

IOT FLOOD MONITORING USING RASPBERRY PI PICO

A Project Report Phase – II

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

A.SAIKUMAR REDDY (39130017)

B. VIVEK VARDHAN REDDY (39130052)



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

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL 2023



SATHYABAMA

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

Accredited with “A” grade by NAAC.

Jeppiaar Nagar, Rajiv Gandhi Salai,

Chennai - 600 119

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 A. SAIKUMAR REDDY (39130017) and B.VIVEK VARDHAN REDDY(39130052) who carried out the project entitled “IOT FLOOD MONITORING SYSTEM USING RASPBERRY PI PICO” under our supervision from November 2022 to April 2023.

Internal Guide

Mr.K.T. ILAYARAJAA, 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, A.SAI KUMAR REDDY (39130017) and B.VIVEK VARDHAN REDDY (39130052) here by declare that the Project Report entitled “IOT FLOOD MONITORING SYSTEM USING RASPBERRY PI PICO” done by us under the guidance of Mr.K.T.ILAYARAJAA M.E.,(Ph.D.),Assistant professor Dept. of Electronics and Communication Engineering SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, CHENNAI, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE:

SIGNATURE OF CANDIDATES

1. 
2. 

ACKNOWLEDGEMENT

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

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

We would like to express my sincere and deep sense of gratitude to our Project Guide **Mr.K.T. ILAYARAJAA, M.E., (Ph.D.)**, Assistant Professor, Dept. of Electronics and Communication Engineering for his valuable guidance, suggestions and constant encouragement for the successful completion of our project work.

We 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

The natural calamities like flooding turn up drastically, and it severely affects standard of living. In this paper the development of flood monitoring system using IOT to keep track of the conditions nearby the reservoir with the help of Raspberry Pi Pico, the prototyping platform, and the compactible sensors such as level, temperature, humidity, and flow distinctly presented. Firstly, the hardware unit is placed in the flood prone areas, the Wi-Fi module (ESP8266) act as the transmitting unit and the sensors associated with the system measures the corresponding parameters. Then the accurately measured parameters are displayed through the LCD display and passed to the IOT web application. Here the thing speak web application is used to store data in private channel and the web application then alerts the authority and people while flood occurs.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	V
	LIST OF FIGURES	VIII
	LIST OF ABBREVIATIONS	XI
1	INTRODUCTION	1
	1.1 Methodology	2
	1.2 Motivation	3
2	LITERATURE SURVEY	4
3	EXISTING AND PROPOSED METHODOLOGY	7
	3.1 Existing system	7
	3.2 Proposed system	7
	3.3 Project design	8
4	SYSTEM ANALYSIS	9
	4.1 Hardware requirements	9
	4.1.1 Raspberry pi Pico	9
	4.1.2 Node MCU	17
	4.1.2.1 History	18
	4.1.2.2 ESP8266 Arduino core	18
	4.1 DHT Sensor	22
	4.1.3.1 Humidity sensor	22
	4.1.3.2 DHT11 Sensor	23
	4.1.4 Ultrasonic (uv) sensors	27
	4.1.4.1 Applications	31
	4.1.5 Water level sensor	34

5	SOFTWARE DESCRIPTION	40
	5.1 Thonny Ide	40
	5.2 Python	40
6	RESULTS AND DISCUSSION	42
7	CONCULSION AND FUTURE SCOPE	44
	REFERENCES	45

LIST OF FIGURES

FIGURE No	FIGURE NAME	PAGE No
3.1	Block Diagram	8
4.1	Raspberry Pi Pico Hardware	9
4.2	Node MCU	17
4.3	NodeMCU Development Board (Version1)	19
4.4	NodeMCU Development Board (Version2)	19
4.5	DHT11 SENSOR	23
4.6	MCU AND DHT11 Signal	24
4.7	Timing difference for transmitting	25
4.8	Data signal in Sequence	25
4.9	Ultrasonic Sensor	27
4.10	The Frequency Range	29
4.11	Distance to the object	29
4.12	Beam Profiles	30
4.13	Flow Detection	33
4.14	Water Sensor	36

LIST OF ABBREVIATIONS

IDE	Integrated Development Environment
LDR	Light Dependent Resistor
USS	Ultra Sonic Sensor
CMUT	Capacitive Micromachined Ultrasonic Transducer
LDV	Laser Doppler Velocimeter
LDA	Laser Doppler Anemometer
PCB	Printed Circuit Board
LED	Light Emitting Diode

CHAPTER 1

INTRODUCTION

Natural calamities happen everywhere in the world, and which affects the human life and economy of the country. Economy and growth of any country depends upon the agriculture; hence the proper alert makes the farmers vigilant to protect the crop from flooding. To detect and avoid flood like disastrous calamities in a timely manner, current world technology plays a vital role. We can prevent natural disaster caused by flood, with the aid of an IOT based early flood related parameter monitoring and detection system and its avoidance using the Arduino project, is proposed as a solution to the mentioned problem. The proposed model is very much utilized for monitoring of the water level, flow variations, humidity, and temperature variation in the river and the same can be used at dam or reservoirs. The measured values are regularly updated on the web server which is very much useful to send flood alerts to authority and people for faster action. The entire system consists of five different Arduino compactible sensors which are temperature, humidity, water level, flow and ultrasonic sensors. Also, it consists of an Arduino controller, a Wi-Fi module, an LCD display an alarm and an IOT remote server-based platform. In this advanced system the initial stage indicates the level of water and the other parameters like flow rate temperature and humidity. Then this information is passed to the web server or the IOT via a Wi-Fi module, here the ESP8266 is used as Wi-Fi module. Which transmit and DHT11 is the temperature and humidity sensor, it is a basic low-cost digital temperature and humidity sensor. And HC-SR04 ultrasonic sensor used as the water level sensor, which works on the SONAR principle. In this paper the main objective is implement a system which covers both the IOT based system and the sensor network interfaced with both ESP8266 and the Arduino Uno R3 board for detecting floods and for sending alert to organizations and the society. The LED and buzzer act as alerting system when there is rise in the water level and the associating parameters. Nowadays at most of the times the ordinary system notifies only the respective governed registered organizations, result in the slowdown of the process in rescuing objectives are to implement a system which covers both the ESP8266- based technology, sensor network.

components, IOT and web applications for detecting the floods for sending an alert to the organization.

1.1 Methodology

After going through literature survey and various research papers we finalized our hardware and software requirements. Various natural factors, which includes humidity, temperature, water level and flow level are observed by system to detect flood. Our system consists of different sensors which helps to collect data for individual parameters.

1. For detecting changes in humidity and temperature the system has a DHT11 Digital Temperature Humidity Sensor. It is a sensor which detects humidity and temperature.
2. The water level is always under observation by an Ultrasonic sensor, which works by constantly monitoring as water levels rise and fall. Once the water level increases beyond threshold, a trigger is generated which sends an Email Alert indicating the rise of Water and possibility of Flood.
3. The Flow sensor on the system keeps eye on the flow of water. The speed changes when water falls on rotor which makes it to rotate.
4. Once the hardware and software are set up, the Raspberry Pi can start collecting data from the sensors. This data can be stored locally on the Pi or transmitted to a cloud server for analysis.

After the successful completion of hardware setup, we move towards software setup and using Arduino IDE and Visual Studio Code. We created a Project Email for sending Email Alerts. Using Python Scripting and interfacing the Arduino outputs with Python, the program reads the inputs from the sensors via Arduino. We set a threshold value for the water level, once the water crosses the threshold level, a trigger causes the program to send an Email Alert notifying people regarding rise in water level and possibility of flood.

1.2 Motivation

Declining quality of water is detrimental to the health, environment, and economy. David Malpas, President of the World Bank cautions about the influence on economy: "Deteriorating water quality is stalling economic growth and exacerbating poverty in many countries". It means that, if the biological oxygen demand, the pointer used for measurement of organic pollution in water, surpasses the threshold, the Gross Domestic Product (GDP) growth of the constituencies surrounded by the allied water basins will decrease by a third. The consequences of water pollution or poor water quality are:

- Destruction of biodiversity: Pollution of water reduces aquatic ecosystems and initiates unrestrained increase of phytoplankton in water resources.
- Food chain contamination: Fishing carried out in polluted water resources and utilization of wastewater for agriculture and livestock husbandry may lead to addition of toxins or contaminants into foods that are injurious to the health after consumption.
- Scarcity of drinkable water: If pollution of water increases or quality of drinking water is not maintained, then there will be no clean water for drinking or public health or sanitization, in rural as well as urban areas.
- Disease: According to WHO (World Health Organization) information, roughly 2 billion people across the world do not have any option for pure water resources, but they have to drink water polluted by excrement, which exposes them to many ailments.
- Infant mortality: As per WHO, diarrheal diseases associated with lacking hygiene results in death of nearly 1,000 children per day across the world. Water quality monitoring is demarcated as the assortment of data at set or desired places and at periodic intervals for providing information that might be accustomed to describing present conditions of water.

CHAPTER 2

LITERATURE SURVEY

[1]. Flomosys: A flood monitoring system

The expansion of the Internet of Things (IoT) has led to numerous innovations in the industry, including improvements to existing systems. Disaster prevention and monitoring systems are a prime example of such systems. Every year, there are significant and preventable financial losses, not to mention the safety hazards caused by floods. To warn people ahead of time, we can deploy low-power wireless sensor nodes to send readings across any terrain to a cloud platform, which can perform pattern analysis, prediction, and alert forwarding to anyone's cellular device. In this paper, we propose Flomosys, a low-cost, low-power, secure, scalable, reliable, and extensible IoT system for monitoring creek and river water levels. Although there are multiple competing solutions to help mitigate this problem, Flomosys fills a niche not covered by existing solutions. Flomosys can be built inexpensively with off-the-shelf components and scales across vast territories at a low cost per sensor node. In this paper, we present the design and implementation of this system as well as real-world test results.

[2]. Design of iot based flood monitoring and alerting system.

The main objective of this paper is the development of an automatic flood detection and prevention system, mainly oriented for residential use, but with possibility to be used in industrial applications as well. According to a study conducted by a German company regarding water damage occurrence and floods in EU households, the participants believed that water damage occurs in only 37% of households. Over 57% of European households suffer from water damage. Different causes can lead to floods and unavoidable water damage. The study participants considered that there are three main causes which may lead to floods: bursting pipes, small cracks due to corrosion, material fatigue or due to frost and defective laundry machines or dishwashers. The occurring ratio for each type of incident has been established as well. The first position is being taken by bursting pipes, the most common incident, responsible for over 48% of the total residential flood incidents. The risk for this type of incident to occur is 21%. Defects caused by corrosion or frost take the second position in the ranking, with a total of 28% of total incidents and the occurring rate is

19%. The last place is being taken by defects caused by small pipe cracks due to material fatigue, the chances for this kind of defect to occur rise to 19% of total. This paper aims to develop an automatic flood detection and prevention system, especially in the residential environment. The aim is to obtain an economical and effective alternative to the technical solutions available on the market. A wireless solution that is not dependent of a connection or a central control unit will be developed.

[3]. IOT based flood detection and notification system using decision tree algorithm.

Floods are basic catastrophic events that allow extreme desolation of any nation. They are typically brought about by precipitation and overflow of waterways, especially during heavy stormy season. This project aims to monitor the flood condition and send alert if there is an occurrence of risk through IOT. The measurement of rising water level is done to detect the flood. The system uses three sensors to detect temperature, humidity and water levels at every stage. The detected sensor values are processed using PIC Microcontroller and it is transmitted to IOT through Wi-Fi module. The system instantaneously uploads and broadcasts sensor values through cloud. The decision tree algorithm is implemented to perform the classification process. The experimental results for Correctly Classified Instances and Minimum Absolute Error indicate that the proposed Decision Tree Algorithm gives 99.6% classification accuracy with Minimum Mean Square Error than Hyper Pipes Algorithm.

[4]. Flood monitoring system using IOT

Flood is one of the catastrophic events that cannot be maintained at a strategic distance from. It happens very quickly and influences such a significant number of lives and properties. Every year, the death rate due to flood increases due to lack of early warning. Majority of the people in general cannot screen, and thus do not know when the flood would occur because they do not have any factual information of the climatic conditions. The proposed system can reduce the few disadvantages of the already created systems. If the general society has a web connection, they would be able to screen what is going on as well as can have an idea whether there is an upcoming flood. The proposed system is less expensive towards design and maintenance.

[5]. Implementation of flood warning system using iot

In this paper, an IoT based system that alerts the authorities about an imminent flood is introduced. The water flow rate is detected using IR Sensor and the water level is detected using Ultrasonic sensor in rivers and drains and sent over Wi-Fi to the main controller which checks these received sensor values and if the values vary beyond a particular threshold, the alarm (buzzer) and also a alert Short Message Service (SMS) is sent to mobile. A website provides the plot of the water levels and the water flow rates. Also, the weather forecast information obtained from the internet is provided on the website. The controller used to process the sensor data, obtain the water level, and flow rate values, and transmit the obtained values is the Node Microcontroller Unit (NodeMCU). The controller used to receive the transmitted data, interpret it, perform the actuation (buzzer and SMS) and update the website is RaspberryPi3(RPi3).

CHAPTER 3

EXISTING AND PROPOSED METHODOLOGY

3.1. Existing system

- Now a day's water is polluted due to many reasons. In this existing system the equipment cost is high, and it takes a lot of time to process.
- Traditional methods have the disadvantages like complicated methodology, long waiting time for results low measurement precision and high cost.
- So, with the implementation in the technology, we use different methods and techniques to check quality of water.
- There is a disadvantage in the existing system that the system has high complexity and low performance.

3.2. Proposed system

To detect a flood the system observes various natural factors, which includes humidity, temperature, water level and flow level. To collect data of mentioned natural factors the system consists of different sensors which collect data for individual parameters. For detecting changes in humidity and temperature the system has a DHT11 Digital Temperature Humidity Sensor. It is an advanced sensor module which consists of resistive humidity and temperature detection components. The water level is always under observation by a float sensor, which works by opening and closing circuits (dry contacts) as water levels rise and fall. It normally rests in the closed position, meaning the circuit is incomplete and no electricity is passing through the wires yet. Once the water level drops below a predetermined point, the circuit completes itself and sends electricity through the completed circuit to trigger an alarm. The flow sensor on the system keeps eye on the flow of water.

3.3. PROJECT DESIGN

Block diagram:

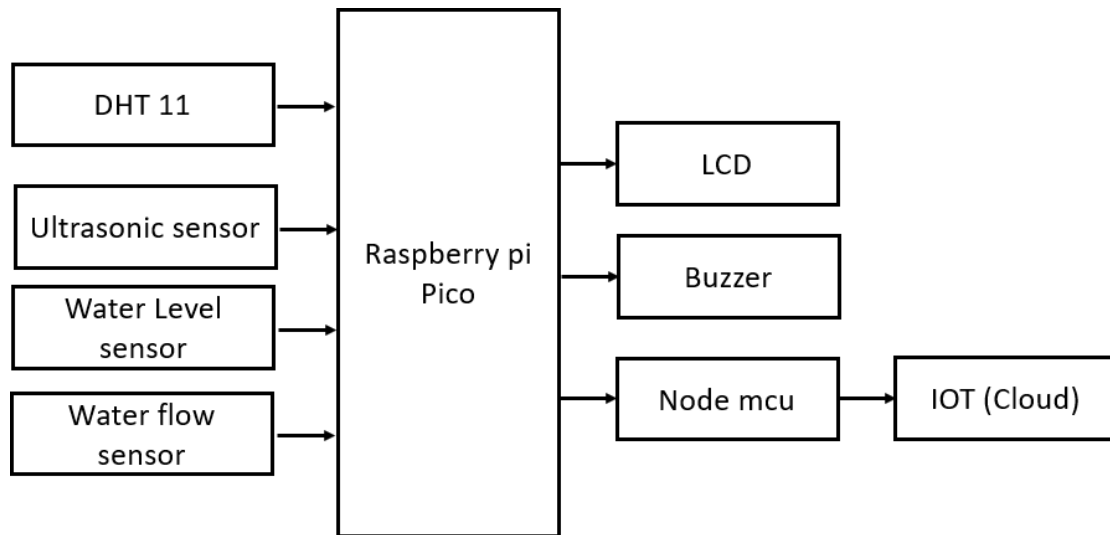


Fig: 3.1 Block diagram of IOT Flood Monitoring

CHAPTER 4

SYSTEM ANALYSIS

4.1. Hardware requirements

- Raspberry pi Pico
- Node MCU
- Ultrasonic sensor
- Water level sensor
- DHT11 sensor
- Water flow sensor
- Buzzer
- LCD

4.1.1. *Raspberry pi Pico*

Raspberry Pi Pico is a microcontroller board based on the Raspberry Pi RP2040 microcontroller chip. It is designed as a low-cost, high-performance microcontroller board with flexible digital interfaces. In the field of MCUs, Raspberry Pi Pico has become one of the most popular topics recently. To help beginners get started with Raspberry Pi Pico quickly, we are releasing the Starter Guide of Raspberry Pi Pico Based on Micro Python. Through four chapters and a total of 16 lessons of study, this course can help you get started with Raspberry Pi Pico easily. With its help, you can learn Micro Python from scratch, and build some interesting projects. You do not need to have knowledge of Micro Python programming or electronics to begin. The course will take you through learning this knowledge step by step, and quickly put it into practice in each project.



Fig: 4.1 Raspberry Pi Pico Hardware Setup

No one expected that Raspberry Pi, the most popular single-board computer maker in the world, would suddenly release a microcontroller of its own. What's more surprising is that Raspberry Pi Pico does not base its design on the common ESP32 or SAMD21, but instead a brand-new microcontroller chip: the RP2040 microcontroller. The RP2040 microcontroller is a microcontroller chip independently designed by Raspberry Pi and is powered by a dual-core ARM Cortex-M0+ processor that runs up to 133Mhz.

GPIO Pins In addition to the powerful new chip, the board of Raspberry Pi Pico exposes 26 multifunction GPIO pins, including 2 * SPI, 2 * I2C, 2 * UART, 3 * 12-bit ADC, and 16 controllable PWM channels. I will explain the functions of these pins in later chapters. In addition to these GPIO pins, Pico also has eight ground pins and a series of power pins. In this course, however, we will not utilize them as we will not be performing tedious wiring with breadboards or DuPont wires when building projects. The expansion board **Grove Shield for Pi Pico** will help you do this instead.

Grove Shield for Pi Pico is a plug-and-play expansion board for Raspberry Pi Pico. It integrates various kinds of Grove connectors on one compact expansion board, including 2 * I2C, 3 * ADC, 2 * UART, 3 * Digital ports, SWD debug interface and an SPI pin. When using Pico and Grove modules to build a prototype or project, you no longer need to do complicated wiring work. You simply need to plug the Pico pins into the interface of the Shield, and directly connect various Grove electronic modules to Pico using Grove wires.

The **Grove Starter Kit of Raspberry Pi Pico** is a learning kit that can help you quickly start building projects with Raspberry Pi Pico. The kit contains 14 handpicked modules, including 5 sensors/ 5 actuators/ 2LED/ 1 LCD display/ 1 Grove Shield for Pi Pico. Let's take a look at these Grove modules one by one.

Grove - Light Sensor Grove - Light Sensor is an analog module, which can output the amount of light detected as various electrical signals to an attached microcontrollerboard.

Grove - Temperature & Humidity Sensor Grove - Temperature & Humidity Sensor is mainly used to detect the temperature and humidity values in the environment. Grove - Rotary Angle Sensor Grove - Rotary Angle Sensor can change the resistance value to achieve different output levels by rotating the 300-degree adjusting knob on the potentiometer. It has a maximum resistance value of 10KΩ. Grove - Sound Sensor Grove - Sound Sensor can detect the sound intensity in the environment. The main component of the module is a simple microphone, which consists of a L358 amplifier and an electret microphone. Grove - Mini PIR Motion Sensor Grove - Mini PIR Motion Sensor uses passive infrared radiation to detect motion, and is often used in projects such as alarm systems, visitor systems, light switches, etc.

Grove - Button Grove - Button is an independent "momentary on/off" button. "Momentary" means that the button rebounds on its own after it is released. The button outputs a high signal when pressed, and a low when released. Grove - LED Pack Grove - LED Pack includes LEDs in four different colors: red, green, blue and white. You can change the light color at any time by swapping out the LED with other LEDs in the LED Pack. There is also a knob on the module to control the LED brightness. Grove - RGB LED (WS2813 mini) Grove - RGB LED (WS2813 Mini) can be set to full-color display and can be connected to other RGB LED rings or RGB LED bars in series. Grove - Passive Buzzer Grove - Passive Buzzer is a low-cost tunable passive buzzer. Grove - Relay Grove - Relay is used to supply greater power to modules that require it, such as an external motor. It can handle currents of up to 5A at 250VAC for long periods of time. When the relay is switched on, the LED on the module will light up. Grove - 16x2 LCD Grove - 16x2 LCD is a blue backlight 16x2 LCD display. It has excellent high contrast and is easy to use. Grove - Servo Grove - Servo is a rotary actuator that allows for precise control of angular position, which makes them suitable for projects where precise position control is needed. Grove - Mini Fan Grove - Mini Fan is equipped with a DC motor, a motor drive board and a soft fan-blade. The soft fan-blade is safe for use, and the easy-to-use Grove connector can let you build your own project prototype quickly.

That is the basic information about the electronic modules included in the Raspberry Pi Pico and the Grove Starter Kits of Raspberry Pi Pico. More interesting information about them will be presented as we go along in the later courses.

Micro python

To build interesting projects with Raspberry Pi Pico, you need to learn how to program it first. Raspberry Pi Pico supports two types of programming languages: C++ and MicroPython. C++ is a general-purpose language developed on the basis of the C language. It is often used in the development of ESP32 and Arduino, so most microcontroller users are familiar with this language. However, to make things simple in this course, we will use another language to program Pico: MicroPython.

Python is an interpreted general-purpose language. Since the first version of Python was designed and released by Guido van Rossum of the National Research Institute for Mathematics and Computer Science in 1991, Python has rapidly become one of the most popular languages in the world with its concise syntax, excellent code readability and hot community resources.

The computer cannot directly understand the high-level language we use when programming. When the computer executes the program, it needs to translate the code written in a high-level language into a machine language that it can understand in some way and then execute it. There are two ways to translate high-level language into machine language: using a compiler or using an interpreter to translate a high-level language. Therefore, we refer to programming languages that use a compiler for translation as a compiled language, while a language that uses an interpreter for translation is known as an interpreted language.

However, the powerful Python is not omnipotent. Although Python performs well on desktop computers, large terminals and servers, it cannot be deployed on microcontrollers with limited resources and small memory. As a result, Micro

Python, a Python-based interpreted language specially designed for microcontrollers, was born. Micro Python optimizes Python cleverly, and the optimized Micro Python can run perfectly in all kinds of resource-constrained microcontrollers. Micro Python also inherits various advantages of Python, such as its ease of use.

Micro Python has made strides in keeping compatibility with ordinary Python as much as possible. If you are a beginner, you will learn a lot about Python while learning MicroPython. On the other hand, if you already have some understanding of Python, you will be able to quickly get started with MicroPython while deepening your knowledge.

Thonny is a Python/ Micro Python IDE (integrated development environment) for programming beginners, developed by the University of Tartu in Estonia. The platform integrates many tools that users need to use when developing. Its interface is simple and easy to understand, which is very suitable for beginners. In this course, all of our development work will be done in Thonny

In this lesson, we will write and run our first program on Pico: Hello World! But before we start, we will need to do some preparatory work. The installation of Thonny is quite simple. Since Python 3.7 is built into Thonny, all you need is a simple installer and you are ready to start learning to program. First, click [Thonny.org](https://thonny.org) to enter the download page, and select the installation package of Thonny to download in the upper right corner of the page according to your operating system. After downloading the installation package, double-click to open it and follow the on-screen instructions to install it.

The main interface of Thonny is simple and can be divided into the following four parts:

1. **Toolbar:** the toolbar offers icon-based quick-access to commonly used program functions, such as new, open, save, run the current script, stop, etc.
2. **Script Area:** the script area is the core area of Thonny, where your Python programs are written.
3. **Python Shell:** The Shell allows you to run console commands and provides information about running programs.

4. Interpreter: the interpreter allows you to change the interpreter version used to run your programs. Click on "Python 3.7.9", look for "Micro Python (Raspberry Pi Pico)" in the menu, click "OK", and switch it to Pico's exclusive interpreter. We can also switch languages in "Theme & Font."

In addition to installing Thonny on the computer, we also need to install the Micro Python firmware on the Pico itself. The Micro Python firmware download link is preloaded inside Pico, so we can directly use this link to enter the download interface. First, hold down the "BOOTSEL" button on Pico; then, while still holding it down, connect Pico to a computer using a USB cable.

At this point, a drive called RPI-RP2 should show up as a removable device on your computer. Navigate to it to find this file: INDEX.HTM.

Open INDEX.HTM to be taken to the download page. Drop down and click on the Micro Python tab where you can find information about Pico and Micro Python. Click on the "Download UF2.file" button in this tab to download the Micro Python firmware.

Finally, drag the downloaded Micro Python firmware into the RPI-RP2 directory to complete the firmware installation.

Just as a young child learns to crawl before they walk, programmers learn how to output "Hello, world" as the first step in learning a new language. Now, let's take a look at how to write your first program. First, connect Pico to the computer using a USB cable; then, open Thonny, and click on the "restart back-end process" button on the toolbar.

After connected, click on the Shell area and type the instruction: `print ()` is a common function used as a printout. When you need to output some data to the Shell, you can use this function. When you press ENTER after you type the instruction, you'll see the execution result of the input instruction is displayed in the Python Shell area, which means, the message "Hello, World!" is printed out in the Python Shell area.

To make your program in the script work, you will have to click the “Run current script” button in the toolbar. If you are running a new program that has never been saved, then when you click the “Run current script” button, Thonny automatically shows you a menu with two options the program to “This computer” or “RaspberryPi Pico”. You can choose where to store the program according to your own needs.

Once your program is saved, you will see the execution result of the program in the Shell area.

Where you save your program depends on your needs. If you want to show your program to your friends, you should save it to your Pico and take it with you. If you simply want to save the program for later use, just save it to your computer.

The next time you want to open your saved program, click the “open” button on the toolbar to find the corresponding program.

However, we so far run the program on Pico without using any other external electronic hardware. In this lesson, we will try to attach other electronic hardware to Pico. More specifically, we will use an LED, and write a program to light up that LED. But before you start programming, let us first learn a little more about the program structure.

The programs that we write with Micro Python are generally composed of three basic program structures. Complex programs are built with a combination of these basic structures, so it is important to learn them first. They are sequence structure, loop structure and selection structure.

Sequence structure is the most basic program structure. In sequence structure, programs run line-by-line, top-to-bottom, in turn. For example, when the following program runs, it prints “Hello” first, and then “World”.

If you look closely, you can find that, unlike other languages where a specific terminator is required to end each line of the statement, you simply need to press ENTER to end a line in MicroPython.

Now, let's go back to the project and practice what we've just learned. To get started with this lesson, we will use the Grove Shield for Pi Pico to connect Pico to other Grove electronics for more interesting projects. Upon observation, you might notice that Pico does not come with the metal pins to connect the Grove Shield for Pi Pico. To do so, you need to solder the headers for Pico. First, get everything you need to solder the headers: a soldering iron, some solder, a cleaning sponge, a stand, two 20-pin headers, a breadboard, and of course, your Pico. For easy soldering, we can use a breadboard to fix the two 20-pin headers first. With the black plastic block on the headers as the boundary, slowly insert the long end of the headers into the breadboard. When inserting, make sure that the two headers are aligned up and down, and spaced the same width as Pico.

Then, turn your Pico right side up and make sure the reserved pinholes on the PCB board are aligned with the headers that are fixed on the breadboard. After alignment, slowly insert the two headers into the reserved hole of Pico and keep pushing until the black plastic block on the headers are sandwiched between your Pico and your breadboard. Currently, you will see a small length of each pin sticking up out of the reserved pinholes on the PCB board.

Under no circumstances should you ever touch the metal parts of a soldering iron. After cleaning, use the tip of the soldering iron to heat the pin and the gold-colored pad beneath it closest to you. Pick up some solder with the other hand, and slowly push it from the opposite direction of the soldering iron into the joint of the pin and the pad. The heated pin and pad melt the solder and make it flow around the pad. After completing the soldering in one direction, continue to push the solder in other directions until the pad is completely covered by the solder.

Attention! Do not use too much solder when soldering. If the solder overflows to adjacent pads, a short circuit will occur later when using the Pico. Well, congratulations on your soldering of the first pin! Now, you just need to solder the remaining 39 pins in the same way. When soldering, remember to wipe your iron with a clean sponge from time to time to keep the tip clean and shiny.

4.1.2. NodeMCU

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 Expressive Systems, and hardware which is based on the ESP-12 module.



Fig:4.2 Node MCU ESP8266

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. 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 Extensa LX106 core, widely used in IoT applications (see related projects).

4.1.2.1. History

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, 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 R committed the Gerber file of an ESP8266 board, named devkit v0.9.^[13] 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.

4.1.2.2. ESP8266 Arduino core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate 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". 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 :4.3 Node MCU Development Board (Version1)

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

Node MCU Dev Kit/board consist of ESP8266 WIFI enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Expressive 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.

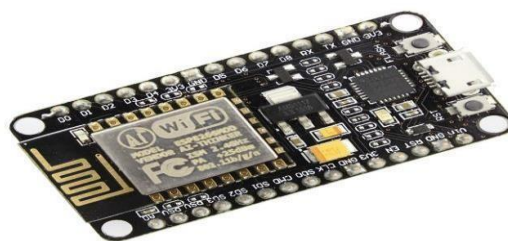


Fig :4.4 Node MCU Development Board (Version2)

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 ESP lorer 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 ESP lorer IDE.

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 executes 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.

4.1.3. DHT sensor

4.1.3.1. 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 perfectly accurate sensors, they are still expensive for hobbyists use. This article discusses the DHT11 sensor which also provides calibrated digital outputs for temperature and humidity but is relatively lot cheaper than the Sensorium 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.

4.1.3.1. 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? s before it releases it.

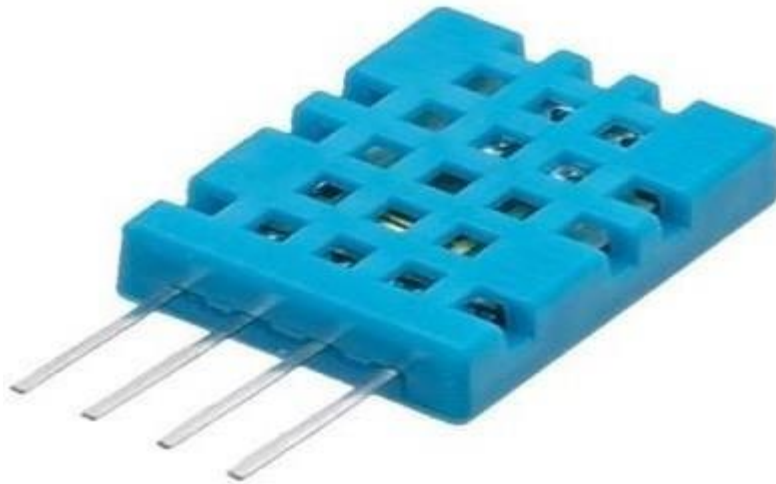


Fig:4.5. DHT11 sensor

Next, the sensor responds to the MCU “Start” signal by pulling the line low for 80? s followed by a logic high signal that also lasts for 80? 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.

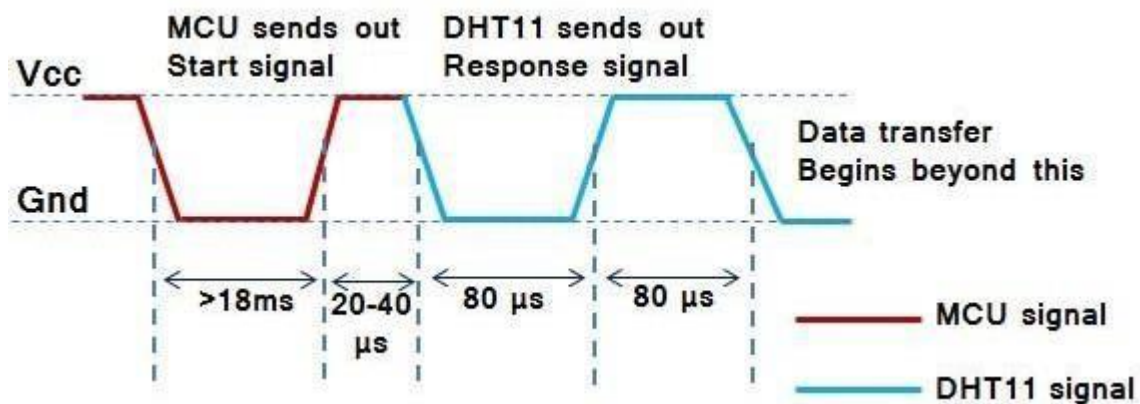


Fig : 4.6.MCU AND DHT11 Signal

"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 give the numeric values of the measured relative humidity (%) and temperature (°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 μs. Then it raises the line to high for 26-28 μs if it must send "0", or for 70 μs if the bit to be transmitted is "1". So, it is the width of the positive pulse that carries information about 1 and 0.

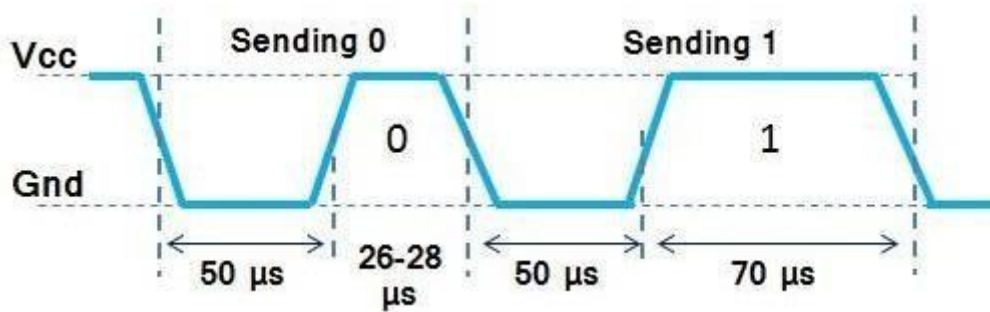


Fig :4.7. Timing difference for transmitting "1s" and "0s"

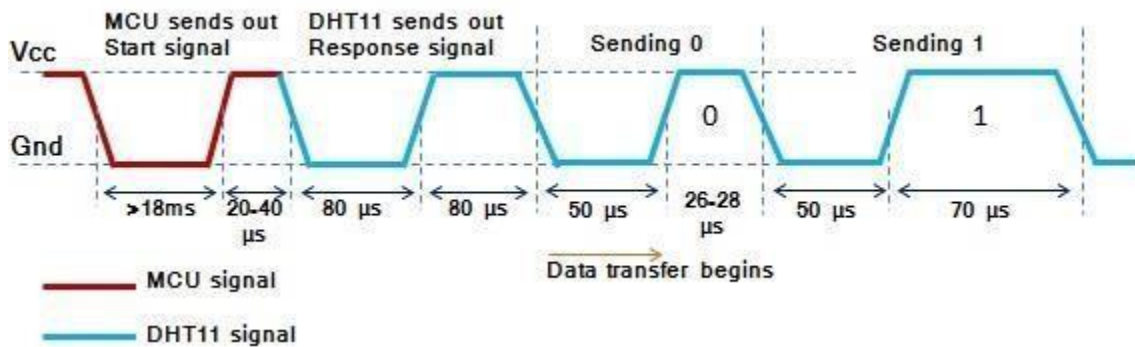


Fig :4.8. Start, Response and Data signals in sequence.

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.

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

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

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

Overview: Item Measurement Range Humidity Accuracy Temperature Accuracy Resolution Package DHT11 20-90%RH 0-50 °C $\pm 5\%RH$ $\pm 2^{\circ}C$ 1 4 Pin Single Row
 Page | 4 Detailed Specifications: Parameters Conditions Minimum Typical Maximum
 Humidity Resolution 1%RH 8 Bit Repeatability $\pm 1\%RH$ Accuracy 25°C $\pm 4\%RH$ 0-50°C $\pm 5\%RH$ Interchangeability Fully Interchangeable Measurement Range 0°C 30%RH 90%RH 25°C 20%RH 90%RH 50°C 20%RH 80%RH Response Time (Seconds) 1/e(63%)25°C, 1m/s Air 6 S 10 S 15 S Hysteresis $\pm 1\%RH$ Long-Term Stability Typical $\pm 1\%RH/year$ Temperature Resolution 1°C 1°C 1°C 8 Bit 8 Bit 8 Bit Repeatability $\pm 1^{\circ}C$ Accuracy $\pm 1^{\circ}C$ $\pm 2^{\circ}C$ Measurement Range 0°C 50°C Response Time (Seconds) 1/e(63%) 6 S 30 S

5. Power and Pin DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering. 5. Communication Process: Serial Interface (Single-Wire Two-Way) Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms. Data consists of decimal and integral parts. A complete data

transmission is 40bit, and the sensor sends higher data bit first. Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the checksum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

Overall Communication Process (Figure 2, below) When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low power-consumption mode until it receives a start signal from MCU again.

4.1.4. Ultrasonic (uv) sensors or Ultrasound sensors

Bats are wonderful creatures. Blind from the eyes and yet a vision so precise that could distinguish between a moth and a broken leaf even when flying at full speed. No doubt the vision is sharper than ours and is much beyond human capabilities of seeing but is certainly not beyond our understanding. Ultrasonic ranging is the technique used by bats and many other creatures of the animal kingdom for navigational purposes. In a bid to imitate the ways of nature to obtain an edge over everything, we humans have not only understood it but have successfully imitated some of these manifestations and harnessed their potential to the greatest extent.



Fig No: 4.9 Ultrasonic (UV) Sensor

The history dates to 1790, when Lazzaro Spallanzani first discovered that bats maneuvered in flight using their hearing rather than sight. Jean-Daniel Colladon in

1826 discovered sonography using an underwater bell, successfully and accurately determining the speed of sound in water. Thereafter, the study and research work in this field went on slowly until 1881 when Pierre Curie's discovery set the stage for modern ultrasound transducers. He found out the relationship between electrical voltage and pressure on crystalline material. The unfortunate Titanic accident spurred rigorous interest into this field because of which Paul Langevin invented

the hydrophone to detect icebergs. It was the first ultrasonic transducer. The hydrophone could send and receive low frequency sound waves and was later used in the detection of submarines in the World War 1. On a note parallel to the SONAR, medical research also started taking interest in ultrasonics. In late 1930's Dr. Karl Dusk used a technique called hyper phonography which recorded echoes of ultrasonic waves on a sensitive paper. This technique was used to produce ultrasound pictures of the brain to help detect tumors and marked the birth of ultrasound imaging. After that, many scientists like Ian Donald, Douglas Howry, Joseph Holmes, John Wild and John Reid improved upon the various aspects of ultrasonic sensors in the medical field which enabled diagnosis of stomach cancers, ovarian cysts, detection of twin pregnancies, tumors etc. Industry too did not waste time in jumping on to the bandwagon and soon developed techniques like ultrasonic welding and non-destructive testing at the outset of the 1960s.

How Ultrasonic Sensors work?

Ultrasonic sensors are devices that use electrical–mechanical energy transformation, the mechanical energy being in the form of ultrasonic waves, to measure distance from the sensor to the target object. Ultrasonic waves are longitudinal mechanical waves which travel as a succession of compressions and rarefactions along the direction of wave propagation through the medium. Any sound wave above the human auditory range of 20,000 Hz is called ultrasound. Depending on the type of application, the range of frequencies has been broadly categorized as shown in the figure below:

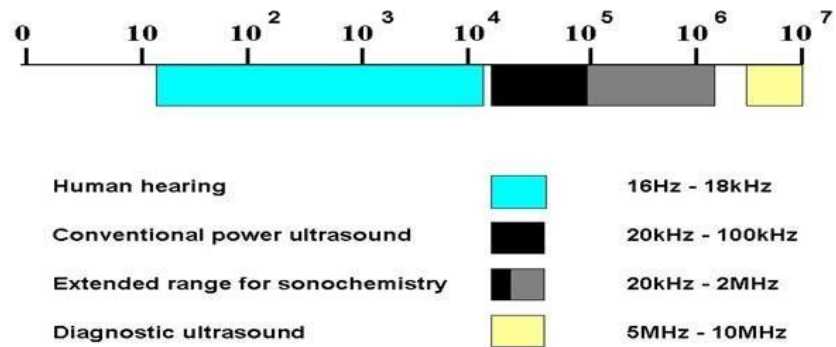


Fig: 4.10 The Frequency Range

When ultrasonic waves are incident on an object, diffused reflection of the energy takes place over a wide solid angle which might be as high as 180 degrees. Thus, some fraction of the incident energy is reflected to the transducer in the form of echoes and is detected. The distance to the object (L) can then be calculated through the speed of ultrasonic waves (v) in the medium by the relation.

$$L = \frac{v t \cos \theta}{2}$$

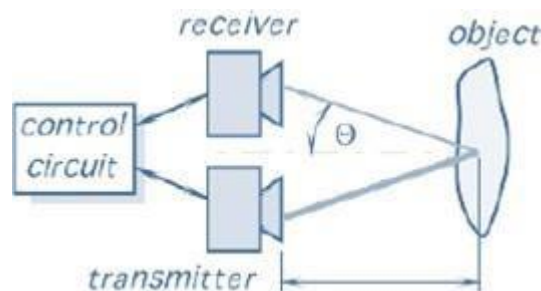


Fig: 4.11 Distance to the object(L)

Where 't' is the time taken by the wave to reach back to the sensor and 'θ' is the angle between the horizontal and the path taken as shown in the figure. If the object is in motion, instruments based on Doppler shift are used. Get all the details about internal structure and working of an ultrasound sensor at [Insight-How Ultrasonic Sensors Work](#).

Generating Ultrasonic Waves

For the generation of such mechanical waves, movement of some surface like a diaphragm is required which can then induce the motion to the medium in front of it

in the form of compression and rarefaction. Piezoelectric materials operating in the motor mode and magnetostrictive materials have been widely employed in the generation of ultrasonic waves at frequency ranges of 1-20 MHz and 20-40 kHz respectively. The sensors employ piezoelectric ceramic transducers which flex when an electric signal is applied to them. These are connected to an electronic oscillator whose output generates the oscillating voltages at the required frequency. Materials like Lead Zirconate Titanate are popular piezoelectric materials used in medical ultrasound imaging. For best results, the frequency of the applied oscillations must be equal to the natural frequency of the ceramic, which produces oscillations readily through resonance. It offers maximum sensitivity and efficiency when operated at resonance.

Piezoelectricity being a reversible phenomenon produces electrical voltages when ultrasonic waves reflect from the target and impinge upon the ceramic structure. In this way, a transducer may work both as a transmitter and a receiver in pulsed mode. When continuous measurement of distances is required, separate transducers may be used for transmission and reception. The sensors when used in industry are generally employed in arrays which may be mechanical arrays consisting of oscillating or rotating sensors, or electronic arrays which may be linear, curved or phased. To visualize the output of an ultrasonic sensor, displays of different kind are used whose shape depends on the type of transducer array used and the function. A sector Field of View is produced by mechanical arrays and curved and phased electronic arrays, while a linear field is generated by linear arrays. The display modes may be linear graphical plotting with amplitude on y-axis and time on x-axis called Amplitude mode or A-mode, or intensity modulated B-scans where the brightness of a spot indicates the amplitude of reflected waves. Other modes include M-mode, Doppler (D) Mode etc.

The parameterization of these sensors is generally done by monitoring the reflected and transmitted signals from the lateral and axial motion of transducer while keeping the target fixed in a specific medium (water in general). The sound beam diverges rapidly; hence care is taken that the transducer produces the smallest possible beams. The narrower the beam pattern, the more sensitive the sensor is. However, the angle possible between the transducer and the surface increases with the beam width. The beam patterns of the kind shown below are observed:

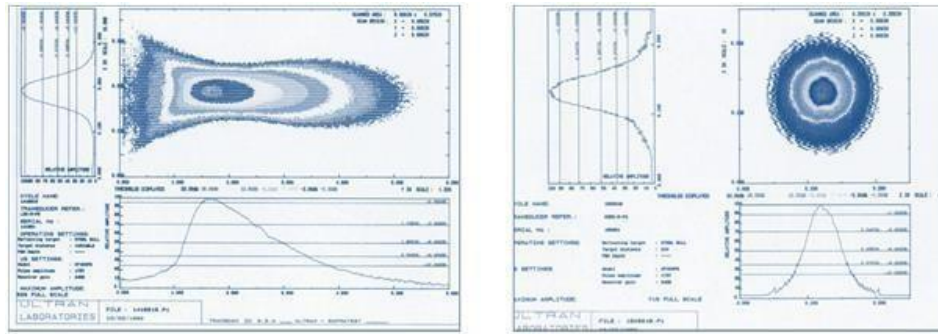


Fig: 4.12 Axial and Cross-Sectional beam profiles

The parameters on which the performance of an ultrasonic sensor is measured include bandwidth, attenuation, dynamic range and resolution like grayscale, axial and lateral resolution. Other parameters are Nominal Frequency, Peak Frequency, Bandwidth center Frequency, Pulse Width, sensitivity, and Signal to Noise Ratio (SNR).

Importance of Ultrasonic Sensors

There are a variety of sensors based on other physical transduction principles like the optical range finding sensors and the microwave-based devices too. Then why should one use ultrasonic transducers in the first place, given that the speed of sound is very slow than the speed of electromagnetic waves? The answer lies in the question itself. Because the EM waves-based devices are too fast. Being slower than the EM waves, the time taken by ultrasonic waves is much longer than that taken by the latter and hence its measurement can be done more easily and less expensively. Because these are based on sound waves rather than EM waves, these would work in places where the latter would not.

For example, in the case of clear object detection and measurement of liquid levels or high glare environments, light based sensors would suffer greatly because of the transmittance of the target or the translucence of the propagating media. Ultrasonic devices being based upon sound propagation would remain practically unaffected. These also function well in wet environments where optical beams may suffer from refraction from the water droplets in the environment. On account of range and accuracy, the ultrasonic sensors may lie in between two EM wave-based sensors, the Infrared rangefinders on the lower end and the LIDARs on the upper end. Not as accurate or long distance as the LIDARs, the Ultrasonic rangefinders fare better than the IR rangefinders which are highly susceptible to ambient conditions and require recalibration when environment changes. Further these devices offer advantage in medical imaging as compared to MRI or X-Ray scans due to

inexpensiveness and portability. No harmful effects of ultrasonic waves at the intensity levels used have been detected in contrast to X-rays or radioactivity-based methods and is particularly suited for imaging soft tissues.

Problems & Concerns

However, Ultrasonic sensors too are not free of all the problems. The speed of sound in a medium increase as the temperature of the medium increases. Thus, even when the target has remained in the same place, it may now seem that it has shifted to a place closer to the sensor. Air currents due to varied reasons may disturb the path of the wave which could lead to 'Missed Detection' or a wrong measurement. Acoustic noise like high pitched sounds created due to whistling or hissing of valves and pneumatic devices at the frequency close to the operating frequency may interfere with the output of the sensor. Electrical noise also affects the performance of the sensor. These may generate artifacts which are not a true representation of the imaged object. Just like the vision starts to blur when the distance of the object from the eye gets too small for the eyes to see it, ultrasonic devices also have a 'dead zone' where the sensor cannot reliably make measurements. This happens due to a phenomenon called ringing which is the continuous vibration of the transducer after emitting the pulse. Thus, when the distance is too small, the transducer has not yet come to rest to be able to differentiate between the vibration due to the incident radiation or the oscillation from the electrical excitation. The dangers of Ultrasonic waves are also well founded. If the intensity is too high, it can cause human tissues to heat and may cause ruptures in people exposed to it.

4.1.4.1. Applications

The applications of ultrasonic sensors can be classified because of the property that they exploit. These can be summarized as:

Domain	Parameter	Applications
Time	Time-of-Flight, Velocity	Density, Thickness, Flaw Detection, Anisotropy, Robotics, Remote Sensing etc.

Attenuation	Fluctuations in reflected and Transmitted Signals	Defect characterization, microstructures, interface analysis
Frequency	Ultrasonic Spectroscopy	Microstructure, grain size, porosity, phase analysis.
Image	Time-of-Flight, velocity, attenuation mapping in Raster C-Scan or SARs	Surface and internal Defect imaging, density, velocity, 2D and 3D imaging.

Research has been going on to overcome the problems of ultrasonic sensors, particularly in medical imaging where it is known as ultrasound. The artifacts of ultrasonic sensors like Acoustic shadowing and Acoustic Enhancement are being exploited to characterize tissues which allow the differentiation between solid and cystic tissues. The industry too has reaped the benefits from ultrasonic sensors in applications like plastic welding, jewelry cleaning, remote sensing and telemetry, assisted parking systems etc. Robotics has been known to use ultrasonic rangefinders as a favorite tool for distance ranging and mapping. Even the fashion industry is using ultrasonic sensors in hair styling like hair extension implants.

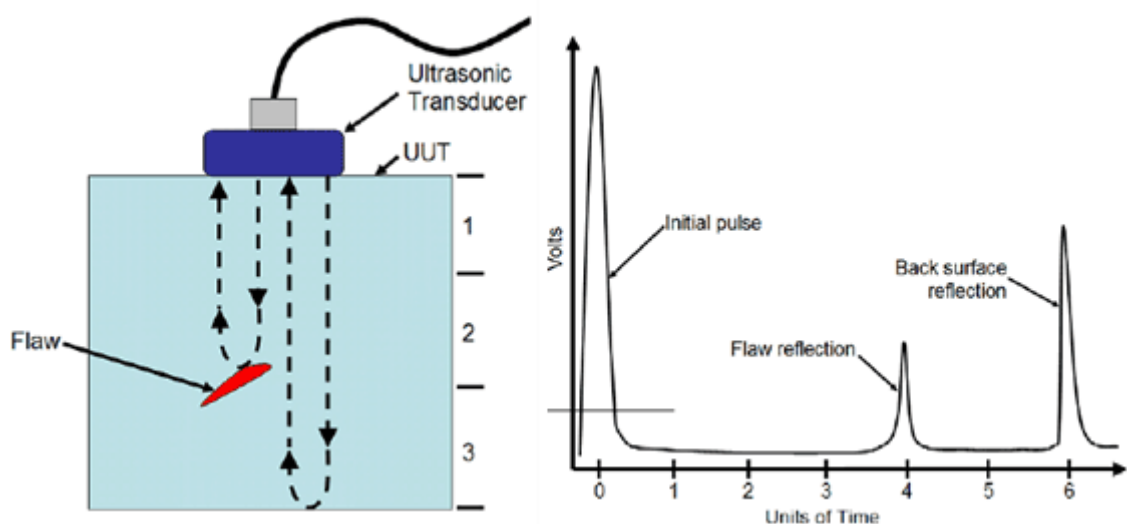


Fig: 4.13 Flow Detection Using Ultrasonic Sensors

Non-destructive testing and flaw detection uses ultrasonic waves in various modes

like the Longitudinal (L-wave) mode and the Shear (S-wave) mode to detect flaws in materials. With the advances in science, new materials offering increased performance at lower voltages like the capacitive micromachined ultrasonic transducers (CMUTs) are being developed which are expected to have higher bandwidth and greater potential for integration with electronic circuits.

These devices provide non-invasive measures for the detection of problems in all kinds of materials, be it a living tissue or non-living manufactured goods. With a healthy history of being able to detect many problems which otherwise left the doctors dazed and the problem untreated, ultrasonic sensors do offer a lot of promise even in the coming times. The environmental and psychological effects of exposure to EM-radiation being rigorously being put under the scanner, ultrasonic applications are expected to thrive and offer a substantial alternative to the contemporary technologies.

4.1.5. *Water level sensor*

Description

The eTape 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 eTape sensor is manufactured using printed electronic technologies which employ additive direct printing processes to produce functional circuits.

Theory of Operation

The eTape 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 eTape 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 ($V_{Max} = 10V$) Sensor Output The eTape can be modeled as a variable resistor ($400 - 2250 \Omega \pm 10\%$).

Connection and Installation

Connect to the eTape by attaching a 4-pin connector with pre-soldered wires to the Crimp flex pins. Do not solder directly to the Crimpflex pins. The inner two pins (pins 2 and 3) are the sensor output (Sense). The outer pins (pins 1 and 4) are the reference resistor (Reef) which can be used for temperature compensation. Suspend the eTape 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 eTape to equilibrate with atmospheric pressure. The vent hole is fitted with a hydrophobic filter membrane to prevent the eTape from being swamped if inadvertently submerged.

Custom Applications

The eTape sensor can be manufactured in custom lengths to fit any application. Contact Milone Technologies if you have an application that requires specific length, configuration or output characteristics.

Water level sensor

Flow measurement is the quantification of bulk fluid movement. Flow can be measured in a variety of ways. Positive-displacement flow meters accumulate a fixed volume of fluid and then count the number of times the volume is filled to measure flow. Other flow measurement methods rely on forces produced by the flowing stream as it overcomes a known constriction, to indirectly calculate flow. Flow may be measured by measuring the velocity of fluid over a known area.



Fig :4.14. Water Sensor

A flow sensor is a device for sensing the rate of fluid flow. Typically a flow sensor is the sensing element used in a flow meter, or flow logger, to record the flow of fluids. As is true for all sensors, absolute accuracy of a measurement requires a functionality for calibration.

There are various kinds of flow sensors and flow meters, including some that have a vane that is pushed by the fluid, and can drive a rotary potentiometer, or similar devices.

Other flow sensors are based on sensors which measure the transfer of heat caused by the moving medium (Thermal mass flow meter). This principle is common formicrosensors to measure flow.

Flow meters are related to devices called velocimeters that measure velocity of fluids flowing through them. Laser-based interferometry is often used for air flow measurement, but for liquids, it is often easier to measure the flow. Another approach is Doppler-based methods for flow measurement. Hall Effect sensors may also be used, on a flapper valve, or vane, to sense the position of the vane, as displaced by fluid flow.

Mechanical flow meters:

A positive displacement meter may be compared to a bucket and a stopwatch. The stopwatch is started when the flow starts, and stopped when the bucket reaches its limit. The volume divided by the time gives the flow rate. For continuous measurements, we need a system of continually filling and emptying buckets to

divide the flow without letting it out of the pipe. These continuously forming and collapsing volumetric displacements may take the form of pistons reciprocating in cylinders, gear teeth mating against the internal wall of a meter or through a progressive cavity created by rotating oval gears or a helical screw.

Laser Doppler flow measurement:

A beam of laser light impinging on a moving particle will be partially scattered with a change in wavelength proportional to the particle's speed (the Doppler effect). A laser Doppler velocimeter (LDV), also called a laser Doppler anemometer (LDA), focuses a laser beam into a small volume in a flowing fluid containing small particles (naturally occurring or induced). The particles scatter the light with a Doppler shift. Analysis of this shifted wavelength can be used to directly, and with great precision, determine the speed of the particle and thus a close approximation of the fluid velocity.

A number of different techniques and device configurations are available for determining the Doppler shift. All use a photodetector (typically an avalanche photodiode) to convert the light into an electrical waveform for analysis. In most devices, the original laser light is divided into two beams. In one general LDV class, the two beams are made to intersect at their focal points where they interfere and generate a set of straight fringes. The sensor is then aligned to the flow such that the fringes are perpendicular to the flow direction. As particles pass through the fringes, the Doppler-shifted light is collected into the photodetector. In another general LDV class, one beam is used as a reference and the other is Doppler-scattered. Both beams are then collected onto the photodetector where optical heterodyne detection is used to extract the Doppler signal.

Electromagnetic, ultrasonic and coriolis flow meters:

Modern innovations in the measurement of flow rate incorporate electronic devices that can correct for varying pressure and temperature (i.e., density) conditions, nonlinearities, and for the characteristics of the fluid.

Magnetic flow meters:

Magnetic flow meters, often called "mag meter's or "electromag"s, use a magnetic field applied to the metering tube, which results in a potential difference proportional to the flow velocity perpendicular to the flux lines. The potential difference is sensed

by electrodes aligned perpendicular to the flow and the applied magnetic field. The physical principle at work is Faraday's law of electromagnetic induction. The magnetic flow meter requires a conducting fluid and a nonconducting pipe liner. The electrodes must not corrode in contact with the process fluid; some magnetic flowmeters have auxiliary transducers installed to clean the electrodes in place. The applied magnetic field is pulsed, which allows the flowmeter to cancel out the effect of stray voltage in the piping system.

Non-contact electromagnetic flow meters:

A Lorentz force velocimetry system is called Lorentz force flowmeter (LFF). A LFF measures the integrated or bulk Lorentz force resulting from the interaction between a liquid metal in motion and an applied magnetic field. In this case the characteristic length of the magnetic field is of the same order of magnitude as the dimensions of the channel. It must be addressed that in the case where localized magnetic fields are used, it is possible to perform local velocity measurements and thus the term Lorentz force velocimeter is used.

Ultrasonic flow meters (Doppler, transit time)

There are two main types of Ultrasonic flow meters: Doppler and transit time. While they both utilize ultrasound to make measurements and can be non-invasive (measure flow from outside the tube, pipe or vessel), they measure flow by very different methods.

Ultrasonic transit time flow meters measure the difference of the transit time of ultrasonic pulses propagating in and against the direction of flow. This time difference is a measure for the average velocity of the fluid along the path of the ultrasonic beam. By using the absolute transit times both the averaged fluid velocity and the speed of sound can be calculated. Using the two transittimes t_{up} and t_{down} and the distance between receiving and transmitting transducers L and the inclination angle α one can write the equations:

$$v = \frac{L}{2 \sin(\alpha)} \frac{t_{up} - t_{down}}{t_{up} t_{down}} \quad \text{and} \quad c = \frac{L}{2} \frac{t_{up} + t_{down}}{t_{up} t_{down}}$$

where v is the average velocity of the fluid along the sound path and c is the speed of sound. With wide-beam illumination transit time ultrasound can also be used to

measure volume flow independent of the cross-sectional area of the vessel or tube.

Ultrasonic Doppler flow meters measure the Doppler shift resulting from reflecting an ultrasonic beam off the particulates in flowing fluid. The frequency of the transmitted beam is affected by the movement of the particles; this frequency shift can be used to calculate the fluid velocity. For the Doppler principle to work there must be a high enough density of sonically reflective materials such as solid particles or air bubbles suspended in the fluid. This is in direct contrast to an ultrasonic transit time flow meter, where bubbles and solid particles reduce the accuracy of the measurement. Due to the dependency on these particles there are limited applications for Doppler flow meters. This technology is also known as acoustic Doppler velocimetry.

CHAPTER 5

SOFTWARE REQUIREMENTS

5.1. Thonny Ide

Thonny (/ˈθɒni/ *THON-ee*) is an integrated environment for Python that is designed for beginners. It was created by Aivar Annamaa, an Estonian programmer. It supports different ways of stepping through code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.

Features

- Line numbers.
- Statement stepping without breakpoints.
- Live variables during debugging.
- Stepping through evaluation of the expressions (expressions get replaced by their values)
- Separate windows for executing function calls (for explaining local variables and call stack)
- Variables and memory can be explained either by using simplified model (name → value) or by using more realistic model (name → address/id → value)
- Simple pip GUI
- Support for Python and Micro Python
- Support for running and managing files on a remote machine via SSH
- Possibility to log user actions for replaying or analyzing the programming process.

5.2. Python

Python has become one of the most popular programming languages in the world in recent years. It's used in everything from machine learning to building websites and software testing. It can be used by developers and non-developers alike.

Python, one of the most popular programming languages in the world, has created everything from Netflix's recommendation algorithm to the software that controls self-driving cars. Python is a general-purpose language, which means it is designed to be

used in a range of applications, including data science, software and web development, automation, and generally getting stuff done.

Let us take a closer look at what Python is, what it can do, and how you can start learning it.

What is Python?

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm Red Monk found that it was the second-most popular programming language among developers in 2021.

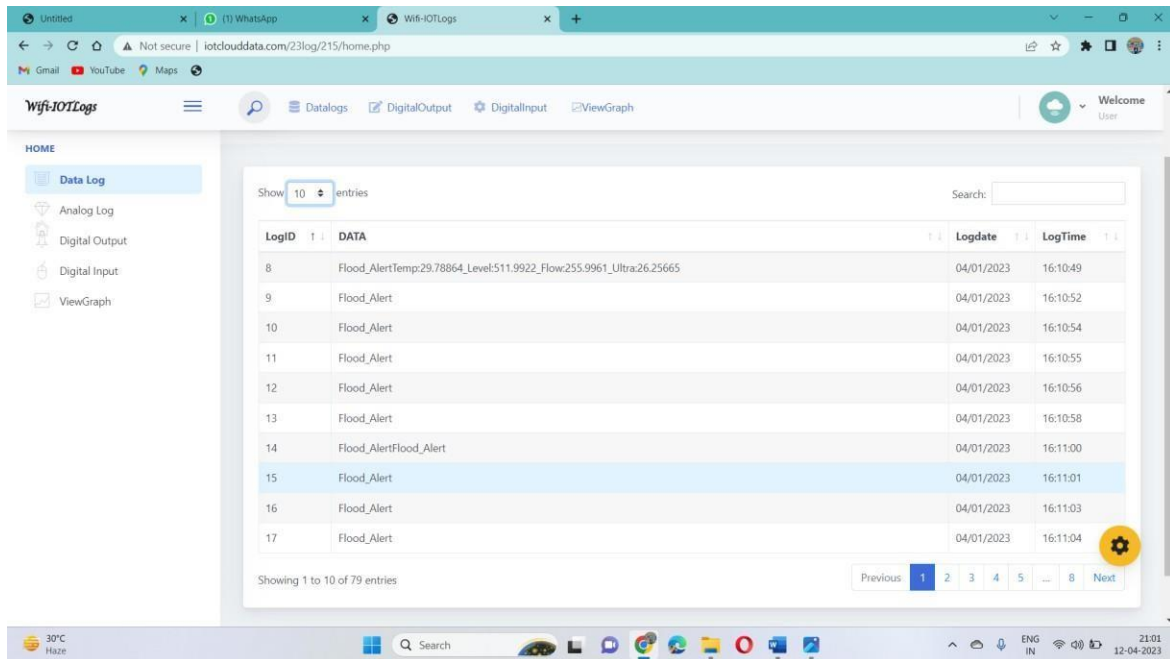
what is Python used for?

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

"Writing programs is a very creative and rewarding activity," says University of Michigan and Coursera instructor Charles R Severance in his book Python for Everybody. "You can write programs for many reasons, ranging from making your living to solving a difficult data analysis problem to having fun to helping someone else solve a problem."

CHAPTER 6

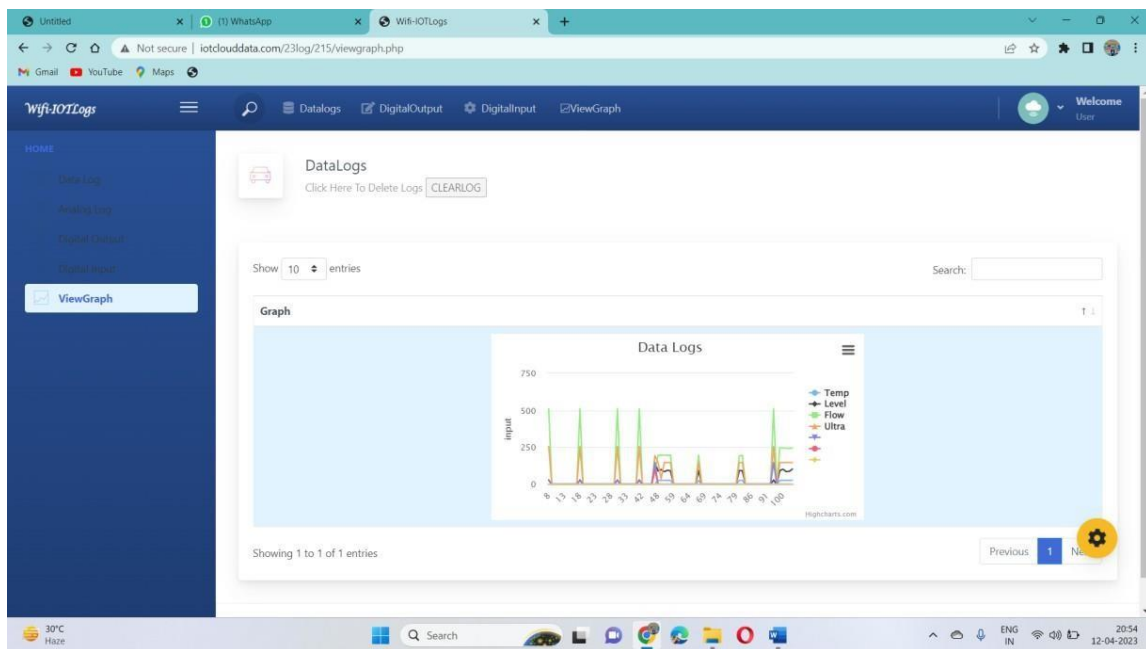
RESULTS AND DISCUSSION



The screenshot shows the 'Data Log' section of the Wifi-IOTLogs application. A table displays log entries with columns for LogID, DATA, Logdate, and LogTime. The table shows entries 8 through 17, all dated 04/01/2023. Entry 15 is highlighted. The interface includes a search bar, a 'Show 10 entries' dropdown, and pagination controls at the bottom of the table.

LogID	DATA	Logdate	LogTime
8	Flood_AlertTemp:29.78864_Level:511.9922_Flow:255.9961_Ultra:26.25665	04/01/2023	16:10:49
9	Flood_Alert	04/01/2023	16:10:52
10	Flood_Alert	04/01/2023	16:10:54
11	Flood_Alert	04/01/2023	16:10:55
12	Flood_Alert	04/01/2023	16:10:56
13	Flood_Alert	04/01/2023	16:10:58
14	Flood_AlertFlood_Alert	04/01/2023	16:11:00
15	Flood_Alert	04/01/2023	16:11:01
16	Flood_Alert	04/01/2023	16:11:03
17	Flood_Alert	04/01/2023	16:11:04

GRAPH:



Thonny - Raspberry Pi Pico - HCSR04.py @ 18:11

File Edit View Run Tools Help

Files

- This computer
 - G:\
 - PerfLogs
 - Program Files
 - Program Files (x86)
 - Users
 - WCH.CN
 - Windows
- Raspberry Pi Pico
 - lib
 - blink.py
 - bt.py
 - dht.py
 - HCSR04.py
 - IR.py
 - main.py
 - tempinbuilt.py

```

1 from machine import Pin
2 import utime
3 trigger = Pin(14, Pin.OUT)
4 echo = Pin(15, Pin.IN)
5 def ultra():
6     trigger.low()
7     utime.sleep_us(2)
8     trigger.high()
9     utime.sleep_us(5)
10    trigger.low()
11    while echo.value() == 0:
12        signaloff = utime.ticks_us()
13    while echo.value() == 1:
14        signalon = utime.ticks_us()
15        timepassed = signalon - signaloff
16        distance = (timepassed * 0.0343) / 2
17        print("The distance from object is ",distance,"cm")
18    while True:
19        ultra()
20        utime.sleep(1)

```

Shell

```

The distance from object is 149.5994 cm
The distance from object is 148.296 cm
The distance from object is 16.4297 cm
The distance from object is 15.778 cm
The distance from object is 18.09325 cm
The distance from object is 23.34115 cm
The distance from object is 21.13535 cm
The distance from object is 21.50055 cm
The distance from object is 22.50455 cm
The distance from object is 21.2097 cm
The distance from object is 26.8569 cm
The distance from object is 27.6458 cm
The distance from object is 26.8912 cm
The distance from object is 27.3028 cm
The distance from object is 27.7144 cm
The distance from object is 148.4332 cm
The distance from object is 148.4647 cm

```

MicroPython (Raspberry Pi Pico)

Thonny - <untitled> @ 20:18

File Edit View Run Tools Help

Run buttons: Run, Stop, Restart, etc.

<untitled> * X

```

1 from machine import Pin
2 import utime
3 trigger = Pin(14, Pin.OUT)
4 echo = Pin(15, Pin.IN)
5 def ultra():
6     trigger.low()
7     utime.sleep_us(2)
8     trigger.high()
9     utime.sleep_us(5)
10    trigger.low()
11    while echo.value() == 0:
12        signaloff = utime.ticks_us()
13    while echo.value() == 1:
14        signalon = utime.ticks_us()
15        timepassed = signalon - signaloff
16        distance = (timepassed * 0.0343) / 2
17        print("The distance from object is ",distance,"cm")
18    while True:

```

Variables

Name	Value
------	-------

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

This project highlights the possibility to provide an alert system that will overcome the risk of flood. As the project is enabled with IOT technology and hence the sensor data can be monitored from anywhere in the world. More sensors can be integrated into the system to create more accurate and efficient flood detection system. It can also contribute to multiple government agencies or authority that ultimately help the society and mankind about the flood like hazardous natural disaster. It will monitor each and every aspect that can lead to flood. If the water level rises along with the speed, it will send an alert immediately. It also ensures increased accessibility in dealing and reverting to this catastrophic incident. In summary, it will help the community in taking quick decisions and planning against this disaster mankind about the flood like hazardous natural disaster.

The Future scope of the project is, flood can also be related to the intensity of rainfall, which is the height of the water layer covering the ground in a period. Hence the development of a rainfall forecasting sensor eventually turns up to the early flood monitoring and detection, Scholarly studies are ongoing and can be implemented to our existing system in future.

REFERENCES

- [1] D. Mysar and M.J. Babu, “Real Time Monitoring of Water Level Variation in Rivers and Flood Alerting System using ARM7”, International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 8, pp. 392-396, 2015.
- [2] S. K. Subramaniam et al., “Flood level indicator and risk warning system for remote location monitoring using Flood Observatory System”, WSEAS Transactions on Systems and Control, vol. 5, no. 3, pp. 153-163, 2010.
- [3] N. Ahmad et al., “Flood Prediction and Disaster Risk Analysis using GIS based Wireless Sensor Networks: A Review”, Journal of Basic and Applied Scientific Research, vol. 3, no. 8, pp. 632- 643, 2013.
- [4] M. Ancona et al., “On the design of an intelligent sensor network for flash flood monitoring, diagnosis and management in urban areas position paper,” Procedia Computer Science, vol. 32, pp. 941–946, 2014.
- [5] M. Borga et al., “Surveying flash floods: gauging the ungauged extremes,” Hydrological processes, vol. 22, no. 18, 2008, pp. 3883-3885.

