# Value Added Course

A Sample Report of the Value Added Course Titled **"Embedded Systems with IoT"** offered for the students of the Department of Electrical and Electronics Engineering is given below

OFFICE OF THE VICE - CHANCELLOR
15 FEB 2024

# AMET

ACADEMY OF MARITIME EDUCATION AND TRAINING
Deemed to be University Under Section 3 of UGC Act 1956

Filo Resubmit on - 15/2/24 Time: 9:45 AM

Inward File No. 3095

*FROM THE OFFICE OF HOD-EEE*

Admin

Date: 15.02.2024

AMET (Deemed to be University) CHENNAI
File No. F152
15 FEB 2024
INWARD

Dr.V.Sridevi
Professor and HoD
Dept. of EEE

To

The Registrar
AMET Deemed to be University.

Sir,

Sub: Renewal of Existing MOU's Reg.

There are 14 MOU's associated with the Department of Electrical and Electronics Engineering to support student activities, FDP, Training programs etc. We would like to renew one of the existing MOU's namely VI Microsystems Pvt.Ltd since their validity period was over in December 2023. I request you to kindly do the needful.

| S.No | Workshop/Value Added Programs | Duration (In Hours) | Max. batch Size | No. of batches per year | Program fee/batch |
|---|---|---|---|---|---|
| 1 | IoT, Embedded and Arduino | 35 | 40 | 1 | INR 40,000/- |
| 2 | Automation | 35 | 40 | 1 | INR 40,000/- |
| 3 | Engineering Design and Additive Manufacturing | 35 | 40 | 1 | INR 40,000/- |
| 4 | Maintenance (Mechanical and Electrical) | 35 | 40 | 1 | INR 40,000/- |

***Note:***

- All the above fee is exclusive of GST
- Boarding & Lodging for the trainees will be additional, only if required.
- Travel, Boarding and Lodging for the trainers to be borne by the University, for all the training execution scheduled at the University.
- University will sponsored Rs. 25000/- for conducting Skill Development or Value Added Training program per semester and the balance amount will be collected from the students (Rs.1000/student).

Thanking you,

Submitted for your approval sir

15/2/24
Dr.V.Sridevi
Professor and HoD

R(E)

HoD-EEE
Dir program
Dr Tmore

15/2/24

AMET UNIVERSITY

(Deemed to be University Under Section 3 of UGC Act 1956)

Report on

Value Added Course

*On*

# Embedded System with IoT

*for*

# I, II and III Year

***Organized By***

Department of Electrical and Electronics Engineering

**Batch 1: 11-03-2024 to 16-03-2024**

**Batch 2: 22-03-2024 to 28-03-2024**

**Batch 3: 1-04-2024 to 06-04-2024**

**Venue: VI Microsystems Pvt.Ltd, Chennai**

## Hands-on Value Added Course

**Topic : Embedded System with IoT**

**Duration : 35 - 40 Hours**

**COURSE OBJECTIVE:**

1. Understanding the concept of Embedded system and real time sensors and Various domains with the hands on session
2. Gaining Advanced knowledge of IoT,
3. Applying the fundamental theories and concepts of Embedded system with IoT
4. Knowledge about Wired and wireless communication
5. experiencing extensive and hands-on in Embedded with IoT concepts

### Course Syllabus:

#### Unit 1: Introduction and Configuration of Embedded system Board

Introduction to Embedded system - Microprocessor and Microcontroller Classification : Different between microprocessor & Microcontroller - Classification based on Architecture-Memory Classification .

Embedded system board architecture - Identify Embedded Platform / simulator – Digital I/O interface, Analog I/O interface – Interrupts – Timer – PWM – Interface of peripheral device,

#### Unit 2: Embedded Software Tools and Simulation Tool

About software introduction – Introduction to Different Types of IDE, Thorny IDE, and Introduction to C and Embedded - Python, micro Python - about Thonny IDE - Thorny Shell. Online

Embedded wokwi pico micropython Simulation Tool.

**Unit 3: Peripherals and Sensors Interfacing**

LED blinking task – Buzzer, Relay and switch Interfacing, 7 segment and LCD Interfacing PWM Generation, different types of Analog and digital sensors are interfacing – LM 35 –Ultrasonic – LDR- IR –Potentiometer and Accelerometer.

**Unit 4: Wired and Wireless Communication Protocols**

Wired communication Communication Protocols Serial Communication: UART – I2C – SPI- wireless communication Protocols : zigbee , Bluetooth ,RF LoRaWAN Wi-Fi and IoT ,wired and wireless communication application interfacing systems.

**Unit 5: Cloud Platforms for IOT**

Introduction to IOT Understanding IoT fundamentals IOT Architecture and protocols Various Platforms for IoT Real time Examples of IoT Overview of IoT components and IoT Communication Technologies Challenges in IOT ,Cloud Platforms for IOT , Study of IOT Cloud platforms ThingSpeak API ,Fire base and Blynk app, Interfacing RP2040 W with Web services

**Course Outcomes:**

Students will be able to

1. Design and develop the embedded system application to acquire the data from sensors like temperature, pressure, humidity, flow etc, and communicate collected signals to the computer through UART, I2C and SPI port.
2. Develop the embedded system to interface with accelerometer, ultrasonic sensor, and encoder to acquire the
3. To study of various IoT Protocols

**5 –Days hands-on training program in Dual core embedded controller**

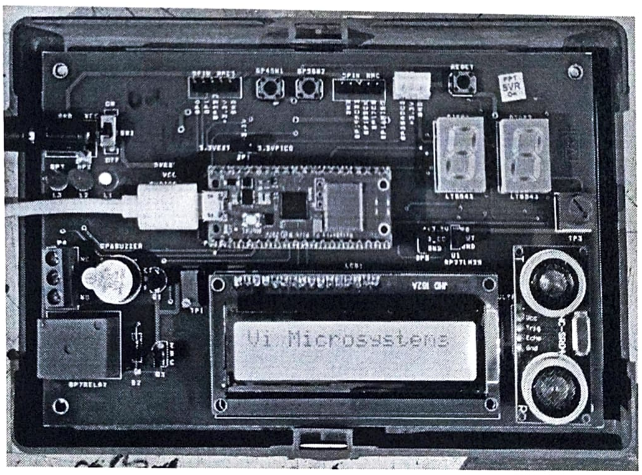
| Day | FN | AN |
|---|---|---|
| 1 | -Introduction to Embedded systems<br>-Difference between Microprocessor Vs Microcontroller ,<br>-Raspberry pi RP 2040 Pico microcontroller Introduction | Hands-on training Raspberry pi Pico Programming overview , LED interfacing , Button Interfacing and different types of IO concept with Example |
| 2 | - GPIO, Timer<br>-Introduction to PWM , PWM Generation, LED Fading Using PWM,<br>-V/F Control of Induction Motor | Dual core , Programmable IO and State Machine concept with Hands-on training |
| 3 | - Introduction to sensors, concept of analog and digital sensor , Hands-on training for Different types of digital sensor interfacing | Introduction to ADC and types<br>Hands-on training for Potentiometer interfacing , Temperature sensor interfacing |
| 4 | Introduction to display unit , types of display , about LCD and 7 segment display with hands-on programming | Hands-on training - Ultrasonic sensor interfacing , Relay interfacing and buzzer interfacing |

| | | |
|---|---|---|
| 5 | Introduction to Wireless technology , About raspberry pi rp2040 W and Introduction to IoT cloud , sensor data send to any cloud service with hands-on | Temperature and Ultrasonic data send to cloud<br>Home automation using IoT - project |

The following Boards and Industrial projects will be used in this course.

**RP2040 W based Carrier Board**

As Raspberry Pi based embedded Controllers become more and more awareness among students, Vi Micro has designed another innovative Carrier Board, based on Raspberry RP2040 Processor, which provides Dual Core Cortex M0+ Microcontroller, 16 GPIO, ADC, etc. to build many Embedded Applications and Study the Interfacing of Various Devices to RP2040



**Hands on Experiments above Trainer kit used**

1. interface a 16*2 LCD Display
2. interface 2*7 Segment Display
3. Interface Various Sensors
4. Acqure ADC Sensor Data and display on a Smart Phone through WiFi

## COURSE OUTCOMES:

Students will acquire basics of Embedded system and Real time IoT Application

- Student get knowledge microprocessor and controller .
- Students get the real time sensors working procedure and get hands-on training Observe surface area and objects on systematic basis and thereby monitor their changes over time.
- IoT Application ( Internet of Things )
- Real time Application.

## REFERENCES:


1.RP2040 Assembling language Programming – Stephen smith

2.Raspberry pi pico Essentials:Program,Build,and Master Over 50 Projects with micro python


Online Link :
https://www.google.com/search?rlz=1C1JZAP_enIN1029IN1029&cs=0&q=Raspberry+Pi+Pico+Essentials:+Program,+Build,+and+Master+Over+50+Projects+with+MicroPython+and+the+RP2040+Microprocessor+Dogan+Ibrahim&stick=H4sIAAAAAAAAAONgVeLVT9c3NCzJMiusSC4uM-IpKjAyMDFQSMrPzy4-xQiRLTA0KkgzSc-D8WGq4fy84oLyPAuTR4yLmLgFXv64Jyw1g2nSmpPXGCcycQn45OcXp-ZUBqXmJJakpoTkCxlysbnmlWSWVAoJSvFzQYwojDfPLUsyyzUSgQpkm2UnZ6RYFAjMfzCNUSiUizs4tSQk3zc_JTOtUshNyIWL0zc1Nym1qNg_TUiZi8s5PycnNbkkMz9

PSFRKmEtQPxkuoA_2ixJfEKd-rr5BUkl8svEBRiYrJg0mpUKjuF2Xpp1jixDc1_7_f6hkklO UhpYgF5tLfm5iZp4gU_lle7kF7-21hLk4Qhlr8vPycysFpdjuFm37f8JeSZHTpp-7IXD-W3tBvVpGhdkzxA5IMCswaDAYXr7G_oAt_td-LQagRU37Vhxi4-BgFGAwYuJgqGLgWcTaxhiUWFwAdHxRpUJAJhAI5yu4FhenAgMlMafYSiGgKD-9KDFXR8GpNDMnRUchMS9FwTexuCS1SMG_DEiYG oCUZAG9V6xQnlmSoeCbmVyUH1BZkpGfB1ZckpGqE BQAjkiwVEFRfnJqcXF-kYJLfnpinoJnUIFiRmbuBDZGABx6pZsCAgAA&sa=X&ved=2ahUKEwjamLKDsNH8AhUx5HMBHeKbDDQQ7fAIegQIABAU

## SOFTWARE REQUIREMENT:

- Thonny IDE
- Python and Micro Python Language
- Wokwi simulator Tool ( Online Simulation Tool )

## HARDWARE REQUIREMENT:

- Processor - ARM Cortex M0 ( Raspberry pi Pico RP2040W)

# *RP2040 CARRIER BOARD*

## *User Manual*

## *Version1.0*

**TechnicalClarification/Suggestion:**

✍/☎

**TechnicalSupport**
**Division,ViMicrosystemsPvt.**
**Ltd.,**
**PlotNo:75,ElectronicsEstate,Perungudi,C**
**hennai-600096,INDIA.**

**Ph:91-44-24961842,91-44-24961852**
**Mail:rnd@vimicrosystems.comW**
**eb:www.vimicrosystem.com01/14/**
**05/10**

[redacted]



[redacted]

## 1. INTRODUCTION TO EMBEDDED SYSTEM

An embedded system is a combination of computer hardware and software designed for a specific function. Embedded systems may also function within a larger system. The systems can be programmable or have a fixed functionality. Industrial machines, consumer electronics, agricultural and processing industry devices, automobiles, medical equipment, cameras, digital watches, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system.

While embedded systems are computing systems, they can range from having no user interface (UI) -- for example, on devices designed to perform a single task -- to complex graphical user interfaces (GUIs), such as in mobile devices. User interfaces can include buttons, LEDs (light-emitting diodes) and touchscreen sensing. Some systems use remote user interfaces as well.

Marketability, a business-to-business (B2B) research firm, predicted that the embedded market will be worth $116.2 billion by 2025. Chip manufacturers for embedded systems include many well-known technology companies, such as Apple, IBM, Intel and Texas Instruments. The expected growth is partially due to the continued investment in artificial intelligence (AI), mobile computing and the need for chips designed for high-level processing.

### Examples of embedded systems

Embedded systems are used in a wide range of technologies across an array of industries. Some examples include:

- **Automobiles.** Modern cars commonly consist of many computers (sometimes as many as 100), or embedded systems, designed to perform different tasks within the vehicle. Some of these systems perform basic utility functions and others provide entertainment or user-facing functions. Some embedded systems in consumer vehicles include cruise control, backup sensors, suspension control, navigation systems and airbag systems.

- **Mobile phones.** These consist of many embedded systems, including GUI software and hardware, operating systems (OSes), cameras, microphones, and USB (Universal Serial Bus) I/O (input/output) modules.
- **Industrial machines.** They can contain embedded systems, like sensors, and can be embedded systems themselves. Industrial machines often have embedded automation systems that perform specific monitoring and control functions.
- **Medical equipment.** These may contain embedded systems like sensors and control mechanisms. Medical equipment, such as industrial machines, also must be very user-friendly so that human health isn't jeopardized by preventable machine mistakes. This means they'll often include a more complex OS and GUI designed for an appropriate UI.

**How does an embedded system work?**

Embedded systems always function as part of a complete device -- that's what's meant by the term embedded. They are low-cost, low-power-consuming, small computers that are embedded in other mechanical or electrical systems. Generally, they comprise a processor, power supply, and memory and communication ports. Embedded systems use the communication ports to transmit data between the processor and peripheral devices -- often, other embedded systems -- using a communication protocol. The processor interprets this data with the help of minimal software stored on the memory. The software is usually highly specific to the function that the embedded system serves.

The processor may be a microprocessor or microcontroller. Microcontrollers are simply microprocessors with peripheral interfaces and integrated memory included. Microprocessors use separate integrated circuits for memory and peripherals instead of including them on the chip. Both can be used, but microprocessors typically require more support circuitry than microcontrollers because there is less integrated into the microprocessor. The term *system on a chip (SoC)* is often used. SoCs include multiple processors and interfaces on a single chip. They are often used for high-volume embedded systems. Some example SoC types are the application-specific integrated circuit (ASIC) and the field-programmable gate array (FPGA).

Often, embedded systems are used in real-time operating environments and use a real-time operating system (RTOS) to communicate with the hardware. Near-real-time approaches are suitable at higher levels of chip capability, defined by designers who have increasingly decided the systems are generally fast enough and the tasks tolerant of slight variations in reaction. In these instances, stripped-down versions of the Linux operating system are commonly deployed, although other OSes have been pared down to run on embedded systems, including Embedded Java and Windows IoT (formerly Windows Embedded).

Characteristics of embedded systems

The main characteristic of embedded systems is that they are task-specific.

Additionally, embedded systems can include the following characteristics:

- typically, consist of hardware, software and firmware;
- can be embedded in a larger system to perform a specific function, as they are built for specialized tasks within the system, not various tasks;
- can be either microprocessor-based or micro controller-based -- both are integrated circuits that give the system compute power;
- are often used for sensing and real-time computing in internet of things (IoT) devices, which are devices that are internet-connected and do not require a user to operate;
- can vary in complexity and in function, which affects the type of software, firmware and hardware they use; and
- are often required to perform their function under a time constraint to keep the larger system functioning properly.

**Structure of embedded systems**

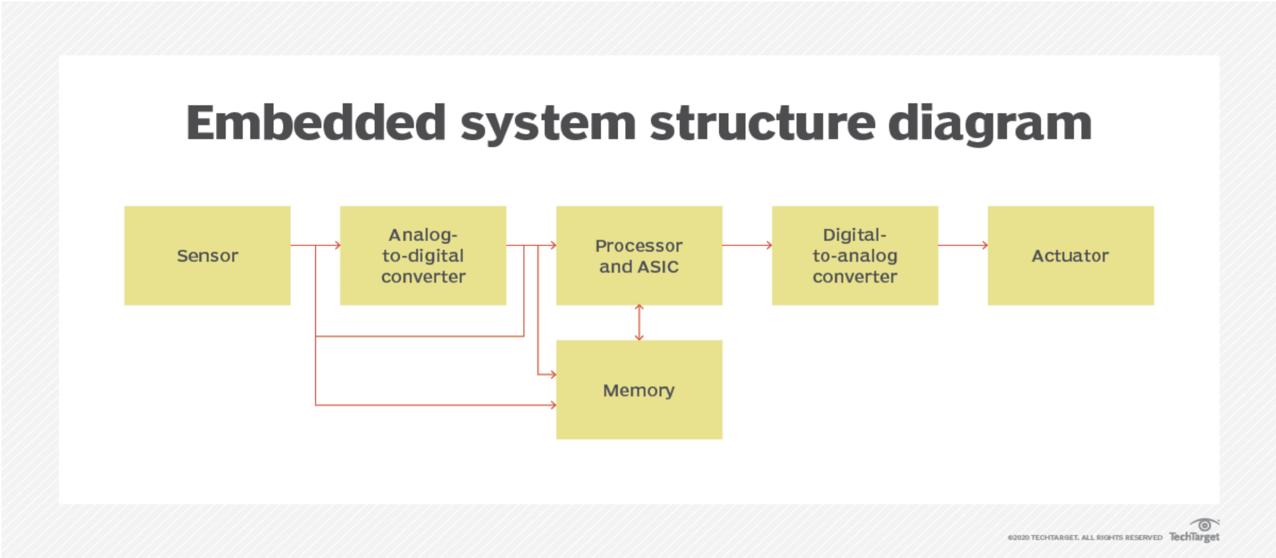
Embedded systems vary in complexity but, generally, consist of three main elements:

- **Hardware.** The hardware of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers and, typically, refer to a CPU (central processing unit) that is integrated with other basic computing components such as memory chips and digital signal processors (DSPs). Microcontrollers have those components built into one chip.
- **Software and firmware.** Software for embedded systems can vary in complexity. However, industrial-grade microcontrollers and embedded IoT systems usually run very simple software that requires little memory.
- **Real-time operating system.** These are not always included in embedded systems, especially smaller-scale systems. RTOSes define how the system works by supervising the software and setting rules during program execution.

In terms of hardware, a basic embedded system would consist of the following elements:

- **Sensors** convert physical sense data into an electrical signal.
- **Analog-to-digital (A-D) converters** change an analog electrical signal into a digital one.
- **Processors** process digital signals and store them in memory.
- **Digital-to-analog (D-A) converters** change the digital data from the processor into analog data.
- **Actuators** compare actual output to memory-stored output and choose the correct one.

The sensor reads external inputs, the converters make that input readable to the processor, and the processor turns that information into useful output for the embedded system.




**2.RASPBERRY PI PICO RP2040**

Designed by Raspberry Pi, RP2040 features a dual-core Arm Cortex-M0+ processor with 264kB internal RAM and support for up to 16MB of off-chip flash. A wide range of flexible I/O options includes I2C, SPI, and - uniquely - Programmable I/O (PIO). These support endless possible applications for this small and affordable package.

Whether you have a Raspberry Pi Pico or another RP2040-based microcontroller board, everything you need to get started is here. You'll find support for getting started with C/C++ or MicroPython on Raspberry Pi Pico, and links to resources for other boards that use RP2040. There are also links to the technical documentation for both the Raspberry Pi Pico microcontroller board and our RP2040 microcontroller chip.

RP2040 is the debut microcontroller from Raspberry Pi. It brings our signature values of high performance, low cost, and ease of use to the microcontroller space.

With a large on-chip memory, symmetric dual-core processor complex, deterministic bus fabric, and rich peripheral set augmented with our unique Programmable I/O (PIO) subsystem, it provides professional users with unrivalled power and flexibility. With detailed documentation, a polished MicroPython port, and a UF2 bootloader in ROM, it has the lowest possible barrier to entry for beginner and hobbyist users.

RP2040 is a stateless device, with support for cached execute-in-place from external QSPI memory. This design decision allows you to choose the appropriate density of non-volatile storage for your application, and to benefit from the low pricing of commodity Flash parts.

RP2040 is manufactured on a modern 40nm process node, delivering high performance, low dynamic power consumption, and low leakage, with a variety of low-power modes to support extended-duration operation on battery power

Key features:

- Dual ARM Cortex-M0+ @ 133MHz
- 264kB on-chip SRAM in six independent banks
- Support for up to 16MB of off-chip Flash memory via dedicated QSPI bus
- DMA controller
- Fully-connected AHB crossbar
- Interpolator and integer divider peripherals
- On-chip programmable LDO to generate core voltage
- 2 on-chip PLLs to generate USB and core clocks
- 30 GPIO pins, 4 of which can be used as analogue inputs
- Peripherals
    - 2 UARTs
    - 2 SPI controller
    - 2 I2C controllers

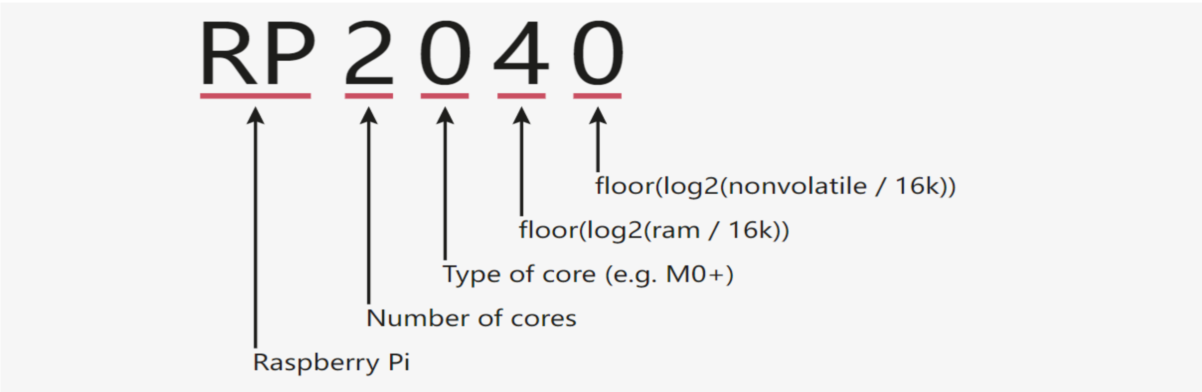
16 PWM channels

USB 1.1 controller and PHY, with host and device support

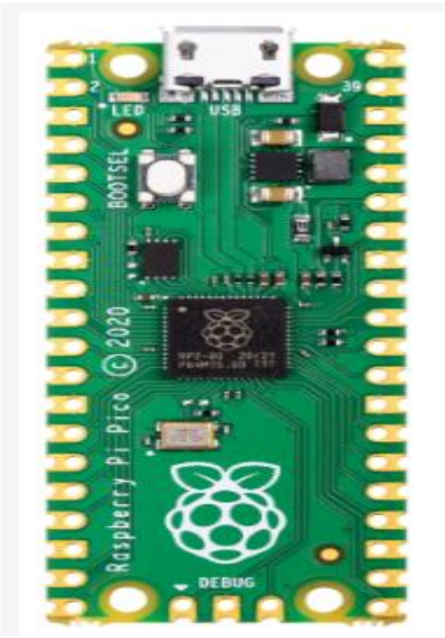
8 PIO state machines

**Features of RP2040 Chip — High performance. Low cost. Small package.**

The RP2040 features a dual-core Arm Cortex-M0+ processor clocked at 133MHz with 264KB internal SRAM and 2MB internal flash storage and can be programmed in both C/C++ and the beginner-friendly MicroPython.




(Picture quoted from Raspberry Pi Official)




Raspberry pi Pico

## Raspberry Pi Pico W and Pico WH

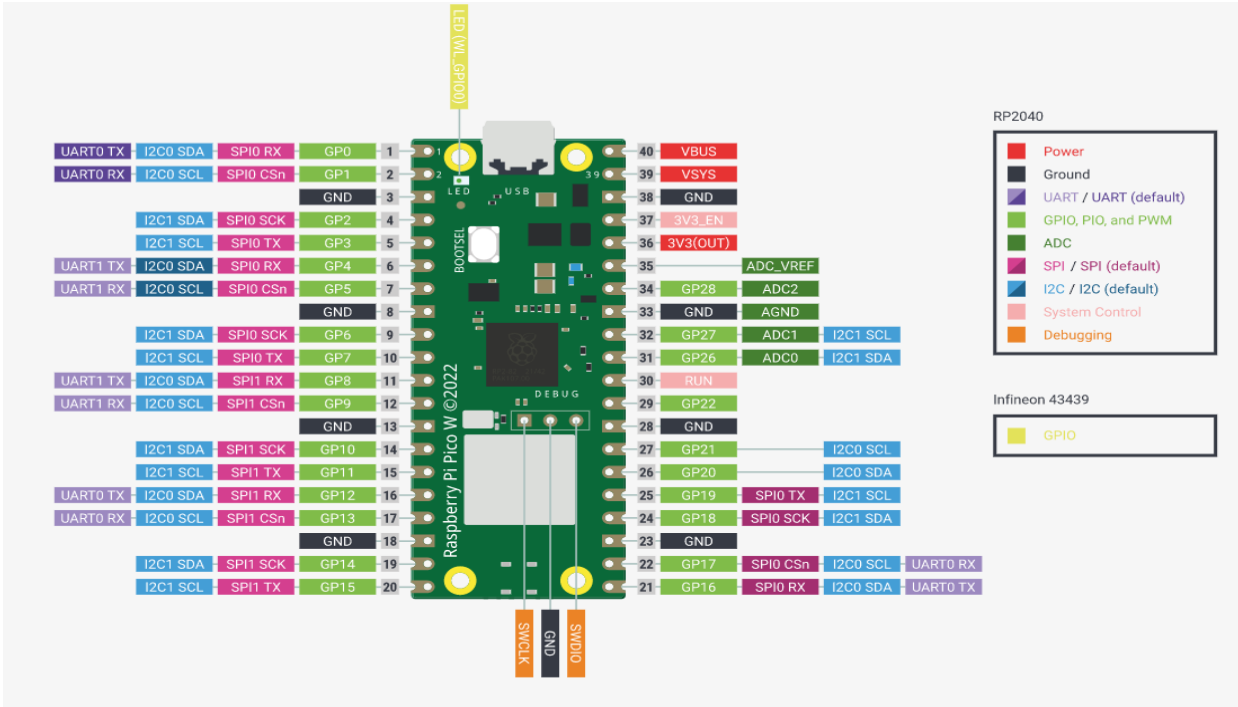
Raspberry Pi Pico W adds on-board single-band 2.4GHz wireless interfaces (802.11n) using the Infineon CYW43439 while retaining the Pico form factor. The on-board 2.4GHz wireless interface has the following features:

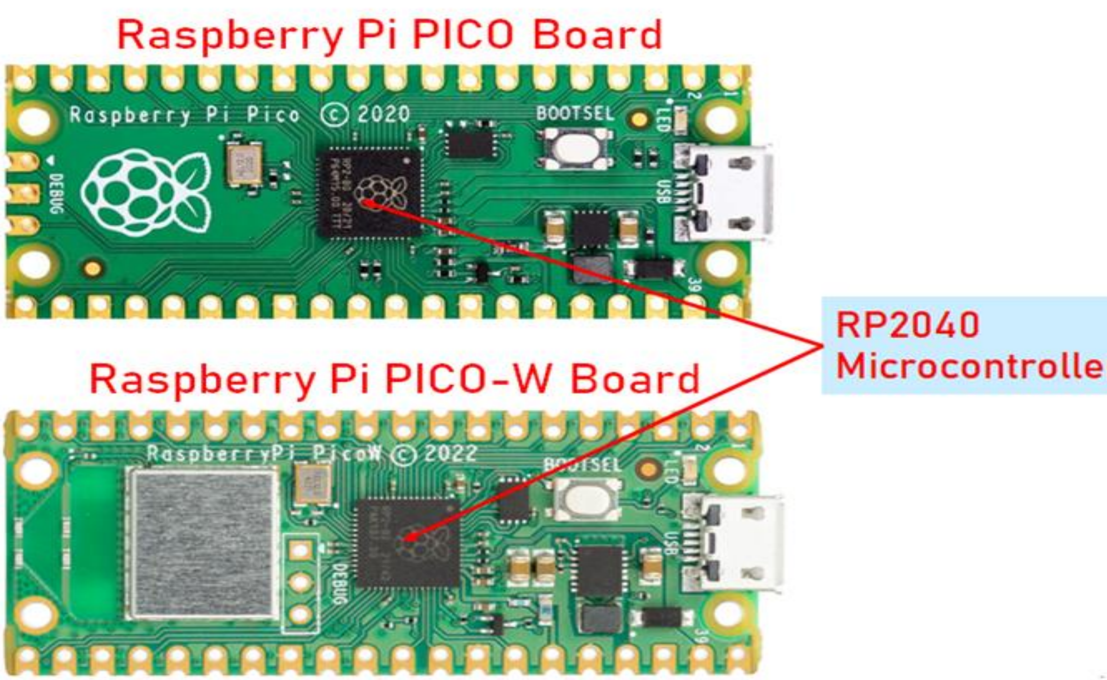
- Wireless (802.11n), single-band (2.4 GHz)
- WPA3
- Soft access point supporting up to four clients

The antenna is an onboard antenna licensed from ABRACON (formerly ProAnt). The wireless interface is connected via SPI to the RP2040 microcontroller.

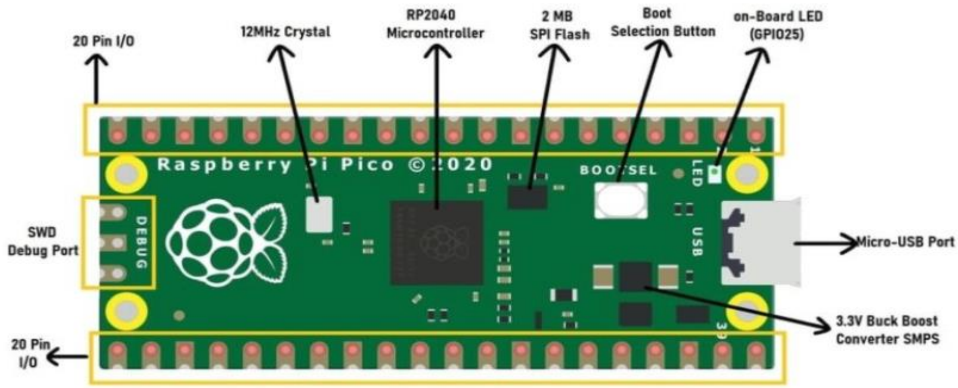
Due to pin limitations, some of the wireless interface pins are shared. The CLK is shared with VSYS monitor, so only when there isn't an SPI transaction in progress can VSYS be read via the ADC. The Infineon CYW43439 DIN/DOUT and IRQ all share one pin on the RP2040. Only when an SPI transaction isn't in progress is it suitable to check for IRQs. The interface typically runs at 33MHz.

For best wireless performance, the antenna should be in free space. For instance, putting metal under or close by the antenna can reduce its performance both in terms of gain and bandwidth. Adding grounded metal to the sides of the antenna can improve the antenna's bandwidth.

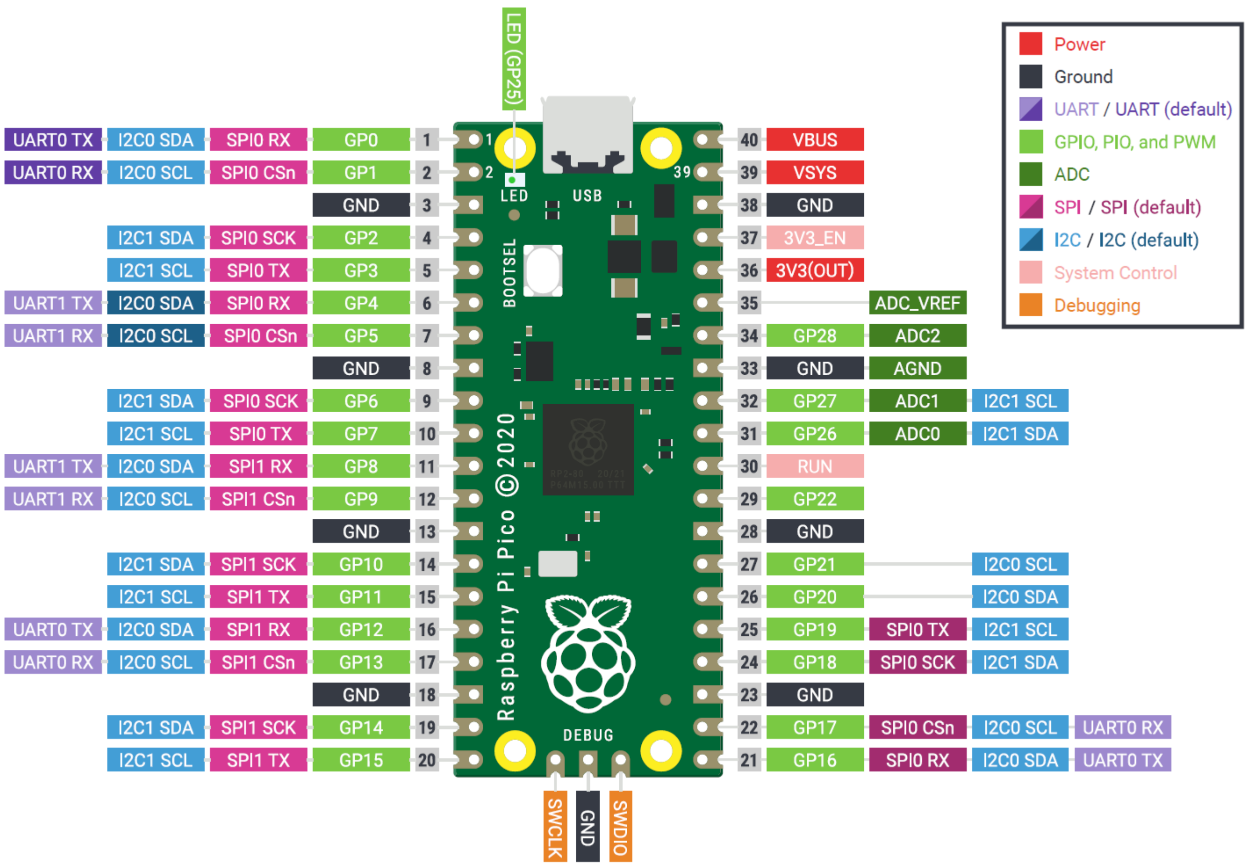



**RP2040 MICROCONTROLLER**




## RASPBERRY PI PICO PINOUT FEATURS




- Micro-USB B port for power and data (and for reprogramming the Flash)

- 12MHz Crystal Oscillator with two PLLs provide a system clock up to 133MHz, and a fixed 48MHz clock for USB or ADC.
- Code may be executed directly from 2MB of on-board Flash memorythrough a dedicated SPI.
- Raspberry Pi Pico is a microcontroller board based on the Raspberry Pi RP2040 - 32 bit microcontroller chip
- 3-pin ARM Serial Wire Debug (SWD) port.
- FourRP2040 GPIO pins are used for internal board functions, these are:

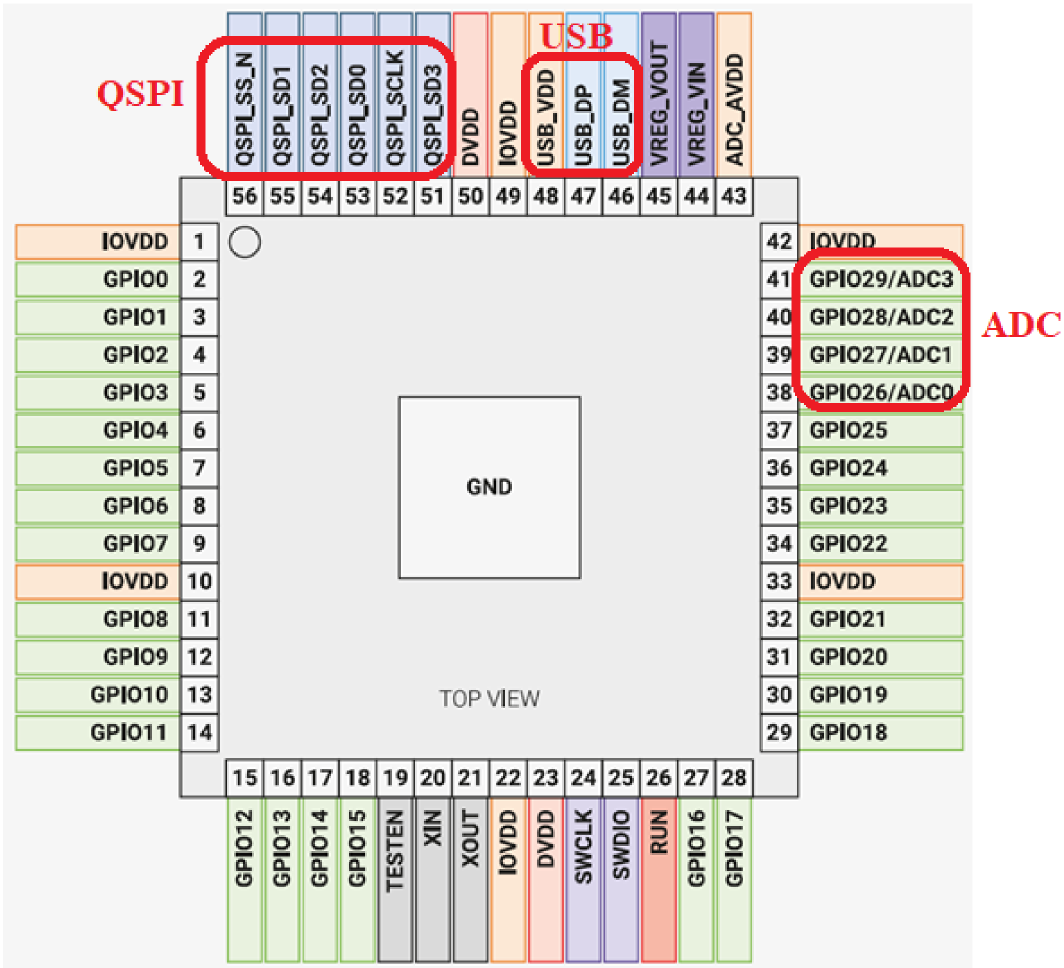
❖ GPIO29 IP (ADC3) Used to measure VSYS/3

❖ GPIO25 OP Connected to onboard LED

❖ GPIO24 IP VBUS sense - high if VBUS is present, else low

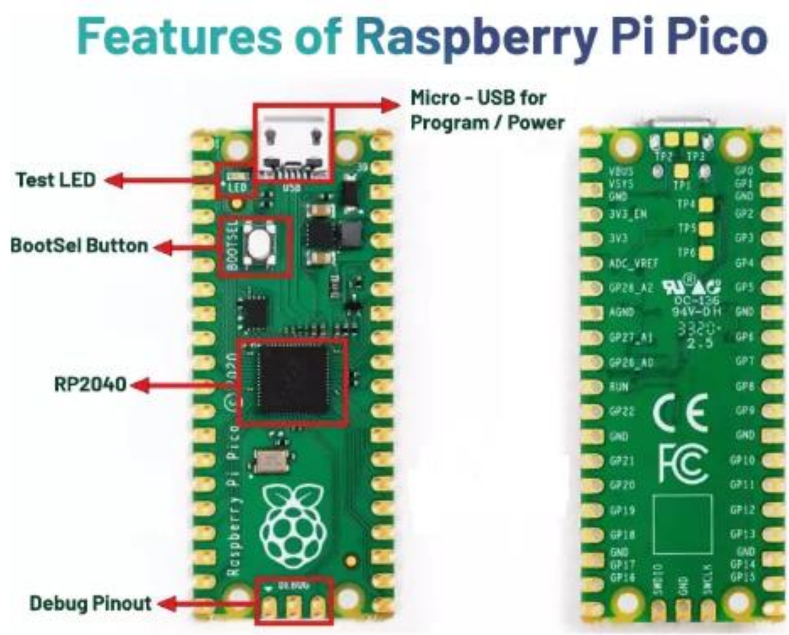
❖ GPIO23 OP Controls the on-board SMPS Power Save pin

**3.RP2040 MICROCONTROLLER**




**POWER SUPPLY:**

At its simplest, RP2040 requires two different voltage supplies, 3.3V (for the IO) and 1.1V (for the chip's digital core).




**CORTEX-M0 PROCESSOR FEATURES**

- The ARM Cortex-M0+ processor brings 32-bit power at an 8-bit Cost.
- Has the Smallest Footprint & Lowest Power requirements, consumes just 9µA/MHz on a low-cost 90nm LP process, around one third of the energy of any 8- or 16-bit processor available today, while delivering significantly higher performance.
- The low-power processor is suitable for a wide variety of applications, including sensors and wearables.
- Single-cycle IO to speed access to GPIO and peripherals.
- Improved debug and trace capability and a 2-stage pipeline.
- The exceptional code density of Cortex-M0+ significantly reduces memory requirements, ideal for use in wearables for healthcare, fitness, and more.
- With its three highly optimized low-power modes, the processor conserves energy to match processing demands.

## 4. RP2040 CARRIER BOARD

The Following On board peripherals  available in the carrier board

1. RP2040 Microcontroller
2. Bush Button
3. Seven Segment display
4. Potentiometer
5. Ultrasonic Sensor
6. 16*2 LCD Display
7. Buzzer
8. Relay
9. USB Connector
10. USB Cable ( B-Type)
11. LED
12. Temperature sensor
13. ADC


Vi Microsystems
1
2
3
4
5
6
7
8
9
10
11
12
13

## 5. SOFTWARE REQUIRED & PROGRAMMING LANGUAGE

**Raspberry Pi Pico Programming – Overview – GPIO Access**

The Raspberry Pi Pico accepts programming with the following programming languages: C/C++, MicroPython, assembly language.

Although the Pico by default is set up for use with the powerful and popular C/C++ language, many beginners find it easier to use MicroPython, which is a version of the Python programming language developed specifically for microcontrollers.

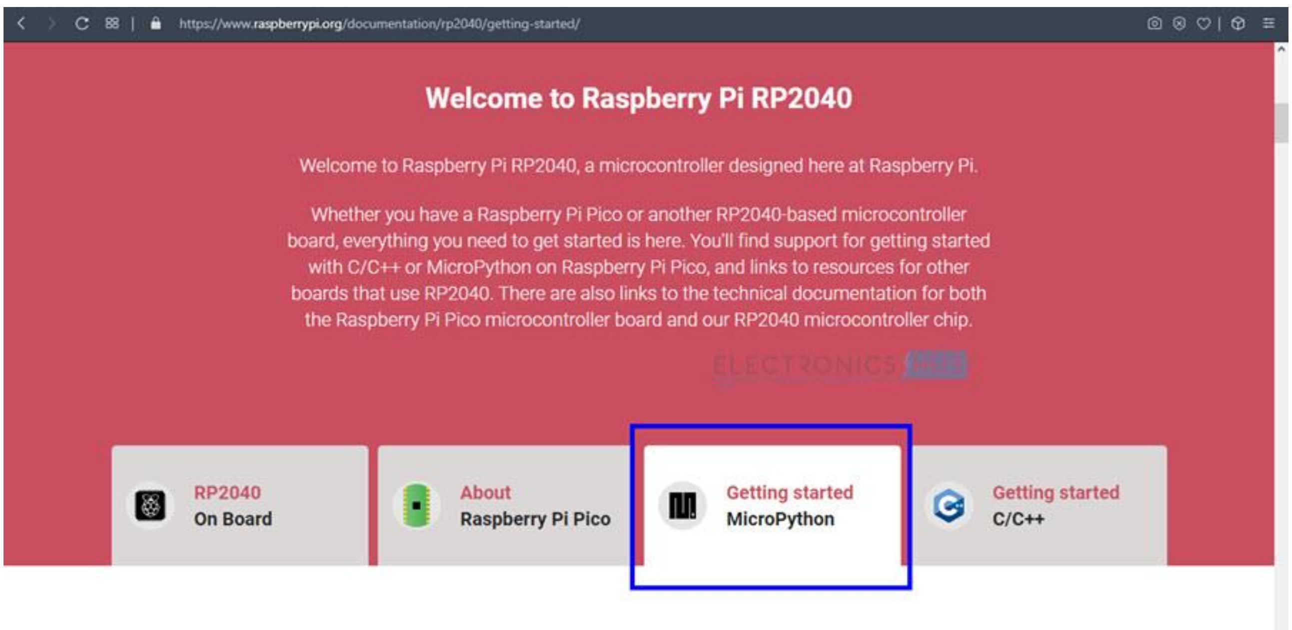
In this Lab we will learn how to install and use the MicroPython programming language. We will be using the Thonny text editor which has been developed specifically for Python programs.

### How to Install MicroPython on Raspberry Pi Pico?

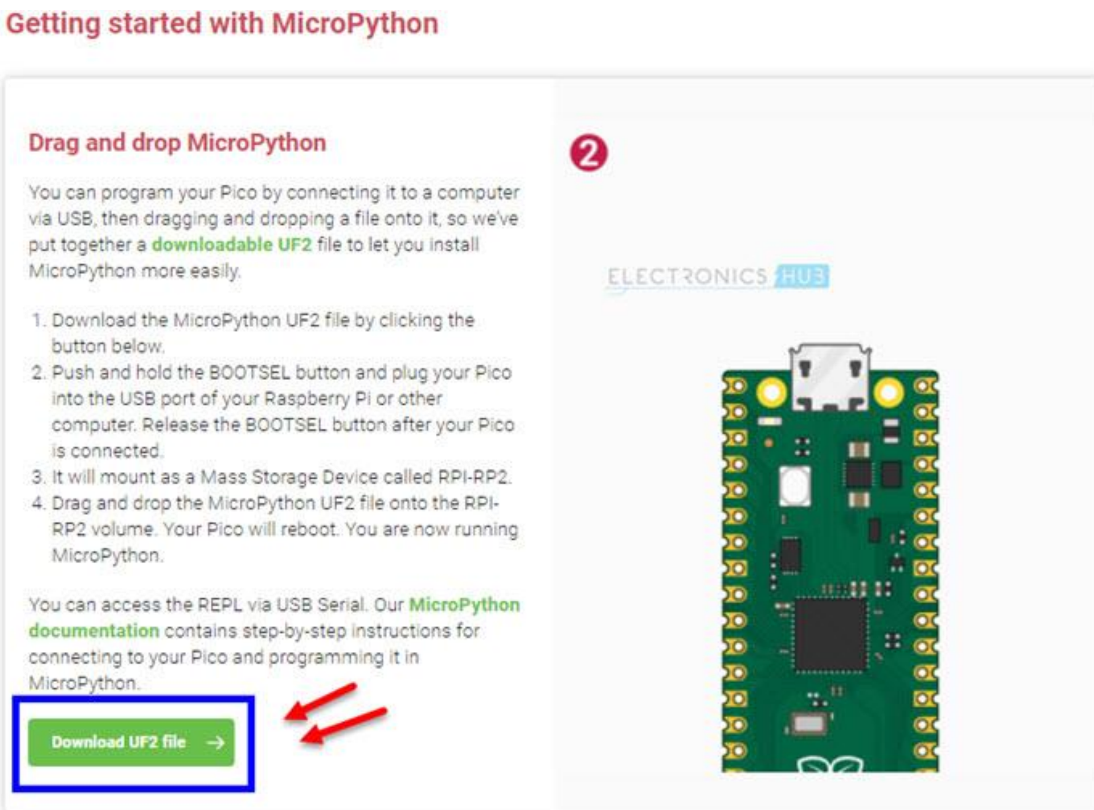
Download MicroPython Binary

Let us now get started with MicroPython on Raspberry Pico. The easiest and fastest way to run MicroPython on Raspberry Pi Pico is to download the prebuilt binary from the official Raspberry Pi Pico's website.

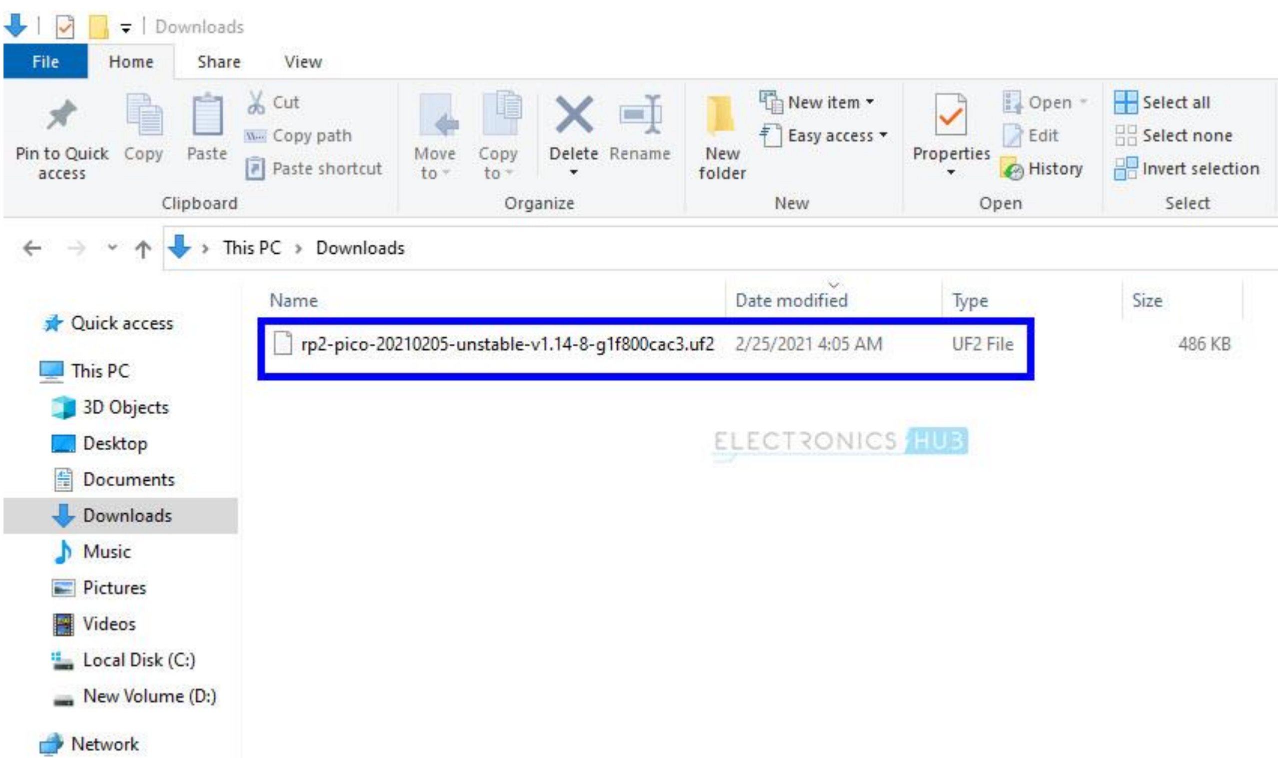
Go to the documentation page of Raspberry Pi Pico and click on "Getting Started MicroPython" tab.




The content below the tab changes according to the selected tab and when you click on "Getting Started MicroPython", a text related to Getting started with MicroPython appears along with a small animation on how to install MicroPython on Raspberry Pi Pico.




Read all the information and click on "Download UF2 file" option. A MicroPython Binary in the form of a .uf2 file will be downloaded.

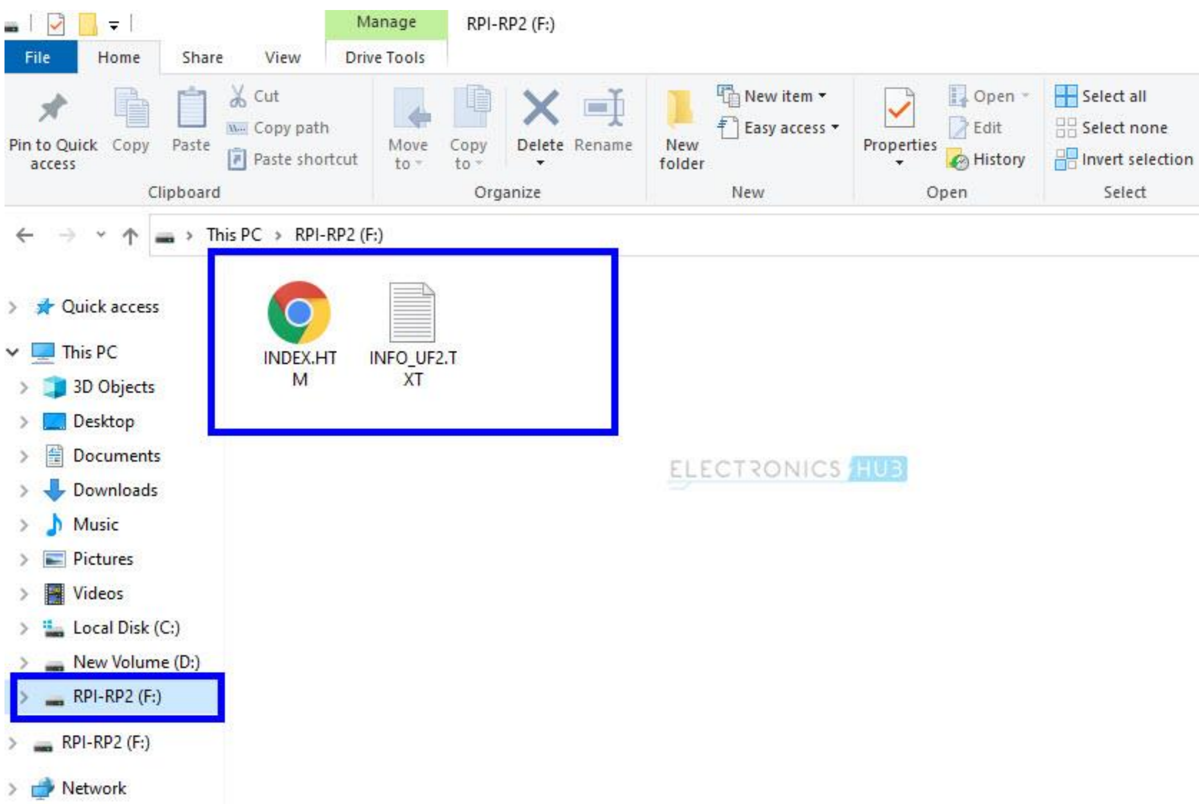


Install MicroPython on Raspberry Pi Pico

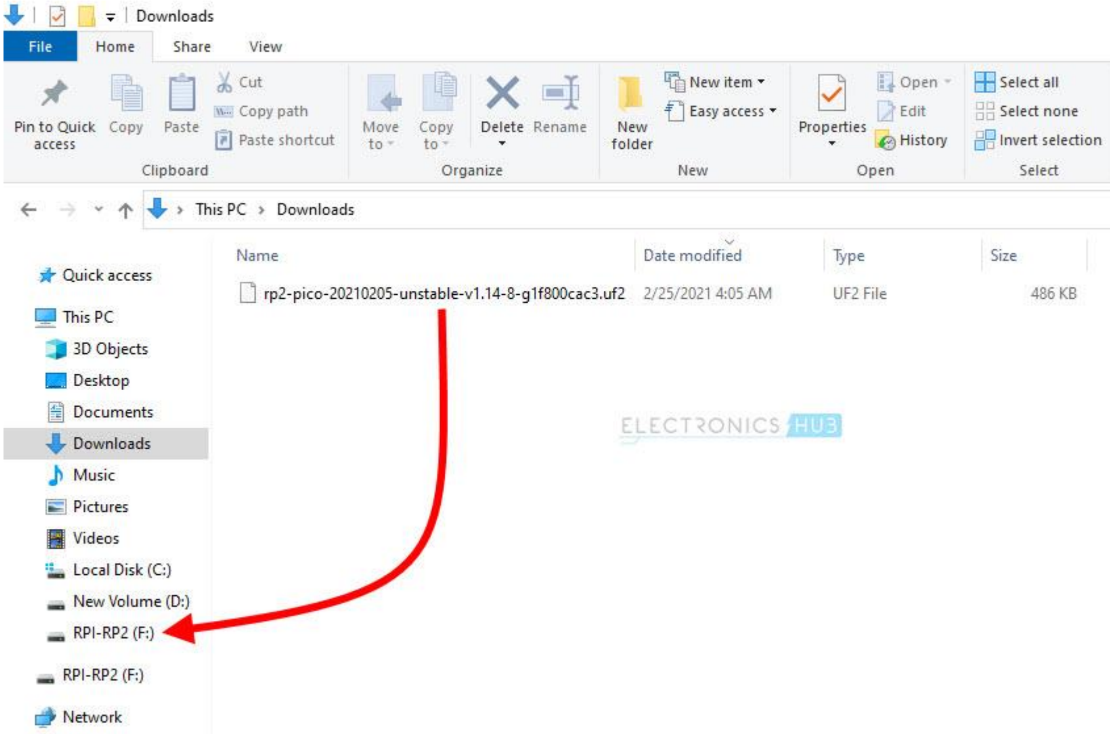
After downloading the MicroPython Binary, we have to upload this firmware in to the Raspberry Pi Pico. For that, first we have to put the Pico in bootloader mode.

To do that, plug-in a micro-USB cable to micro-USB port of Raspberry Pi Pico. Now, hold the BOOTSEL button on the Pico and plug-in the other end of the USB cable to a USB port of the host computer (while holding the BOOTSEL button).

You can release the button after a couple of seconds when the Raspberry Pi Pico appears as a Mass Storage Device with name “RPI-RP2”. If you open it, you will see a text file and an HTML file.

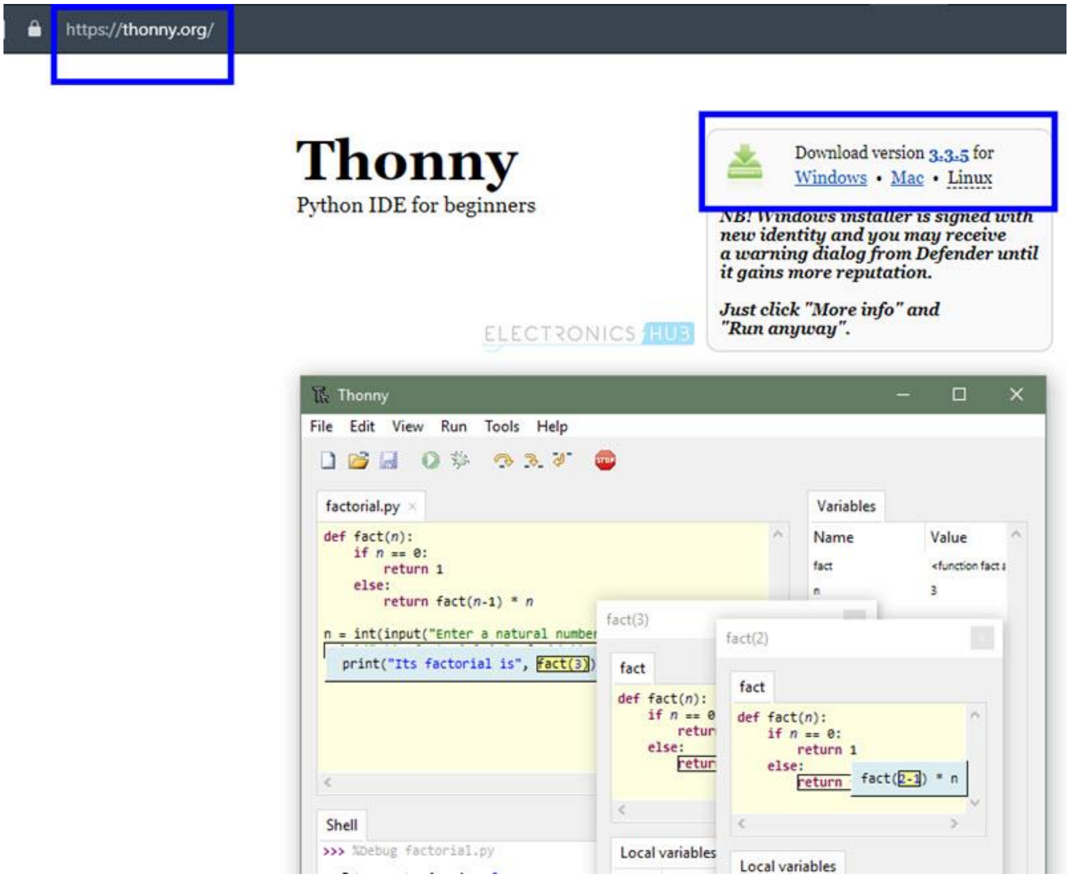



Now, go to the downloads folder and drag-and-drop the downloaded MicroPython UF2 file onto RPI-RP2 device. After copying, the Raspberry Pi Pico will restart and run MicroPython. The mass storage device will disappear after you copy the MicroPython UF2 file.




Your Raspberry Pi Pico is now running MicroPython. You are now ready to program Raspberry Pi Pico with MicroPython.

Downloading Thonny




Thonny is a simple Python IDE available for Windows, Mac and Linux. The Raspberry Pi OS comes with Thonny preinstalled. Since I am using a Windows system, I downloaded the Windows version of Thonny. An executable called “thonny-3.3.5.exe” is downloaded.

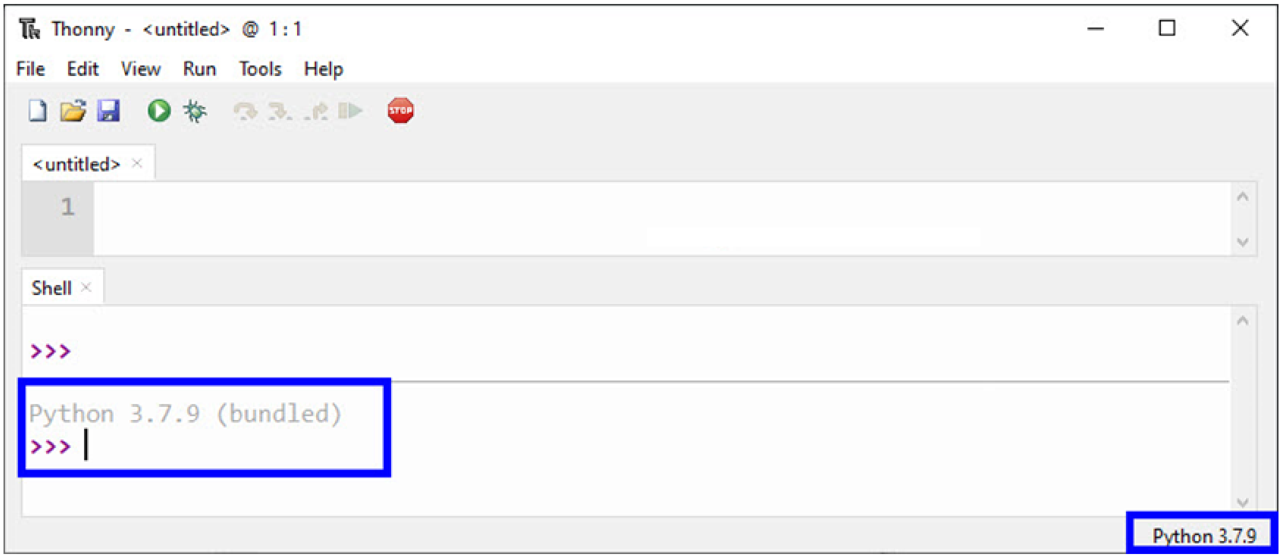
Double click on the downloaded executable and install Thonny. There is nothing special with this installation and it is very straight forward. Optionally, you can select to create a desktop shortcut.

### Configuring Thonny

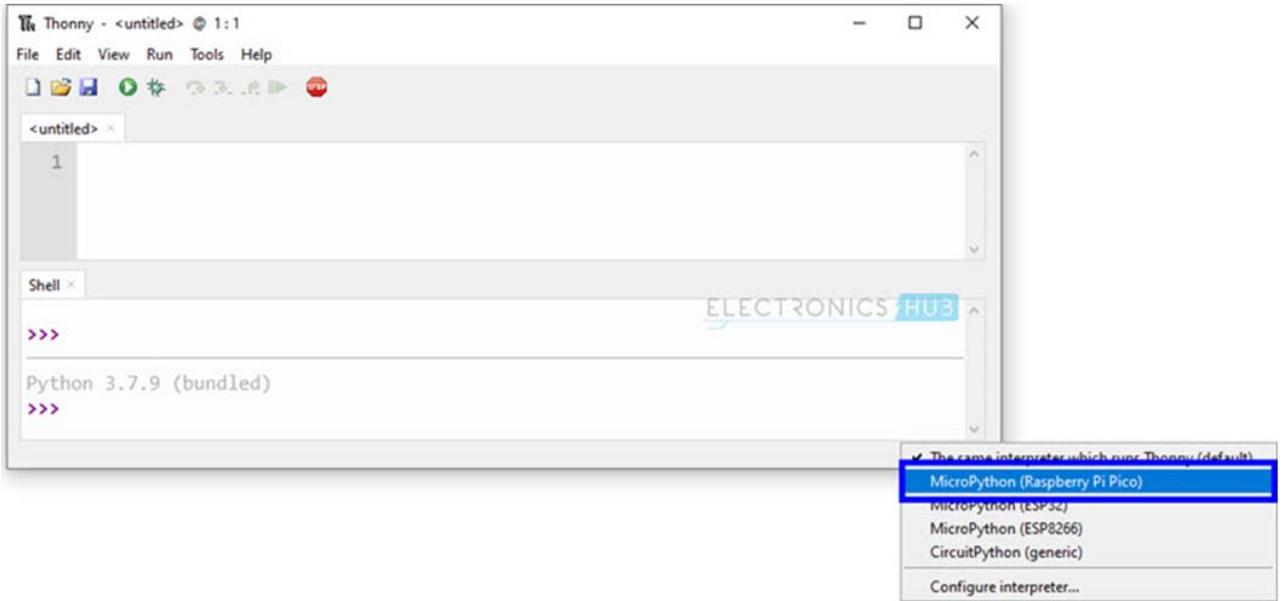
After downloading and installing Thonny IDE, open it. Make sure that Raspberry Pi Pico is already plugged into the host computer. Thonny IDE is very simple. Its layout can be divided into four parts: Toolbar, Script Area, Shell, Interpreter.


- The Toolbar: Contains icons for saving, running and stopping the programs.
- The Script Area: This is where you write the Python Programs.
- The Shell: The Python Shell is an interactive REPL (Read-Evaluate-Print-Loop) block where you can give individual commands to the interpreter and it will execute them.
- The Interpreter: Select the right interpreter from the bottom right of the IDE.

By default, Thonny IDE is configured to interpret Python 3.x.x.



Click on Python 3.7.9 (or whatever the version is) and select MicroPython (Raspberry Pi Pico) interpreter. As soon as you select the MicroPython interpreter, the shell at the bottom changes to MicroPython.



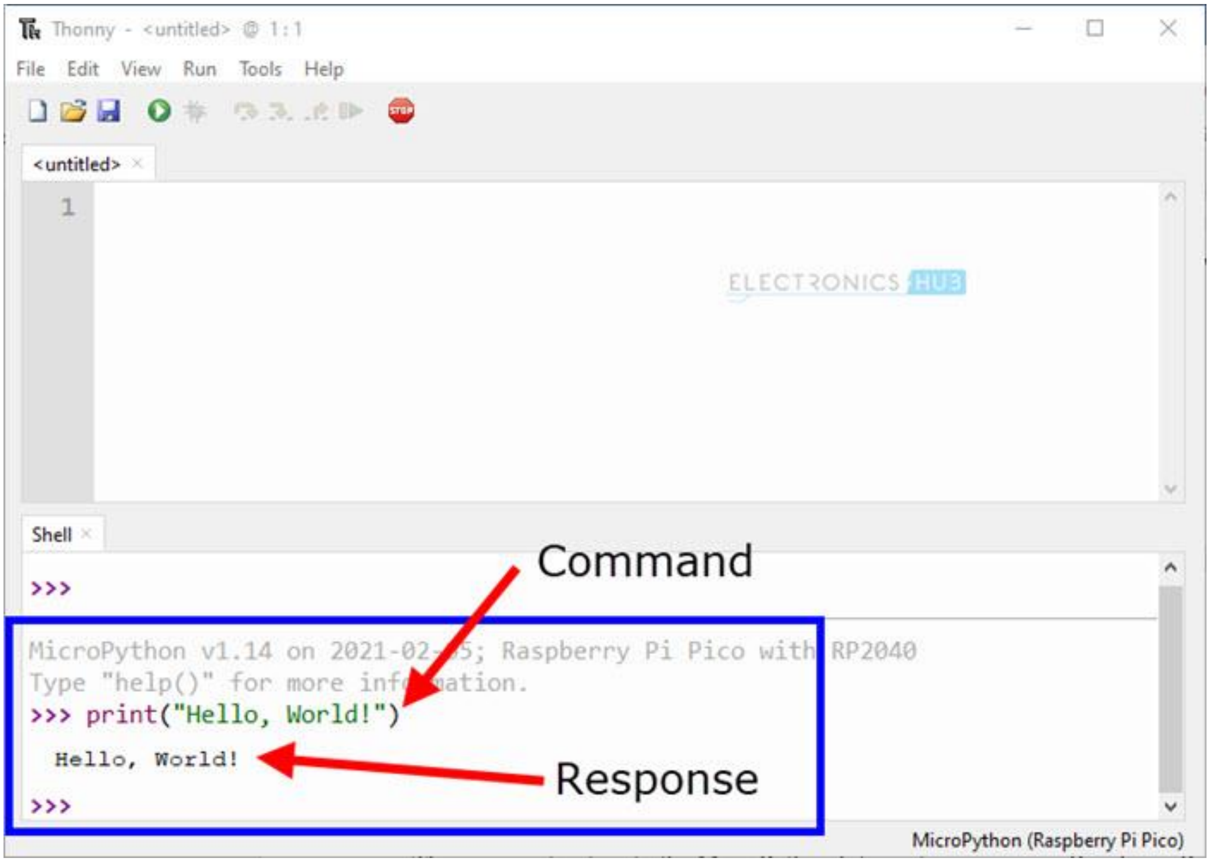

Since MicroPython supports interactive REPL, you can enter commands in the shell and Raspberry Pi Pico will execute them. Let us try this. We will start with Hello World of programs which is to print Hello World.

Programming Raspberry Pi Pico with MicroPython

In the Shell, type the following next to “>>>” symbol and hit enter.

```
print(“Hello, World!”)
```

This is an instruction to the MicroPython Interpreter running on Raspberry Pi Pico. Up on receiving this command, the MicroPython will respond with the message “Hello, World!” and prints it on the shell itself.




If you remember the layout of the Raspberry Pi Pico, an LED is connected to GPIO 25. We can try to turn this LED ON and OFF from the shell.

For that, first we have to import a special library called ‘machine’. The machine library in MicroPython is used to control the hardware of a board, Raspberry Pi Pico in this case. You can reset the microcontroller, put it to sleep, enable or disable interrupts, wake it from sleep using machine module.

Some of the classes of machine module are: Pin, Signal, ADC, UART, SPI, I2C, RTC, Timer, WDT, SD Card and so on.

We will learn about all the modules and their classes as and when we use them. The MicroPython documentation is good place to begin with if you want to explore further on MicroPython Libraries.

Since we want to use the GPIO block, we can import the ‘pin’ class from the ‘machine’, which is used to control the IO pins of the Raspberry Pi Pico.

```
from machine import Pin
```

Next, we create an object of class Pin and set the GPIO number and its direction i.e., Input or Output.

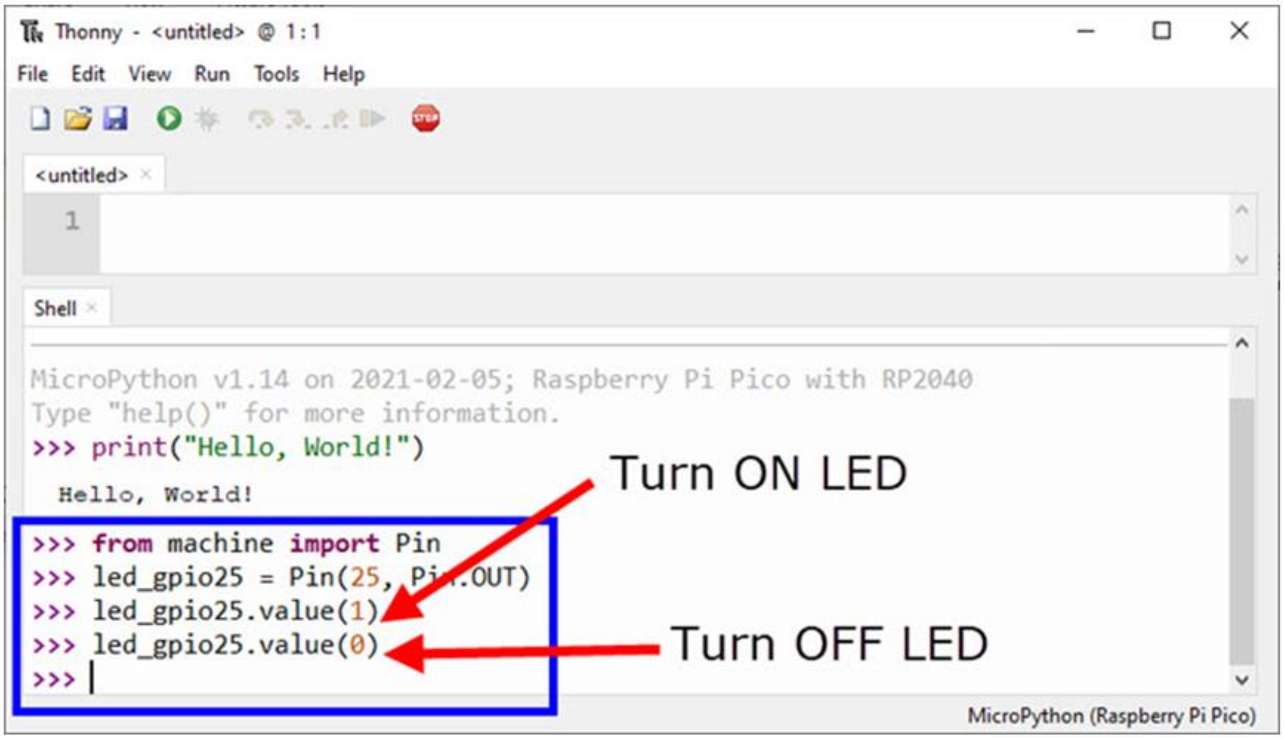
```
led_gpio25 = Pin(25, Pin.OUT)
```

To turn ON the LED, we have to set its value to 1.

```
led_gpio25.value(1)
```

Type the above lines one after the other in the shell. You can see the LED turned ON. To turn the LED OFF, set the value of the pin to 0.

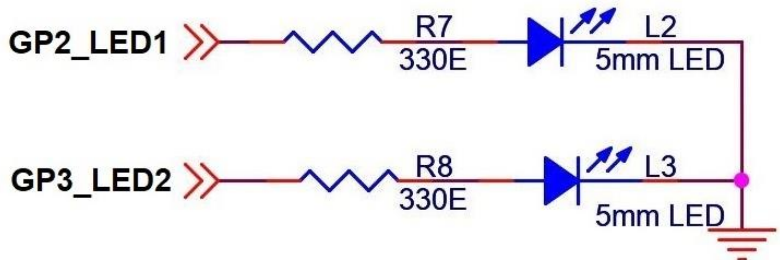
```
led_gpio25.value(0)
```




## Blink an LED

Executing commands from shell is good but what if you want to write a complete Python program? That is why you have the script area. Let us now see how can we write our first Python Program for Raspberry Pi Pico and Blink an LED.

In the board, 5mm red LEDs L2 and L3 are connected to GPIO 2 and GPIO 3 of Raspberry Pi Pico with the help of a 330Ω current limiting resistor.

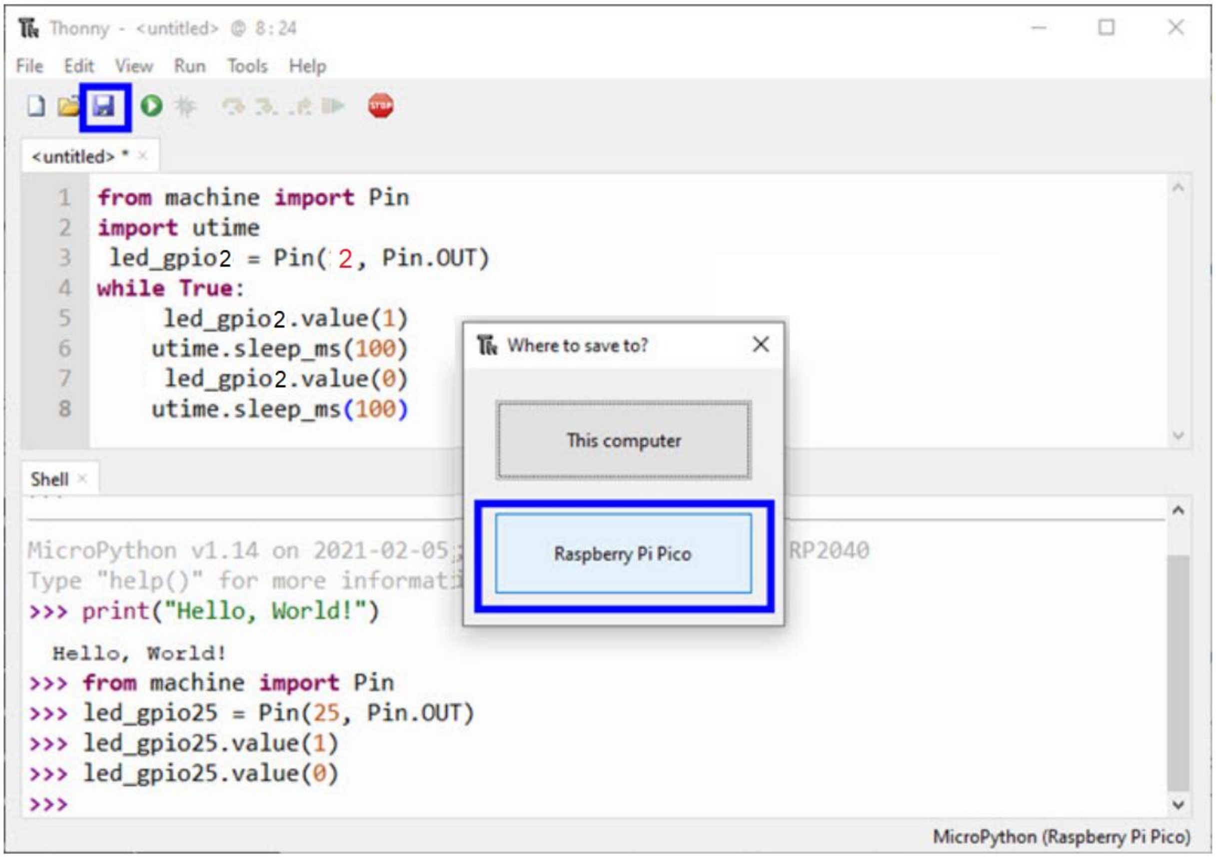



Now, in the script area type the following program. The code is commented for detailed explanation on what each line does. You can ignore the comments.

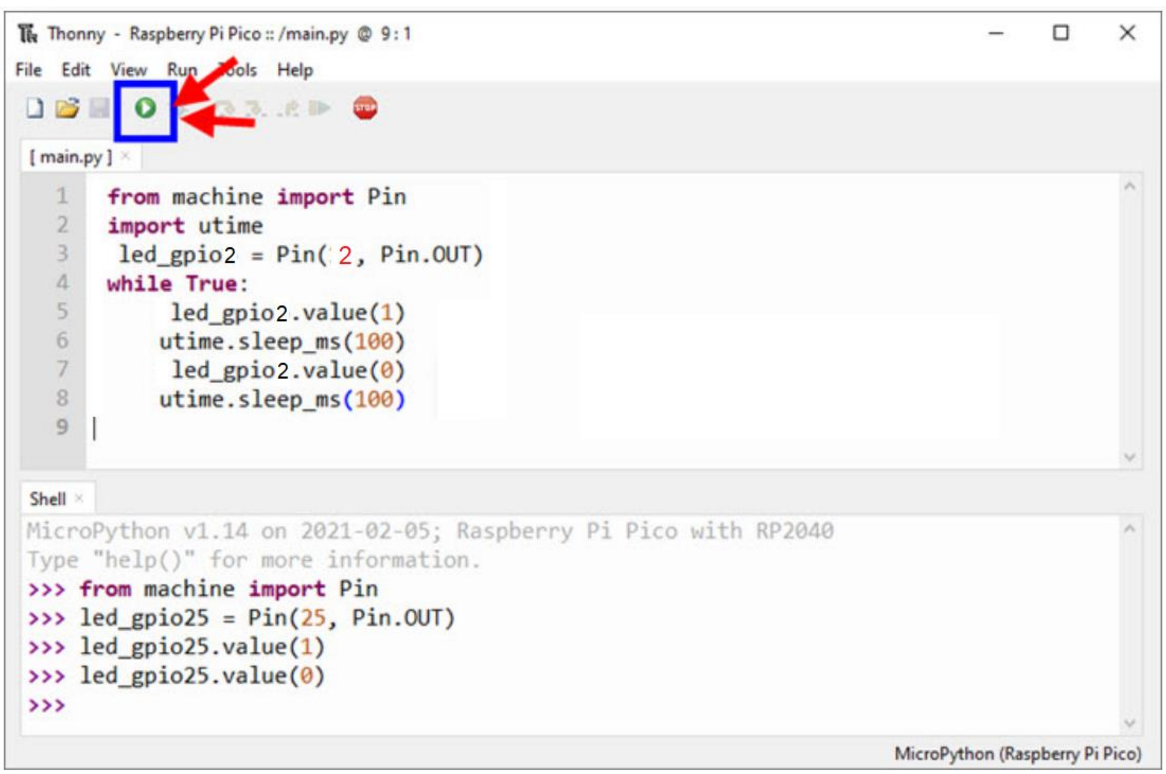
```
from machine import Pin          #Import Pin class from machine library to configure GPIO Pins.
import utime                     #Import utime library to implement delay
 led_gpio2 = Pin( 2 , Pin.OUT)  #create an object of Pin class and set GPIO Parameters (GPIO Pin, Direction).
while True:                           #Create an infinite loop. This is similar to while(1) in C.
      led_gpio2.value(1)              #Set value to 1 to turn ON LED.
     utime.sleep_ms(100)              #sleep_ms function provides delay in milliseconds.
      led_gpio2.value(0)              #Set value to 0 to turn OFF LED.
     utime.sleep_ms(100)              #Provide another 100ms delay to see the LED Blinking.
```

Raspberry-Pi-Pico-Demo.py hosted with ❤ by GitHub view raw

Click on save and select Raspberry Pi Pico, when asked.




Give a name as “main.py” to the file and click on OK.



### Reason for Naming main.py

When you reset any microcontroller (either power down completely and power it on or button reset), you expect the microcontroller to execute the program once again. If you want the same thing to happen in Raspberry Pi Pico, then you have to name the Python script as ‘main.py’. You can provide any name for the Python program when saving like ‘blinky.py’ but it will not execute if you remove the power and reconnect it. For this, you have to name the Python Program as ‘main.py’.
Even if you have multiple Python Scripts in Raspberry Pi Pico, if there is a file named main.py, then MicroPython will execute that.

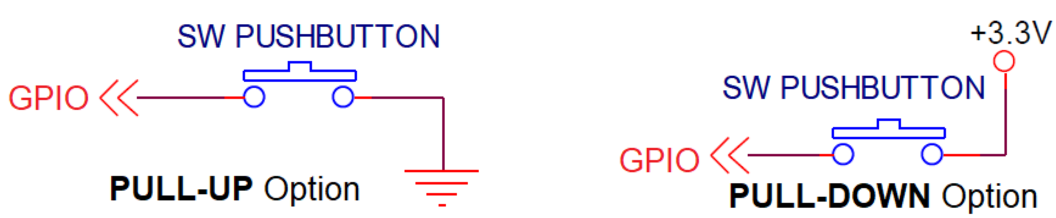
Instead of setting the value to 1 and 0, you can use the toggle function to reduce the code.

```
from machine import Pin        #Import Pin class from machine library to configure GPIO Pins.
import utime                   #Import utime library to implement delay
led_gpio2 = Pin(2 , Pin.OUT)  #create an object of Pin class and set GPIO Parameters (GPIO Pin, Direction).
while True:                         #Create an infinite loop. This is similar to while(1) in C.
        led_gpio25.toggle()    #Toggle the status of LED.
        utime.sleep_ms(100)    #sleep ms function provides delay in milliseconds.
```

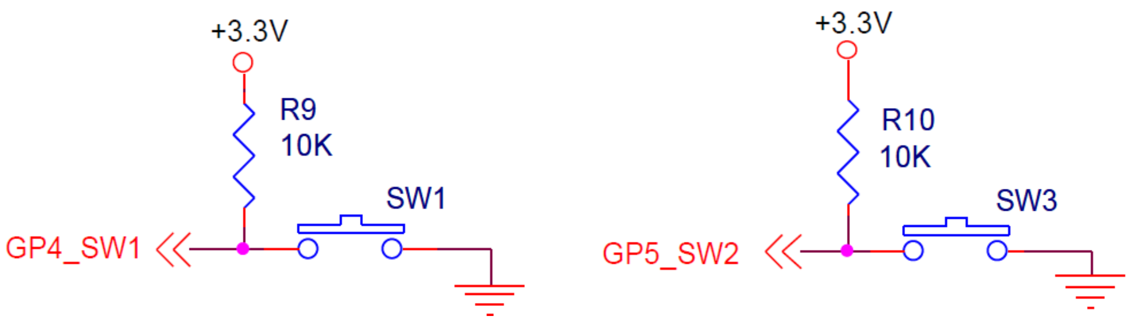
Raspberry-Pi-Pico-Demo.py hosted with ❤ by GitHub view raw

## Read from Button

We have seen how to set Raspberry Pi Pico’s GPIO Pin as Output and Blink an LED. Let us now extend this by setting a GPIO Pin as an Input and connecting a Button to the GPIO Pin. We will read the status of the Button and toggle the state of the LED.




In the **Board,**




GPIO 4 & GPIO 5, can be used as an Input Pin and connected to a simple momentary push button switch pulled up through a 10K Resistor, as shown above. So, normally the Pico reads HIGH on the button pin but when the button is pushed, Pico reads LOW on the button pin.

```
from machine import Pin     #Import Pin class from machine library to configure GPIO Pins.
import utime                #Import utime library to implement delay
ledPin = Pin( 2 , Pin.OUT)  #create an object of Pin class and set GPIO Parameters (GPIO Pin, Direction).
buttonPin = Pin( 4, Pin.IN)
while True:                 #Create an infinite loop. This is similar to while(1) in C.
    if buttonPin.value() == 1:
        utime.sleep_ms(20)
        ledPin.toggle()
```

Raspberry-Pi-Pico-Read-Button.py hosted with ❤ by GitHub view raw

## 6.PROGRAM

### 6.1 LED

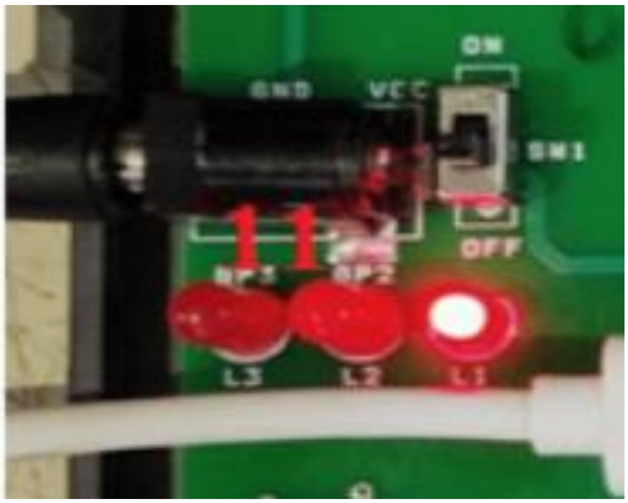
Light Emitting Diodes are the outputdevice connected with Gpio pin GP2 and GP3 of RP2040 .

**PROGRAM :**

```
from machine import Pin
import utime
led1 =Pin(2,Pin.OUT)
led2 =Pin(3,Pin.OUT)
delay = .50
while True:
  led1.value(1)
```

```
led2.value(0)
print("Led1 On")
utime.sleep(delay)
led1.value(0)
led2.value(1)
print("Led2 On")
utime.sleep(delay)
led1.value(1)
led2.value(1)
print("Led1 & Led2 On")
utime.sleep(delay)
led1.value(0)
led2.value(0)
print("Led1 & Led2 Off")
utime.sleep(delay)
```

Carrier Board  LED Connection




## Output:

```
Shell
>>> %Run -c $EDITOR_CONTENT
 Led1 On
 Led2 On
 Led1 & Led2 On
 Led1 & Led2 Off
 Led1 On
 Led2 On
 Led1 & Led2 On
 Led1 & Led2 Off
 Led1 On
 Led2 On
```

## 6.2. BUTTON WITH LED

**PROGRAM :**

```
from machine import Pin
from utime import sleep_ms
button1 = Pin(5, Pin.IN, Pin.PULL_UP)#Internal pull-up
button2 = Pin(4, Pin.IN, Pin.PULL_UP)
led1 = Pin(2, Pin.OUT)
led2 = Pin(3, Pin.OUT)                #0 means that the light is currently off
if __name__ == '__main__':
  while True:
    if button1.value() == 0:    #key press
        led1.value(1)
    else:
        led1.value(0)
    if button2.value() == 0:    #key press

        led2.value(1)

    else:
```

```
led2.value(0)
```

### 6.3. LCD DISPLAY:

- Add  GpioLcd.py  library for lcd display
- Must save as Gpiolcd.py in RP2040

**PROGRAM:**

```
from machine import Pin
from gpio_lcd import GpioLcd
import time
count=0
lcd = GpioLcd(rs_pin = Pin(8),
        enable_pin = Pin(9),
        d4_pin = Pin(10),
        d5_pin = Pin(11),
        d6_pin = Pin(12),
        d7_pin = Pin(13))
lcd.move_to(0,0)
lcd.putstr("PERSON COUNTER")
lcd.move_to(0,1)
lcd.putstr("TOTAL COUNT :")
while (1):
  count=count+1
  lcd.move_to(13,1)
  lcd.putstr(str(count))
  time.sleep(1)
```

**Output:**


## 6.4.INTERFACING LM35 WITH LCD DISPLAY

**PROGRAM :**

```
from machine import Pin
from gpio_lcd import GpioLcd
import time
import utime
conversion_factor = 3.3/(65536)
adc2 = machine.ADC(27)
lcd = GpioLcd(rs_pin = Pin(8),
        enable_pin = Pin(9),
        d4_pin = Pin(10),
        d5_pin = Pin(11),
        d6_pin = Pin(12),
        d7_pin = Pin(13))
while True:
    val2 = adc2.read_u16()
    temp = (val2 * conversion_factor)*100
    temp1 = int(temp)
    temp2 = str(temp1)
```

```
print("==============================")
print("temperature: ",temp1)
lcd.move_to(0,0)
lcd.putstr('TEMP in Degree :')
lcd.move_to(0,1)
lcd.putstr(temp2)
time.sleep(0.8)
```

## Output:


### 6.5. SEVEN SEGMENT DISPLAY: ( connected suing i2c , GP20 and Gp21 SDA ,SCL)

Add library for seven segment …Must save as pcf8574.py save in RP2040

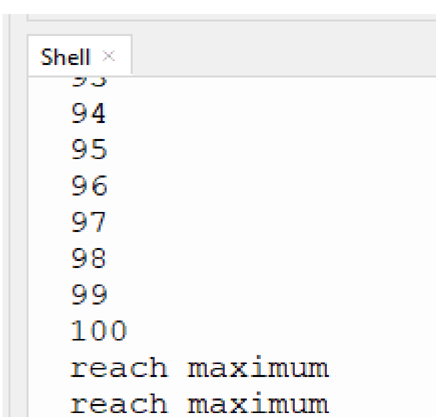
### PROGRAM : ( Count digital number 0 to 100 )

```
import pcf8574
from machine import I2C, Pin
```

```
import time
import array as arr
count=0
mod=0
mod1=0
m = arr.array('i', [0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x67])

i2c = I2C(id=0,scl=Pin(21),sda=Pin(20),freq=100000)
pcf = pcf8574.PCF8574(i2c, 0x21)
pcf.port =0x00
pcf = pcf8574.PCF8574(i2c, 0x20)
pcf.port =0x00
# for n in range(11):
#     pcf.port =m[n]
#     time.sleep(0.5)
while(1):
    if(count<=99):
        pcf = pcf8574.PCF8574(i2c, 0x21)
        count=count+1
        mod = count % 10
        mod1 = count / 10
        pcf.port =m[int(mod)]
        print(count)
if(count==10):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]
        if(count==20):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]
        if(count==30):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]
```

```
    if(count==40):
     pcf = pcf8574.PCF8574(i2c, 0x20)
     pcf.port =m[int(mod1)]
    if(count==50):
     pcf = pcf8574.PCF8574(i2c, 0x20)
     pcf.port =m[int(mod1)]
    if(count==60):
     pcf = pcf8574.PCF8574(i2c, 0x20)
     pcf.port =m[int(mod1)]
    if(count==70):
     pcf = pcf8574.PCF8574(i2c, 0x20)
     pcf.port =m[int(mod1)]
    if(count==80):
     pcf = pcf8574.PCF8574(i2c, 0x20)
     pcf.port =m[int(mod1)]
    if(count==90):
     pcf = pcf8574.PCF8574(i2c, 0x20)
     pcf.port =m[int(mod1)]
else:
    print("reach maximum")
time.sleep(1)
```

**Output:**



**6.6. POT WITH SEVEN SEGMENT INTERFACING**

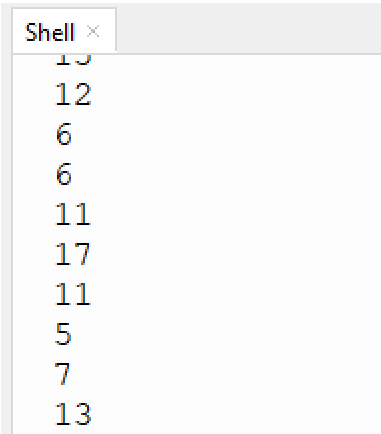
**PROGRAM :**

```
import pcf8574
from machine import I2C, Pin
import array as arr
import machine
import utime
analog_value = machine.ADC(28)
count=0
mod=0
mod1=0
m = arr.array('i', [0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x67])

i2c = I2C(id=0,scl=Pin(21),sda=Pin(20),freq=100000)
pcf = pcf8574.PCF8574(i2c, 0x21)
pcf.port =0x00
pcf = pcf8574.PCF8574(i2c, 0x20)
pcf.port =0x00
while(1):
    reading = analog_value.read_u16()
    value_in_digi =((reading/65536)*100)
    count= int(value_in_digi)
    print(count)
    mod = count % 10
    mod1 = count / 10
if(count<=99):
     pcf = pcf8574.PCF8574(i2c, 0x21)
     pcf.port =m[int(mod)]
```

```
        if(count>=10)or(count<=19):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

        if(count>=20)or(count<=29):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

        if(count>=30)or(count<=39):
  pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

        if(count>=40)or(count<=49):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

        if(count>=50)or(count<=59):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

if(count>=60)or(count<=69):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

        if(count>=70)or(count<=79):
         pcf = pcf8574.PCF8574(i2c, 0x20)
         pcf.port =m[int(mod1)]

        if(count>=80)or(count<=89):
```

```
        pcf = pcf8574.PCF8574(i2c, 0x20)
        pcf.port =m[int(mod1)]
       if(count>=90)or(count<=99):
        pcf = pcf8574.PCF8574(i2c, 0x20)
        pcf.port =m[int(mod1)]
 else:
        print("reach maximum")
 utime.sleep(0.5)
```

**Output:**




**6.7. ULTRASONIC SENSOR**

Download the following library and upload it to Raspberry Pi Pico board with the name of hcsr04.py under the lib folder.

Must hcsr04.py save as RP2040

Inside the pico Board…

**PROGRAM :**

```
from hcsr04 import HCSR04 # we have to add this library file in the same folder or else it may not work sometimes

from time import sleep

sensor = HCSR04(trigger_pin=15, echo_pin=14, echo_timeout_us=10000)

while True:
```

```
    distance = sensor.distance_cm()
    print('Distance:', distance, 'cm')
    sleep(1)
```

**Output:**

```
Shell
>>> %Run -c $EDITOR_CONTENT
  Distance: 97.43986 cm
  Distance: 98.24742 cm
  Distance: 97.47423 cm
  Distance: 4.037801 cm
  Distance: 4.570446 cm
  Distance: 14.0378 cm
  Distance: 24.22681 cm
  Distance: 38.19588 cm
  Distance: 9.140893 cm
  Distance: 97.35395 cm
  Distance: 96.47766 cm
  Distance: 96.06529 cm
  Distance: 98.23024 cm
```

**5.8. ULTRASONIC WITH LCD**

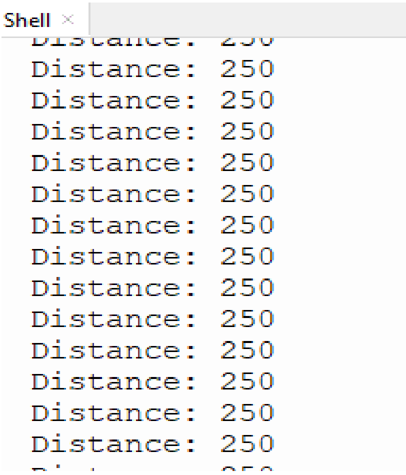
**PROGRAM:**

```
from machine import Pin
from gpio_lcd import GpioLcd
import time
from hcsr04 import HCSR04
```

```
import utime
sensor = HCSR04(trigger_pin=15, echo_pin=14, echo_timeout_us=10000)
lcd = GpioLcd(rs_pin = Pin(8),
        enable_pin = Pin(9),
        d4_pin = Pin(10),
        d5_pin = Pin(11),
        d6_pin = Pin(12),
d7_pin = Pin(13))
def countDigits(n):
   ans = 0
   while (n):
      ans = ans + 1
      n = n // 10
   return ans
while True:
   distance = sensor.distance_cm()
   distance1 = int(distance)
   print('Distance:', distance1)
   n=countDigits(distance1)
   if(n==1):
    lcd.move_to(1,1)
    lcd.putstr('  ')

   if(n==2):
    lcd.move_to(2,1)
    lcd.putstr(' ')
    lcd.move_to(0,0)
    lcd.putstr('Distance in Cm :')
    lcd.move_to(0,1)
    lcd.putstr(str(distance1))
```

```
    time.sleep(0.5)
```

**Output:**

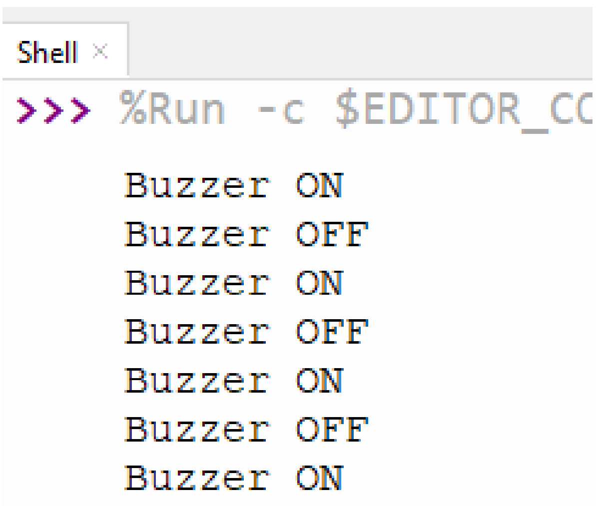

**6.8. BUZZER**

**PROGRAM :**

```
from machine import Pin
import time
buzzer = Pin(6,Pin.OUT)
while True:
    buzzer.value(1)
    print("   Buzzer ON")
```

```
    time.sleep(0.5)
    buzzer.value(0)
    print("   Buzzer OFF")
    time.sleep(0.5)
```

**Output:**



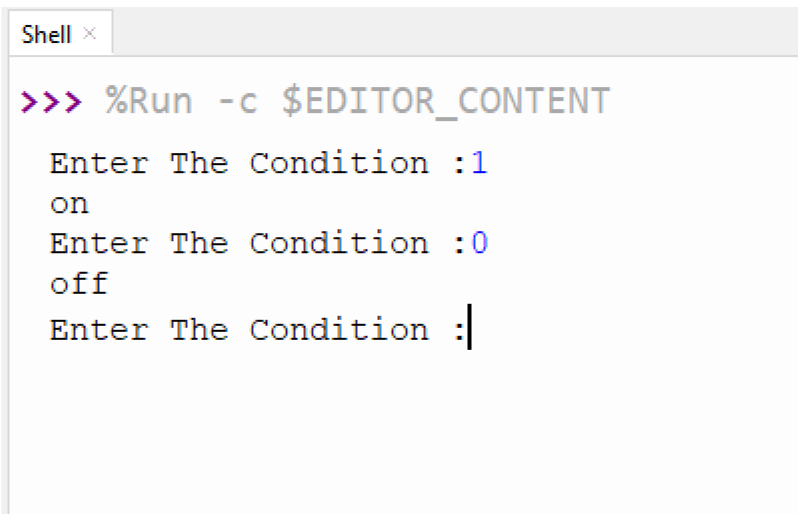

**6.9. RELAY WITH SERIAL**

**PROGRAM :**

```
from machine import Pin
import utime
m=0
led=Pin(7,Pin.OUT)
led.low()
while(1):
    m=int(input("Enter The Condition :"))
    if(m==1):
```

```
        led.value(1)
        print("on")
    elif(m==0):
        led.value(0)
        print("off")
    else:
        print("unknown charachters")
```

**Output:**




**6.10.SWITCH INCREMENT WITH LED**

**PROGRAM :**

```
from machine import Pin
import time
Btn_Pin1 = 4
Btn_Pin2 = 5
counter = 0
def setup():
    global sw_BtN1
```

```
    global sw_BtN2
    global led
    sw_BtN1 = Pin(Btn_Pin1,Pin.IN, Pin.PULL_UP)
    sw_BtN2 = Pin(Btn_Pin2,Pin.IN, Pin.PULL_UP)
    led = Pin(2, Pin.OUT)
def loop():
    while True:
        global sw_BtN1
        global sw_BtN2
        global counter
        global led
        if sw_BtN2.value()==0:
            print("Button Pressed2")
            led.value(1)
            counter+=1
            print("Count={}".format(counter))
while(1):
                if sw_BtN2.value()==1:
                    time.sleep(0.1)
                    break
if sw_BtN1.value()==0:
            print("Button Pressed1")
            led.value(1)
            counter-=1

            if(counter<=0):
              counter=0
            print("Count={}".format(counter))
            while(1):
                if sw_BtN1.value()==1:
```

```
        time.sleep(0.1)
        break


if __name__ == '__main__':
    setup()
    loop()
```

**Output:**

```
Shell
>>> %Run -c $EDITOR_CONTENT
  Button Pressed2
  Count=1
  Button Pressed1
  Count=0
  Button Pressed1
  Count=0
  Button Pressed2
  Count=1
```

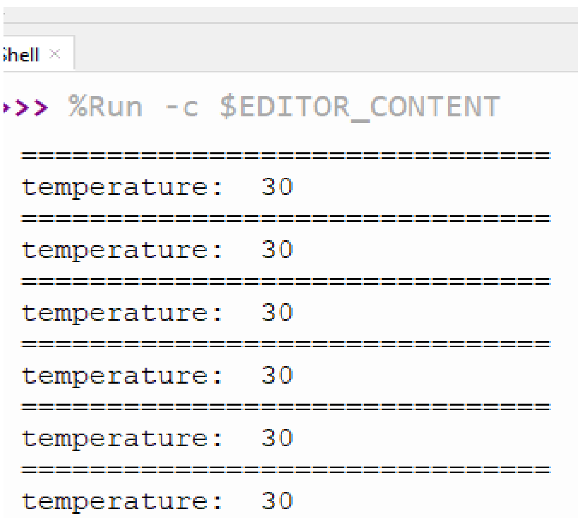
### 6.11. TEMPERATURE SENSOR

**PROGRAM :**

```
from machine import Pin
import time
import utime
conversion_factor = 3.3/(65536)
adc2 = machine.ADC(27)
while True:
    val2 = adc2.read_u16()
    temp = (val2 * conversion_factor)*100
    temp1 = int(temp)
```

```
temp2 = str(temp1)
print("=============================")
print("temperature: ",temp1)
time.sleep(0.8)
```

**Output:**



```
Shell
>>> %Run -c $EDITOR_CONTENT
=============================
temperature:  30
=============================
temperature:  30
=============================
temperature:  30
=============================
temperature:  30
=============================
temperature:  30
=============================
temperature:  30
```

**6.12. ADC**

**PROGRAM :**

```
import machine
import utime

analog_value = machine.ADC(28)

while True:
    reading = analog_value.read_u16()
    #value_in_digi ="{:.0f}".format(reading/660)
```

```
    print(reading)
    utime.sleep(0.1)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT

 POT Value :  8610
 Voltage Value :  0.4335546
 POT Value :  10642
 Voltage Value :  0.5358756
 POT Value :  9346
 Voltage Value :  0.4706158
 POT Value :  5137
 Voltage Value :  0.2586725
 POT Value :  3200
 Voltage Value :  0.1611353
 POT Value :  4689
 Voltage Value :  0.2361135
```

**6.13. PWM**

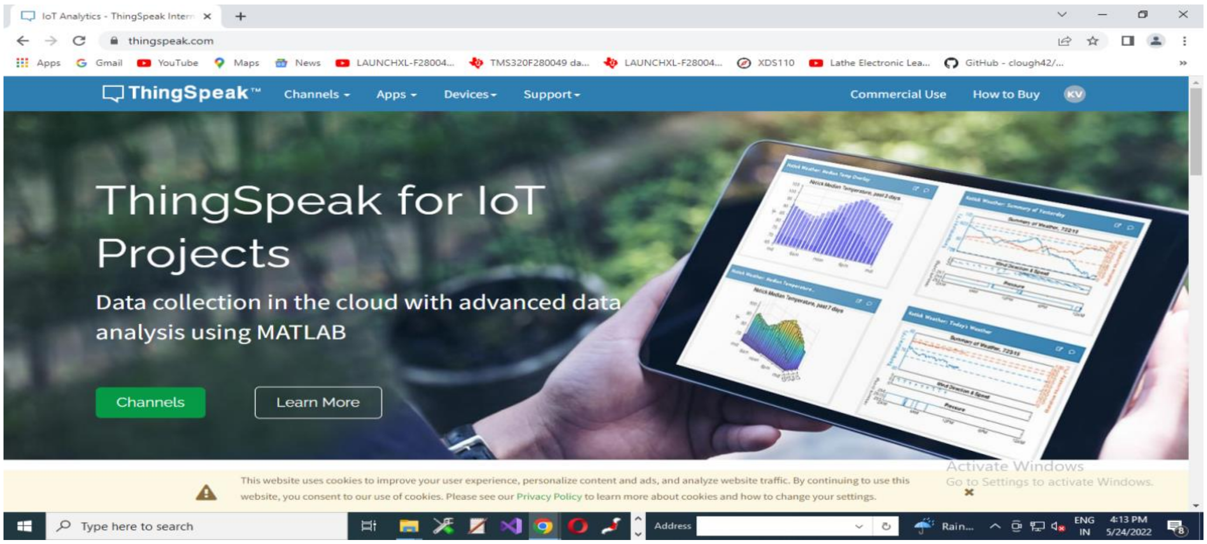
**PROGRAM :**

```
from machine import Pin,PWM
from time import sleep
led=PWM(Pin(2))
led.freq(150)
while True:
    for duty in range(0,65535):
        led.duty_u16(duty)
        sleep(0.0001)

    for duty in range(65535,0,-1):
        led.duty_u16(duty)
        sleep(0.0001)
```

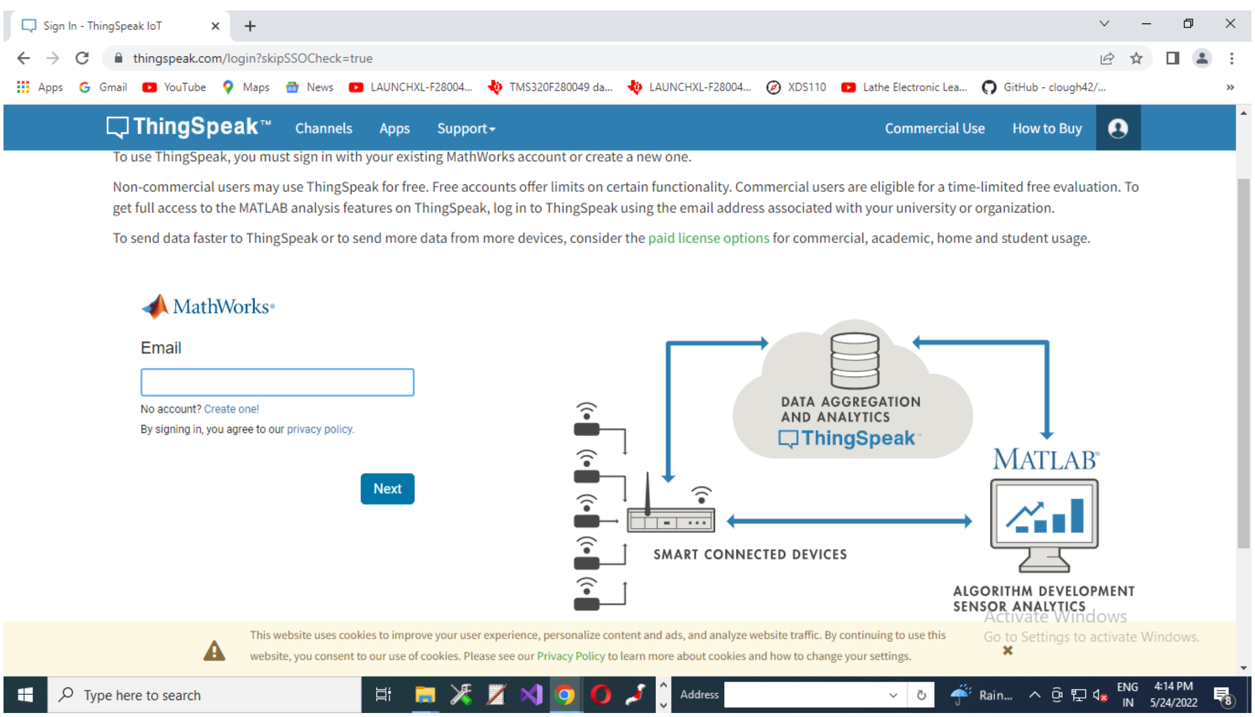
# 7. PROCEDURE IN (THINK SPEAK CLOUD)

**STEP 1:** GO TO THINK SPEAK CLOUD WEBSITE.

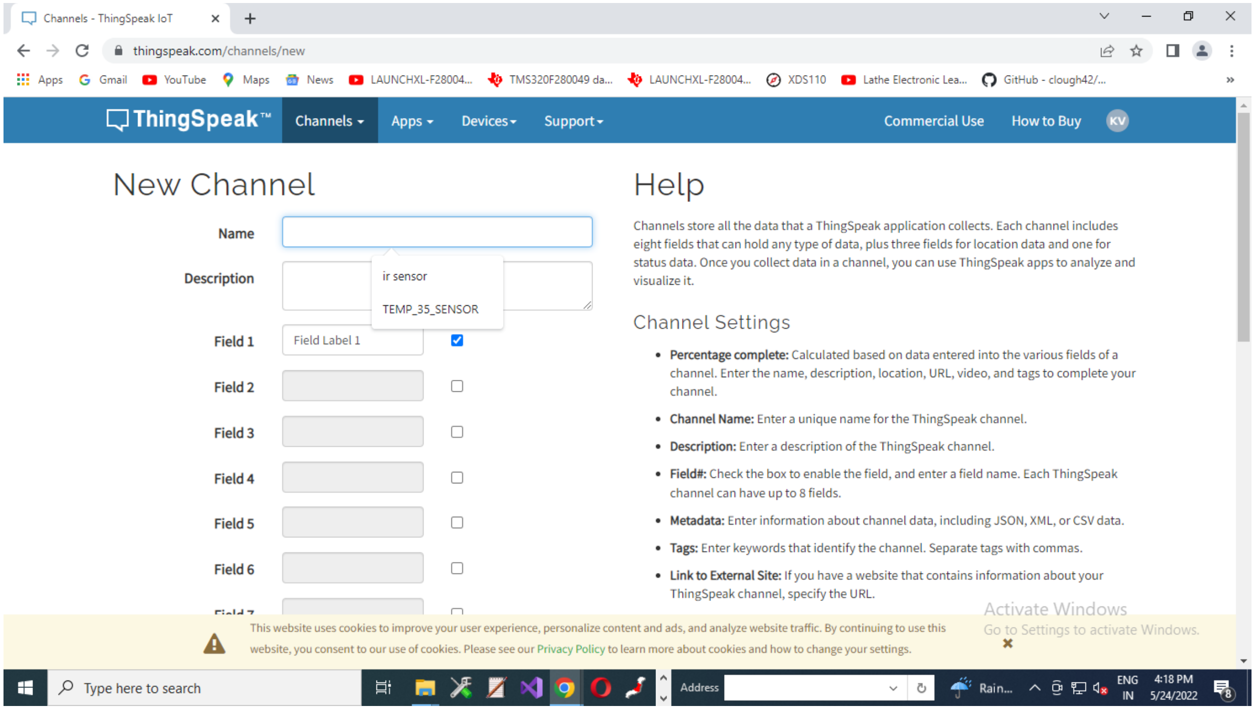
**STEP 2:** login the account.



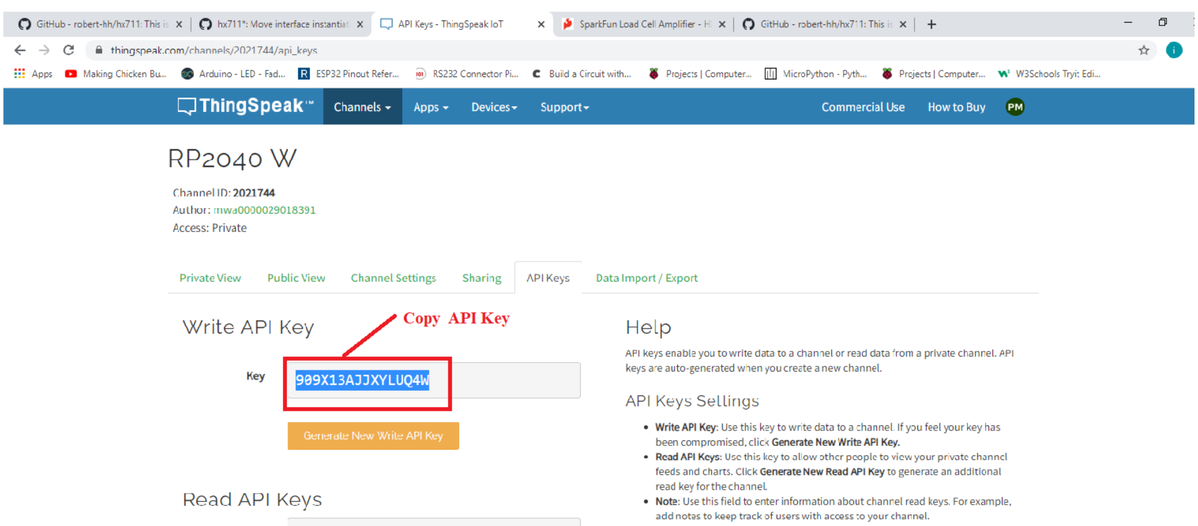

**STEP4:** Enter the email ID.




**STEP 5 :** CREATE NEW CHANNEL ➔ENTER THE NAME ➔CLICK SAVE



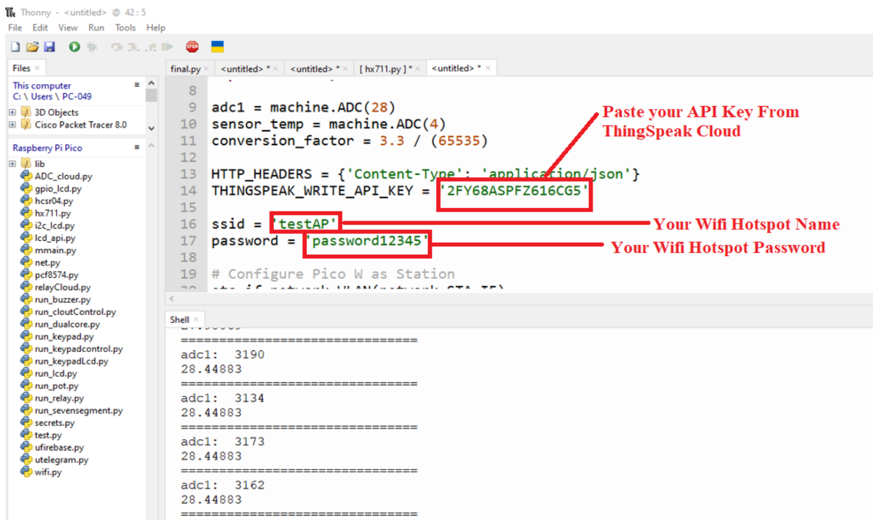

**STEP 6:** CHANNEL ID CREATED,API KEY CREATED.




**STEP 7:CHANNEL ID** →in think cloud speak,**API KEY**→ in think cloud speak Enter in the program.

**STEP 8:** Switch on the wifi hot spot in our smart phone and get the SSID and password.

- ➢ This 4 line program is a header file program included in the main program.
- ➢ Channel ID no and API key from think speak cloud website enter in the header file.




**STEP 9 :** upload the program

**STEP10:** view the sensor value in the graph.

### 7.1. ADC VALUE SEND TO THINGSPEAK CLOUD

**PROGRAM :**

```
from machine import Pin, ADC

import utime

import machine

import urequests

import machine

from machine import Pin

import network, time


adc1 = machine.ADC(28)

sensor_temp = machine.ADC(4)
```

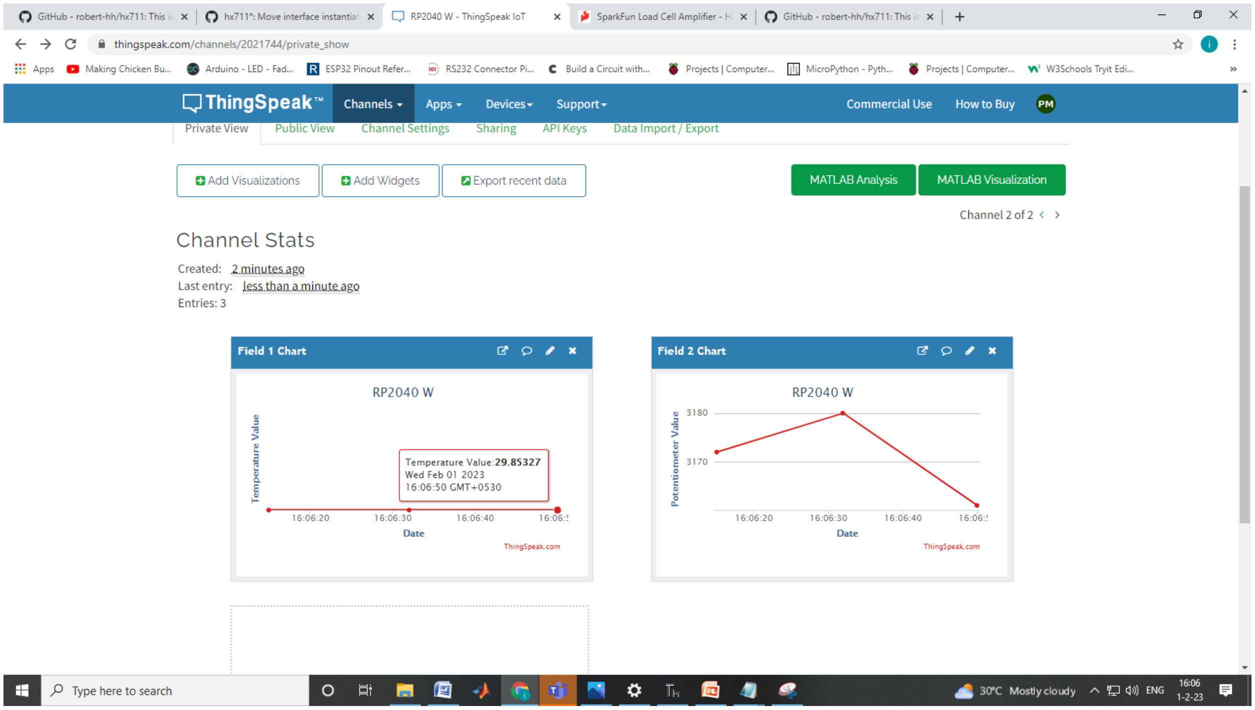
```
conversion_factor = 3.3 / (65535)

HTTP_HEADERS = {'Content-Type': 'application/json'}
THINGSPEAK_WRITE_API_KEY = '909X13AJJXYLUQ4W'

ssid = 'testAP'
password = 'password12345'

# Configure Pico W as Station
sta_if=network.WLAN(network.STA_IF)
sta_if.active(True)

if not sta_if.isconnected():
    print('connecting to network...')
    sta_if.connect(ssid, password)
    while not sta_if.isconnected():
     pass
print('network config:', sta_if.ifconfig())

while True:
    #time.sleep(5)
    val1 =  adc1.read_u16() >> 4
    print("===============================")
    print("adc1: ",val1)
    reading = sensor_temp.read_u16() * conversion_factor
    temperature = 27 - (reading - 0.706)/0.001721
    print(temperature)
    time.sleep(2)
    temp_readings = {'field1':temperature,'field2':val1 }
```

request = urequests.post( 'http://api.thingspeak.com/update?api_key=' + THINGSPEAK_WRITE_API_KEY, json = temp_readings , headers = HTTP_HEADERS )

request.close()

**OUTPUT :**

```
Shell
>>> %Run -c $EDITOR_CONTENT
 network config: ('192.168.43.66', '255.255.255.0', '192.168.43.1', '192.168.43.1')
 ================================
 adc1:  3172
 29.85327
 ================================
 adc1:  3186
 29.85327
 ================================
 adc1:  3194
 29.85327
 ================================
 adc1:  3182
 29.85327
```



# 8. INTRODUCTION TO RTOS

## What is An RTOS?

*"Provide a free product that surpasses the quality and service demanded by users of commercial alternatives"*

Dedicated FreeRTOS developers have been working in close partnership with the world's leading chip companies for more than 15 years to provide you market leading, commercial grade, and completely free high quality RTOS and tools ...but what is an RTOS?

This page starts by defining an operating system, then refines this to define a real time operating system (RTOS), then refines this once more to define a real timer kernel (or real time executive).

See also the FAQ item "why an RTOS" for information on when and why it can be useful to use an RTOS in your embedded systems software design.

**What is a General Purpose Operating System?**

An operating system is a computer program that supports a computer's basic functions, and provides services to other programs (or *applications*) that run on the computer. The applications provide the functionality that the user of the computer wants or needs. The services provided by the operating system make writing the applications faster, simpler, and more maintainable. If you are reading this web page, then you are using a web browser (the application program that provides the functionality you are interested in), which will itself be running in an environment provided by an operating system.

**What is an RTOS?**

Most operating systems appear to allow multiple programs to execute at the same time. This is called multi-tasking. In reality, each processor core can only be running a single thread of execution at any given point in time. A part of the operating system called the scheduler is responsible for deciding which program to run when, and provides the illusion of simultaneous execution by rapidly switching between each program.

The type of an operating system is defined by how the scheduler decides which program to run when. For example, the scheduler used in a multi user operating system (such as Unix) will ensure each user gets a fair amount of the processing time. As another example, the scheduler in a desk top operating system (such as Windows) will try and ensure the computer remains responsive to its user. [Note: FreeRTOS is not a big operating system, nor is it designed to run on a desktop computer class processor, I use these examples purely because they are systems readers will be familiar with]

The scheduler in a Real Time Operating System (RTOS) is designed to provide a predictable (normally described as *deterministic*) execution pattern. This is particularly of interest to embedded systems as embedded systems often have real time requirements. A real time requirements is one that specifies that the embedded system must respond to a certain event within a strictly defined time (the *deadline*). A guarantee to meet real time requirements can only be made if the behaviour of the operating system's scheduler can be predicted (and is therefore deterministic).

Traditional real time schedulers, such as the scheduler used in FreeRTOS, achieve determinism by allowing the user to assign a priority to each thread of execution. The scheduler then uses the priority to know which thread of execution to run next. In FreeRTOS, a thread of execution is called a *task*.

**What is FreeRTOS?**

FreeRTOS is a class of RTOS that is designed to be small enough to run on a microcontroller - although its use is not limited to microcontroller applications.

A microcontroller is a small and resource constrained processor that incorporates, on a single chip, the processor itself, read only memory (ROM or Flash) to hold the program to be executed, and the random access memory (RAM) needed by the programs it executes. Typically the program is executed directly from the read only memory.

Microcontrollers are used in deeply embedded applications (those applications where you never actually see the processors themselves, or the software they are running) that normally have a very specific and dedicated job to do. The size constraints, and dedicated end application nature, rarely warrant the use of a full RTOS implementation - or indeed make the use of a full RTOS implementation possible. FreeRTOS therefore provides the core real time scheduling functionality, inter-task communication, timing and synchronisation primitives only. This means it is more accurately described as a real time kernel, or real time executive. Additional functionality, such as a command console interface, or networking stacks, can then be included with add-on components.

# Vi Microsystems Pvt.Ltd., Chennai-96
## INTERNSHIP TRAINING PROGRAM

**College Name:** AMET University, Chennai - 603113.
**Duration:** 1 Week ( 11.03.2024 to 16.03.2024 ) Morning

| S. No | Students Name / Date | 11.03.2024 | 12.03.2024 | 13.03.2024 | 14.03.2024 | 15.03.2024 | 16.03.2024 |
|---|---|---|---|---|---|---|---|
| 1. | DIVYANSHU OJHA | | | | | | |
| 2. | VARUN CHITRANSH | | | | | | |
| 3. | ARYAN SINGH BORA | | | | | | |
| 4. | FARHAN JAWED ANSARI | Farhan | Farhan | Farhan | AB | Farhan | |
| 5. | ANGSUKH BAITALIK | | | | AB | | |
| 6. | JIGNESH VIJAY KATE | JKate | JKate | JKate | JKate | JKate | JKate |
| 7. | AKASH S | | | | | | |
| 8. | ADHITHYAN P | | | | | | |
| 9. | DEEKSHATH V | | | | | | |
| 10. | GIDEON MATHEW - A | | | | | | |
| 11. | HEMACHANDRAN R | | | | | | |
| 12. | KIRUBAKARAN B | B.K | B.K | B.K | B.K | | B.K |
| 13. | ARAVIND KUMAR A | A.Arvind | A.Arvind | A.Arvind | A.Arvind | A.Arvind | A.Arvind |
| 14. | ARISH BALA D | D. | D. | D. | D. | D. | D. |
| 15. | ADHAVAN M | M. | M. | M. | M. | M. | M. |

B.K

| S. No | Students Name / Date | | | | | | |
|---|---|---|---|---|---|---|---|
| 16. | JAGADEESH.P | | | | | | |
| 17. | DHINESH S | | | | | | |
| 18. | KALAI.G | | | | | | |
| 19. | VARSHIINI S | | | | | | |
| 20. | SAKSHI D | | | | | | |
| 21. | KANNIKA KESAR .R | | | | | | |
| 22. | KARUNYA V | | | | | | |
| 23. | HEMANTH KUMAR G | | | | | | |
| 24. | JAGANADHA GANDHI~~I~~E | | | | | | |
| 25. | ADHITHYAN A | | | | | | |
| 26. | VIGNESH V | | | | | | |
| 27. | LAVITHA V S | | | | | | |
| 28. | ARYAKI PRASHANT NARKAR | | | | | | |
| 29. | ARUN PANDI V | | | | | | |
| 30. | GOVINDA RAJA PERUMAL R | | | | | | |
| 31. | KALIMUTHU S | | | | | | |
| 32. | ARSLAN ~~MALICK~~ MALLICK | | | | | | |

| S. No | Students Name / Date | | | | | | |
|---|---|---|---|---|---|---|---|
| 33. | ISHAAQ AHAMED KHAN | Ishaaq | Ishaaq | Ishaaq | Ishaaq | Ishaaq | |
| 34. | AJIN JAMES | Ajin | Ajin | Ajin | Ajin | Ajin | Ajin |
| 35. | JOBIN GEORGE | Jobingg | Jobingg. | Jobingg.. | Jobingg. | Jobingg. | Jobingg |
| 36. | ABHINAV K | Abhinav | Abhinav | Abhinav | Abhinav | Abhinav | Abhinav |
| 37. | HARISH.M | MR. Harish | MR. Harish | MR. Harish | MR. Harish | MR. Harish | MR. Haris |
| 38. | ANTHONY ~~M~~ BHARATHI . M | M. Anthony | M. Anthony | M. Anthony | M. Anthony | M. Anthony | |
| 39. | V.P ASHWA~~[illegible]~~NTH (V.P. ASHWANTH) | [illegible] | [illegible] | Ashwy | [illegible] | [illegible] | |
| 40. | DHINESH . B | Dhinesh . B | Dhinesh | B Dhinesh | B Dhinesh | B Dhinesh . B | |
| 41. | ARJUN VIJAY | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

# Vi Microsystems Pvt.Ltd., Chennai-96
## INTERNSHIP TRAINING PROGRAM

**College Name:** AMET University, Chennai - 603113.
**Duration:** 1 Week ( 11.03.2024 to 16.03.2024 ) Afternoon

| S. No | Students Name / Date | 11.03.2024 | 12.03.2024 | 13.03.2024 | 14.03.2024 | 15.03.2024 | 16.03.2024 |
|---|---|---|---|---|---|---|---|
| 1. | DIVYANSHU OJHA | | | | | | |
| 2. | VARUN CHITRANSH | | | | | | |
| 3. | ARYAN SINGH BORA | | | | | | |
| 4. | FARHAN JAWED ANSARI | | | | | | |
| 5. | ANGSUKH BAITALIK | | | | | | |
| 6. | JIGNESH VIJAY KATE | | | | | | |
| 7. | AKASH S | | | | | | |
| 8. | ADHITHYAN P | | | | | | |
| 9. | DEEKSHATH V | | | | | | |
| 10. | GIDEON MATHEW | | | | | | |
| 11. | HEMACHANDRAN R | | | | | | |
| 12. | KIRUBAKARAN B | | | | | | |
| 13. | ARAVIND KUMAR A | | | | | | |
| 14. | ARISH BALA D | | | | | | |
| 15. | ADHAVAN M | | | | | | |

| S. No | Students Name / Date | | | | | | |
|---|---|---|---|---|---|---|---|
| 16. | JAGADEESH F | | | | | | |
| 17. | DHINESH S | | | | | | |
| 18. | KALAI G | | | | | | |
| 19. | VARSHIINI S | | | | | | |
| 20. | SAKSHI D | | | | | | |
| 21. | KANNIKA KESAR . R | | | | | | |
| 22. | KARUNYA V | Karunya | Karunya | Karunya | Karunya | Karunya | Karunya |
| 23. | HEMANTH KUMAR G | Hemanth | Hemanth | Hemanth | Hemanth | Hemanth | |
| 24. | JAGANADHA GANDHI E | | | | | | |
| 25. | ADHITHYAN A | | | | | | |
| 26. | VIGNESH V | | | | | | |
| 27. | LAVITHA V S | | | | | | |
| 28. | ARYAKI PRASHANT NARKAR | | | | | | |
| 29. | ARUN PANDI V | Arun Pandi | Arun Pandi | Arunpandi | Arunpandi | Arunpandi | Arun Pandi |
| 30. | GOVINDA RAJA PERUMAL R | Perumal | Perumal | Perumal | Perumal | Perumal | Perumal |
| 31. | KALIMUTHU S | | | | | | |
| 32. | ARSLAN ~~MALICK~~ MALLICK | Arslan Mallick | Arslan Mallick | Arslan Mallick | Arslan Mallick | Arslan Mallick | |

| S. No | Students Name / Date | | | | | | |
|---|---|---|---|---|---|---|---|
| 33. | ISHAAQ AHAMED KHAN | Ishaaq | Ishaaq | ISlaq | Ishaaq | Ishaaq | |
| 34. | AJIN JAMES | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 35. | JOBIN GEORGE | [illegible] | Jobin.. | Jobin.. | Jobin.. | Jobin.. | Jobin |
| 36. | ABHINAV K | Abhinav | Abhinav | Abhinav | Abhinav | Abhinav | Abhinav |
| 37. | HARISH M | MR [illegible] | MR. [illegible] | MR. [illegible] | MR. [illegible] | MR. [illegible] | [illegible] |
| 38. | ANTHONY M | M. Anthony | M. Anthony | M. Anthony | M. Anthony | M. Anthony | |

39 DHINESH . B — Dhinesh . B, Dhinesh . B, Dhinesh . B, Dhinesh . B, Dhinesh . B

40 V.P Ashwanth

50 Arjun Vijay

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

**COLLEGE NAME:** AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

**YEAR** : III
**SESSION:** AN

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 1. | SAMMAR | | | | | | | |
| 2. | SIDHARTH ROY | | | | | | | |
| 3. | KAVIN R K | | | | | | | |
| 4. | MURALIDHARAN R | | | | | | | |
| 5. | HARISH K | | | | | | | |
| 6. | SHANMUGARAJ M | | | | | | | |
| 7. | LIKITH J | | | | | | | |
| 8. | SHYAM | | | | | | | |
| 9. | RITHIK VIGNESH M.M | | | | | | | |
| 10. | JOE JAMES SERAPIN | | | | | | | |
| 11. | THARUN | | | | | | | |
| 12. | DHYNAESHWAR | | | | | | | |
| 13. | SEBOVIN | | | | | | | |
| 14. | ABILASH | | | | | | | |
| 15. | ATHUL KRISHNA | | | | | | | |

S Saranth

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 16. | JASUVA JENIS JOVIN | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 17. | ~~SHAHANAWAZ KHAN~~ SHAHNAWAZ KHAN | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 18. | MOHAMMED KAIF | [illegible] | [illegible] | [illegible] | [illegible] | M. Kaif | Mohammed | |
| 19. | SURYAKANTH | R. Suryakanth | R. Suryakanth | R. Suryakanth | R. Suryakanth | R. Suryakanth | R. Suryakanth | |
| 20. | | | | | | | | |
| 21. | | | | | | | | |
| 22. | | | | | | | | |
| 23. | | | | | | | | |

S [illegible]

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

**COLLEGE NAME:** AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

**YEAR** : II
**SESSION:** AN

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 1. | SAMAN SINGH | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 2. | NAVANEETHAN P | P. Nav[illegible] | P. Nav[illegible] | [illegible] | P. Nav[illegible] | P. Nav[illegible] | | |
| 3. | RANJITH M | M. Ranjith | [illegible] | [illegible] | M. [illegible] | [illegible] | | |
| 4. | NAVEEN KUMAR | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | |
| 5. | SHAMER BASHA C | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Shamer | |
| 6. | SIVA KUMAR I | I. [illegible] | [illegible] | I. [illegible] | I. [illegible] | I. [illegible] | Siva. | |
| 7. | ABISHEK A | A. Abishek | [illegible] | A. [illegible] | A. [illegible] | A. Abishek | Abishek | |
| 8. | SANJEEVI KUMAR S | Sanjeevi Kumar | [illegible] | Sanjeevi [illegible] | [illegible] | [illegible] | Sanjeevi K | |
| 9. | BRUHAN MALIKDEEN A | A. Bruhan Malikdeen | A. Bruhan Malikdeen | A. [illegible] | A. Bruhan Malikdeen | Bruhan A. Malikdeen | A. Bruhan Malikdeen. | |
| 10. | MOHAN RAJ M | M. Mohan[illegible] | [illegible] | M. Mohan[illegible] | M. Mohan[illegible] | M. Mohan[illegible] | M. Mohan [illegible] | |
| 11. | GOKUL PANDIT R | R. Gokul | R. Gokul | R. [illegible] | R. [illegible] | R. [illegible] | | |
| 12. | PRAVEEN C | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | |
| 13. | POSWAL KHUSHI S | Khushi | Khushi | [illegible] | Khushi | | | |
| ~~14.~~ | ~~RANJITH~~ | M. Ranjith | [illegible] | [illegible] | | | | |
| 15. | MARIA ROSHAN M | M. M. Roshan | M. M. Roshan | M. M. Roshan | M. M. Roshan | M. [illegible] | M. M. Roshan | |

S. [illegible]

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 16. | VISHWAS S | V. Vish | S.Vish | S. Vish | S. Vish | S. Vish | S. Vi | |
| 17. | HARI PRASATH S | S.Hari Pra | S.Hari Pra | S. Hari P | S. Hari Prasath | S. Hari Pra | S.Hari Pra | |
| 18. | SOOROJ | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | |
| 19. | ARAVINDAN S | S. Ara | S. Aravin | S. Ara | S. Aravindan | S. Aravindan | | |
| 20. | AMARNATH G | G. Amarnath | G. Amarnath | G. Amarnath | G. Amarnath | G. Amarnath | | |
| 21. | GANESAN M | Ganesh | Ganesh | Ganesh | Ganesh. | Ganesh | | |
| 22. | SELVAKUMAR S | S. Selva | S. Selva | S. Sel | S. Selva | S. Selva | S. Sel | |
| 23. | ASHWIN R | Ashwin R. | Ashwin R | Ashwin R | Ashwin R | Ashwin | | |
| 24. | THARUN KUMAR | Tharun | Tharun | [illegible] | Tharun | Tharunth | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

S. [illegible]

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

**COLLEGE NAME**: AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

**YEAR** : II
**SESSION**: FN

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 1. | SAMAN SINGH | [illegible] | [illegible] | od | [illegible] | [illegible] | [illegible] | |
| 2. | NAVANEETHAN P | P. Navaneethan | P. Navaneethan | P. Navaneethan | P. Navaneethan | P. Nav | | |
| 3. | RANJITH M | M. Ranjith | Ranjith | M. Ranjith | M. Ranjith | M. Ranjith | | |
| 4. | NAVEEN KUMAR | Naveen | Naveen | Naveen | Naveen | | | |
| 5. | SHAMER BASHA C | [illegible] | Shamer | [illegible] | [illegible] | S. [illegible] | Shamer. | |
| 6. | SIVA KUMAR I | I. Siva | I. Siva | I. Siva | I. Siva | I. [illegible] | Sivakumar | |
| 7. | ABISHEK A | A. Abishek | A. Abishek | A Abishek | A. Abishek | A. Abishek. | Abishek | |
| 8. | SANJEEVI KUMAR S | Sanjeevikumar | Sanjeevikumar | sanjeevikumar. | Sanjeevikumar | Sanjeevikumar. | Sanjeevi | |
| 9. | BRUHAN MALIKDEEN A | Bruhan Malikdeen | A. Bruhanmalik deen | A. Bruhanmalik deen | A. Bruhanmalik deen | A. Bruhanmalik deen | A. Bruhan Malikdeen | |
| 10. | MOHAN RAJ M | M. Mohanraj | M. Mohan Raj | M. Mohanraj | M. Mohanraj | M. Mohan | M Mohan | |
| 11. | GOKUH PANDIT R | R. Gokul | R. Gokul Pandit | P. Gokul Pandit | P. Gokul Pandit | Gokul Pandit | | |
| 12. | PRAVEEN C | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | |
| 13. | POSWAL KHUSHI S | khushi | khushi | od | khushi. | khushi. | | |
| ~~14.~~ | ~~RANJITH~~ | M. Ranjith. | M. Ranjith | M. Ranjith | | | | |
| 15. | MARIA ROSHAN M | M. M. Roshan | M. M. Roshan | M. M. Roshan | M. M. Roshan | M. M. Roshan | M. M. Roshan | |

S. [illegible]

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 16. | VISHWAS S | S. Vish | S.Vishwa | S.Vishwa | S.Vishwa | S. Vish | S. Vish | |
| 17. | HARI PRASATH S | S. Hari [illegible] | S. Hari [illegible] | S. Hari [illegible] | S. Hari [illegible] | S. Hari [illegible] | [illegible] | |
| 18. | SOORQJ | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | |
| 19. | ARAVINDAN S | S. Aravind | S. Aravind | S. Aravind | S. Ara[illegible] | S. [illegible] | | |
| 20. | AMARNATH G | G. Amarnath | G. Amarnath | G. Amarnath | G. Amarnath | | G. Amrth | |
| 21. | GANESAN M | Ganesh | Ganesh | Ganesh. M | Ganesh | Ganesh | | |
| 22. | SELVAKUMAR S | S. Selva | S. Selva | S. Selva. | S. Selva | S. Selva | S. Selva | |
| 23. | ASHWIN R | Ashwin. R | Ashwin. R | Ashwin. R | Ashwin. R | Ashwin R | | |
| 24. | THARUN KUMAR | [illegible] | Tharun | [illegible] | Tharun | Tharun | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

S [illegible]

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

**COLLEGE NAME:** AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

**YEAR** : III
**SESSION:** FN

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 1. | SAMMAR | Sammar | Sammar | Sammar | Sammar | Sammar | Sammar | |
| 2. | SIDHARTH ROY | Sidharth | Sidharth | [illegible] | S.Roy | Sidhartha | Sidharth | |
| 3. | KAVIN R K | Kavin. | Kavin. | Kavin. | Kavin. | [illegible] | Kavin | |
| 4. | MURALIDHARAN R | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 5. | HARISH K | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 6. | SHANMUGARAJ M | M.Shanmugaraj | M.Shanmugaraj | M. [illegible] | M. [illegible] | [illegible] | M. [illegible] | |
| 7. | LIKITH J | J. [illegible] | J. [illegible] | J. [illegible] | J. [illegible] | [illegible] | J. [illegible] | |
| 8. | SHYAM | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 9. | RITHIK VIGNESH M·M | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 10. | JOE JAMES SERAPIN | P. [illegible] | P. [illegible] | P. [illegible] | P. [illegible] | [illegible] | P. [illegible] | |
| 11. | THARUN | Tha | | Tha | [illegible] | [illegible] | [illegible] | |
| 12. | DHYNAESHWAR | [illegible] | | [illegible] | [illegible] | Dhy | Dhy | |
| 13. | SEBOVIN | Sebovin | Sebovin | Sebovin | Sebovin | Sebovin. | Sebovin | |
| 14. | ABILASH | R. [illegible] | R. [illegible] | R. [illegible] | R. [illegible] | R. [illegible] | [illegible] | [illegible] |
| 15. | ATHUL KRISHNA I.V | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

S [illegible]

| S. No | Students Name / Date | 22.03.24 | 23.03.24 | 25.03.24 | 26.03.24 | 27.03.24 | 28.03.24 | 29.03.24 |
|---|---|---|---|---|---|---|---|---|
| 16. | JASUVA JENIS JOVIN | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 17. | SHAHANAWAZ KHAN × SHAHNAWAZ KHAN | Shams | S.Kh | S.Kh | S.Kh | S.Kh | [illegible] | |
| 18. | MOHAMMED KAIF | [illegible] | [illegible] | [illegible] | M.Kaif | [illegible] | [illegible] | |
| 19. | SURYAKANTH | R.Suryakanth | R.Suryakanth | R.Suryakanth | R.Suryakanth | [illegible] | R.Suryakanth | |
| 20. | | | | | | | | |
| 21. | | | | | | | | |
| 22. | | | | | | | | |
| 23. | | | | | | | | |

S. [illegible]

**Vi Microsystems Pvt.Ltd, Chennai-96**
**INTERNSHIP TRAINING PROGRAM**

**COLLEGE NAME:** AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

**YEAR** : I
**SESSION:** AN

| S. No | Students Name / Date | 01.04.24 | 02.04.24 | 03.04.24 | 04.04.24 | 05.04.24 | 06.04.24 | 07.04.24 |
|---|---|---|---|---|---|---|---|---|
| 1. | RANJITH C | C.Ranjith | C.Ranjith | C.Ranjith | C.Ranjith | C.Ranjith | C.Ranjith | |
| 2. | SRI SABARI BALASAKTHI S | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 3. | MUTHU SURESH S | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 4. | YUVARAJ T | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 5. | MATHI BALA K | K.Mathi Bala | K.Mathi Bala | K.Mathimala | K.Mathibala | K.Mathibala | K.Mathi Bala | |
| 6. | RAKESH R S | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 7. | VASANTH C | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 8. | MASILAMANI V | V.masilamani | V.masilamani | V.masilamani | V.masilamani | V.masilamani | | |
| 9. | LOKESHWARAN K | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | |
| 10. | SAHAYA JABIN BIJO | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 11. | HARRIS CHRISTOPHER | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 12. | MOKESH T | T.Moff | T.Moff | T.Moff | T.Moff | T.Moff | T.Moff | |
| 13. | RAMANA A | A.Ramana | A.Ramana | A.Raman | A.Ramana | A.Ramana | | |
| 14. | MATHEW FRANKLIN P D | D.Mathew | D.Mathew | D.Mathew | D.Mathew | D.Mathew | D.Mathew | |
| 15. | KIRUBANATHAN J | J.Kirubanathan | J.Kirub | J.Kirub | J.Kirub | J.Kirub | J.Kirubanathan | |

| S. No | Students Name / Date | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 16. | VEERA GANESH | | | | | | | |
| 17. | MANOJE KUMAR | C. Manoje | C. Manoje | C. Manoje | C. Manoje | C. Manoje | C. Manoje | |
| 18. | PRASANA A V | A.V. Prasanna | A. V. Prasanna | A. V. Prasanna | | | | |
| 19. | LOGESH G | G. | G. | G. | G. | G. | G. | |
| 20. | JEEVIDOSS J | Jeevidoss J | J. Jeevidoss J | J. Jeevidoss | | J. Jeevidoss | J. Jeevidoss | |
| 21. | VARUN KUMAR M | Varun | Varun | Varun | Varun | Varun | Varun | |
| 22. | NAVIN E | | | | | | | |
| 23. | MONISH P | Monish P | Monish P | Monish P | Monish. P | Monish. P | Monish. P | |
| 24. | SRIMAN S | Sriman | | | | | | |
| 25 | A. VIMAL KANNAN | | | | | | | |
| 26. | S. RAVIN KUMAR | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

YEAR : I
SESSION: AN

**COLLEGE NAME:** AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

| S. No | Students Name / Date | 01.04.24 | 02.04.24 | 03.04.24 | 04.04.24 | 05.04.24 | 06.04.24 | 07.04.24 |
|---|---|---|---|---|---|---|---|---|
| 25. | MOHAMMED SHAJI S | | | | | | | |
| 26. | ROHAN | | | | | | | |
| 27. | ABHISHEK SATHEESH | | | | | | | |
| 28. | PRANAV D | | | | | | | |
| 29. | RIYAZ KHAN N | | | | | | | |
| 30. | MUHAMMED SHARIF | | | | | | | |
| 31. | ADHITHYAN K S | | | | | | | |
| 32. | SULTHAN MOHAMMED HAREERI | | | | | | | |
| 33. | MANTHAN VHADGIRE | | | | | | | |
| 34. | ZAID GHANI | | | | | | | |
| 35. | TEJAS SURYE | | | | | | | |
| 36. | NIKHIL JETHWA | | | | | | | |
| 37. | PRANAV PRINCE | | | | | | | |
| 38. | ANGITH B | | | | | | | |
| 39. | Shan Joseph | | | | | | | |

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

**YEAR** : I
**SESSION:** FN

**COLLEGE NAME:** AMET UNIVERSITY, CHENNAI - 603113.
**DURATION:** 1 WEEK.

| S. No | Students Name / Date | 01.04.24 | 02.04.24 | 03.04.24 | 04.04.24 | 05.04.24 | 06.04.24 | 07.04.24 |
|---|---|---|---|---|---|---|---|---|
| 1. | RANJITH C | C.Ranjith | C.Ranjith | C.Ranjith | C.Ranjith | C.Ranjith | C.Ranjith | |
| 2. | SRI SABARI BALASAKTHI S | [illegible] | Sabari | Sabari | Sabari | Sabari | Sabari | |
| 3. | MUTHU SURESH S | Suresh | Suresh | Suresh | [illegible] | [illegible] | [illegible] | |
| 4. | YUVARAJ T | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 5. | MATHI BALA K | K.Mathi Bala | K.Mathi Bala | K.Mathi Bala | K.Mathi Bala | K.Mathi Bala | K.Mathi Bala | |
| 6. | RAKESH ~~K~~ S | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 7. | VASANTH C | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 8. | MASILAMANI V | V.Masilamani | V.Masilamani | V.Masilamani | V.Masilamani | V.Masilamani | V.Masilamani | |
| 9. | LOKESHWARAN K | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 10. | SAHAYA JABIN BIJO | Sahaya | Sahaya | Sahaya | [redacted] | Sahaya | [illegible] | |
| 11. | HARRIS CHRISTOPHER | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 12. | MOKESH T | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 13. | RAMANA A | A.Raman | A.Raman | A.Raman | A.Raman | A.Raman | A.Raman | |
| 14. | MATHEW FRANKLIN ~~P~~ D | D.Mathew | D.Mathew | [illegible] | D.Mathew | D.Mathew | D.Mathew | |
| 15. | KIRUBANATHAN J | [illegible] | [illegible] | J.[illegible] | [redacted] | [illegible] | J.[illegible] | |

| S. No | Students Name / Date | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 16. | VEERA GANESH | [illegible] | [illegible] | [illegible] | [illegible] | Veer | [illegible] | |
| 17. | MANOJE KUMAR | Manoje Ku | Manoje | Manoje | Manoje | C. Manoje | C. Manoje | |
| 18. | PRASANA A V | A.V Prasanna | A.V Prasanna | A.V. Prasanna | [illegible] | | | |
| 19. | LOGESH G | G.[illegible] | G.[illegible] | G.[illegible] | G.[illegible] | G.[illegible] | G.[illegible] | |
| 20. | JEEVIDOSS J | Jeevidoss J. | Jeevidoss J | J. Jeevidoss. | [illegible] | J. Jeevidoss | J. Jeevidoss | |
| 21. | VARUN KUMAR M | Varun | Varun | Varun | Varun | Varun | Varun | |
| 22. | NAVIN E | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 23. | MONISH P | Monish.P | Monish.P | Monish.P | Monish.P | Monish.P | Monish.P | |
| 24. | SRIMAN.S | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 25. | A.VIMAL KANNAN | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| 26. | S.RAVINKUMAR | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

# Vi Microsystems Pvt.Ltd, Chennai-96
## INTERNSHIP TRAINING PROGRAM

**COLLEGE NAME**: AMET UNIVERSITY, CHENNAI - 603113.
**DURATION**: 1 WEEK.

**YEAR** : I
**SESSION**: FN

| S. No | Students Name / Date | 01.04.24 | 02.04.24 | 03.04.24 | 04.04.24 | 05.04.24 | 06.04.24 | 07.04.24 |
|---|---|---|---|---|---|---|---|---|
| 25. | MOHAMMED SHAJI S | | | | | | | |
| 26. | ROHAN | | | | | | | |
| 27. | ABHISHEK SATHEESH | | | | | | | |
| 28. | PRANAV D | | | | | | | |
| 29. | RIYAZ KHAN N | | | | | | | |
| 30. | MUHAMMED SHARIF | | | | | | | |
| 31. | ADHITHYAN K S | | | | | | | |
| 32. | SULTHAN MOHAMMED HAREERI | | | | | | | |
| 33. | MANTHAN VHADGIRE | | | | | | | |
| 34. | ZAID GHANI | | | | | | | |
| 35. | TEJAS SURYE | | | | | | | |
| 36. | NIKHIL JETHWA | | | | | | | |
| 37. | PRANAV PRINCE | | | | | | | |
| 38. | ANGITH B | | | | | | | |
| 39. | Shan Joseph | | | | | | | |

# Vi Microsystems Pvt. Ltd.,

**Plot No: 75, Electronics Estate,**
**Perungudi,**
**Chennai - 600 096.**

## CERTIFICATE OF COMPLETION

**This is to certify that Mr/Ms** RAKESH S [AEC23056] **Studying** II$^{nd}$ Year, B.E - Department of ECE [Electrical and Computer Engineering] - AMET University **has successfully completed his/her Internship training programme on** Embedded System with IoT **in our company from** 24 June 2024 **to** 06 July 2024 **We appreciate his/her interest and the efforts taken to do this training programme. During the period his/her conduct and the performance is satisfactory.**

***We wish success in all future endeavors.***

Vi Microsystems Private Limited
Plot No.75, Electronics Estate, Perungudi, Pin : 600 096
Chennai

V. D. B. [signature]

Manager

Vi Microsystems Pvt Ltd., Chennai - 96.

# Vi Microsystems Pvt. Ltd.,

Plot No: 75, Electronics Estate,
Perungudi,
Chennai - 600 096.

## CERTIFICATE OF COMPLETION

This is to certify that Mr./~~Ms.~~ P. DHYANESHWAR Register No: AEE21018L studying in the III EEE-M year of the Department of EEE at AMET University - Chennai, has successfully completed a 35 Hours Value added training program on "EMBEDDED SYSTEM WITH IoT" at our company From 22.03.2024 To 28.03.2024

We appreciate their interest and the efforts taken to complete this training program. Throughout the period, their conduct and performance were satisfactory.

We extend our best wishes for their future endeavors.

Vi Microsystems Pvt. Ltd. Chennai

J. Anandhan

Manager

Vi Microsystems Pvt Ltd., Chennai - 96.