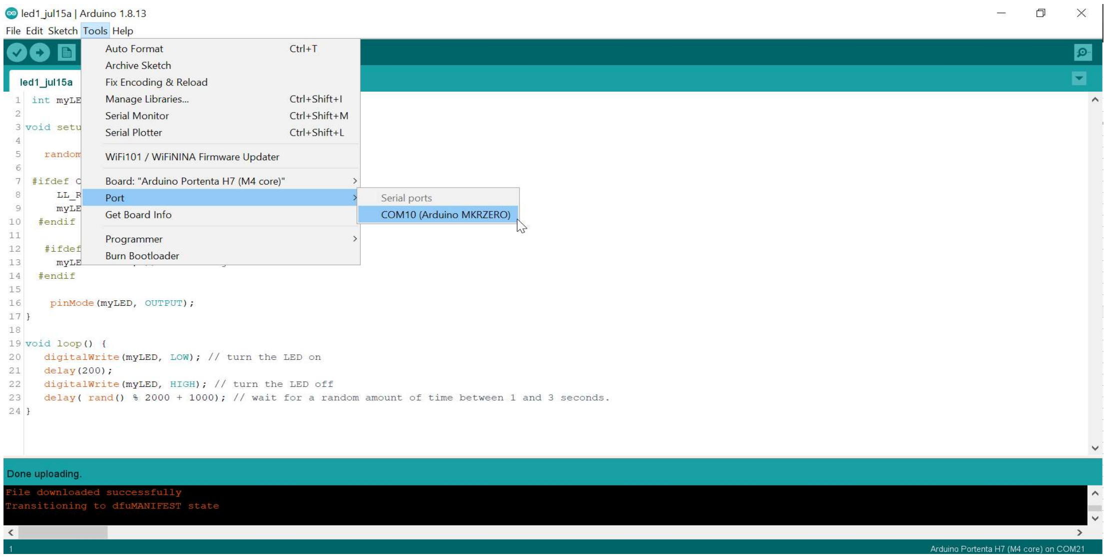
6.Upload the sketch to Arduino using upload.

7 Open the serial monitor in case of any outputs to be displayed on it by serial monitor.



```
/*
  Blink
  Turns                    for one second, repeatedly.

  Most A
  Leonar
  pin th
  the do

  This e
```

Figure 7.2: Arduino Board selection Tool options



led1_jul15a | Arduino 1.8.13
File Edit Sketch Tools Help
led1_jul15a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Portenta H7 (M4 core)"
Port
Get Board Info
Programmer
Burn Bootloader
Serial ports
COM10 (Arduino MKRZERO)

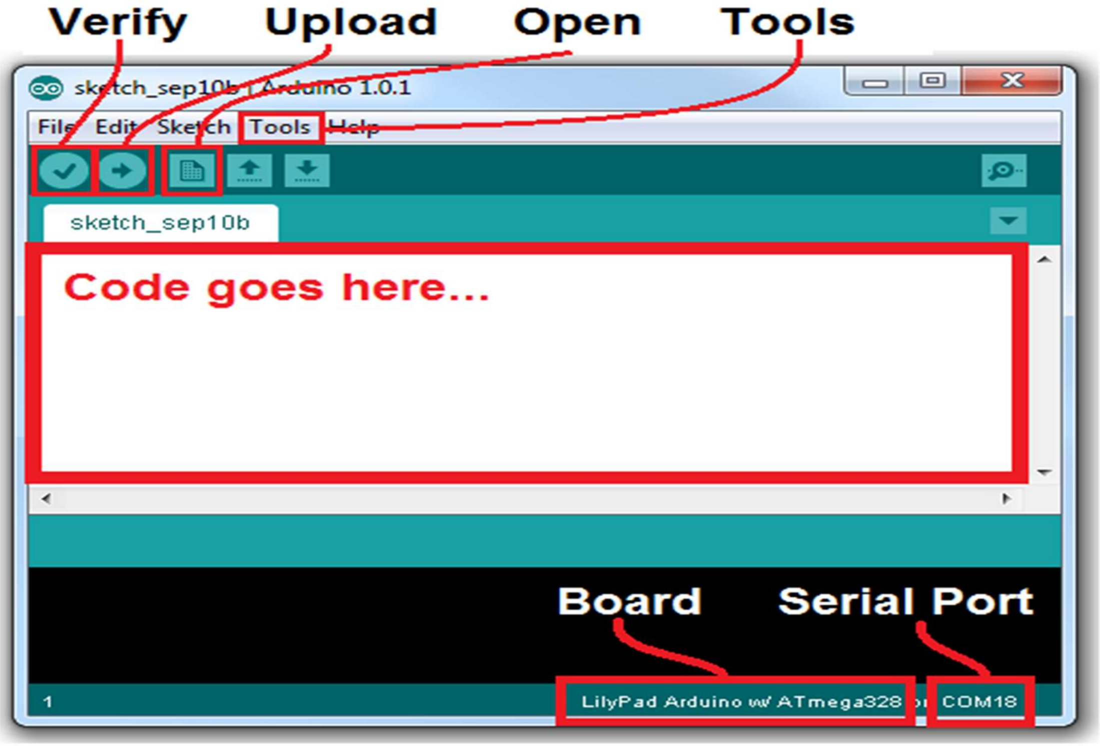
```
int myLE

void setu

   random

 #ifdef C
     LL_R
     myLE
  #endif

   #ifdef
     myLE
  #endif

    pinMode(myLED, OUTPUT);
}

void loop() {
   digitalWrite(myLED, LOW); // turn the LED on
   delay(200);
   digitalWrite(myLED, HIGH); // turn the LED off
   delay( rand() % 2000 + 1000); // wait for a random amount of time between 1 and 3 seconds.
}
```

Done uploading.

```
File downloaded successfully
Transitioning to dfuMANIFEST state
```

Arduino Portenta H7 (M4 core) on COM21

Figure 7.3: Port selection



LilyPad Arduino w/ ATmega328 on COM18

Figure 7.4: Representation of Controllable options in Arduino Software

7.4 ADVANTAGES

1. Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help.

2. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

3. Uses high speed serial communication.

4. Allows to manage sketches with more than one file. These can be norm1al Arduino

code files(no visible extension), c files(.c extension), c++ files(.cpp extension), or header files (.h)

5. Library functions can be uploaded to board

6. Displays an error message in case of any mistakes in the sketches.

CHAPTER – 8

BLYNK APPLICATION

8. BLYNK APPLICATION

Blynk allow us to create applications and then use it to control Arduino board connected to a PC with internet access, from anywhere in the world, (for instance, control LED, servos, receive data, etc), with a smartphone. The connection can also be established by Bluetooth between smartphone and Arduino board, but this will be not presented in this work. Blynk can be download from Google play store (for Android). & app store (for apple), providing us the dashboard as well as the connectivity to Arduino, (it is a virtual connectivity). Programming Blynk is very simple, push and drag widgets form the tools bar and allocating them pins on Arduino board. For such project can be used an ordinary Arduino board, without internet shield, connected to a PC with internet access and a smartphone. The role of PC is to ensure the connectivity of the Arduino board to the internet and to upload the Arduino code. For this purpose, it is necessarily to install Blynk library on the PC and made some settings. The Arduino codes used for Blynk have the same structure as ordinary codes but contain specific parts to communicate with Android devices.

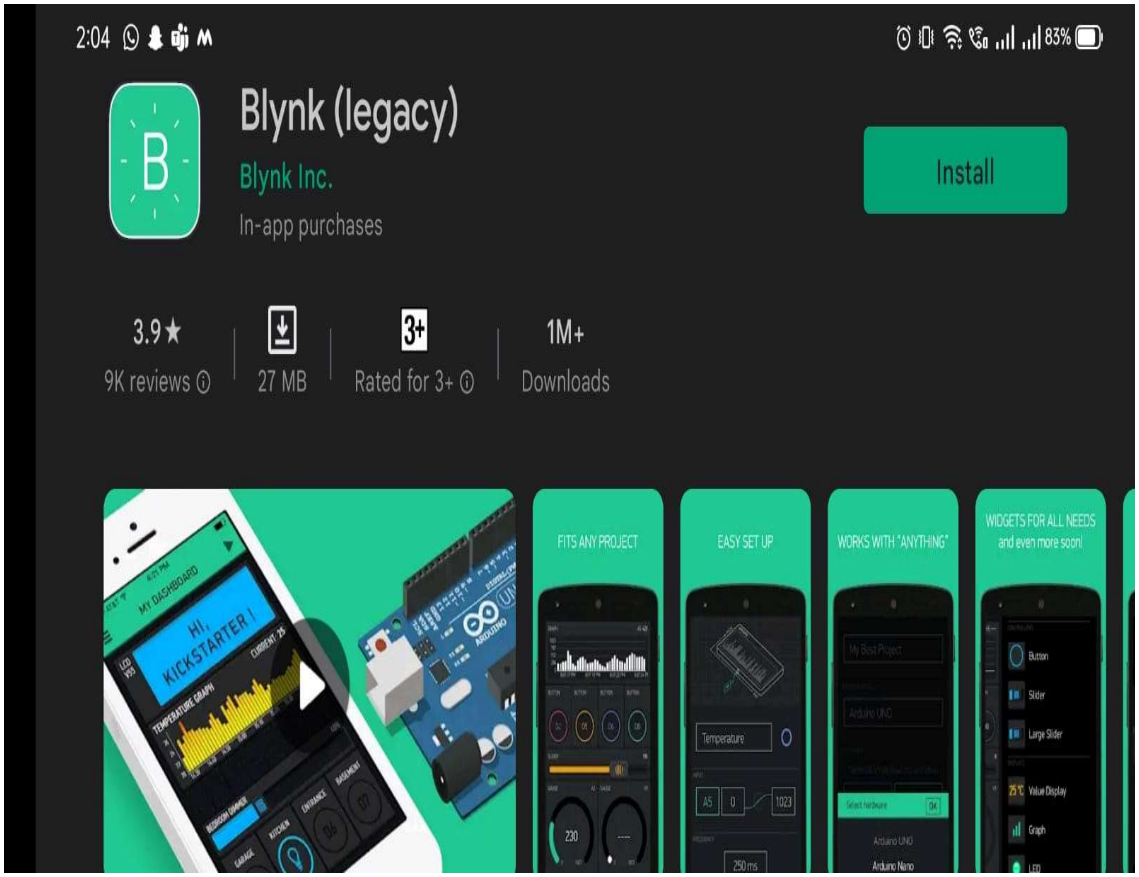


Figure 8.1: Blynk app for Android

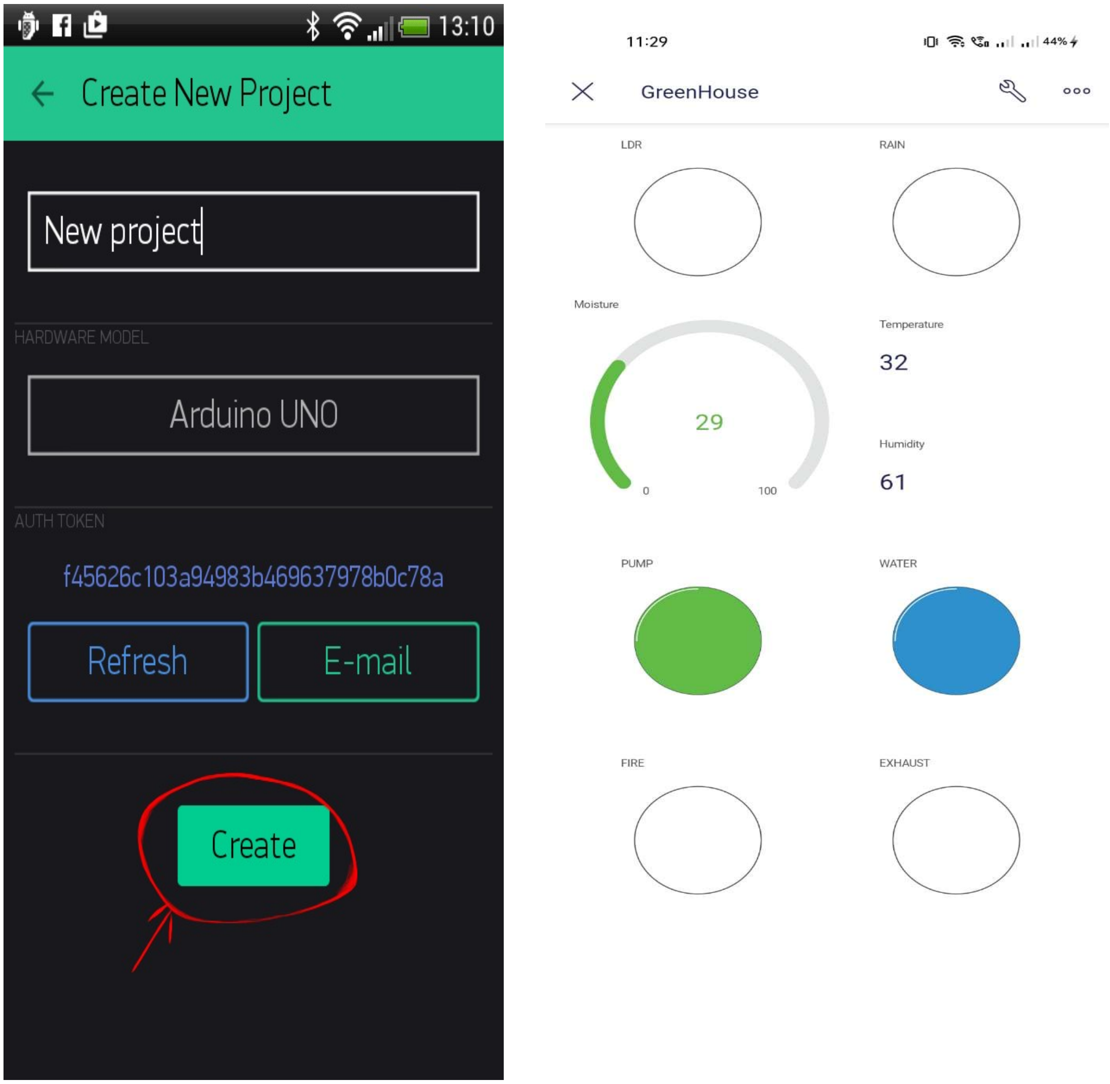
To create our first project, we need to follow many steps:

1. Install Blynk app on your smartphone and create your simplest application, for instance control one led.

2. Download Blynk v0.3.1 library and install on your PC. Set the right port of the PC to communicate with the Arduino board, (very important), from the script of the Blynk library.

3. Upload the Arduino code, (from the examples presented in this work, or other).

4. Run the Blynk on the smartphone



13:10

Create New Project

New project

HARDWARE MODEL

Arduino UNO

AUTH TOKEN

f45626c103a94983b469637978b0c78a

Refresh E-mail

Create

11:29 44%

GreenHouse

LDR

RAIN

Moisture 29 (0 – 100)

Temperature 32

Humidity 61

PUMP

WATER

FIRE

EXHAUST

Figure 8.2: Working of Blynk app

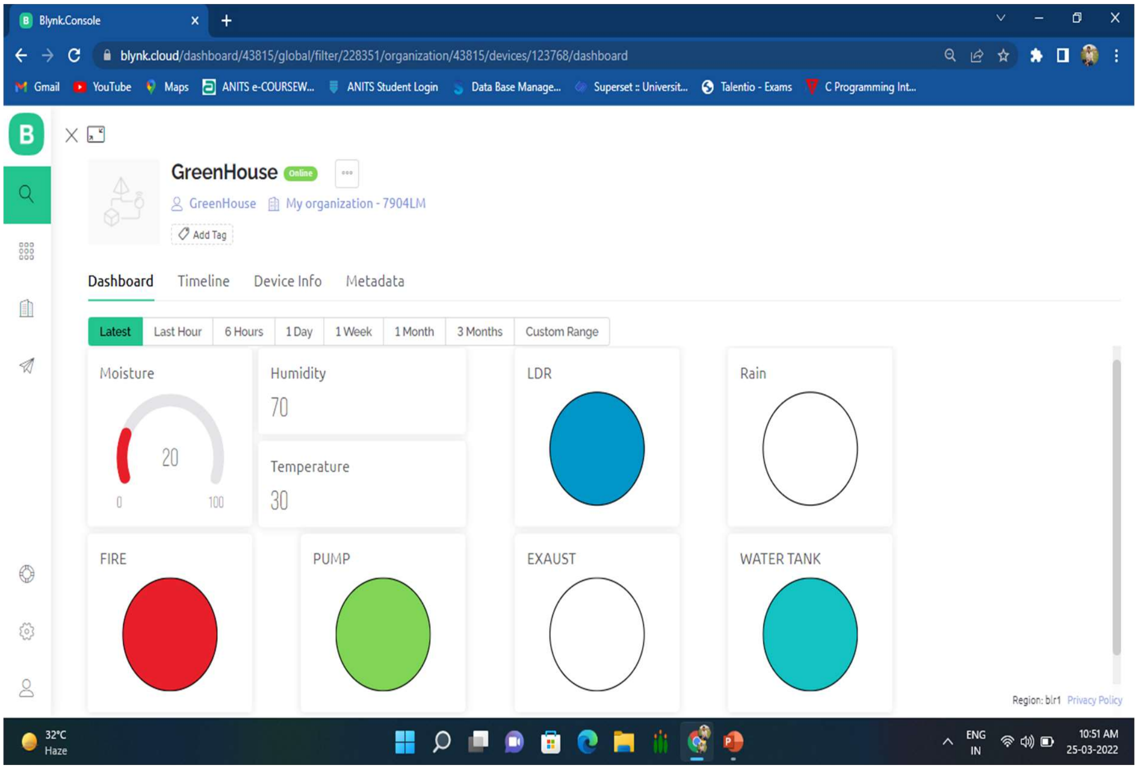
CHAPTER-9

RESULTS

RESULTS

After successful connection to the Blynk server we can monitor the changes in the parameters that are required for the growth of plant. This data is retrieved from the sensors which are continuously monitored and are mounted on the Arduino UNO.

The following figure shows the output in the Blynk server:



GreenHouse Online

Moisture 20

Humidity 70

Temperature 30

Figure 9.1: Blynk Web Interface

From the figure we can observe the Moisture level of the soil, Percentage of Humidity and Temperature. The blue colour of the Water tank indicates that the Water tank contains enough water.

Also, from the figure we can conclude the state of the sensors

LDR – ON

Rain Sensor - OFF

Fire – ON

Pump - ON

Exhaust – OFF

CONCLUSION

The IoT based greenhouse monitoring and controlling system is designed. The main sensors used in this model are Soil Moisture sensor, DHT11 sensor, LDR sensor, Rain sensor, Water Level Indicator sensor and fire sensor involved in controlling and monitoring environmental parameters. This system mainly developed using Arduino UNO and Node MCU. Ethernet is also used to send all the parameters to online web application (Blynk) and mobile phone (Blynk IoT App). This model reduces the power consumption, maintenance, and complexity. This project is mainly developed to be used in agricultural fields, in nursery and in greenhouse-floriculture.

FUTURE SCOPE

Future work using this system equipped with modern sensors, smart greenhouses automatically capture and deliver information 24/7 on the surroundings and crop. By unlocking massive crop insights, a smart greenhouse allows us to minimize labor work, improve efficiency in resource and chemical use while optimizing yield rates. The system can be used to maintain ideal micro-climatic conditions, control infection, and avoid disease outbreak in plants and improve security of the greenhouse. Rather than switching ON/OFF the device the power module can be enhanced and improved to solar charging using solar panel and rechargeable batteries to avoid the manual charging of power source. For example, this model can be easily implemented in small nurseries to monitor plants and also small scale farmers can easily afford this for seasonal crops as it is cost efficient and provides easy monitoring from web application.

REFERENCES

1. **George Mois, Teodora Sanislav and Silviu C. Folea,** “A Cyber Physical System for Environmental Monitoring”, IEEE Transactions on Instrumentation and Measurement, Vol. 65, No. 6 June 2016.
2. **Ullas S Patel, Saiprasad, Shravankumar and Veerabhadra K J,** “Green House Monitoring and Controlling Using Android Mobile App”, International Journal of Combined Research & Development (IJCRD), Vol. 5, Issue 5, May 2016.
3. **Remya Koshy, M D Yaseen, Fayis K, Nisil Shaji, Harish N J and Ajay M,** “Greenhouse Monitoring and Control Based on IOT Using WSN”, ITSI Transactions on Electrical and Electronics Engineering, Vol. 4, Issue 3,2016.
4. **Bulipe Srinivas Rao, Dr. K. Srinivasa Rao and N. Ome,** “Internet of Things (IOT) Based Weather Monitoring System”, International Journal of Advanced Research in Computer and Communication Engineering, , Vol. 5, Issue 9, September 2016.
5. **Amber Katyal, Ravi Yadav and Manoj Pandey,** “Wireless Arduino Based Weather Station”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 4, April 2016.
6. **Ashwini Bhosure, Mayur Bhosure and Rakeshkumar Sharma,** “Web Based Greenhouse Monitoring and Controlling System using Arduino Platform”, International Journal of Scientific Engineering and Applied Science (IJSEAS), Vol. 2, Issue 2, February 2016.
7. **Joaquin Gutierrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Angel Porta Gandara,** “Automated Irrigation System Using a Wireless Sensor Network and GPRS Module”, IEEE Transactions on Instrumentation and Measurement, Vol. 63, No. 1 January 2014.
8. **Abdullah Tanveer, Abhishek Choudhary, Divya Pal, Rajani Gupta and Farooq Husain,** “Automated Farming using Microcontroller and Sensors”, International Journal of Scientific Research and Management Studies (IJSRMS), Vol.2, Issue 1, 2015.
9. **Choppara Manendra Babu and S. Saidarao,** “Modern Agricultural Management and Greenhouse Monitoring System based on Wireless Communication”, International Journal of Engineering Research and Technology (IJERT), Vol. 5, Issue 2, February 2016.
10. **Dipak V. Sose and, Dr. Ajij D. Sayyad,** “Weather Monitoring Station: A Review”, International Journal of Engineering Research and Application, Vol. 6, Issue 6, June 2016.