

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



Inward File Co. 3095 FROM THE OFFICE OF HOD-EEE

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

То

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,

Dr.V.Sridevi Professor and HoD

Submitted for your approval for P. New part of RCRO Holl-tisk procern In 2 2



6





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 <u>COURSE OBJECTIVE:</u>

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:

1

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

3

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 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_enIN1029I N1029&cs=0&q=Raspberry+Pi+Pico+Essentials:+Progr am,+Build,+and+Master+Over+50+Projects+with+Micro Python+and+the+RP2040+Microprocessor+Dogan+Ibra him&stick=H4sIAAAAAAAAAAONgVeLVT9c3NCzJMiusS C4uM-IpKjAyMDFQSMrPzy4-xQiRLTA0KkgzSc-D8WGq4fy84oLyPAuTR4yLmLgFXv64Jyw1g2nSmpPX GCcycQn45OcXp-ZUBqXmJJakpoTkCxlysbnmIWSWVAoJSvFzQYwojDfP LUsyyzUSgQpkm2UnZ6RYFAjMfzCNUSiUizs4tSQk3zc

JTOtUshNyIWL0zc1Nym1qNg_TUiZi8s5PycnNbkkMz9

PSFRKmEtQPxkuoA_2ixJfEKdrr5BUkl8svEBRiYrJg0mpUKjuF2Xpp1jixDc1_7_f6hkklO UhpYgF5tLfm5iZp4gU_lle7kF7-21hLk4Qhlr8vPycysFpdjuFm37f8JeSZHTpp-7lXD-W3tBvVpGhdkzxA5IMCswaDAYXr7G_oAt_td-LQagRU37Vhxi4-BgFGAwYuJgqGLgWcTaxhiUWFwAdHxRpUJAJhAl5yu 4FhenAgMIMafYSiGgKD-9KDFXR8GpNDMnRUchMS9FwTexuCS1SMG_DEiYG oCUZAG9V6xQnImSoeCbmVyUH1BZkpGfB1ZckpGqE BQAjkiwVEFRfnJqcXFkYJLfnpinoJnUIFiRmbuBDZGABx6pZsCAgAA&sa=X&v ed=2ahUKEwjamLKDsNH8AhUx5HMBHeKbDDQQ7fAI

egQIABAU

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:<u>rnd@vimicrosystems.com</u>W eb:<u>www.vimicrosystem.com</u>01/14/ 05/10



6.3	LCD DISPLAY	34
6.4	INTERFACING LM 35 WITH LCD	35
6.5	SEVEN SEGMENT DISPLAY	36
6.6	POT WITH 7 SEGMENT INTERFACING	40
6.7	ULTRASONIC SENSOR	42
6.8	BUZZER	45
6.9	RELAY WITH SERIAL	45
6.10	SWITH INCREMENT WITH LED	46
6.11	TEMPERATURE SENSOR	48
6.12	ADC	49
6.13	PWM	50
7	PROCEDURE IN THINK SPEAK	51



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 realtime 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 (<u>IoT</u>) 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 (<u>DSPs</u>). 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



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 <u>page</u> 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.

Getting started with MicroPython



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

Tione	Share view		Trans.		1 Million I
to Quick Copy P access	Cut Copy path Paste shortcut	Move Copy to * Copy	New item ▼ 1 Easy access ▼ New folder	Properties	Select all
Clipt	oard	Organize	New	Open	Select
> • 🛧 🖊	> This PC > Download	5			
	Name		Date modified	Туре	Size
📌 Quick access	C m2 mice 20	210205-unstable-v1.14-8-g1f800cac	3.uf2 2/25/2021 4:05 AM	UE2 File	486 KB
	IDC-DICO-20				
This PC				ales of sources	
This PC		J			
This PC 3D Objects Desktop			ELECTRONICS	/HUB	
 This PC 3D Objects Desktop Documents 			ELECTRONICS	НИЗ	
This PC This P			ELECTRONICS	(HUB)	
This PC This PC Dobjects Desktop Cournents Downloads Music			ELECTRONICS	HUB	
 This PC 3D Objects Desktop Documents Downloads Music Pictures 			ELECTRONICS	НИВ	
This PC This P		, , , , , , , , , , , , , , , , , , ,	ELECTRONICS	HUB	
This PC This PC This PC This PC This PC This Polyments This Polyments This Polyments This Pictures T		, , , , , , , , , , , , , , , , , , ,	ELECTRONICS	HUB	
 This PC 3D Objects Desktop Documents Downloads Music Pictures Videos Local Disk (C:) New Volume (D:)	, , , , , , , , , , , , , , , , , , ,	ELECTRONICS		

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.

Image: Image	nloads Share View					
Pin to Quick Copy P access	Cut Copy path aste Paste shortcut	Move to * Copy to * Organize	e Rename New fold	New item ▼ ↑ Easy access ▼ New	Properties v Open	Select all Select none Invert selection Select
← → • ↑ ↓	> This PC > Download	s				
Ouick access	Name			Date modified	Туре	Size
This DC	📄 rp2-pico-20	210205-unstable-v1.14-8-	-g1f800cac3.uf2	2/25/2021 4:05 AM	UF2 File	486 KB
3D Objects						
Desktop						
Documents					ANTINES	
🕹 Downloads			E	LECTRONICS	721023	
Music						
E Pictures						
Videos						
🏪 Local Disk (C:)		/				
🛖 New Volume (I	D:)					
🕳 RPI-RP2 (F:) 🗲						
RPI-RP2 (F:)						
Artwork						

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

Downloading Thonny

https://thonny.org/				
Thonny Python IDE for beginners		NB: Wina new iden a warnin it gains n	Download ver Windows • Iows install tity and you g dialog fro nore reputa	sion 3.3.5 for <u>Mac</u> • <u>Linux</u> er is signed with a may receive m Defender until tion.
ELECTRON		Just click "More info" and "Run anyway".		
译。Thonny				- 🗆 X
File Edit View Run Tools Help				
factorial.py ×			Variables	
<pre>def fact(n): if n == 0: return 1 else: return fact(n_1) = n</pre>		^	Name fact	Value ^ <function a<br="" fact="">3</function>
<pre>n = int(input("Enter a natural number</pre>	fact(3)	fact(2)	-	
print("Its factorial is", <u>fact(3)</u>)	fact	fact		
c	if n == 0 retur else: retur	def fact if n else	(n): == 0: return 1 : return fact	(2-3) * n
Shell	<	¢		>
>>> %Debug factorial.py	Local variables	Local var	iables	

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 Thonny - <untitled> @ 1:1 File Edit View Run Tools Help</untitled>			-		×
	0				
<untitled> ×</untitled>	~				
1	2	ELECTRONICS HUB			^ ~
Shell ×					
>>>					^
Python 3.7.9 (bundled) >>>	3			4	
				Pytho	n 3.7.9

- 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.

The Thonny - <untitled> @ 1:1</untitled>	-		×
<untitled> ×</untitled>			
1			^
			~
Shell ×			
>>>			^
Python 3.7.9 (bundled)			
			v
		Pytho	n 3.7.9

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.

🙀 Thonny - <untitled> 🕲 1:1</untitled>	- 🗆 X
File Edit View Run Tools Help	
0 😂 🖬 0 🏘 🗇 3 : 🕒 😄	
<untiled> ×</untiled>	
1	^
Shell ×	
>>>	ELECTRONICS HUB
Python 3.7.9 (bundled)	
>>>	
	The same interpreter which ours Thonny (default
	MicroPython (Raspberry Pi Pico)
	MicroPython (ESP32)
	CircuitPython (generic)
	Configure interpreter

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, <u>UART</u>, 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.

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



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.



6.PROGRAM

6.1 LED

Light Emitting Diodes are the output device connected with Gpio pin GP2 and GP3 of $\ensuremath{\mathsf{RP2040}}$.

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 Off

Led1 On

Led2 On

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:

Chall X
Shell A
94
05
90
96
97
98
99
100
reach maximum
reach maximum

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
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)]</pre>
```

```
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)]</pre>
```

else:

print("reach maximum")

utime.sleep(0.5)

Output:

$\mathbf{Shell} \times$	
10	
12	
6	
6	
11	
17	
11	
5	
7	
13	

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:

$\mathsf{Shell} \times$

>>> %Run -	c \$EDITO	R_CONTENT
Distance:	97.43986	cm
Distance:	98.24742	cm
Distance:	97.47423	cm
Distance:	4.037801	cm
Distance:	4.570446	cm
Distance:	14.0378 0	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:



Shell ×

Distance.	200
Distance:	250
- · ·	050

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:

```
Shell ×
>>> %Run -c $EDITOR_CONTENT
Enter The Condition :1
on
Enter The Condition :0
off
Enter The Condition :
```

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:

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 \times

```
>>> %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.

Sign In - ThingSpeak IoT x +	\vee	-	٥	×
← → C	e	☆		:
🔛 Apps 🔓 Gmail 📭 YouTube 💡 Maps 🍵 News 😰 LAUNCHXL-F28004 👆 TMS320F280049 da 👆 LAUNCHXL-F28004 🧭 XDS110 📭 Lathe Electronic Lea 🎧 GitHub - clough4	!/			»
ThingSpeak™ Channels Apps Support- Commercial Use How to Buy	0			^
io use Thingspeak, you must sign in with your existing Mathworks account or create a new one.				
Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evalu get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.	tion. 1	Го		
To send data faster to ThingSpeak or to send more data from more devices, consider the paid license options for commercial, academic, home and student usage				
MathWorks Email No account? Create anel By signing in, you agree to our privacy policy. Next	}"]]	T		
SENSOR ANALYTICS	lows			
This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our Privacy Policy to learn more about cookies and how to change your settings.	activat	te Win	dows.	Ŧ
📲 🔎 Type here to search 🛛 🗄 🛄 💥 🎽 🖓 🧿 🚺 🎜 dress 🔍 💆 🖑 Rain ^ @ 🖫	d <mark>∞</mark> EN ∥	IG 4∷ N 5/2	14 PM 4/2022	8

STEP 5 : CREATE NEW CHANNEL \rightarrow ENTER THE NAME \rightarrow CLICK SAVE

Channels - ThingSpeak loT x + ← → C a thingspeak.com/channels/ ∷ Apps G Gmail	/new	004 💠 TMS320F28	0049 da	 ー の × ピ ☆ □ ④ : AUNCHXL-F28004 ② XDS110 Lathe Electronic Lea 〇 GitHub - clough42/ »
□ , ThingSpeak™	Channels - Apps -	Devices - Sup	port -	Commercial Use How to Buy 🔯
New Chanr	nel			Help
Name				Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for
Description	ir sensor			status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.
	TEMP_35_S	INSOR	10	Channel Settings
Field 1	Field Label 1			 Percentage complete: Calculated based on data entered into the various fields of a channel. Enter the name description location. URL video and tags to complete your
Field 2				channel.
Field 3				 Channel Name: Enter a unique name for the ThingSpeak channel. Description: Enter a description of the ThingSpeak channel.
Field 4				 Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
Field 5				Metadata: Enter information about channel data, including JSON, XML, or CSV data.
Field 6				Tags: Enter keywords that identify the channel. Separate tags with commas.
		0		Link to External Site: If you have a website that contains information about your ThingSpeak channel, specify the URL. Activate Windows
This w	vebsite uses cookies to improve y	ur user experience, perso	onalize con	tent and ads, and analyze website traffic. By continuing to use this Go to Settings to activate Windows.
P Type here to search		🗶 🗾 刘 🧧		よい Index Bood Coonce and Non to Change your actungs.

STEP 6: CHANNEL ID CREATED, API KEY CREATED.

🔉 GitHub - robert-hh/hx711: This is: 🗙 🕤 hx711*: Move interface instantis: 🗴 🗔 API Keys - ThingSpeak IoT 🗙	👂 SparkFun Load Cell Amplifier - H. 🗙 🔘 GitHub - robert-hhyhx711: This is: 🗙 🕂 🗧 🗗
← → C 🔒 thingspeak.com/channels/2021744/api_keys	x 🔹
III Apps 💶 Making Chicken Bu 🛞 Arduino - LED - Fad 限 ESP32 Pinout Refer 🛞 RS232 Connector Pi C	Build a Circuit with 🐞 Projects Computer 🔟 MicroPython - Pyth 🐞 Projects Computer 💘 W3Schools Tryi: Edi
ThingSpeak Channels - Apps - Devices - S	upport~ Commercial Use How to Buy PM
Channel ID: 2021744 Author: mwa0000029018391 Access: Private Private View Public View Channel Settings Sharing API	Keys Data Import / Export
Write API Key Key 909X13AJJXYLUQ4W Generate New Write API Key	Help AFI keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel. API Keys Sellings • Write API Key: Use this key to write data to a channel. If you feel your key has been compromised. click Generate New Write API Key. • Read API Key: Use this key to allow other popule to view your private channel read key for the channel. • Read API Key the channel.
Read API Keys	 Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

STEP 7:CHANNEL ID \rightarrow in think cloud speak, **API KEY** \rightarrow 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 }

 $\label{eq:request} requests.post('http://api.thingspeak.com/update?api_key=' + THINGSPEAK_WRITE_API_KEY, json = temp_readings , headers = HTTP_HEADERS)$

request.close()

OUTPUT:

S	heli ×
>	>> %Run -c \$EDITOR_CONTENT
	<pre>network config: ('192.168.43.66', '255.255.255.0', '192.168.43.1', '192.168.43.1')</pre>
	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

57

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.

	15	14	13	12	:-	10	.9	o:	.7	6	S	4	ω	N	-		5			
		·	•	•								-					No			College
	ADHAVAN M	ARISH BALA D	ARAVIND KUMAR A	KIRUBAKARAN B	HEMACHANDRAN R	GIDEON MATHEW - A	DEEKSHATH V	ADHITHYAN P	AKASH S	JIGNESH VIJAY KATE	ANGSUKH BAITALIK	FARHAN JAWED ANSARI	ARYAN SINGH BORA	VARUN CHITRANSH	DIVYANSHU OJHA	Date		Students Name	III: 1 WEEK (11.03.2024 to 16.03.202	Name: AMET University, Chennai -
W TO STILL AND S	NO STAND AND THE MAN AND NO DI	Detal Detal Ditter 11. 11. 11. 11. 11. 11. 11. 11. 11. 11	A. Aray	D. I. C. D. D. D. C. M. D. M.	2 min in the start of the start of the second of the secon	OK ROCK MICHA CKAMA, VERTA, 1400.	The state where we will be a set of the set		L'AND AND AND AND	Thenter which the first	Purhan Forhan transform The of the Faultain		annun annun annun annun a	a difference of the of				11.03.2024 12.03.2024 13.03.2024 14.03.2024 15.03.2024	2024) Morning	ai - 6031113.
A A						4		: \	R				lest	¥			3.2024			

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

	32.	31.	30.	29.	28.	27.	26.	25.	24.	23.	22.	21.	20.	19.	18.	17.	16.		S. No
	ARSLAN MALICK MAL LICK	KALIMUTHU S	GOVINDA RAJA PERUMAL R	ARUN PANDI V	ARYAKI PRASHANT NARKAR	LAVITHA V S	VIGNESH V	ADHITHYAN A	JAGANADHA GANDHILE	HEMANTH KUMAR G	KARUNYA V	KANNIKA KESAR , R	SAKSHI D	VARSHIINI S	KALAI,G	DHINESH S	JAGADEESH,P	Data	Students Name
mich mat	And Haller And Haller	The providence of the providen	Marine 1 1980	How of the state	War - OLT.	And the stand		2 Art Argent	Handa Home	Karuman Karumu	12. May Riman.		areas Atom	V. Lui G.Kh	A CONT	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-			
Junton II Jondon Hours	And And And	Perman Perma	Ham Asking 1	Andre, Hutter	Addy Ordy.	Contraction of the second seco	*	The KAZ	Hander Hanner	Kanna Kanna	R.K.M. 12.KM	AL AN	Marine Marine	6 for a Contra		1. C. 1. C.			
and a stranger		Commel Journel	Superdi Agunte	The second is the second secon	QAL DAT.	A Barton		N X1		Kastimia Kasima		Mar Win.	AND CUMP	Contraction of the second					

4. 38. 37. 35. 36. 40 DHINE SHE SHE SHE ARJUN WIDY 37. V.P PSHUPPEN NTH (V.P. ASHWANTH) That and shurt 34. 3 S. No HARISH.M ANTHONY & BHARATHI, M ABHINAV K JOBIN GEORGE AJIN JAMES **Students Name** ISHAAQ AHAMED KHAN Date Things B offices Blines & Bhinesh & M. Andano M. Androva H. Andronz M. Andronza NR. World Mr. Horsey to have of Blance Andrehold A is heavy A Pro-Abunda genniege 1 Sharay Delinger AT I Island Alimater MR. Warde Antinor Johnner I's han ACMIT MA. X Collect M. And the Hinesour A Annual Annal . A. MRCHON Aspendage . Honorand A-T

S. No Date 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 11. HEMACHANDRAN R 12. KIRUBAKARAN B 13. ARAVIND KUMAR A 14. ARISH BALA D 15. ADHAVAN M	College N Duration	Vame: AMET University, Chennai - 60 1: 1 Week (11.03.2024 to 16.03.2024 Students Name	3113.) Afternoon 11.03.2024	12.03.2024	13.03.2024	14.03.2024	<u></u>
DIVYANSHU OJHA Office 1. VARUN CHITRANSH Office 3. ARYAN SINGH BORA Office 4. FARHAN JAWED ANSARI John Chitran 5. ARGSUKH BAITALIK Amgsudb.l 6. JIGNESH VIJAY KATE Transmithe 7. AKASH S Jignesh VIJAY KATE 8. ADHITHYAN P Jignesh VIJAY KATE 9. DEEKSHATH V Jignesh VIJAY 10. GIDEON MATHEW Jignesh VIJAY 11. HEMACHANDRAN R M. Jignesh VIJAY 12. KIRUBAKARAN B Lirubakaan Endated 13. ARAVIND KUMAR A Angaray - Arthon 14. ARISH BALA D P. Jignesh M. July	. No	Date	-		1	11	
2. VARUN CHITRANSH 3. ARYAN SINGH BORA 4. FARHAN JAWED ANSARI 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	1.	DIVYANSHU OJHA	1	100 M	.	Olla-	alle alle
 ARYAN SINGH BORA FARHAN JAWED ANSARI FARHAN JAWED ANSARI FARHAN JAWED ANSARI ANGSUKH BAITALIK ADHITHYAN P AKASH S ADHITHYAN P ALLAR ADHITHYAN P ADHITHYAN P ALLAR ADHITHYAN P ALLAR ADHITHYAN P ALLAR ANGUND MATHEW ARAVIND KUMAR A ANAWAN B <li< td=""><td>2.</td><td>VARUN CHITRANSH</td><td>Con states</td><td>and the</td><td></td><td></td><td>Will Man</td></li<>	2.	VARUN CHITRANSH	Con states	and the			Will Man
 4. FARHAN JAWED ANSARI 5. ANGSUKII 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 	ູ.	ARYAN SINGH BORA	af s			Je.	A.
5. ANGSUKH BAITALIK Amgsubb-8 Agsubb-8 6. JIGNESH VIJAY KATE JIGNESH VIJAY KATE JIGNESH VIJAY KATE 7. AKASH S JIGNESH VIJAY KATE JIGNESH VIJAY 7. AKASH S JIGNESH VIJAY KATE JIGNESH VIJAY 8. ADHITHYAN P JIGNESH ATH V JIGNESH ATH V 9. DEEKSHATH V JIGNEON MATHEW JIGNEON MATHEW 10. GIDEON MATHEW VIJAA VIJAA 11. HEMACHANDRAN R M. JUSE (A JIGNESH VIJAY 12. KIRUBAKARAN B KIRUBAKARAN B KIRUBAKARAN B 13. ARAVIND KUMAR A A.Annaut - A.Annaut - 14. ARISH BALA D P. JIA JIA 15. ADHAVAN M M. JULA JIA	4.	FARHAN JAWED ANSARI	forthan -	anter	-	farther	for hun the work
 G. JIGNESH VIJAY KATE JIGNESH VIJAY KATE JIGNESH VIJAY KATE JIGNESH VIJAY KATE JIGNESH SHATH V 9. 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 M. JULL 	5.	ANGSUKH BAITALIK	Amasulp. 8	Angsulth-C	0	3 Angsolah, B.	3 Argsoldh B. Anduspart 8
7. AKASHS User 8. ADHITHYAN P Multiple 9. DEEKSHATH V World M 10. GIDEON MATHEW World M 11. HEMACHANDRAN R M. M. Starley M 12. KIRUBAKARAN B Kirubabaun Entruta 13. ARAVIND KUMAR A A. Anger S 14. ARISH BALA D P. M. S 15. ADHAVAN M M. Acuto	6.	JIGNESH VIJAY KATE	- Headle	the state	- Y	- Tuester	- Theater Provide
 8. ADHITHYAN P 9. DEEKSHATH V 9. DEEKSHATH V 10. GIDEON MATHEW 11. HEMACHANDRAN R 12. KIRUBAKARAN B 13. ARAVIND KUMAR A 14. ARISH BALA D 15. ADHAVAN M 15. ADHAVAN M 	7.	AKASH S	yhe-	Al-		B	R.
 9. DEEKSHATH V 10. GIDEON MATHEW 11. HEMACHANDRAN R 12. KIRUBAKARAN B 13. ARAVIND KUMAR A 14. ARISH BALA D 15. ADHAVAN M 14. ADHAVAN M 	8.	ADHITHYAN P	Milton	ATT -	İ		All Mar
10. GIDEON MATHEW Cull Gull 11. HEMACHANDRAN R M. M. Le. M. M. 12. KIRUBAKARAN B Kirubabaan Enated 13. ARAVIND KUMAR A A. Analysis 14. ARISH BALA D P. M. J. Kirubabaan Enated 15. ADHAVAN M M. J. Le.	9.	DEEKSHATH V	THE N	HALD.V	ן יכ	N QUA	~ V DUS VDUS.
11. HEMACHANDRANR M. H. Les Value 12. KIRUBAKARANB Kirubabaun 13. ARAVIND KUMARA A.Annaut 14. ARISH BALAD P. Jack 15. ADHAVANM M. Aud	10.	GIDEON MATHEW	Grandh	Sunth	, F	Curth 1.	Curth of Cr. M.
12. KIRUBAKARAN B Kirubakan Prada 13. ARAVIND KUMAR A Aranad - Araad 14. ARISH BALA D D. And Araad 15. ADHAVAN M M. Acut	11.	HEMACHANDRAN R	M A Culo	the byte	·K	the Mark	Why Cherry Why
13. ARAVIND KUMARA A.Aray: A.An 14. ARISH BALA D D. M.An 15. ADHAVAN M M. ACC M.Acc	12.	KIRUBAKARAN B	Kirubakaw	n tointal	cer	en kinderbern	an kinder being kitubakara
14. ARISH BALA D D. D. M.	13.	ARAVIND KUMAR A	A. Aray:	- A.Anz	tí	L' A.Amy.	1: A.Amy: A.Ang
15. ADHAVANM M. Ace	14.	ARISH BALA D	D.	X	D	D. A.	J b. A Glad
	15.	ADHAVAN M	M. Ald	Mixee	01	er notet	er notit Model

J

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

52.	د د	31.	30.	29.	28.	27.	26.	25.	24.	23.	22.	21.	20.	19.	18.	17.	16.	3. INO	
MALLICK	ARSLAN MALICK	KALIMUTHU S	GOVINDA RAJA PERUMAL R	ARUN PANDI V	ARYAKI PRASHANT NARKAR	LAVITHA V S	VIGNESH V	ADHITHYAN A	JAGANADHA GANDHI E	HEMANTH KUMAR G	KARUNYA V	KANNIKA KESAR . R	SAKSHI D	VARSHIINI S	KALAI G	DHINESH S	JAGADEESH F	Date	Students Name
(Jaylon Par	1 n white	Sol-	nound	Alun Pondi	And and	Art	June	B	744	Hama	Kasumun	R. AUT	AAA.	Lation	C.Klui		P.F.A.		
monet	OF THE WAR	Samo	format	Abun Parti	there was a	Det .	General .	A	Add	Henry	Karump	R. VEDX -	all a	Liferen	(m/Ch-		D. Fred		
Jonar	A LANDI	. Manul :	former	Asupporti	the for the	Outon .	A.	A	LA-L	Homanth	Kanunga	12. Kart.	and the	Advention	Cifeli		N.C.		
(Jennomen (The second second	Xam.	Joursof	Brunhard (Harton .	Part .	13		the wit	Hundart	Kanunja	R. Werts.	S. Set	Charter	しんし		N SA	A	
Day on the		sont-	Connot.	A - Juntumer	Providence of the second	· A	Lage.	Þ	they	Aman -	Kasumer	D. Mary	afe	Anin	6. Kh	CO. XX	(J)	i Ol	
	1	T T	Compo	Vour Vour	And the state	A.		P			Kounya			1 Avin	Call	A PARA	FUR	100	

.....
to v.P Alshwanth S. No ယ ပာ 5 36. 35. 34. 38. 37. **Students Name** ISHAAQ AHAMED KHAN JOBIN GEORGE AJIN JAMES ABHINAV K HARISH M ANTHONY M DHINESH.B Date Thing in thing is things & things is things is things is things is the state of the MR (Hlaces Albunar CAR Americ T M. Firstrong M. M. M. Andler M. S. W. H. Andler blas Advinant MR. Claure -Advine 1 Tishaard A A A A A . MR. Maart Haindry sols? deprindent deprindent. All . Mr. Wiener A A 1 TShare Alderman A WAY DO Mahanawad THIT topiner, Thorn No. it Astrabel Que

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

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

VFAR

15.	14.	13.	12.	11.	10.	9.	8.	7.	6.	5.	4.	دی	2.	1.	S. No
ATHUL KRISHNA	ABILASH	SEBOVIN	DHYNÆSHWAR	THARUN	JOE JAMES SERAPIN	RITHIK VIGNESH M.M	SHYAM	LIKITH J	SHANMUGARAJ M	HARISH K	MURALIDHARAN R	KAVIN R K	SIDHARTH ROY	SAMMAR	Students Name Date
And the second	p R.	Scherin Stopp	the item	those of the	the the the	APP P	Leeb teep	J. h. Mart F. M.	M.S. ; Alt	Part there	AP LOA	Kowin: alloal	Studenter Sin At	Sanner Sporman	22.03.24 23.03.24
A AR	De	. Sebout	A		Plant	P	A A A		Ch - L	AAA		havin			25.03.24
H	RA	A chovin	m	ab	Dund	Ditta	Accel (11			CO	Pervin		De L	26.03.24
AW	2.4	Selogiis	AT &	A C	X	XXX	freed	fay's	T.		5	have	0.11 h	Carment	27.03.24
- Allen	Reve	Sepanna	Ren		p. j.	10.1	Part.	A. MAA	· L			Komp	Oundry	Church	28.03.24
A A A A A A A A A A A A A A A A A A A							_								29.03.24

Samo

	23.	22.	21.	20.	19.	18.	17.	16.	S. No
		-			SURYA	MOHAN	SHAHA	JASUVA	Studen
					KANTH	AMED KA	IAWAZ K	JENIS JO	ts Name
						IF	HAN	VIN	Date
				2 au Duch	A.S. white	Mar	202	and h.	22.03.24
				1 AN	RUN	A	Ma	hedr-	23.03.24
					Pro Jemp my	¥.	- Charles	the	25.03.24
					2 R. Sury	- Atril	A MA	The second	26.03.24
op go					unt Richt	M. Kend	Str.	2	27.03.24
h					A. Aunt	Maharak	24-	the second	28.03.24
					F				29.03.24
		1	I			I			

INTERNSHIP TRAINING PROGR	Vi Microsystems Pvt.Ltd, Chenna
GRAM	nnai-96

7 DCITV CHENNIAI - 603113

> YEAR : II

DURATI	GE NAME: AMET UNIVERSITY, CHEN ON: 1 WEEK.	INAI - 603113.					SI	ESSION: 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	Just Non .	hour	P	A A	P	VA/	
2.	NAVANEETHAN P	P. Naund	P. Namythat	P. Mart	PN amunilly	P. New		
د.	RANJITH M	h. Rauth		R P	H. Car	P		Ň
4.	NAVEEN KUMAR	Na l	Non-Alth	mily	Nauntry	Nouhin		
S.	SHAMER BASHA C	of bat		NA A	. Advert	SEA.	Shanes	
6.	SIVA KUMAR I	T. M. E	Stolen !	Pril	· Hitem	J. Acu	Stra.	
7.	ABISHEK A	A Abolinat	A Allow .	D. Ahard	A-Diaula	A. Abher	Abichel	
8.	SANJEEVI KUMAR S	Sonjour Kune		Busenila	Jumiliu	Summer	Seijawile	
9.	BRUHAN MALIKDEEN A	A rith allow	lever ben	-orter Mard- Mi	#- Kunder	A-mellidered	Actionspice .	Y.
10.	MOHAN RAJ M	M.Molida.	the fi	M. Mabyla.	Mi Mahul	M.Mohuky	march -	
11.	GOKUL PANDIT R	S Cand	Rand	Pack	The Chart	RAN		
12.	PRAVEEN C	E.	F		A C	JAN A		
13.	POSWAL KHUSHI S	Khurshi	Webs -		Khughar?			
A	RANNEL	1 Allow I		at (and)		
15.	MARIA ROSHAN M	N. M. OCHI	H.M. L.	M MLORA	N-N-Q	THE LI	Ny man	

				24.	23.	22.	21.	20.	19.	18.	17.	16.	S. No	
				THARUN KUMAR	ASHWIN R	SELVAKUMAR S	GANESAN M	AMANATH G	ARAVINDAN S	SOOROJ	HARI PRASATH S	VISHWAS S	Students Name Da	2
													ite \	
				There	Shin P.	S. A.	At .	cr. And	N. AND	P	Shipp.	H. Vier	22.03.24	
				that .	Almuto. P	Shop-	'hant'	C. Amert	S. ANDUR	All and a second	S. mi kut.	ANAL	23.03.24	
				and a start	Annin 2	S. S.S.	Granest	G. Mont	S. How		the is is	is. Kil	25.03.24	
		12.7		Ammy t	Ashini-12	&. Shop 1	Crowner.	C. Amost	S. NYONALLS	R	J. Juiland	s. what	26.03.24	
90 00				Renth	Admilia	2 Dal	Opensh	g. took	7. A vankan	R.	Sidam Kul	s. vie	27.03.24	
h	The second second		and the second	14		XA					SHAK .	S MA	28.03.24	
			9										29.03.24	

15.		13.	12.	11.	10.	9.	8.	7.	6.	5.	.12	ເມ	2.	1.		S. No	COLLE DURAT
MARIA ROSHAN M	RANIFI	POSWAL KHUSHI S	PRAVEEN C	GOKUH PANDIT R	MOHAN RAJ M	BRUHAN MALIKDEEN A	SANJEEVI KUMAR S	ABISHEK A	SIVA KUMAR I	SHAMER BASHA C	NAVEEN KUMAR	RANJITH M	NAVANEETHAN P	SAMAN SINGH	Date	Students Name	GE NAME: AMET UNIVERSITY, CH ION: 1 WEEK.
N. M. R. A. H. M. Bari K	M. And M. M.	Howester Howester	HAR I AND	R. Gones R. Consul Fault 1	M. M. altri M. Maker R	But Alcuir A-Barbarretter A	Schipeeritumes Sogivervitume S	A ABIEN A ABIEN A	J. L. Vol. I	Alinhat Shunden Q	Mand 1 maren	M. Keyl, Aron I	P. Namethal P. Namethal	Vindhar . Commandor . =		22.03.24 23.03.24	IENNAI - 603113.
Mary H.	The free of the second	and he was here	AL -	Gobul Rudit P. 600	TRANSPORT M. M.	. Bubundrefile r. Bu	anjohnm. Carla	HARVE A.A	F.Shur H.C	humper Sha	Man A.	S. Prost M. (P. Nambal D. N			25,03.24 26.	_
2 Ray M. ME	1	ingen + might	THAN THAN	halmidtly a hent 1	whither in much	derindite A. Burk	famos Songer	A. Aleri	And F. R	2 A		N AN AN	and of the	und and gave)	.03.24 27.03.	
Strank Pr		1		Bundy	& Mrally	un Delleder	ing Soijeen?	we prichele	the sideker	A channer	8	A.		· Infre .	1	24 28.03.24	SX
																× 29.03.24	(EAR : II ;ESSION: FN

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

CS OF

15.

••



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

£,

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

SESSIO	YEAR
2	
FZ	III

15.	14.	13.	12.	11.	10.	9.	8.	7.	6.		4.	نی •	2.) .		S. No	
ATHUL KRISHNA 1.~	ABILASH	SEBOVIN	DHYNASHWAR	THARUN	JOE JAMES SERAPIN	RITHIK VIGNESH M·M	SHYAM	LIKITH J	SHANMUGARAJ M	HARISH K	MURALIDHARAN R	KAVIN R K	SIDHARTH ROY	SAMMAR	Date	Students Maine	Studente Name
Aller Aller	20	Schouin S	B 1		P. H.		o tea	I. P. Mard T.	All and a line		Den 10%	Kernhur, K	Soulinet A	Saraha 2		22.03.24 23	
- free	F	ebour of			ie l			Man of the				~/im.	dhuadha /	Sunder S		3.03.24 2	5
Atreto	R.	ebovin s	A A	De la	D. O. A.		Sugal -	PUP 1	1.00		- mar -	Variation 1	" Elman a	Journer 1		25.03.24	1
Add .		elovin -					and a		A DT .	J Hark			your a	anne		26.03.24	
ALAS	RACE		9			R C	Alwy T				A CAL		Convert			27.03.24	
	ROD	Selamo	OM		All C		N. Marken	M. Com		the top		(Incl with	X	CA MAN		28.03.24	S
× JAM	×							A CONTRACT OF A						*		29.03.24	ESSION: FN

SAG

h

CLARKER STALL

	23.	22.	21.	20.	19.	18.	17.	16.	S. No
					SURYAKANTH	MOHAMMED KAIF	SHAHANAWAZ KHAN ×	JASUVA JENIS JOVIN	Students Name Date
				N. M W. W.	ed at for a		Shime Si	A C A	22.03.24 23
	1345			TO A TRACT	C ALL P	A A	S.	-	.03.24 25
				and themas	XARD	aft mile	1-2 - 2 - 2 - 1 1 1	A Mar	.03.24 26.
Co 10					the plan			R	03.24 27.0
ap A					1 + R. Qorth	ALL RANGE		H- H- H	3.24 28.03.24
	_								29.03.24

COLLEGE NAME: AMET UNIVERSITY, CHENNAI - 603113.

YEAR : 1 SESSION: AN

15.	14.	13.	12.	11.	10.	9.	8.	7.	6.	5.	4.	3.	2.	1.	S. No	DURAT
KIRUBANATHAN J	MATHEW FRANKLIN P.D	RAMANA A	MOKESH T	HARRIS CHRISTOPHER	SAHAYA JABIN BIJO	LOKESHWARAN K	MASILAMANI V	VASANTH C	RAKESH KS	MATHI BALA K	YUVARAJ T	MUTHU SURESH S	SRI SABARI BALASAKTHI S	RANJITH C	Students Name Date	ION: 1 WEEK.
J. laidet	PINE	A. Rent.	1. att.	2 Cod	Sald-	toot	Vimastancemi	V-40	Junelys	H. mater Dale	finit	A	Stow	C.Rayijith	01.04.24	
J. half	D. Mitur	Aller .	1. Mat	All of	Sal da		V. nourlanswi	Virtury.	Katij	k. Maki Ente	A.F.	All and a second	50	c. RanjiR	02.04.24	
J. King	D. Nother	A. Paron		- Have	- AUA-		v.mourlaman,	Vite.	Rushis	K- Mathimela	ym. fr	Start	K.	C. Ranjith	03.04.24	
J. Luso	12- Hathew	A fend.	- THE	D all'	B K de	Logent	1 Hasilaman	VAC-1.	2 April	and retranged	A.	giosp-	and a	C. Ray ith	04.04.24	
J. Yub	D. Mathau	A. Ation	1.MAT	the state	Fordal		V-masilanoni	Vitten.	(Produced	L. manifeda	4.	E.	and a	CRangitt	05.04.24	
J. Jein	DUNG		Tight	000	10Peb			Und Cr.	SAF	K. Mall Cal	₩.	Sug	Arg	CiRanit	06.04.24	
1			J.		~				A A A	1218					07.04.24	ESSION: AN

Anches 194

			26.	20	24.	23.	22.	21.	20.	19.	18.	17.	16.	S. No	
			S. RAUIN KUMPR	A. UMAL KANNAN	SRIMAN S	MONISH P	NAVIN E	VARUN KUMAR M	JEEVIDOSS J	LOGESH G	PRASANA A V	MANOJE KUMAR	VEERA GANESH	Date	Students Name
			G4.	· tyle	Stimute	Honish 1	D. }- (Vert	The se don's		0.11.77 mart		and		
		1	Q-4 .	· ty-b)	P T	How when P	N -	- MON	S IT Tour day		11 () XI, ST.			l	
				10-1-4-1-		Moninh	n n n n	Prod Cont	2 A. Jange		C.Wash				
				PA-	A. (Rain T	A - A	V and) (internet	Nr.		
		(C)		Prof.	Port of IT	5	Var A		UN AND		· Friday		And		
		4	\$) -	AND I	MONISH.P	ting	Vast	altered , Do		Ð	C.March				
7	The second s										p				

MET UNIVERSITY.	
CHENNAI - 603113.	Vi Microsystems Pvt.Ltd, Chennai-9 INTERNSHIP TRAINING PROGRAM

YEAR : I SESSION: AN

33. 32. 31. 30. 29. 27. S. No 28. 26. 25. 34. 37. 36. 35 DURATION: 1 WEEK. COLLEGE NAME: AN ა .8 **Students Name** ROHAN MOHAMMED SHAJI S MANTHAN VHADGIRE SULTHAN MOHAMMED MAREERI MCHAMMED SHARIF ADHITHYAN K S RIYAZ KHAN 📈 PRANAV D ABHISHEK SATHESH ZAID GHANI NIKHIL JETHWA **TEJAS SURYE** Shan, PRANAV PRINCE ANGITH B y dasel Date Had then A.A. ß Sang 01.04.24 S SIG 02.04.24 A HA Temp 1 and Joseph Land ğ いませ Mard Collins 14ANE E PT-FA 03.04.24 Ř CUR F NE: Later. X TA Prof NA IT 04.04.24 Mitte the second 800 S ALAN I A Curr SALL I Q 05.04.24 12 APridia the second 06.04.24 10 Ŕ E Ś 07.04.24

Z	Vil
TEF	Mic
RNS	rosy
HIP	/stei
TR.	ms]
AIN	vt.
ΝG	Ltd
PRO	, Ch
OGF	lenn
RAN	lai-
	96

TENNIA I 211209

YEAR : I

	15.	14.	13.	12.	11.	10.	9.	8.	7.	6.	5.	4.	<u>з</u> .	2.	1.	S. No		DURATI
	KIRUBANATHAN J	MATHEW FRANKLIN #.D	RAMANA A	MOKESH T	HARRIS CHRISTOPHER	SAHAYA JABIN BIJO	LOKESHWARAN K	MASILAMANI V	VASANTH C	RAKESH	MATHI BALA K	YUVARAJ T	MUTHU SURESH S	SRI SABARI BALASAKTHI S	RANJITH C	Date	Students Name	ON: 1 WEEK.
(1 alt	Dula	A. Rent	1.64	Baset	Alder	Logod	y. mar il mari	V-J-K	(Jezd S)	K. Hatt- Lato	A.	Annal		C. Panyith		01.04.24	UNAI - 003113
4	1 APC	DiMute	Allert	T	alleson	Salya	Alero-	V. Mary Ilaman I	NAC.	Check	Jr. nothit al	A A	Same		C.Ranyith		02.04.24	
	J.C.A.	Lint	A. Quedr		2 Reco	Subort	Kard A	V. Marsila MANI	Utle.	(testand)	p.nortal	J.	Sort A	Saha-	C. Buy ST-		03:04.24	
	Contraction of the	D.Mary	Allera	1.000	H.	The second second	Nord I	V.M.ds James	Vart 1	5.	- M. Now	2	Stal	School School	C.Ran i 1		04.04.24	
	J. left	D. Lettro	A.K.			Salya &		V. may - of	N.M.C.	- Aller	K wet	ACCERT	GURIE	Sel		C. Ranyith	05.04.24	
2	J. Kurth	D. Marton	A. ho	T.	A BE	CONTRACT OF ANY		V.Marianan	1-1(-J.	and	Kinath fale	97	Superior	Sabar	C.Kay 1Th	5	06.04.24	S
	and the second																07.04.24	ESSION: FN

		26.	26.	24.	23.	22.	21.	20.	19.	18.	17.	16.		S. No	
		S.PAUINKUMAR	A.VIMAL KANNAN	SRIMAN.S	MONISH P	NAVIN E	VARUN KUMAR M	JEEVIDOSS J	LOGESH G	PRASANA A V	MANOJE KUMAR	VEERA GANESH	Date		Students Name
•		¢ Į	(a)-14.	dit.	HOMMED	0	Vary	Junio arter	a.P	A.V.Passan	no sol	ala)			
		R I I	· HARD	P. T	Monsh-P	P	Vand	Drest of est of	G.A	P.V Russey	mh oi	AD			95 5 - 50
		9-9-9-	· tyke	p.p.	Homint 7	P	Vort	inder I L	5	N. J. Promov	A DI				
		50	Querky	ia A	Momish P		Maril		(m. Con		HA	A CONTRACTOR			
		Stor	P14.	A.	Honia P	08		J. Turiders	G P		mode	Arean .			e M
		B-C).	A KA	À, Ŧ.	V-10- A.P	Pp.	Mar A	are front O. B	G.D						
	1														

a company in the same and the same the	38. ANGITH B Charles Colored Colored Colored	37. PRANAV PRINCE Changer Changer, Changer, Changer,	36. NIKHIL JETHWA Attract JURKAN Nitem Autom	35. TEJAS SURYE AND AND THE SURYE	34. ZAID GHANI gid than - Joid than	33. MANTHAN VHADGIRE Automatic Automatic Automatic Automatics	32. SULTHAN MOHAMMED HAREERI Survey Southern Sou	31. ADHITHYANKS	30. MUHAMMED SHARIF	29. RIYAZ KHAN W WAR WAR WAR WAR WAR WAR	28. PRANAVD (1) (1) (1)	27. ABHISHEK SATHESH	26. ROHAN	25. MOHAMMED SHAJI S Show Show Show Show Show	S. No Date Date	Students Name 01.04.24 02.04.24 03.04.24 04.04.	COLLEGE NAME: AMET UNIVERSITY, CHENNAI - 603113. DURATION: 1 WEEK.	INTERNSHIP TRAINING PROGRAM
Att Strangt	All Color	Burner (Presser	Mitan Althout	- Hutter firmed	Low How Big How	North and the White was a	State Caller				Con Con .	A south and the		And .		03.04.24 04.04.24		NING PROGRAM
The loss	ALL ALL	+ Attant	Y AHAT ATT	The Tree	Cresh ind material	Show of Car		The state of the s				And		J. My .	X	4.24 05.04.24 06.04.2		
			:				<u></u>			A			th.			,4 07.04.24	YEAR : I SESSION: FN	

Ś

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



Vi Microsystems Pvt. Ltd.,

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

CERTIFICATE OF COMPLETION

This is to certify that Mr/Ms. <u>RAKESH</u>.<u>S</u> <u>[AEC23056]</u>. Studying <u>Ind Yean</u>, <u>B.E.</u> <u>Department</u> of <u>ECE</u> <u>[Electroleal</u> and <u>Computer</u>. <u>Engineering</u> <u>]</u> <u>-</u> <u>AMET</u> <u>University</u> has successfully completed his/her Internship training programme on <u>Errohedded</u> <u>System</u> with <u>IaT</u> in our company from <u>24</u>.<u>JUDE</u> <u>2024</u>. to <u>Ob</u>.<u>July</u> <u>2024</u>. 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.

V. O. A. Quand

Manager Vi Microsystems Dvt Idd., Chennai - 96.



Vi Microsystems Pvt. Ltd.,

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

CERTIFICATE OF COMPLETION

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.



Manager Vi Microsystems Dvt Ltd., Chennai - 96.

