

IOT BASED ON AGRICULTURE MONITORING SYSTEM

Submitted in partial fulfilment of the requirements for the award of

**Bachelor of Engineering Degree in
Electronics and Communication Engineering**

By

MURALA AJAY BABU (38130143)

PENUGONDA MANOJ KUMAR (38130162)



**DEPARTMENT OF ELECTRONICS & COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAICHENNAI – 600119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MURALA AJAY BABU (38130143)**, and **PENUGONDA MANOJ KUMAR(38130162)** who have done the project work as a team who carried out the project entitled "**IOT BASED ON AGRICULTURE MONITORING SYSTEM**" under mu supervision from September 2021 to April 2022.

Internal Guide

Dr. N.R. KRISHNAMOORTHY, M.E., Ph.D.

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **MURALA AJAY BABU (38130143)**, and **PENUGONDA MANOJ KUMAR (38130162)** hereby declare that the Project Report entitled “**IOT Based ON AGRICULTURE MONITORING SYSTEM**” done by us under the guidance of **Dr. N.R. KRISHNAMOORTHY M.E., Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1) MURALA AJAY BABU

DATE: 07/05/2022

2) PENUGONDA MANOJ KUMAR

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA. M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI. M.E., Ph.D.**, Head of the Department, Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. N.R. KRISHNAMOORTHY M.E.,Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This IOT based Agriculture monitoring system makes use of wireless sensor networks that collect data from different sensors deployed at various nodes and send it through the IOT Technology. This smart agriculture using IOT system is powered by Arduino with Temperature sensor, Moisture sensor, water level sensor, DHT11 Sensor, water motor, buzzer and node MCU. When the IOT based agriculture monitoring system starts it checks the water level, humidity and moisture level. It sends SMS alert on the phone about the levels. Sensors sense the level of water if it goes down, it automatically starts the water pump. If the temperature goes above the level, water motor starts. When any animal reaches the agricultural land, the buzzer we attached to the arduino will make the sound alert system. So that the animal will leave the land.

Finally all data will be transferred to the IOT cloud system using node MCU module. In 1995, “thing to thing” was coined by BILL GATES. In 1999, IoT (Internet of Things) was come up by EPC global. IOT interconnects human to thing, thing to thing and human to human. The goal of IoT is bring out a huge network by combining different types connected devices. IoT targets three aspects Communication, automation, cost saving in a system. IOT empowers people to carry out routine activities using internet and thus saves time and cost making them more productive. IOT enables the objects to be sensed and/or controlled remotely across existing network model. IOT in environmental monitoring helps to know about the air and water quality, temperature and conditions of the soil, and also monitor the intrusion of animals in to the field. IOT can also play a significant role in precision farming to enhance the productivity of the farm.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1.	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 NEED OF AUTOMATIC IRRIGATION	2
	1.3 EXISTING SYSTEM	3
	1.4 PROBLEM STATEMENT	3
	1.5 PROPOSED SYSTEM	3
	1.6 ADVANTAGES	4
	1.7 MOTIVATION	4
	1.8 OBJECTIVE	6
2	LITERATURE SURVEY	7
	2.1 EXPERIMENTATION AND RESULTS	18
	2.2 WORKING	20
3	PROPOSED METHODOLOGY	21
	3.1 BLOCK DIAGRAM	22
	3.2 HARDWARE REQUIREMENTS	23
	3.3 SOFTWARE REQUIREMENTS	23
4	HARDWARE DESCRIPTION	24
	4.1 ARDUINO	24
	4.2 SUMMARY	26
	4.3 PIN CONFIGURATION	26
	4.4 COMMUNICATION	29
	4.5 ARDUINO SOFTWARE	31
	4.6 NODE MCU	36

CHAPTER NO.	TITLE	PAGE NO.
5	SOFTWARE DESCRIPTION	49
	5.1 ARDUINO SOFTWARE	49
	5.2 PROTEUS	50
6	CONCLUSION	65
	REFERENCES	66

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	Block Diagram	22
4.1	Arduino	24
4.2	ESP8266 Development board	38
4.3	NodeMCU Development Board	39
4.4	DHT11 sensor	45
4.5	"Start" and "Response" signals	46
4.6	Timing difference for transmitting "1s" and "0s"	47
4.7	MUC start signal and DHT signals response signal	47

LIST OF ABBREVIATIONS

AIT	-	Agricultural Information Technology
DIP	-	Dual In-Line Package
GUI	-	Graphical User Interface
IoT	-	Internet of Things
RIMS	-	Remote Intelligent Monitoring System
SV	-	Synthetic Variables
VSM	-	Virtual System Modelling
WIFI	-	Wireless Transmission
WSN	-	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

The Internet of things (IOT) is remodeling the agriculture enabling the farmers with the wide range of techniques such as precision and sustainable agriculture to face challenges in the field. IOT technology helps in collecting information about conditions like weather, moisture, temperature and fertility of soil, Crop online monitoring enables detection of weed, level of water, pest detection, animal intrusion in to the field, crop growth, agriculture. IOT leverages farmers to get connected to his farm from anywhere and anytime. Wireless sensor networks are used for monitoring the farm conditions and micro controllers are used to control and automate the farm processes. To view remotely the conditions in the form of image and video, wireless cameras have been used. A smart phone empowers farmer to keep updated with the ongoing conditions of his agricultural land using IOT at any time and any part of the world. IOT technology can reduce the cost and enhance the productivity of traditional farming.

1.1 OVERVIEW

Agriculture is major source of income for the largest population in India and is major contributor to Indian economy. In past decade it is observed that there are not much crop development in agriculture sector. Food prices are continuously increasing because crop rate declined. There are number of factor which is responsible for this it may be due to water waste, low soil fertility, Fertilizer abuse, climate change or diseases etc. It is very essential to make effective intervention in agriculture and the solution is IOT in integration with wireless sensor network. Internet of things (IOT) is a method of connecting everything to the internet- it is connecting object or things(such as car, home, electronic devices, etc. ...) which are previously not connected with each other main purpose of IOT is ensuring delivery

of right information to right people at right time. In agriculture irrigation is the important factor as the monsoon rain falls are unpredictable and uncertain.

1.2 NEED OF AUTOMATIC IRRIGATION

- Simple and easy to install and configure.
- Saving energy and resources, so that it can be utilized in proper way and amount.
- Farmers would be able to smear to right amount of water at the right time by automatic irrigation.
- Avoiding irrigation at the wrong time of day, reduce runoff from overwatering saturated soils which will improve crop performance.
- Automated irrigation system uses valves to turn motor ON and OFF. Motors can be automated easily by using controllers and no need of labour to turn motor ON and OFF.
- It is precise method for irrigation and a valuable tool for accurate soil moisture control in highly specialized greenhouse vegetable production.
- It is time saving, the human error elimination in adjusting available soil moisture levels.

This project uses IOT technology in agriculture, gathering crops growth environmental parameters in a fixed place to help farmers find problems in time. Agriculture experts give guidelines with specific information to increase the farmer's income and help them in the prevention and control of crop diseases and pests. Through the custom development of mobile phone apps, it has been implemented with agriculture technology promotion and expert online FAQ. The system development composes three parts: The server, Android client and PC client to achieve scalability, high reliability, security, compatibility of technical requirement.

1.3 EXISTING SYSTEM

- Existing systems based on the different technologies and also focuses on generic automated irrigation system based on WSN with GSM-ZigBee for remote monitoring and controlling devices.
- The objective is to make use of wireless sensor network and communication technology such as ZigBee and GSM in industrial field to make low-cost automated irrigation system to monitor the condition of the soil and to lower the energy consumption.
- The system helps the farmer to monitor and control the parameters of the soil such as air temperature, humidity, soil moisture.
- At any abnormal condition, the farmer is informed and will be able to take actions remotely by using GSM. Due to its lower energy consumption and low cost, the system has the potential to be useful in semiarid or arid areas.

1.4 PROBLEM STATEMENT

- The automation is not used for operations of the sensors and motors
- Monitoring is not possible in our existing system
- High cost for installation of the modules
- Hard implementation
- Low accuracy compared to the proposed system.

1.5 PROPOSED SYSTEM

- The whole Arduino, Node MCU, Soil Moisture sensor, water level sensor, DHT 11 Sensor. Soil Moisture sensor measures moisture content of the soil.
- When soil moisture sensor goes low the water motor will be on and exceeds a defined level, the water motor will off automatically. When

the water level is low in the well it will automatically detected by ultrasonic sensor and the details about the water in the well are updated in a webpage.

- The user can monitor and control parameters through webpage. This device is very much helpful to the farmers to monitor and control environmental parameters at their farms. The farmers need not to go their farms.
- All sensor's details uploaded to the IOT cloud storage system with updated datas.

1.6 ADVANTAGES

- The project is built from easily available and reasonably priced components. Therefore, the cost is reasonable and maintenance is easy.
- Using this project, the status of crops can be viewed remotely on a smartphone or laptops using the internet. This helps to keep the farmer up to date even when he is away.
- The project keeps the farmer updated regarding the status of the crop via SMS notifications. This keeps the farmer updated and conscious about the status of his crop.
- Additional agricultural, chemical and weather-related sensors can be added to the system in order to achieve more effective and accurate monitoring from the smart system.

1.7 MOTIVATION

India, being a country of farmers mainly focuses on the improvements in the cultivation and the methods of farming. Indian agriculture sector has growth of 2% for the year 2019-20 as compared to 5.1% of 2018-19. In the recent budget Rs. 2.83 Lac Cr is allocated to the agriculture and the allied fields. The Indian agriculture sector despite of the several uncertainties and the environmental challenges is

improving in terms of the farming methodologies implemented. The country is mainly depended on the agriculture and allied sector for employment. The uncertain rainfall and floods are the reasons behind less production of the various agriculture products on the other hand the sector is also affected due to less water availability in many regions of country. Despite of these challenges there are many examples in the country where the farmers have broken the records of per hector production every year. Indian farmers are becoming smart, although the percentage is less but now a days farmer have started implementing the smart systems for better results. The technology is developed to control the utilisation of resources and improving the per hector cultivation. Farmers are facing the situation where the effects of atmospheric conditions are severe on the crop. On the other hand the improper planning of the type of crop to be cultivated may lead to the low returns from market. Many of the farmers are illiterate, and Indian Government is guiding them in order to understand the farming operations for improvement in the cultivation of crop. Government now a days also providing the soil card for the farmers, it helps in understanding the situation of soil and the type of crop for which the soil is most suitable. The Smart systems for monitoring the crop and soil help the farmer to enhance the cultivation. The IoT based system can be implemented to monitor the crop and soil in order to control the use of resources depending upon the requirement.

Many times the major concern with farming is the improper management of water. Water management with IoT can not only improves per hector production but also enables the cultivation twice or thrice per year. The field monitoring with drone cameras helps the farmers to understand the situation of the crops over the longer areas. The sensors are useful for the monitoring of soil condition such as humidity and the moisture.

Agriculture sector in India is facing several problems with environmental hazards. The situations like floods, uneven rain, lack of technology to control the farming operations and conventional irrigation and other crop related operations etc. The researchers are working on providing the solution for the agriculture related operation with automation and IoT. IoT has made it possible to monitor and control the crop for better cultivation. The sensors help in sensing the exact situation of the

crop and the soil. The various parameters will be checked and compared with the standards for controlling the flow of resources such as water.

1.8 OBJECTIVE

The objectives behind the study carried out are:

- To study the various agriculture monitoring systems available.
- Understand the requirements of the IoT based agriculture monitoring system
- Develop the framework for agriculture monitoring system development.
- Selection of the proper components for development of the system.

CHAPTER 2

LITERATURE SURVEY

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

Liu Dan [2], Joseph Haule, Kisangiri Michael [3] and Wang Weihong, Cao Shuntian [38] carried out experiments on intelligent agriculture greenhouse monitoring system based on ZigBee technology. The system performs data acquisition, processing, transmission and reception functions. The aim of their experiments is to realize greenhouse environment system, where the of system efficiency to manage the environment area and reduce the money and farming cost and also save energy. IOT technology here is based on the B-S structure and cc2530 used like processing chip to work for wireless sensor node and coordinator. The gateway has Linux operating system and cortex A8 processor act as core. Overall the design realizes remote intelligent monitoring and control of greenhouse and also replaces the traditional wired technology to wireless, also reduces manpower cost.

Joseph Haule [3], Dragoş Mihai Ofrim, Bogdan AlexandruOfrim and Dragoş Ioan Săcăleanu [18] have proposed an experiment that explains the use of WSN used in automating irrigation. Irrigation control and rescheduling based on WSN are powerful solutions for optimum water management through automatic

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

Vijay Kumar [4], Lin Zhang, Min yuan, Deyi Tai, Xia Oweixu, Xiang Zhan, Yuanyuan Zhang [13] studied the work of rural farming community that replaces some of the traditional techniques. The sensor nodes have several external sensors namely leaf wetness, soil moisture sensor, soil pH, atmospheric pressure sensors attached to it. Based on the soil moisture sensor the mote triggers the water sprinkling during the period of water scarcity and switches off after adequate water is sprinkled. This results in water conservation and soil pH is sent to the base station and in turn base station intimates the farmer about soil pH via SMS using GSM model. This information helps the farmers to reduce quantity of fertilizers used. A development of rice crop monitoring using WSN is proposed to provide a helping hand to farmers in real time monitoring and increasing the rice production. The automated control of water sprinkling and ultimate supply of information is implemented using wireless sensor network.

G. Nisha [5], Chun-ling Fan, Yuan Guo [10] proposed a wireless sensor based automated irrigation system to optimize water use for agricultural purpose. The system consists of distributed wireless sensor network of soil moisture, and temperature sensors mounted in the crop field. Zigbee protocol is used to handle the sensor information and water quantity programming using algorithm with threshold values of the sensors sent to a micro controller for irrigation system. Data inspection is done using by using solar panel and cellular internet interface. A

wireless camera is fixed in crop field to monitor the disease area using image processing technique.

Meng Ji-hua [6] conducted a research on growth of cereal crop seedlings, as well as the status and trend of their growth. This paper introduced the design, methods used and implementation of a global crop growth monitoring system, which satisfies the need of the global crop monitoring in the world. The system uses two methods of monitoring, which are real-time crop growth monitoring and crop growing process monitoring. Real-time crop growth monitoring could get the crop growing status for certain period by comparing the remote sensed data (NDVI, for example) of the period with the data of the period in the history (last year, mostly). The differential result was classified into several categories to reflect the condition at difference level of crop growing. In this system, both real-time crop growth monitoring and crop growing process monitoring are carried out at three scales, which are state (province) scale, country scale and continent scale. Global crop growth monitoring system was found in this design and built a system that can monitor the global crop growth with remote sensing data. The system showed the characteristics of fast, effective, high credibility and operational in its run.

Alan Main-waring [7], A. Sivasankari, S. Gandhimathi[36] have provided an in-depth study of applying wireless sensor networks to real-world habitat monitoring. A set of system design requirements are developed that cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management. To evaluate this implementation, have deployed an initial prototype network at the James San Jacinto Mountains Reserve (JMR) in Idyllwild, California. JMR is a 29-acre ecological preserve, representing just one of the University of California System Natural Reserve System's 34 land holdings. JMR climate is different from GD and weather changes can exists for long time. The data collection can be made easy from previously inaccessible using a micro-measurement scale.

Lei Xiao [8] Fiona Edwards Murphy, Emanuel Popovici, adraig Whelan, and Michele Magno [21] Proposed agriculture monitoring system using wireless sensor network (WSN). The conditions can be monitored in real time are

temperature, light intensity, and humidity. The experiment involves the hardware and software design of the built modules, network topology and network communication protocol with the challenges. Design explains how the node can achieve agricultural condition information collection and transmission. The system is compact in frame work, lightweight, good in performance and operation. It improves the agricultural production efficiency automatically.

Ling-ling LI [9], Wen-Yao Zhuang, Miguel Costa Junior, Pedro Cheong, Kam-Weng Tam [12] have proposed system uses ZigBee technology. This research deals with hardware and the software of the network coordinator node and the sensor nodes. The theoretical and practical results show that the system can efficiently capture greenhouse environmental parameters, including temperature, humidity, and carbon dioxide concentration and also clears the normal communication between nodes and the network coordinator, good network stability. The implementation explored values used in the complex greenhouse environmental monitoring.

Fu Bing [11], V. Sandeep, K. Lalith Gopal, S. Naveen, A. Amudhan, L. S. Kumar [23] have proposed the transition from precision agriculture to modern agriculture in China. The agriculture intelligent system was based on IOT which is introduced for organic melons and fruits production and quality. Many of the technologies were used in the system, such as RFID, sensors and etc., The system contains three platforms to monitor agriculture and fruits. The intelligence agricultural system based on internet has been applied to the melon and fruit production, it plays a role which is not only that the farmers have lesser working hours, but also to improve the ability to save costs, improve the quality of fruits.

Yunseop Kim [14], R. Balamurali, K. Kathiravan [15] have proposed the design for wireless sensor network (WSN) for a water irrigation control and monitoring that is composed of a number of sensor nodes with a networking capability that is deployed for an ad-hoc for the purpose of ongoing monitoring. The parameters used in the water reservation control are water levels and motor movement of the gate controlling the flow of water which is measured by the sensors, which will sense the condition and forward it to base station or control room.

This proposed system offers a low power consumption with high reliability based on the result. The use of high power WSN is suitable for tasks in industries involving huge area monitoring like manufacturing, mining constructing, etc., The system discussed here is very easy to install and the base station can be placed at the local residence close to the area of monitoring where a person requires minimal training at the beginning of the system installation.

Giuseppe Anastasi [16] designed a WSN-based system to monitor the productive cycle of high-quality wine in a sicilian winery. This project aimed to ensure overall good quality of the production. The design incorporates accurate planning in field, the stored product preservation. Wireless Sensor Networks are deployed as the sensing infrastructure of distributed system to control prototype productive chain, nodes have been deployed both in the field and in the cellar, where wine aging is produced. The data is collected at a main unit in order to process inferences that suggest timely interventions that preserve the grapes quality.

Rwan Mahmoud [17], Chen XianYi, Jin Zhi Gang, Yang Xiong [33] describes the security issues of Internet of Things which are directly related to the wide application of the system. Beginning with the architecture and features of IOT, expands many security issues that exist in three layered architectures, and came up with solutions to the issues. The safety measures concerned with it, the ones about perception layer are particularly viewed, including algorithm and key management, security routing protocol and data fusion technology, as well as authentication and access control, etc.

Dragoş Mihai Ofrim, Zulhani Rasin, Hizzu Hamzah Mohd, Shahrieel Mohd Aras designed an improved system for environmental monitoring and controlling in terms of efficiency, flexibility and performance. Some parameters that have been taken into consideration are resolution, accuracy, acquisition rate, energy consumption, flexibility etc., The designed system allows multi-point monitoring at any location, without any need of wired connection and have intelligent sensors. The measuring point density offers high accurate data even from the remote locations. A split is created, in terms of physical connection, between the measuring, monitoring and control parts, making the system extremely flexible. The

disadvantage of this system is regarding power consumption, which is a key factor of wireless sensor networks. Therefore, the sensor nodes require a good resource management in network. This paper uses Zig bee protocol. Improvements and further developments of this system predicts: alternative energy resources, algorithms for energy saving, increased connectivity and reduced traffic. To monitor the parameters from a greater distance, this system could be supplied with GSM or Wi-Fi transmitters, to be able to transfer the information through existing telecommunication networks.

Rachel Cardell-Oliver [19] described the design and implementation of a reactive event driven network for environmental monitoring of soil moisture and evaluates the effectiveness of this solution. A novel feature is to create a solution is its reactivity to the environment: when rain falls and soil moisture is changing rapidly, measurements are collected frequently, whereas during dry periods between rainfall event measurements are collected much less often. Reactivity allows to focus on dynamic responses and limit the amount of useless data gathered, as well as improving robustness and network lifetime. The main aim of this experiment is to demonstrate a reactive sensor network that can deliver useful data on soil moisture responses to rainfall. The Pin-jar network meets the goal of providing useful data on dynamic responses of soil moisture to rainfall. Future work will focus on addressing the limitations of the current prototype in robustness of packet delivery and network longevity, and in guaranteeing network response to events of interest. Authors plan to generalize event-condition-action framework for programming reactive sensor networks.

Duan Yan-e [20] explained the agricultural information technology (AIT) widely applied to every part of agriculture and is going to become the most efficient means and tools for enhancing agricultural production and for making use of complete agricultural resources. Agriculture Information Management affects the range of agricultural information and efficiency of agricultural production. In this experiment, on the count of introducing the concept of agricultural information management and analyzing some of the features of agricultural data, the design method and architecture of Intelligent Agriculture MIS was designed in detail, finally, the proposal gives an implementation illustration of system in agricultural production.

Fiona Edwards Murphy [21] proposed system uses Wireless Sensor Network (WSN) technology to monitor a honeybee colony and collect information about activity within a beehive as well as its surrounding area. The project uses low power WSN technologies, including novel sensing techniques, energy neutral operation, and multi radio communications including cloud computing to monitor the conditions within the colony. WSN is modern new technology, it is a important concept of the Internet of Things. A complete solution is presented including a smart hive communication with data aggregation and visualization tools. Future work will focus on improving the energy performance of the system, introducing a more specialized set of sensors, implementing a machine learning algorithm to extract meaning from the data without human supervision; and securing additional deployments of the system.

P. Tirelli, N.A. Borghese [22] found that monitoring pest insect population is currently a issue in crop protection. The system here is currently based on a distributed imaging device operated via a wireless sensor network that is able to automatically capture and transmit images of trapped areas to a remote host station. The station validates the density of insect evolution at different farm locations and produces an alarm when insect density goes over threshold. The client nodes are spread in the fields, which act as monitoring stations. The master node coordinates the network and retrieves captured images from the client nodes. During a monitoring period of four weeks the network operating regularly, predicts a pest insects' population curve correlated to daily evaluation obtained by visual observations of the trap and hence the feasibility is determined.

Nguyen Tang Kha Duy [24], K.Sathish Kannan and G.Thilagavathi [35] found a versatile solution in an effort of improving the accuracy in monitoring the environmental conditions and reducing human power for industrial household's shrimp farming. The component is capable of collecting, analyzing and presenting data on a Graphical User Interface (GUI), programmed with Lab VIEW. The proposed system saves the cost of hiring labor as well as the electricity usage. The system design proposes a versatile, low-cost, and commercial version which will function best for small to medium size farming operations as it does not require any renovation or reconstruction of the pond. This method is updating the sensor

information and reflecting the real factors of environmental shrimp farming. At present this system also provides many options which are user friendly allowing the updates to farmer with all the necessary farming factors resulting in good production. The system has high scalability for households or farming businesses on a large scale. Authors proceed to test the reliability of the system in real life applications.

Zulhani Rasin [25], Wang Weighing and Cao Shuntian [39] presented a remote intelligent monitoring system (RIMS) based on ZIGBEE Wireless Sensor Network (WSN). In RIMS data is transmitted to the controlling center through ZIGBEE Mesh Network and the remote control will be available. The redundancy router nodes are designed to improve the transmitting reliability. This paper uses the wireless mesh network instead of placing the cables in the field, to avoid cable problems and deployment. Except the wireless central receiver, the system will keep on working as usual in case of failure of any node.

R. Balamurali [26], Narut Soontranon, Panwadee Tangpattanakul, Panu Srestasathien, Preesan Rakwatin [32], Chen XianYi, Jin Zhi Gang, Yang Xiong [33] have discussed precision agriculture for real-time monitoring of environmental conditions of a farm like temperature, humidity, soil PH etc. The values of monitored parameters are communicated to the remote server in order to take appropriate action, instead an actuator or an automated system can also be used to take appropriate action based on the measured parameters over a period of time. This paper analyzes the various routing protocols like AOMDV, AODVDSR and Integrated MAC and Routing protocol (IMR) for precision agriculture using WSN. This analysis draws conclusions that Integrated MAC and Routing Algorithm is best suitable for multi-hop routing for precision agriculture using Wireless Sensor Network (WSN) in-terms of Network life time. The network lifetime is considered as the time at which the first node in the WSN dies. The work may be hanced to analyze other network parameters like throughput and end-end delay.

Sonal Verma [27], Chen XianYi, Jin Zhi Gang, YangXiong [33], A. Sivasankari and S. Gandhimathi [36] have experimented the potential utility of an inexpensive camera observation system called crop phonology recording system which is as an alternative approach for the observation of crop growth condition. The

design explored the availability of daytime exposure values recorded in the header region of EXIF formatted JPEG files by RGB and NIR-cameras and proposed using vegetation indices, ev-NDVI, ev-SR, and ev-CIgreen, which were calculated from the combination of daytime exposure values and cDN. The study found that ev-VARI worked the best for maize and ev-CIgreen for soybeans when estimating green LAI. ev-VARI was also the best in estimating green leaf biomass in maize and NRBINIR in soybean. This camera-based vegetation index has possibility to estimate a wide variety bio-physical parameters, a choice for high frequency observations at many locations of vegetation.

Elias Yaacoub [28], K. Sathish Kannan, G. Thilagavathi [34] proposed the deployment of a wireless sensor network to monitor and analyze air quality in Doha. Data stored on the server is subjected to intelligent processing and analysis in order to present it in different formats for different categories of end users. This experiment brings out a user-friendly computation of an air quality index to disseminate the data to the general public and also the data presentation for environmental experts using dedicated software tools, for example- the R software system and its Open-air package. Depending on the target end-user the stored data can be accessed and displayed in different formats.

Jinhu Liao [30], Chen XianYi, Jin Zhi Gang, Yang Xiong[37], Weimin Qiu, Linxi Dong, Haixia Yan, Fei Wang [38] have proposed a remote monitoring system, which can monitor agricultural land in real time and makes good decisions. The system collects data from a farm by using ZigBee modules, which makes data fused by using high performance controller ARM micro controller and transfers the data to a remote computer by using GPRS modules to take an informed management decisions by using computer. This is not only a solution to improve the level of agricultural production, but also to reduce human the costs very effectively.

Nelson Sales [31] experimented with interconnection of smart objects embedded with sensors that enabled them to interact with the environment and among themselves, forming a Wireless Sensor Network (WSN). These network nodes perform acquisition, collection and analysis of data, such as temperature and soil moisture. This type of data can be applied to automate the irrigation process in

agriculture for decreasing the water consumption, which would result in monetary and environmental benefits. Authors proposed to use cloud computing which has the high storage and processing capabilities, the rapid elasticity and pay-per-use characteristics makes an attractive solution to the provided might help researchers to highlight issues in the agriculture domain

Smart Agriculture Monitoring System Using IOT Dr. Sanjay N. Patil, Madhuri B. Jadhav Associate Professor, Electronics Engg. Department, PVPIT Budhgaon, Sangli, Maharashtra PG Student, Electronics Engg. Department, PVPIT Budhgaon, Sangli, Maharashtra

Agriculture is basic source of livelihood People in India. It plays major role in economy of country. But now days due to migration of people from rural to urban there is hindrance in agriculture. Monitoring the environmental factor is not the complete solution to increase the yield of crops. There are no of factors that decrease the productivity to a great extent. Hence Automation must be implemented in agriculture to overcome these problems. An automatic irrigation system thereby saving time, money and power of farmer. The Traditional Farm land irrigation techniques require manual intervention. With the automated technology of irrigation the human intervention can be minimized. Continuous sensing an monitoring of crops by convergence of sensors with Internet of things (IOT) and making farmers to aware about crops growth, harvest time periodically and in turn making high productivity of crops and also ensuring correct delivery of products to end, consumers at right place and right time. So to overcome this problem we go for smart agriculture technique using IOT. This Project includes sensors such as temperature, humidity, soil moisture and rain detector for collection the field data and processed. These sensors are combined with well established web technology in the form of wireless sensor network to remotely control and monitor data from the sensors. Keywords: Arduino Uno, ESP8266 (Wi-Fi module), Automation of Irrigation System, Sensors

- 1) Sensor data acquisition :- The sensor is interface with Arduino Uno such as DHT11 Temperature, Humidity, Soil moisture and Rain detection sensor is used.

- 2) Wireless data transmission :- The data acquired from sensors are transmitted to the web server using wireless transmission (WIFI module ESP8266).
- 3) Data processing and Decision making :- The data processing is the task of checking various sensors data received from the field with the already fixed threshold values. The motor will be switched ON automatically if the soil moisture value falls below the threshold and vice- versa. The farmer can even switch ON the Motor from mobile using mobile application.
- 4) Automation and irrigation system :- The irrigation system automated once the control received from the web application or mobile application. The relays are used to pass control form web application to the electrical switches using Arduino microcontroller. The circuits with low power signal can be controlled using relay.
- 5) Web application :- The web application will be designed to monitor the field and crops from anywhere using internet connection. To control the Arduino processing IDE is used, the webpage can be communicated using the processing IDE.
- 6) Mobile Application :- The mobile application will be developed in android. The mobile application helps to monitor an controlled filed from anywhere.

Smart Agriculture Monitoring System using IoT P. Prema, B. Sivasankari, M. Kalpana and R. Vasanthi Assistant Professor, Tamil Nadu Agricultural University, Agricultural College and Research Institute, Madurai 3,4Assistant Professor, Tamil Nadu Agricultural University, Coimbatore

The agricultural growth is enhanced with the increase in the productivity and upgradation of the plantation systems. The application of Internet of Things (IoT) technology in agriculture could have the greatest impact for increase the productivity. The global population is increased every year. So, to feed this much population, the farming industry must use advanced technology like IoT for

increased profit. Smart farming based on IoT technologies will enable growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made. In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, PIR Sensor) and automating the irrigation system. The farmers can monitor the field conditions from anywhere. IoT-based smart farming is highly efficient when compared with the conventional approach. Keywords: IoT, PIR Sensor, Light, Humidity, Temperature

Hardware Arduino Uno-MICROCONTROLLER: The Arduino Uno is one of the most popular microcontrollers in the industry. It is user convenient and easier to handle. The coding or programming of this controller is also easy. The device has capabilities to be connected the internet and act as a server too, this way the handling of information and data. **Soil Moisture Monitoring** Soil moisture is the water held within the soil pores. Soil moisture is one of the main factors deciding crop yield, as it affects the water uptake of the plant. So measuring soil moisture always plays an important role in successful farm management. The conventional method to evaluate soil moisture content of 'look and feel' can be highly inaccurate. The use of moisture sensors helps to evaluate soil moisture helps to make relevant irrigation decisions. The hardware is consists of an Arduino microcontroller (here an Arduino Uno) and a pre-wired soil moisture sensor module. The soil moisture sensor module, built around the LM393 comparator, gives an active-low (L) level output when the soil is dry (determined by a pre-setted threshold value). This digital output (wet soil → L / dry soil → H) is routed to one I/O terminal (D2) of the Arduino microcontroller. Based on this input (at D2) arduino gives an active-high (H) output through D13 when soil is dry, and an active-low (L) output when soil is wet.

The sensor read the temperature and humidity and displays the reading in the LCD. It can measure the environment temperature and humidity to meet the high demand.

2.1 EXPERIMENTATION AND RESULTS

The hardware is interfaced with all the sensors in the board. The hardware components include the microcontroller, a water pump, relay, 12 V battery,

Wi-fi sensor and the soil moisture sensor is interfaced and power supply has provided. The system has been tested for watering a plant in a garden. In the field section, sensors are deployed in the field like soil moisture. The data collected from these sensors are sent to the Database via the android application. In control section, the system is turned on using the application, this is done using the on or off buttons in the application. Also, this system is turned on automatically when the moisture of the soil is low, the pump is turned on and depending on the moisture content. The application has a future feature of taking the time from the user and irrigates the field when the time comes. In manual mode, there is a manual switch in the field to make sure that if the system fails, one can turn off the water supply manually.

IoT Based Smart Agriculture Monitoring System Harika Pendyala, Ganesh Kumar Rodda , Anooja Mamidi , Madhavi Vangala Sathyam Bonala, Keerti Kumar Korlapati

In every country agriculture is done from ages which are considered to be science and also art of cultivating plants. In day today life, technology is updating and it is also necessary to trend up agriculture too. IoT plays a key role in smart agriculture. Internets of Things (IoT) sensors are used to provide necessary information about agriculture fields. The main advantage of IoT is to monitor the agriculture by using the wireless sensor networks and collect the data from different sensors which are deployed at various no des and send by wireless protocol. By using IoT system the smart agriculture is powered by NodeMCU. It includes the humidity sensor, temperature sensor, moisture sensor and DC motor. This system starts to check the humidity and moisture level. The sensors are used to sense the level of water and if the level is below the range then the system automatically stars watering. According to the change in temperature level the sensor does its job. IoT also shows the information of humidity, moisture level by including date and time. The temperature level based on type of crops cultivated can also be adjusted. Keywords: IoT, Soil, Moisture and Temperature sensors, Relay, Wi-Fi module ESP8266, ThingSpeak

2.2 WORKING

The smart agriculture monitoring system is tested under various conditions. The soil moisture sensor is used to test the soil for all climatic conditions and results are interpreted successfully. The moisture output readings at different weather conditions is taken and updated. Wi-Fi is used to achieve the wireless transmission. The values of soil moisture sensor purely depend on the resistivity of the soil. The value of the sensor at beginning of wet condition is 0. The sensed value is sent to microcontroller through NodeMCU and motor pump gets OFF in this condition. The maximum threshold value upon dry soil is 1023. When the sensed value by sensor reaches the threshold value, the microcontroller trigger the relay and motor gets ON. When sufficient amount of water is supplied to plants, the motor pump is turned ON and is turned OFF automatically

OVERVIEW OF IOT BASED AGRICULTURE MONITORING SYSTEM SYEDA FASIHA FATIMA M.Tech Student, E&TC Department CSMSS Chh.Shahu College of Engineering, Kanchanwadi, Aurangabad, (MS), India PROF. SUMERA ALI Assistant Professor, E&TC Department CSMSS Chh.Shahu College of Engineering, Kanchanwadi, Aurangabad, (MS), India DR. ULHAS SHINDE Principal and Professor, E&TC Department CSMSS Chh.Shahu College of Engineering, Kanchanwadi, Aurangabad, (MS), India

The Indian agriculture sector is contributing 18% to the GDP. About 50% of Indian workforce has got an employment through agriculture and allied sector. Indian economy is dependent on agriculture sector since many decades and the contribution of the sector always important for the development of India. In ancient days the conventional farming was the only method implemented in India. With development of India in last few decades and availability of the technology at affordable cost has opened up the doors of opportunities for Indian agriculture sector. Internet of things (IoT) technology being simple and with availability of the open source hardware and software is contributing for the smart management of the farming activities. This paper presents the overview of agriculture monitoring by means IoT based systems. **KEYWORDS:** IoT, Sensors, Wi-Fi module, ZCD, Optical Isolator, TSOP Receiver, IR Remote, etc.

CHAPTER 3

PROPOSED METHODOLOGY

The Arduino based system for monitoring and controlling the agriculture operations is proposed to be developed by Authors. Authors have proposed to use the sensors such as humidity, moisture and temperature sensor. The objective of the work is to develop the low cost solution for automation of agriculture operation.

In Agriculture field, the farmers use manual method for checking the checking the soil parameters. The system mainly focuses to developing devices to manage and alert the farmers by the wireless sensor system (Lakshmisudha, 2011). It proposes a low cost and efficient wireless sensor network technique to acquire the soil moisture and temperature from various locations of farm and as per the need of crop controller to take the decision whether the irrigation is requirement (Nandurkar et al., 2014). The author (Chetan, 2015) proposes a novel technique for smart farming by linking a smart sensing system and smart irrigator system through wireless communication technology. The atmospheric conditions are monitored and controlled online (Vidya Devi & Meenakumari, 2013). The IoT technologies are used for smart agriculture (Gondchawar & Kawitkar, 2016). The highlighting features are smart GPS based remote controlled robot to perform tasks like weeding, spraying, moisture sensing, human detection and keeping vigilance. The cloud computing devices that can create a whole computing system from sensors to tools that observe data from agricultural field images and from human actors on the ground and accurately feed the data into the repositories along with the location as GPS coordinates (Gayatri et al., 2015). The author proposes (Joaquín Gutiérrez, 2013) an idea about how automated irrigation system was developed to optimize water use for agricultural crops. In addition, a gateway unit handles sensor information. The IoT based monitoring system (Meonghun Lee, 2013) to analyze crop environment and the method to improve the efficiency of decision making by

analyzing harvest statistics. Image processing (Jhuria et al., 2013) is used as a tool to monitor the diseases on fruits during farming, right from plantation to harvesting. The variations are seen in color, texture and morphology. The soil parameters such as soil moisture content and surrounding temperature are crucial factors for the proper plant growth. The system provides a real time system which monitors soil temperature and soil moisture efficiently. The system valves are turn ON or OFF automatically depending upon the moisture content. The system provides a real time analysis to the owner to monitor variation in the parameters. Using this system, one can save manpower, water to improve production and ultimately increase profit.

3.1 BLOCK DIAGRAM

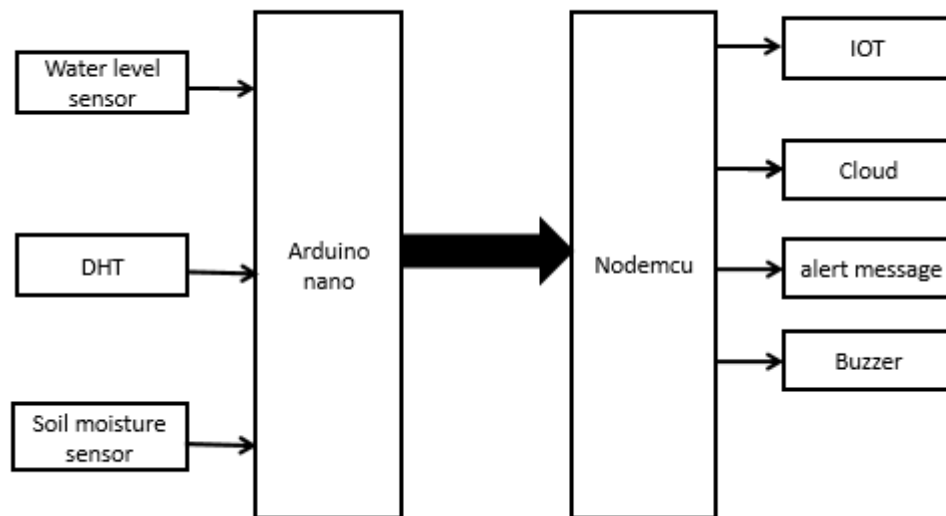


Fig. 3.1: Block Diagram

The figure below shows the basic blocks of the proposed system for IoT based agriculture monitoring. The sensors for identification of various parameters affecting the crop are shown in the figure. The Water level sensor mainly detects the rainfall related information and its effect on the crop can be studied. Temperature and moisture sensors are used for real time monitoring of the parameters. The water pump is controlled through ATmega 2560 microcontroller. Wi-Fi module enables the storage of real time parameters on cloud. The parameters can also be displayed on the LCD on real time basis. The feedback signal is useful to control the operation of the pump there by controlling the flow of water. The system is found suitable to

control the water flow depending upon the need of the cultivation. The programming enables the operations in different environments and various soil types to control the water flow. The moisture of the soil is checked through the sensor and the flow of water can be controlled through the circuit.

3.2 HARDWARE REQUIREMENTS

- Arduino UNO Microcontroller
- Node MCU module.
- Power Supply unit
- Water level
- DHT
- Soil Moisture sensor

3.3 SOFTWARE REQUIREMENTS

- Arduino IDE
- Embedded C
- Proteus

CHAPTER 4

HARDWARE DESCRIPTION

4.1 ARDUINO

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

This guide covers the Arduino Uno board (Spark fun DEV-09950, \$29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the

This is what the Arduino board looks like.



Fig. 4.1: Arduino

The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

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

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

4.2 SUMMARY

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

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

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

4.3 PIN CONFIGURATION

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

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

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

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** 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.
- **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 Arduino 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 for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digital Write( )`, and `digital Read( )` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-

up resistor (disconnected by default) of 20-50 kohms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and Transmit (TX) TTL serial data.

These pins are connected to the corresponding pins of the ATmega8U2 USB-to- TTL Serial chip.

- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite () function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built- in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

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

- **I²C: A4 or SDA pin and A5 or SCL pin.** Support I²C communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference ().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

4.4 COMMUNICATION

Microcontrollers depend on a host computer for developing and compiling programs. The software used on the host computer is known as an integrated development environment, or IDE. For the Arduino, the development environment is based on the open source Processing platform (www.processing.org) which is described by its creators as a “programming language and environment for people who want to program images, animation, and interactions.” The Arduino programming language leverages an open source project known as Wiring (wiring.org.co). The Arduino language is based on good old- fashioned C. If you are unfamiliar with this language, don’t worry; it’s not hard to learn, and the Arduino IDE provides some feedback when you make mistakes in your programs.

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

A 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; see the documentation for details. For SPI communication, use the SPI library.

As you go through the list of programming statements available in the Arduino IDE (choose Help->Reference), you might think there isn’t much power for doing things like running servos, operating stepper motors, reading potentiometers,

or displaying text on an LCD. Like most any language based on C, the Arduino supports the notion of “libraries” code.

Repositories that extend core programming functionality. Libraries let you re- use code without having to physically copy and paste it into all your programs. The standard Arduino software installation comes with several libraries you may use, and you can download others from the Arduino support pages and from third-party websites that publish Arduino library code. A good example of a library you'll use with the Robot and likely many other robot projects is

Servo. This library allows you to connect one or more hobby R/C servos to the Arduino's digital I/O pins. The Servo library comes with the standard Arduino installation package Library->Servo. This adds the line

```
#include <Servo.h>
```

Which tells the Arduino IDE that you wish to include the Servo library in your sketch. With the functionality of the library now available to you, you can use its various functions to control one or more servos. For example, you can use the write function to rotate a servo to a specific position, from 0 to 180 degrees. The following code

```
myServo.write(90);
```

Moves a servo to its midpoint, or 90 degree position. Structurally, Arduino sketches are very straightforward and are pretty easy to read and understand. The Arduino program contains two main parts: setup () and loop (). These are programming functions that do what their names suggest: setup () sets up the Arduino hardware, such as specifying which I/O lines you plan to use, and whether

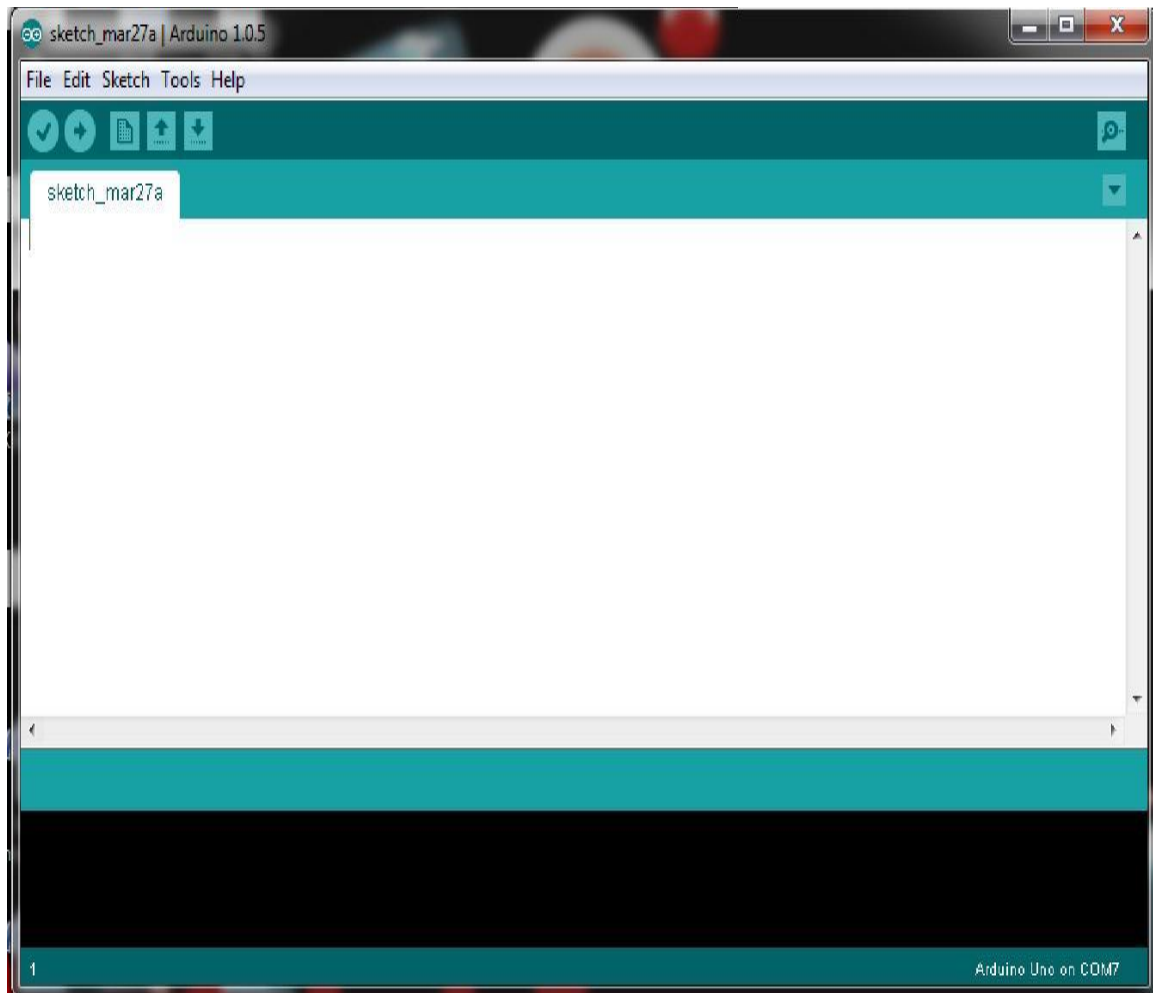
They are inputs or outputs. The loop () function is repeated endlessly when the Arduino is operating.

Arduino IDE (Integrated development environment) is used to write the program and dump into the Arduino board.

4.5 ARDUINO SOFTWARE

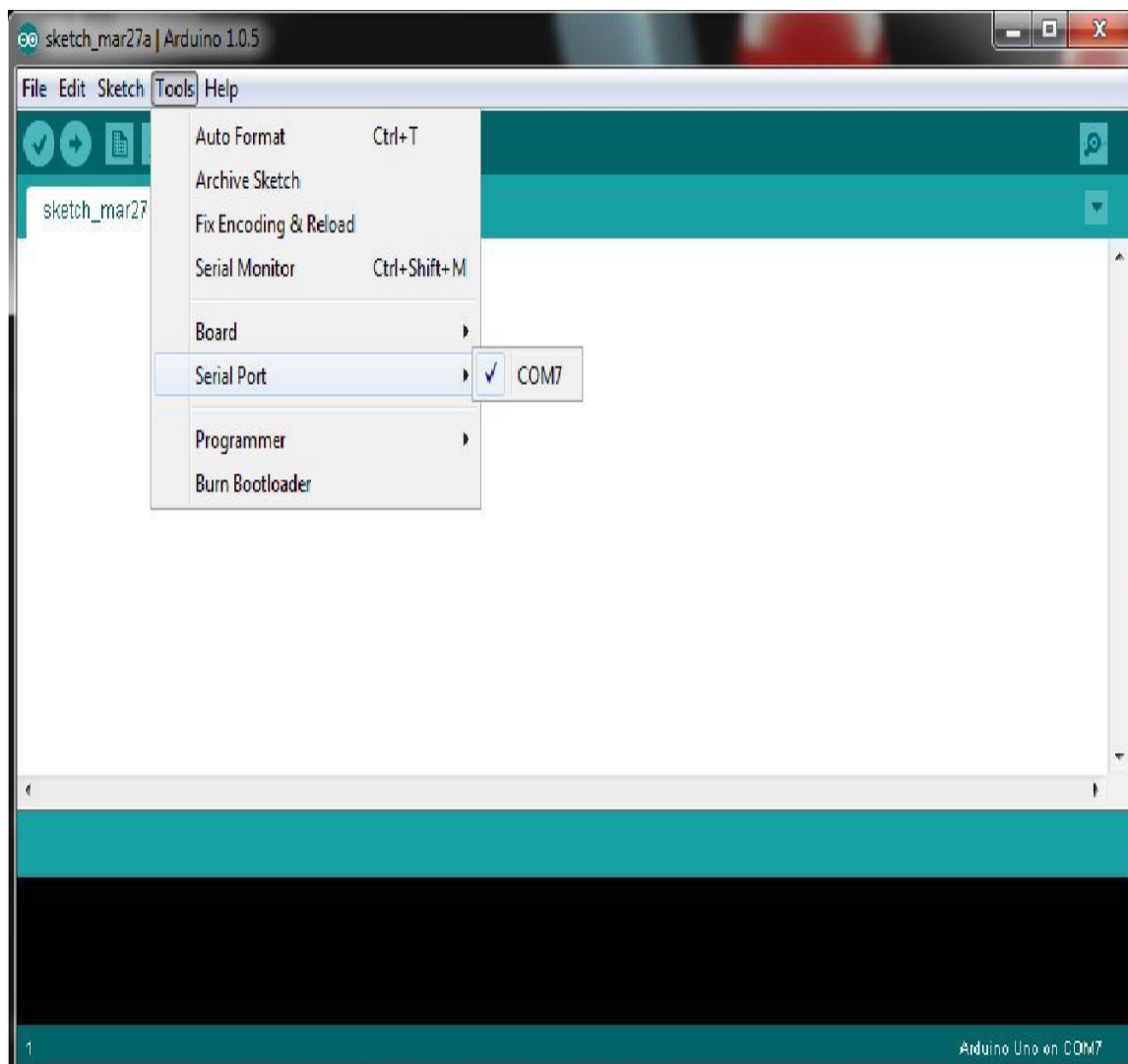
1. Open Arduino IDE as shown below

Open Arduino IDE



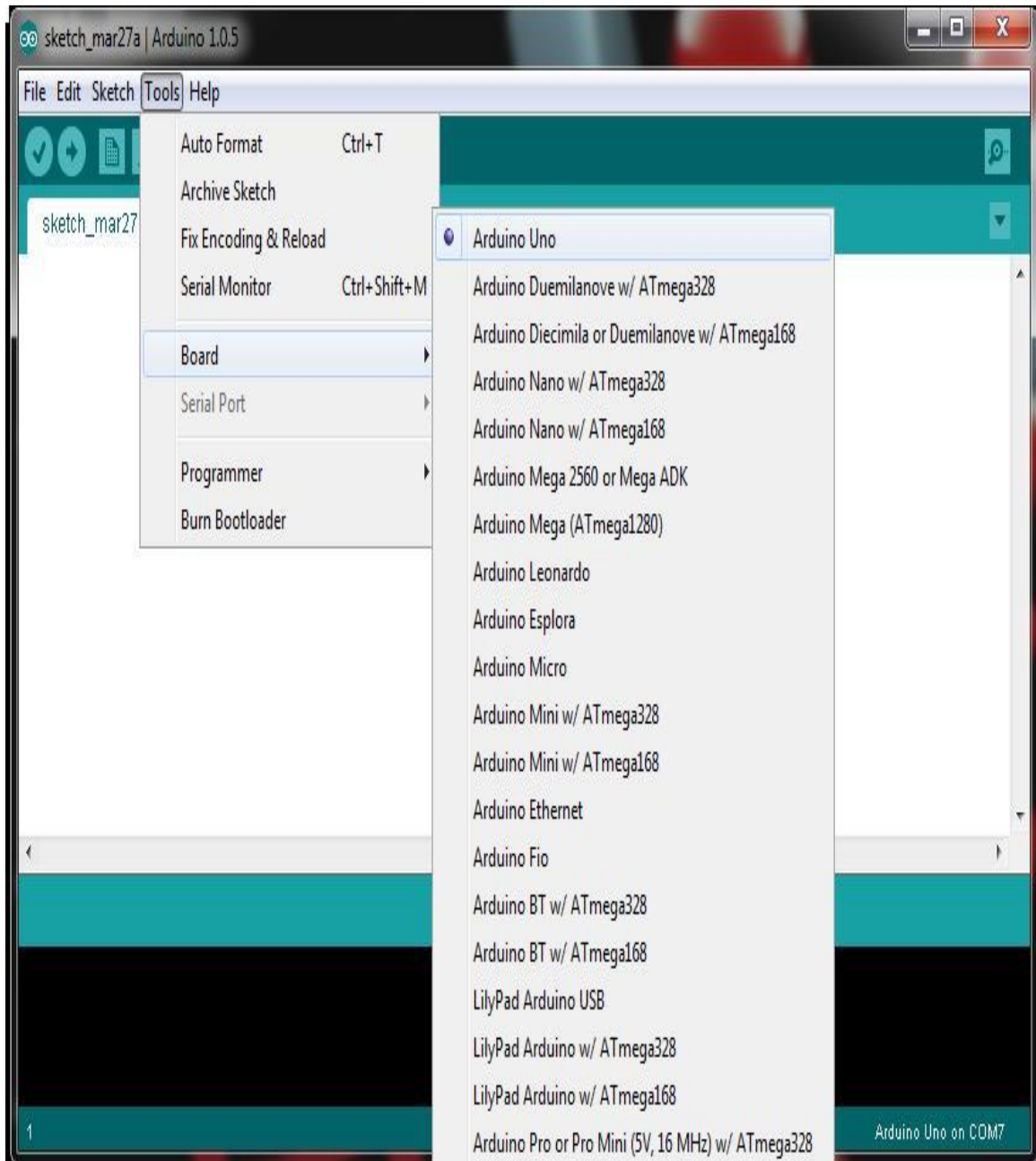
2. Select the COM Port from tool

Select the COM Port



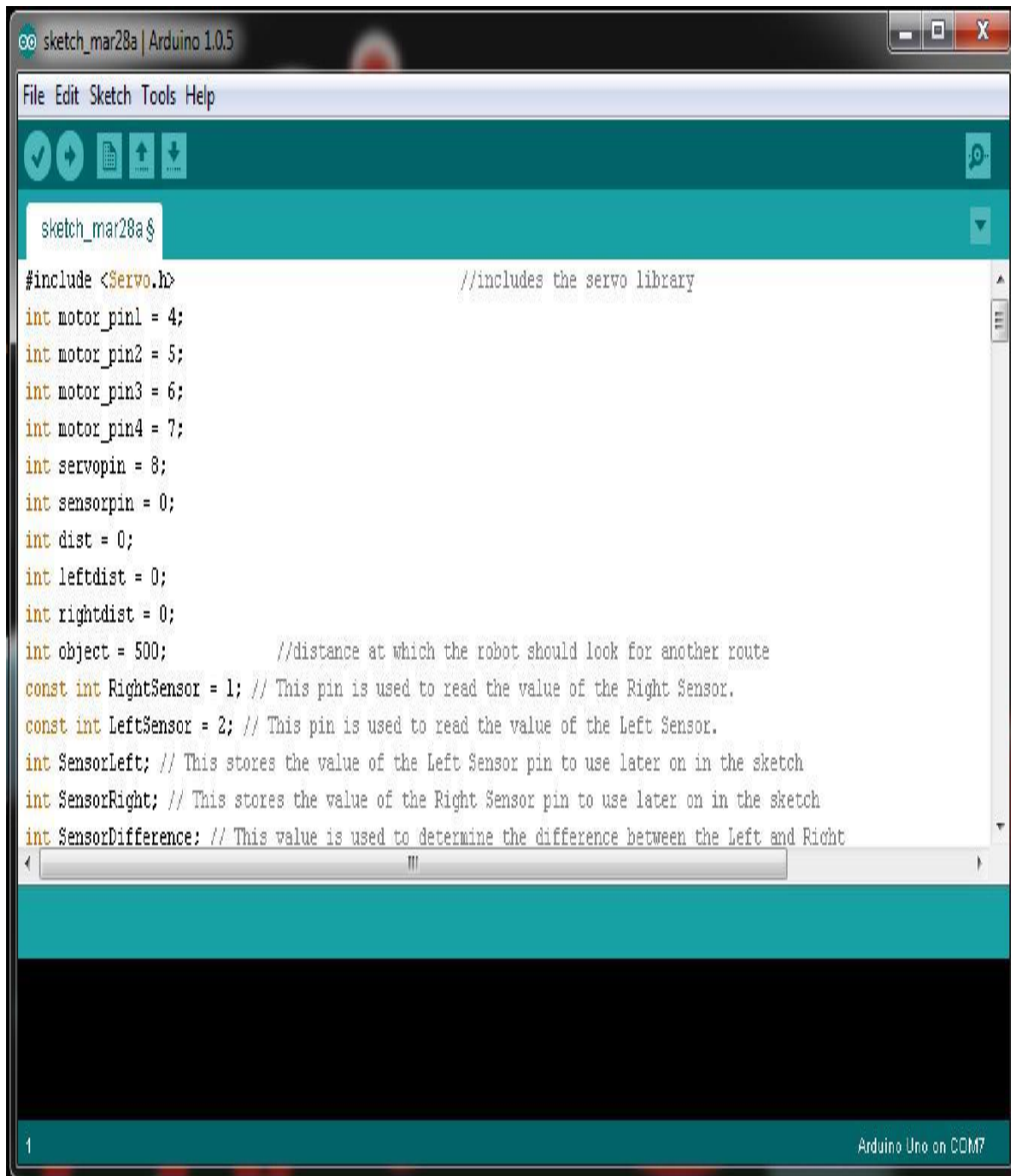
3. Select the required Arduino board from Tools as shown below

Select the required Arduino board



4. Write the sketch in Arduino IDE

Sketch in Arduino DE



```
sketch_mar28a $
#include <Servo.h> //includes the servo library
int motor_pin1 = 4;
int motor_pin2 = 5;
int motor_pin3 = 6;
int motor_pin4 = 7;
int servopin = 8;
int sensorpin = 0;
int dist = 0;
int leftdist = 0;
int rightdist = 0;
int object = 500; //distance at which the robot should look for another route
const int RightSensor = 1; // This pin is used to read the value of the Right Sensor.
const int LeftSensor = 2; // This pin is used to read the value of the Left Sensor.
int SensorLeft; // This stores the value of the Left Sensor pin to use later on in the sketch
int SensorRight; // This stores the value of the Right Sensor pin to use later on in the sketch
int SensorDifference; // This value is used to determine the difference between the Left and Right
```

1 Arduino Uno on COM7

5. Compile and upload the Sketch to Arduino board upload the Sketch to Arduino board



4.6 NODE MCU

NodeMCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module

Programming NodeMCU ESP8266 with Arduino IDE

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use.

Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this [Getting Started Tutorial for NodeMCU](#) to prepare your Arduino IDE for NodeMCU.

NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.

Both the firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with

a Tensilica Xtensa LX106 core, widely used in IoT applications (see related projects).

History

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. NodeMCU started on 13 Oct 2015, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki 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 the NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

In the summer of 2015 the original creators abandoned the firmware project and a group of independent contributors took over. By the summer of 2016 the NodeMCU included more than 40 different modules.

ESP8266 Arduino Core[edit]

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 toolchains 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 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".^[17] This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.

NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board.

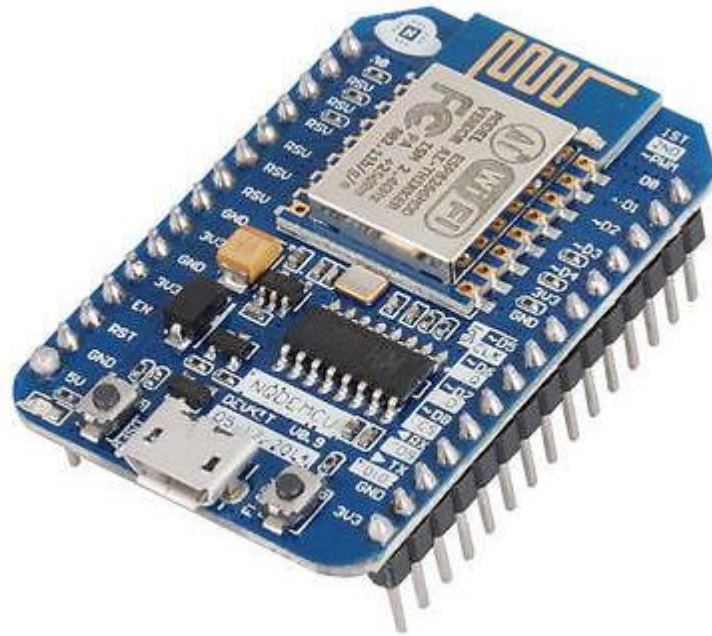


Fig. 4.2: ESP8266 Development board

NodeMCU Development Board/kit v0.9 (Version1)

Since NodeMCU is an open-source platform, its hardware design is open for edit/modify/build.

NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 WiFi Module.

There is Version2 (V2) available for NodeMCU Dev Kit i.e. **NodeMCU Development Board v1.0 (Version2)**, which usually comes in black colored PCB.

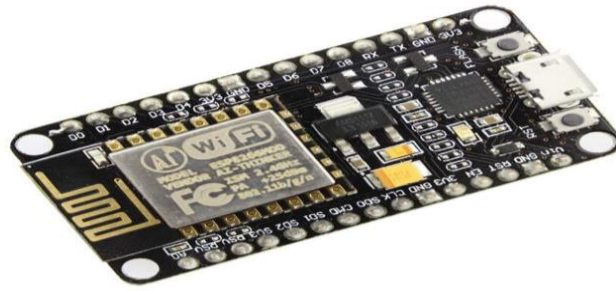


Fig. 4.3: NodeMCU Development Board

NodeMCU Development Board/kit v1.0 (Version2)

For more information about NodeMCU Boards available in the market refer to NodeMCU Development Boards.

NodeMCU Dev Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board.

It supports serial communication protocols i.e. UART, SPI, I2C, etc.

Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards, etc.

How to start with NodeMCU?

NodeMCU Development board is featured with wifi capability, analog pin, digital pins, and serial communication protocols.

To get started with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There are online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

To know more about how to build custom NodeMCU firmware online and download it refer to [Getting started with NodeMCU](#).

How to write codes for NodeMCU?

After setting up ESP8266 with Node-MCU firmware, let's see the IDE (Integrated Development Environment) required for the development of NodeMCU.

NodeMCU with ESPlorer IDE

Lua scripts are generally used to code the NodeMCU. Lua is an open-source, lightweight, embeddable scripting language built on top of C programming language.

For more information about how to write Lua script for NodeMCU refer to [Getting started with NodeMCU using ESPlorerIDE](#)

NodeMCU with Arduino IDE

Here is another way of developing NodeMCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on NodeMCU using the Arduino development environment. This makes it easy for Arduino developers than learning a new language and IDE for NodeMCU.

For more information about how to write Arduino sketch for NodeMCU refer to [Getting started with NodeMCU using Arduino IDE](#).

The difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing an application for NodeMCU using ESPlorer IDE and Arduino IDE.

We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE.

Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executed sequentially. It will not build a binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get executed.

In Arduino IDE when we write and compile code, the ESP8266 toolchain in the background creates a binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware.

That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got an error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware.

Since Arduino IDE compiles and upload/writes complete firmware, it takes more time than ESPlorer IDE.

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz

- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

Water Level Sensor

- Water Sensor. The water sensor or water level sensor is **used to detect water leakage, rainfall, tank overflow, or to measure the water level.**
- This sensor works on the principle of variable resistance. The sensor consists of a series of parallel exposed conductors. Together this series acts as a variable resistor, whose resistance varies according to the water level in the water tank.
- As more water sensor is submerged in, the better is the conductivity and the lower is the resistance. The less the water sensor is submerged in, the poor is the conductivity and the higher is the resistance.
- The output of the water level sensor is according to the resistance of the water produced. i.e. it will produce a voltage proportional with resistance.

The Water level sensor is a solid state, continuous (multi-level) fluid level sensor for measuring levels in water, non-corrosive water based liquids and dry fluids (powders). The Water level sensor is manufactured using printed electronic technologies which employ additive direct printing processes to produce functional circuits. Theory of Operation The Water level sensor's envelope is compressed by hydrostatic pressure of the fluid in which it is immersed resulting in a change in resistance which corresponds to the distance from the top of the sensor to the fluid surface. The Water level sensor provides a resistive output that is inversely proportional to the level of the liquid: the lower the liquid level, the higher the output resistance; the higher the liquid level, the lower the output resistance.

Specifications

Sensor Length: 14.1" (358 mm)

Resolution: < 0.01" (0.25 mm)

Thickness: 0.015" (0.381mm)

Actuation Depth: Nominal 1" (25.4 mm)

Width: 1.0" (25.4 mm)

Reference Resistor (Rref): 2250 , $\pm 10\%$

Active Sensor Length: 12.4" (315 mm)

Connector: Crimpflex Pins Sensor

Output: 2250 empty, 400 full, $\pm 10\%$

Temperature Range: 15°F - 150°F (-9°C - 65°C)

Resistance Gradient: 150 /inch (59 /cm), $\pm 10\%$

Power Rating: 0.5 Watts (VMax = 10V)

Sensor Output The Water level can be modeled as a variable resistor (400 – 2250 $\Omega \pm 10\%$).

Connect to the Water level by attaching a 4 pin connector with pre-soldered wires to the Crimpflex pins. Do not solder directly to the Crimpflex pins. The inner two pins (pins 2 and 3) are the sensor output (Rsense). The outer pins (pins 1 and 4) are the reference resistor (Rref) which can be used for temperature compensation. Suspend the Water level sensor in the fluid to be measured. To work properly the sensor must remain straight and must not be bent vertically or longitudinally. For best results install the sensor inside a section of 1-inch diameter PVC pipe. Double sided adhesive tape may be applied to the upper back portion of the sensor to suspend the sensor in the container to be measured. However, the liquid must be allowed to interact freely with both sides of the sensor. The vent hole located above the max line allows the Water level to equilibrate with atmospheric

pressure. The vent hole is fitted with a hydrophobic filter membrane to prevent the Water level from being swamped if inadvertently submerged.

Specification

- Operating voltage: DC3-5V.
- Operating current: less than 20mA.
- Sensor Type: Analog.
- Detection Area: 40mmx16mm.
- Operating temperature: 10°C-30°C
- Humidity: 10% -90% non-condensing.

Humidity sensor

Measurement and control of temperature and relative humidity finds applications in numerous areas. These days devices are available which have both temperature and humidity sensors with signal conditioning, ADC, calibration and communication interface all built inside them. The use of such smart sensors greatly simplify the design and reduces the overall cost. We discussed in past about Humidity and temperature measurements with Sensirion's SHT1x/SHT7x sensors. These sensors are capable of measuring both temperature and relative humidity and provide fully calibrated digital outputs. While SHT1x/SHT7x are very accurate sensors, they are still expensive for hobbyists use. This articles discusses the DHT11 sensor which also provides calibrated digital outputs for temperature and humidity but is relatively lot cheaper than the Sensirion sensors. The DHT11 sensor uses a proprietary 1-wire protocol which we will be exploring here and implementing with the PIC16F628A microcontroller that will receive the temperature and humidity values from the sensor and display them on a 16×2 character LCD.

About DHT11 sensor

The DHT11 sensor comes in a single row 4-pin package and operates from 3.5 to 5.5V power supply. It can measure temperature from 0-50 °C with an accuracy of $\pm 2^{\circ}\text{C}$ and relative humidity ranging from 20-95% with an accuracy

of $\pm 5\%$. The sensor provides fully calibrated digital outputs for the two measurements. It has got its own proprietary 1-wire protocol, and therefore, the communication between the sensor and a microcontroller is not possible through a direct interface with any of its peripherals. The protocol must be implemented in the firmware of the MCU with precise timing required by the sensor. The following timing diagrams describe the data transfer protocol between a MCU and the DHT11 sensor. The MCU initiates data transmission by issuing a “Start” signal. The MCU pin must be configured as output for this purpose. The MCU first pulls the data line low for at least 18 ms and then pulls it high for next 20-40 μ s before it releases it.

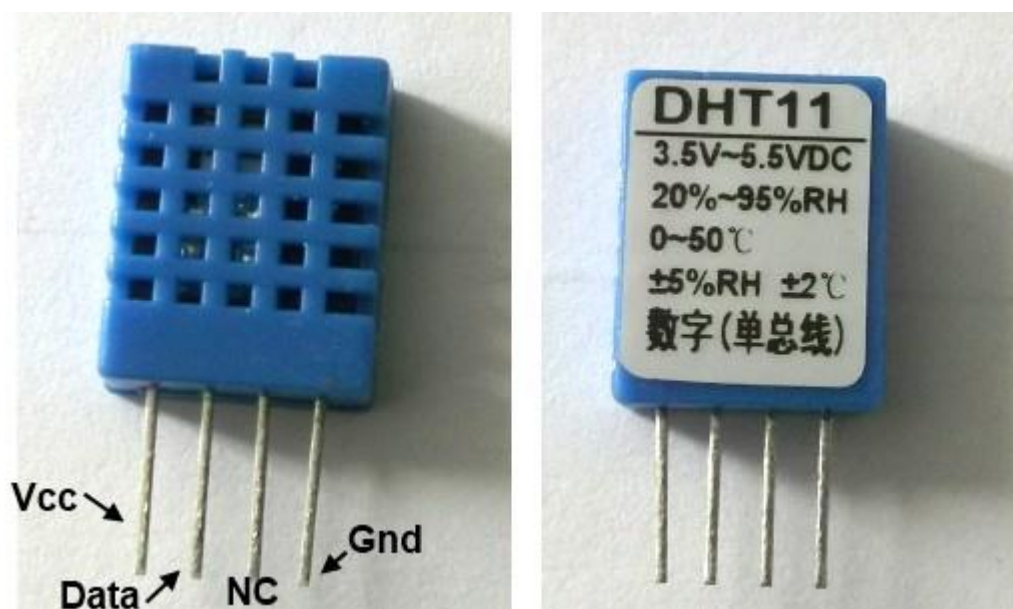


Fig. 4.4: DHT11 sensor

- DHT11 sensor comes in a single row 4-pin package
- Next, the sensor responds to the MCU “Start” signal by pulling the line low for 80 μ s followed by a logic high signal that also lasts for 80 μ s. Remember that the MCU pin must be configured to input after finishing the “Start” signal. Once detecting the response signal from the sensor, the MCU should be ready to receive data from the sensor. The sensor then sends 40 bits (5 bytes) of data continuously in the data line. Note that while transmitting bytes, the sensor sends the most significant bit first.

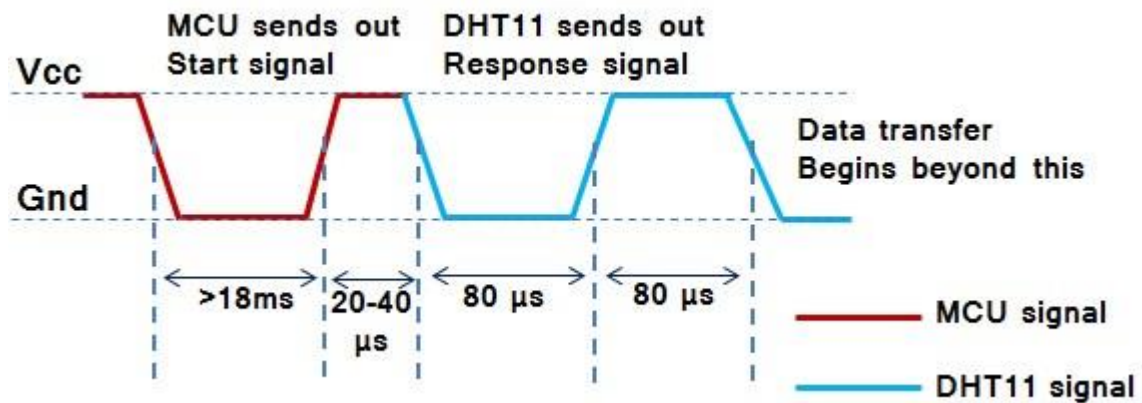


Fig. 4.5: "Start" and "Response" signals

- The 40-bit data from the sensor has the following structure.
- Data (40-bit) = Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp. + Checksum Byte
- For DHT11 sensor, the decimal bytes of temperature and humidity measurements are always zero. Therefore, the first and third bytes of received data actually give the numeric values of the measured relative humidity (%) and temperature ($^{\circ}\text{C}$). The last byte is the checksum byte which is used to make sure that the data transfer has happened without any error. If all the five bytes are transferred successfully then the checksum byte must be equal to the last 8 bits of the sum of the first four bytes, i.e.,
- Checksum = Last 8 bits of (Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp.)
- Now let's talk about the most important thing, which is signalling for transmitting "0" and "1". In order to send a bit of data, the sensor first pulls the line low for $50\text{ }\mu\text{s}$. Then it raises the line to high for $26-28\text{ }\mu\text{s}$ if it has to send "0", or for $70\text{ }\mu\text{s}$ if the bit to be transmitted is "1". So it is the width of the positive pulse that carries information about 1 and 0.

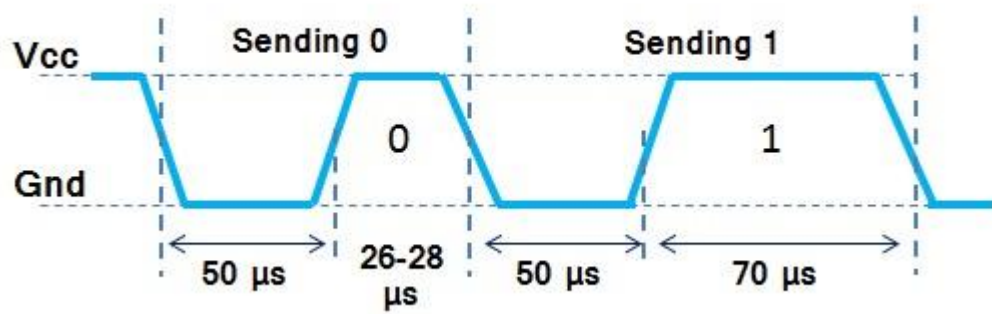


Fig. 4.6: Timing difference for transmitting "1s" and "0s"

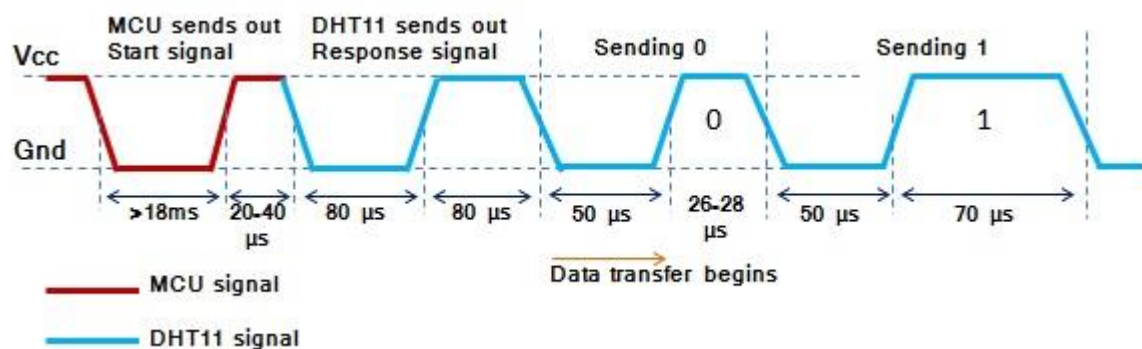


Fig. 4.7: MUC start signal and DHT signals response signal

- Start, Response and Data signals in sequence
- At the end of the last transmitted bit, the sensor pulls the data line low for 50 μs and then releases it. The DHT11 sensor requires an external pull-up resistor to be connected between its Vcc and the data line so that under idle condition, the data line is always pulled high. After finishing the data transmission and releasing the data line, the DHT11 sensor goes to the low-power consumption mode until a new "Start" signal arrives from the MCU.

Moisture Sensor

Soil moisture sensors measure the volumetric water content in soil. [1] Since the direct 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. 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. Soil moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in soils called water potential; these sensors are usually referred to as soil water potential sensors and include tensio meters and gypsum blocks. Measuring soil moisture is important for agricultural applications to help farmers manage their irrigation systems more efficiently. Knowing the exact soil moisture conditions on their fields, not only are farmers able to generally use less water to grow a crop, they are also able to increase yields and the quality of the crop by improved management of soil moisture during critical plant growth stages. Soil moisture sensors are used in numerous research applications, e.g. in agricultural science and horticulture including irrigation planning, climate research, or environmental science including solute transport studies and as auxiliary sensors for soil respiration measurements.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 ARDUINO SOFTWARE

ARDUINO

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

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

Why Arduino?

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 of 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:

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 \$50

Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

Simple, clear programming environment - The Arduino programming 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 and 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.

5.2 PROTEUS

Proteus software

Proteus 8 is a best simulation software for various designs with microcontroller. It is mainly popular because of availability of almost all microcontrollers in it. So it is a handy tool to test programs and embedded designs for electronics hobbyist. You can simulate your programming of microcontroller in Proteus 8 Simulation Software. After simulating your circuit in Proteus 8 Software you can directly make PCB design with it so it could be a all in one package for students and hobbyists. So I think now you have a little bit idea about what is proteus software.

Proteus is a Virtual System Modelling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and LED and LCD displays and, if attached to the PC, switches and buttons.

Proteus comes with extensive debugging features, single stepping and variable display for a neat design prior to hardware prototyping. Proteus is the program to use when you want to simulate the interaction between software running on a microcontroller and any analog or digital electronic device connected to it.

Proteus was initially created as a multiplatform (DOS, Windows, Unix) system utility, to manipulate text and binary files and to create CGI scripts. The language was later focused on Windows, by adding hundreds of specialized functions for: network and serial communication, database interrogation, system service creation, console applications, keyboard emulation, ISAPI scripting (for IIS). Most of these additional functions are only available in the Windows flavour of the interpreter, even though a Linux version is still available.

Proteus was designed to be practical (easy to use, efficient, complete), readable and consistent.

Its strongest points are:

- powerful string manipulation;
- comprehensibility of Proteus scripts;
- Availability of advanced data structures: arrays, queues (single or double), stacks, bit maps, sets, AVL trees.
- The language can be extended by adding user functions written in Proteus or DLLs created in C/C++.

Get Proteus 8 Software?

After getting introduction about what is proteus software if you wish to download and give a try to proteus 8 software then labcenter is giving latest version of proteus, proteus 8 demo version for free of cost. The latest Proteus Software version is Proteus 8. You can download it from here.

Explanation

If you have clear idea about what is proteus 8 simulation software then you can start proteus 8 simulations at this point of time. I am sure that you have your proteus 8. If you don't have then download from above link. To learn Proteus 8 You can start with a series of video tutorials or you can download a Proteus 8 Software Tutorial PDF. But main and most important thing with this software is hands on experiments. The more you work the better you become There is no shortcut.

After making your Schematic you have to do PCB designing so here is the video tutorial which describes Single layer auto routing.

At first sight, Proteus may appear similar to Basic because of its straight syntax, but similarities are limited to the surface:

Language Features

Proteus has a fully functional, procedural approach; variables are untyped, do not need to be declared, can be local or public and can be passed by value or by reference; all the typical control structures are available (if-then-else; for-next; while-loop; repeat-until; switch-case); new functions can be defined and used as native functions.

Data types supported by Proteus are only three: integer numbers, floating point numbers and strings. Access to advanced data structures (files, arrays, queues, stacks, AVL trees, sets and so on) takes place by using handles, i.e. integer numbers returned by item creation functions.

Type declaration is unnecessary: variable type is determined by the function applied – Proteus converts on the fly every variable when needed and holds previous data renderings, to avoid performance degradation caused by repeated conversions.

There is no need to add parenthesis in expressions to determine the evaluation order, because the language is fully functional (there are no operators).

Proteus includes hundreds of functions for:

- accessing file system;
- sorting data;
- manipulating dates and strings;
- interacting with the user (console functions)
- Calculating logical and mathematical expressions.

Proteus supports associative arrays (called sets) and AVL trees, which are very useful and powerful to quickly sort and lookup values.

The functional approach and the extensive library of built-in functions allow to write very short but powerful scripts; to keep them comprehensible, medium-length keywords were adopted. The user, besides writing new high-level functions in Proteus, can add new functions in C/C++ by following the guidelines and using the templates available in the software development kit; the new functions can be invoked exactly the same way as the predefined ones, passing expressions by value or variables by reference. Proteus is an interpreted language: programs are loaded into memory, pre-compiled and run; since the number of built-in functions is large, execution speed is usually very good and often comparable to that of compiled programs. One of the most interesting features of Proteus is the possibility of running scripts as services or ISAPI scripts. Running a Proteus script as a service, started as soon as the operating system has finished loading, gives many advantages: no user needs to login to start the script; a service can be run with different privileges so that it cannot be stopped by a user. This is very useful to protect critical processes in air environments (data collection, device monitoring), or to avoid that the operator

inadvertently closes a utility (keyboard emulation). The ISAPI version of Proteus can be used to create scripts run through Internet Information Services and is equipped with specific functions to cooperate with the web server.

For intellectual property protection Proteus provides: digital signature of the scripts, by using the development key (which is unique); the option to enable or disable the execution of a script (or part of it) by using the key of the customer. Proteus is appreciated because it is relatively easy to write short, powerful and comprehensible scripts; the large number of built-in functions, together with the examples in the manual, keep low the learning curve.

The development environment includes a source code editor with syntax highlighting and a context-sensitive guide. Proteus does not need to be installed: the interpreter is a single executable (below 400 Kb) that does not require additional DLLs to be run on recent Windows systems.

How to Setup Proteus?

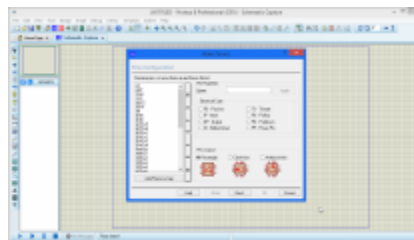
1. Start-up the Microsoft Windows.
2. Place the ECOM 4315 CD into CD ROM drive.
3. Double click on the software tools folder and then choose Proteus Simulator after that double click Proteus 7.1 folder and run the setup.
4. Press next until you reach the window which ask for the key.
5. From browse for key; browse until you reach the same folder where the setup exists, and then open the second folder and chose the MAXIM_LICENCE folder.
6. Then click on the top icon which is MAXIM and then press install.
7. Then choose yes and then close the window, after that browse and install the program.
8. After that run the patch which exist in the same folder where the MAXIM_LICENCE exists.

9. Then browse for c:\Program files\Labcenter electronics \Proteus.
10. Finally chose next, and then finish.

Get Started To Proteus

1. From start menu chose the PROTEUS, and then chose ISIS(blue).
2. To get a part click on Devices (P).
3. Write the name of the PIC16F84A.
4. To get LED write led then chose green led.
5. To get a resistor write 1k and chose the first part. Device Libraries

Proteus capture comes with pre-supplied libraries for TTL, CMOS, ECL, Microprocessor, Memory and Analogue ICs plus libraries with hundreds of named Bipolar, FET and Diode discrete semiconductors. Also included are manufacturer specific libraries from National Semiconductor, Philips, Motorols, Teccor, Texas Instruments, Dallas and Zetex amongst others - an expanding total, currently over 10,000 parts.



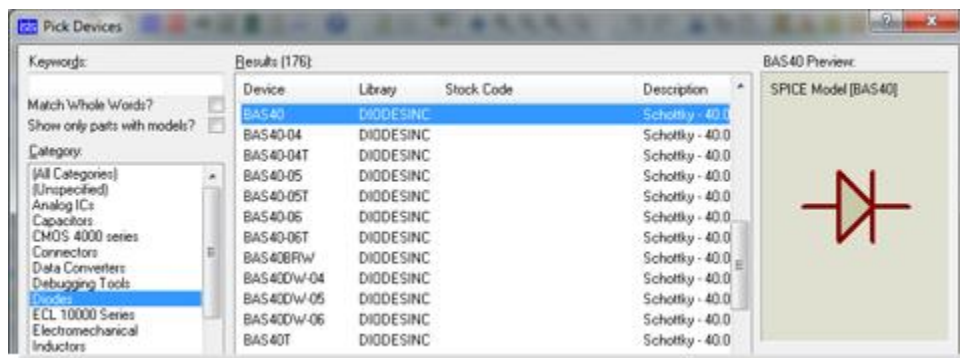
Importing Components from BSDL

Direct import of BSDL (Boundary Scan Description Library) files provides a semi-automatic way to quickly create and package larger components.

Libraries provided with Proteus also contain default properties for PCB packaging and/or simulator models. Both homogenous and heterogeneous multi-element devices are supported so you can place a relay's coil and contacts on separate parts of the drawing. Connector pins can also be distributed. Device editing is performed directly on the drawing using the general editing tools. Devices may be

constructed from lines, boxes, circles, arcs, text and special pin objects. Non-electrical symbols may also be created for use in small mechanical or block diagrams.

A summary listing of included embedded peripheral libraries can be found [here](#)



UBIDITS

Ubidots is an IoT Platform empowering innovators and industries to prototype and scale IoT projects to production. Use the Ubidots platform to send data to the cloud from any Internet-enabled device. You can then configure actions and alerts based on your real-time data and unlock the value of your data through visual tools. Ubidots offers a REST API that allows you to read and write data to the resources available: data sources, variables, values, events and insights. The API supports both HTTP and HTTPS and an API Key is required.

Your data will be protected with two more replication, encrypted storage and optional TLS/SSL data support. You can also customize permission groups to each module of the platform, making sure the right information is shown to the right user.

Born as an engineering services firm in 2012, Ubidots delivered end-to-end IoT solutions in tandem with its partner and co-founding company Netux, to remotely monitor, control, and automate processes for healthcare clients as well funded startups and Fortune 1,000s in the American Southeast and across Latin America.

Between 2012 and 2014, Ubidots accomplished countless internet connected projects across Healthcare, Energy / Utilities, Manufacturing, Transportation, and Retail – learning the many small characteristics of IoT and Cloud enablement that digital transformation experts cannot speak to unless they have gotten their hands dirty in the field.

With a strong backbone and a steadfast determination to become a product-led company, Ubidots joined the Boston MassChallenge Accelerator in 2013, pivoting into a global tech startup, gaining US venture and support, and leaving behind the local services business.

Since Ubidots pivot in 2014, Ubidots has become known within hardware, software, embedded engineering, and maker circles as the affordable, reliable, and most usable platform in the IoT Application Enablement ecosystem.

In line with Ubidots guiding principles, in 2018 Ubidots spun out and repurposed its first ever cloud version as Ubidots STEM Rather than retiring this shared-resources platform, Ubidots instead empowers 60,000+ students, makers and researchers to explore and develop technology and solutions, giving all users their first 3 devices free, always!

Ubidots Basics: Devices, Variables, Dashboards, and Alerts

The basics components of any Internet of Things application powered by Ubidots are: **Devices, Variables, Synthetic Variables Engine, Dashboards, and Events**. Within this article we will address each of these concepts as they relate to Ubidots IoT Development and Deployment Platform and how you can better organize your Ubidots Apps to best connect with the users.

Once your devices, variables, and general Application is assembled, give your App some layers with Ubidots Device Management to learn more about Ubidots internal architecture and how you can use Apps, Organizations, and Users to efficiently connect your data with those who should be using it.

1. Devices
2. Variables
3. Synthetic Variables Engine
4. Dashboard and Visualizations
5. Events.

1. Devices

A Ubidots' device is a virtual representation of a data-source or simply, an asset taking sensor data and transmitting said data through a connection protocol to Ubidots' cloud. [Click here](#) for current firmware examples and tutorials for connecting your device to Ubidots.

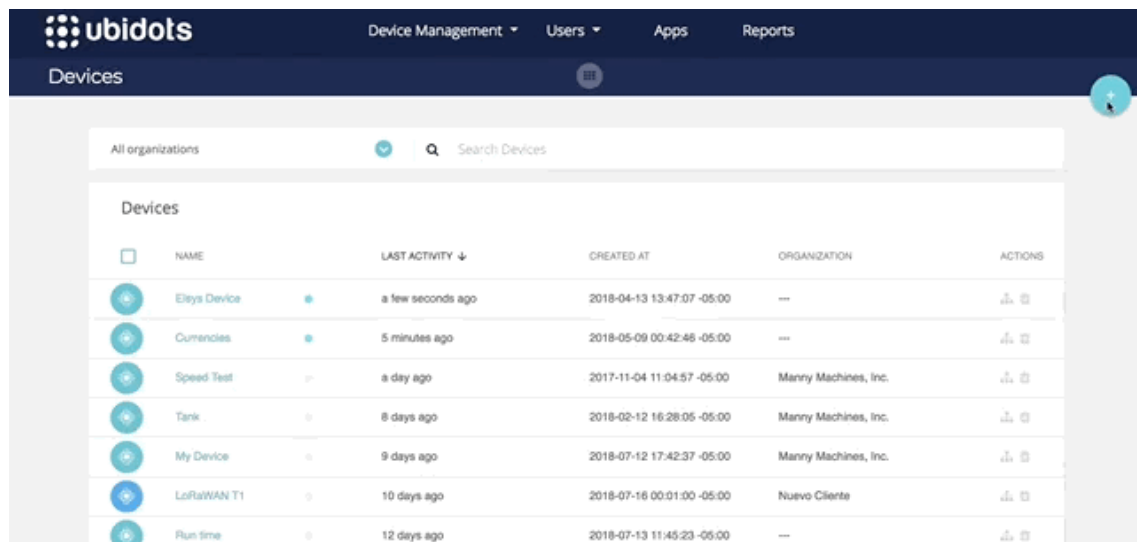
All devices are different, but the standard setup on any device entails:

- A library that should be installed in the device's IDE. (not required)
- Filling parameters for authentication and connection such as a TOKEN (i.e. the unique ID for each account or user), device and variable labels (i.e. devices and variables unique identifiers in Ubidots), Wi-Fi SSID and Password, depending on the device and the requirements.
- Making API request (i.e. a call from a device to the web server). Ubidots' libraries spare the need to manually make these requests. However, if you're working with a device not currently in the list of supported hardware, you can choose a connectivity protocol such as MQTT or HTTP and make a request using the API accordingly.

Creating devices: *there are three ways to create devices:*

1. Devices are created automatically in Ubidots the first time a dot is received to a user's private TOKEN or an Organizational TOKEN.

2. Clicking the "+" icon in the top right corner of any single device's screen.
3. Creating a Device Type which then creates a new device with pre-determined device properties, appearance, and variables and automate the on-boarding of thousands of devices.



	NAME	LAST ACTIVITY ↓	CREATED AT	ORGANIZATION	ACTIONS
	Elays Device	a few seconds ago	2018-04-13 13:47:07 -05:00	---	
	Cumencles	5 minutes ago	2018-05-09 00:42:46 -05:00	---	
	Speed Test	a day ago	2017-11-04 11:04:57 -05:00	Manny Machines, Inc.	
	Tank	8 days ago	2018-02-12 16:28:05 -05:00	Manny Machines, Inc.	
	My Device	9 days ago	2018-07-12 17:42:37 -05:00	Manny Machines, Inc.	
	LoRaWAN T1	10 days ago	2018-07-16 00:01:00 -05:00	Nuevo Cliente	
	Run time	12 days ago	2018-07-13 11:45:23 -05:00	---	

2. Variables

Once a device is created and receiving data from your hardware or another 3rd party data-source, the data will be presented in its raw or calculated form as a **variable**.

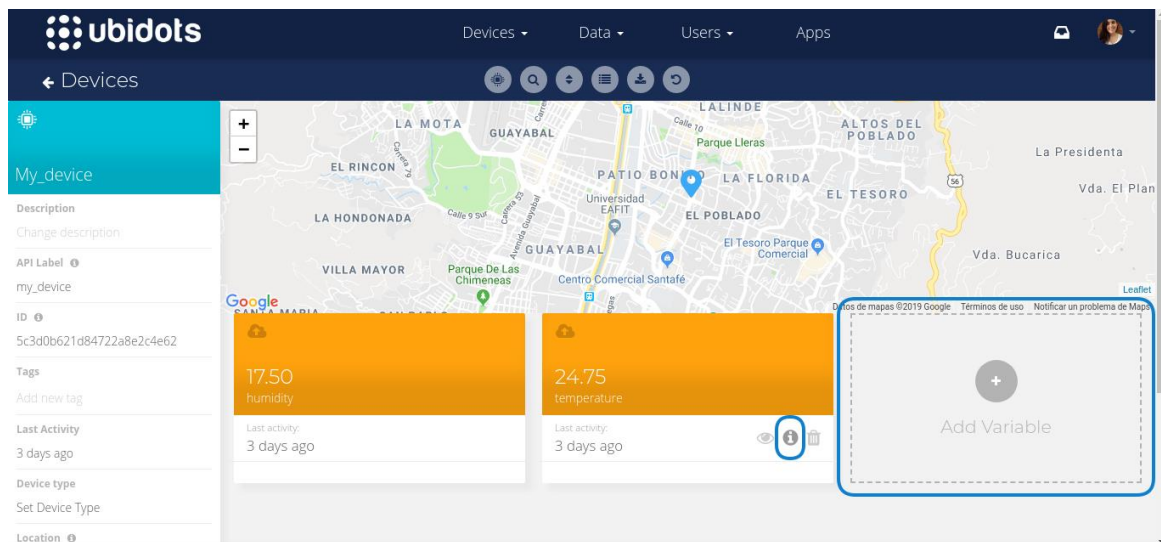
Types of Variables

- **Default** - raw data coming from devices (people counted).
- **Synthetic** - correspond to statistical or arithmetical operations of default variables in a determined time-frame (e.g. average daily traffic this month).

Below you can read more about synthetic variable setups.

Creating variables: there are two ways to create variables.

1. Assigning a label as a variable within your hardware's code.
2. Clicking the "+" icon found in any single device screen and assign a name which will also correspond to the variable's label. By default, Ubidots assigns an additional unique identifier to the variable called Variable ID. To find both the variable's ID and variable's label, select the "i" icon from the variables card.



3. Synthetic Variables Engine

Ubidots Synthetic Variables (SV) Engine is a simple and effective way to create insights from the raw data stored in Ubidots' Cloud. Using Ubidots SV Engine, users can create **Synthetic Variables** that calculate and enhance data to give data-driven decision to user of an Ubidots powered application.

For example, to calculate Average (mean) Energy consumed this week, we will need to create a synthetic variable called Energy to calculate the product of Current and Voltage ($P=C*V$): the **default variable Current is multiplied by Voltage**. Then average the product every week with built-in function **mean()**.

Synthetic Variables Analytics

Energy

Synthetic Variable

Enter your synthetic variable expression. [Learn more about synthetic variables.](#)

1 mean(**current** * **voltage** , "W")

Timezone:

America/Bogota

Click on a variable to add it to your expression:

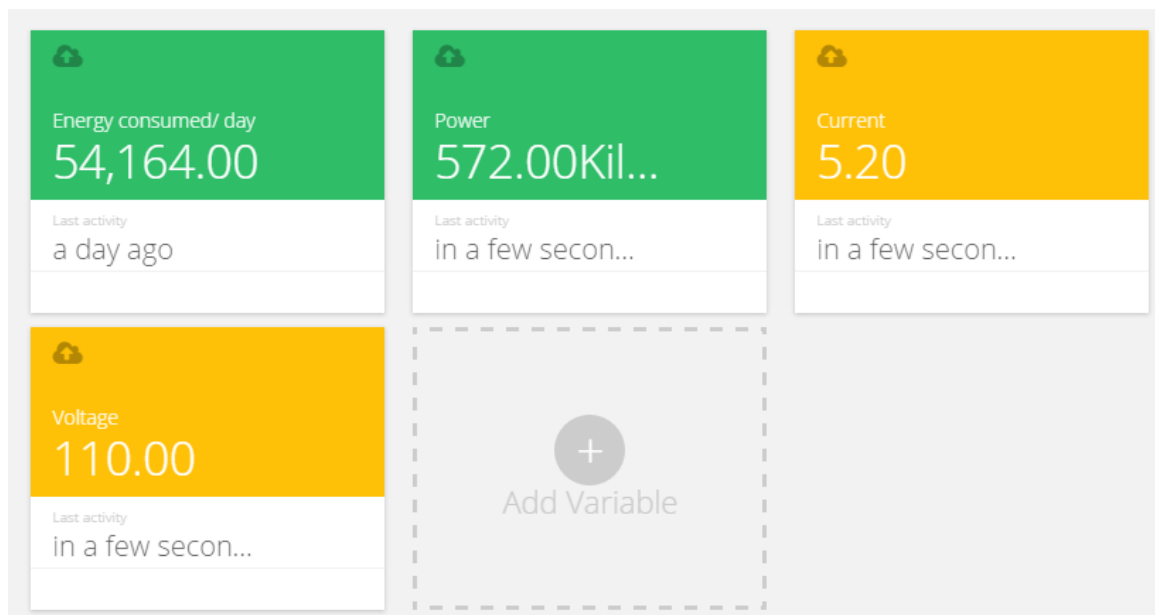
Current

Voltage

Cancel

Accept

Here is a complete list of available functions, such as **cos(x)** or **ceil(x)** functions, and more!



Note that default variables are yellow, and Synthetic Variables are green.

4. Dashboards and Visualizations

Dashboards are the human-machine interfaces where data is easily visualized. Your Ubidots account will let you create as many dashboards as needed, containing widgets and data-visualizations to comprehend your data at a glance. For additional details on your dashboard's construction, check out this tutorial.

5. Events

In Ubidots, Events are messages triggered and delivered through Email, SMS, Telegrams, Slack, Voice Call or webhook messages based on a customized design rule created in the application. If you wish to create alerts for your IoT application, check out this tutorial for additional support.

3 Event types:

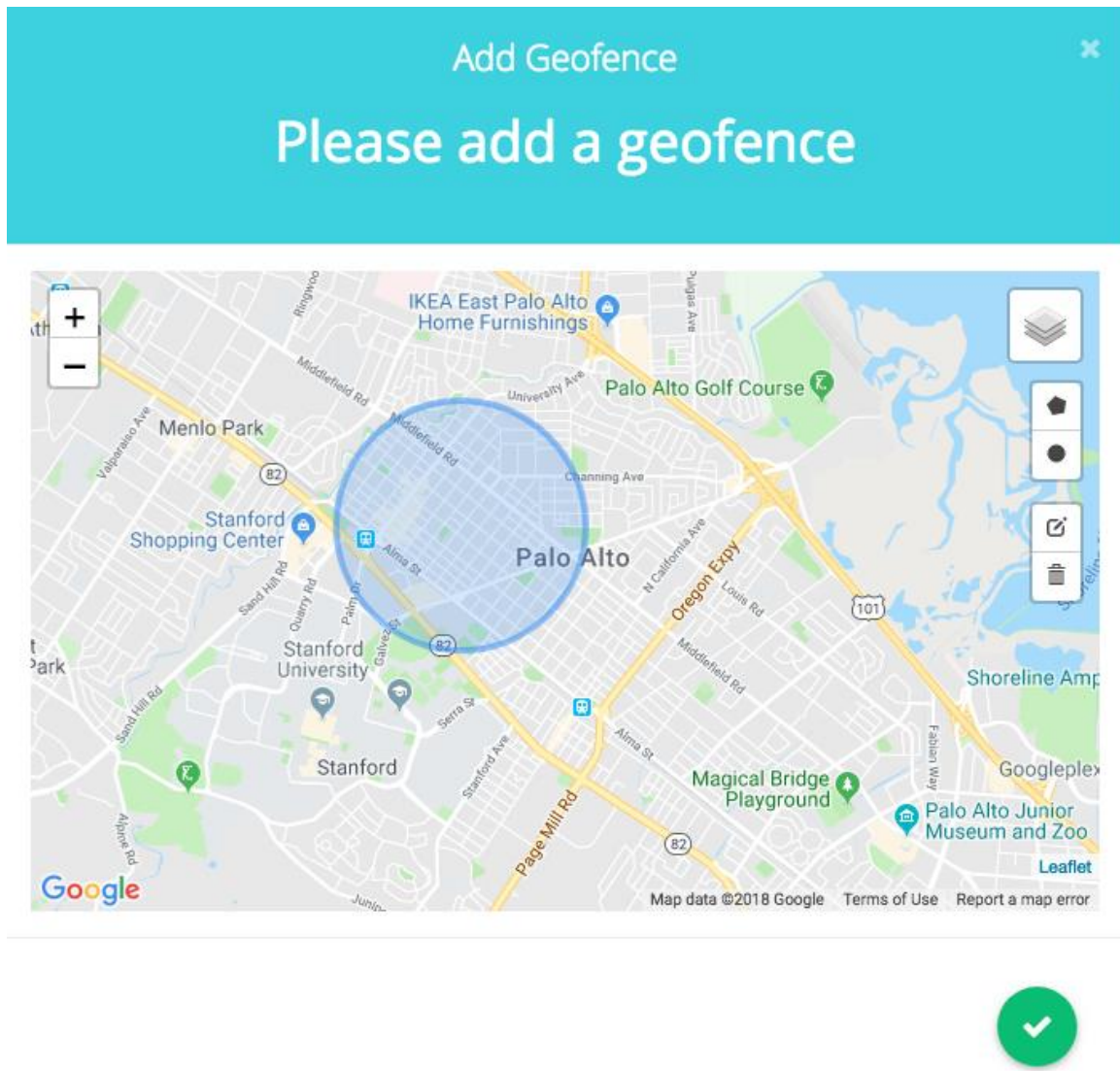
- When sensor readings reach a specific value:

The screenshot shows the 'If triggers' section of the Ubidots event configuration interface. A rule is being configured: 'demo' (with a plus icon) 'Value' (with a dropdown arrow) 'is' (with a dropdown arrow) 'Equal to' (with a dropdown arrow) '10' 'for' '0' 'minutes' (with a trash icon). A red box highlights the dropdown menu for the comparison operator, which includes the following options: 'Equal to', 'Greater than', 'Greater than or equal to', 'Less than', 'Less than or equal to', and 'Not equal to'.

- Variable is inactive during a specific amount of time:

The screenshot shows the 'If triggers' section of the Ubidots event configuration interface. A rule is being configured: 'demo' (with a plus icon) 'Has been inactive' (with a dropdown arrow) 'for' '10' 'minutes' (with a dropdown arrow). Below the rule, there are buttons for '+ And' and '+ Or' to add more conditions.

- When a device enters or exits a specific area (geofence):



You can also determine the format of alerts by choosing between emails, SMS, Telegrams or Webhook.

Applications

- Simple an easy to install and configure.
- Saving energy and resources, so that it can be utilized in proper way and amount.
- Farmers would be able to smear to right amount of water at the right time by automatic irrigation.

- Avoiding irrigation at the wrong time of day, reduce runoff from overwatering saturated soils which will improve crop performance. Automated irrigation system uses vales to turn motor ON and OFF. Motors can be automated easily by using controllers and no need of labour to turn motor ON an OFF.
- It is precise method for irrigation and a valuable tool for accurate soil moisture control in highly specialized greenhouse vegetable production.
- It is time saving, the human error elimination in adjusting available soil moisture levels

CHAPTER 6

CONCLUSION

This paper describes automated irrigation system using IOT. Internet of things and cloud computing collectively makes a system that control agriculture sector effectively. This system will sense all the environmental parameters and send the data to the user via cloud. User will take controlling action according to that this will be done by using actuator. This asset allows the farmer to improve the cultivation in a way the plant need. It leads to higher crop yield, prolonged production period, better quality and less use of protective chemicals.

REFERENCES

- [1] Asha, H. V., et al. "Smart Agricultural Monitoring System Using Internet of Things." *Emerging Research in Computing, Information, Communication and Applications*, Springer, Singapore, 2019, pp. 473-482.
- [2] Dasgupta, Ajanta, et al. "Smart irrigation: IOT-based irrigation monitoring system." *Proceedings of International Ethical Hacking Conference 2018*, Springer, Singapore, 2019.
- [3] Devan, P. Arun Mozhi, et al. "IoT Based Water Usage Monitoring System Using LabVIEW." *Smart Technologies and Innovation for a Sustainable Future*. Springer, Cham, 2019, pp. 205-212.
- [4] Garg, Bhumiika, et al. "IoT based Smart Agriculture Monitoring System", 2019.
- [5] Haque, Md Shadman Tajwar, et al. "Design and Implementation of an IoT based Automated Agricultural Monitoring and Control System." *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*. IEEE, 2019.
- [6] Kamienski, Carlos, et al. "Smart water management platform: lot-based precision irrigation for agriculture," *Sensors*, Vol. 19.2, 2019, p. 276.
- [7] Khattab, Ahmed, et al. "An IoT-based cognitive monitoring system for early plant disease forecast." *Computers and Electronics in Agriculture*, Vol. 166, 2019, p. 105028.
- [8] Raj, Shivang, et al. "IoT based model of automated agricultural system in India." *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019.
- [9] Ruan, Junhu, et al. "A life cycle framework of green IoT-based agriculture and its finance, operation, and management issues," *IEEE communications magazine*, Vol. 57.3, 2019, pp. 90-96.
- [10] Sambath, M., et al. "IoT Based Garden Monitoring System." *Journal of Physics: Conference Series*, IOP Publishing, Vol. 1362, No. 1, 2019.