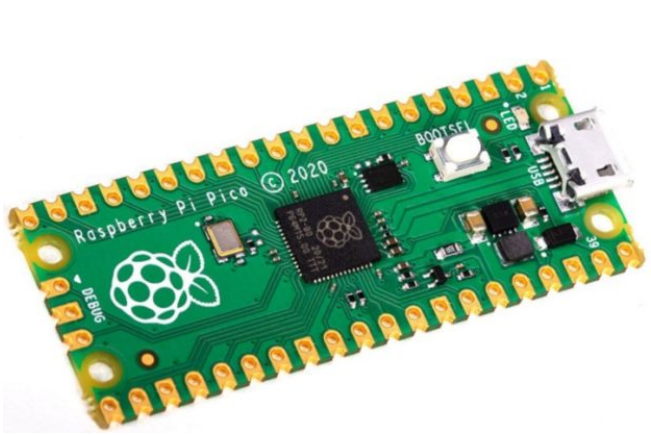
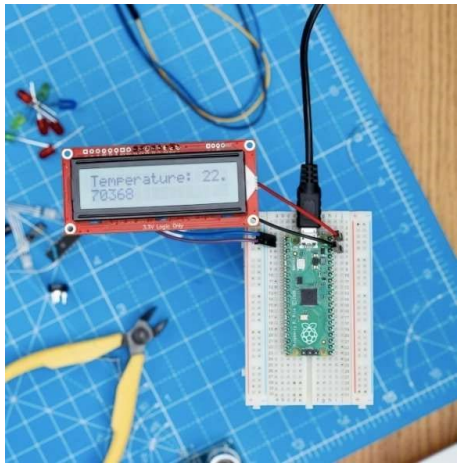
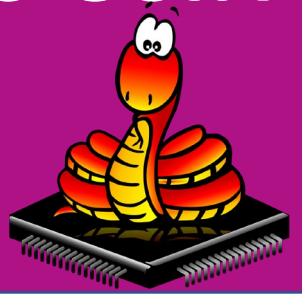


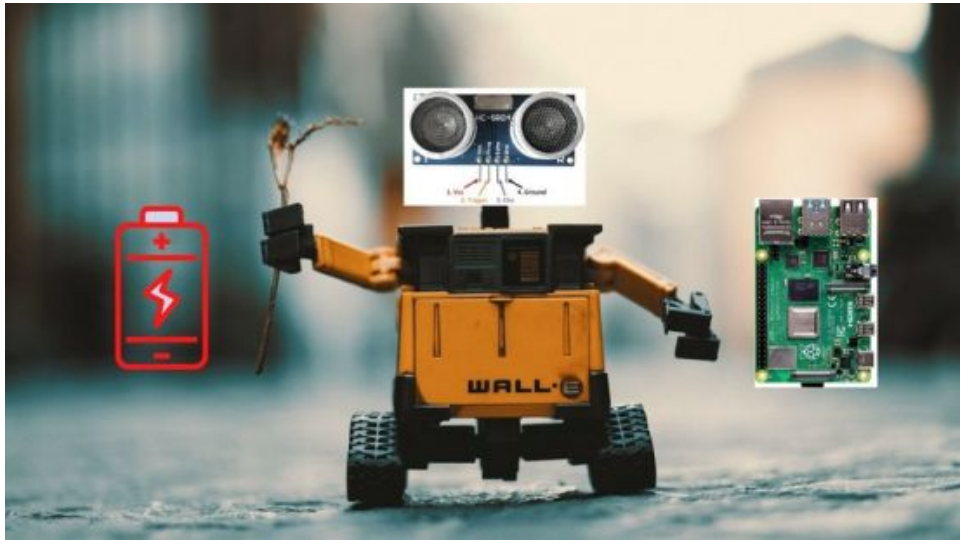


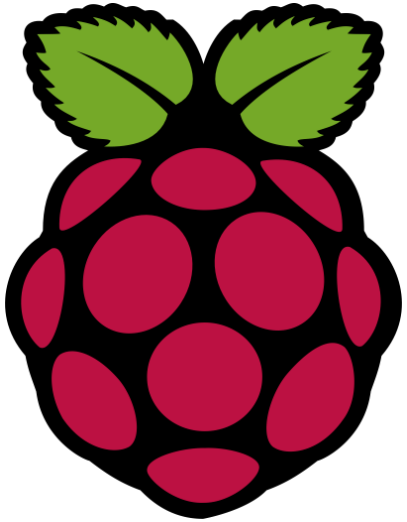
INTERNET OF THINGS (IOT) PROJECTS USING PYTHON

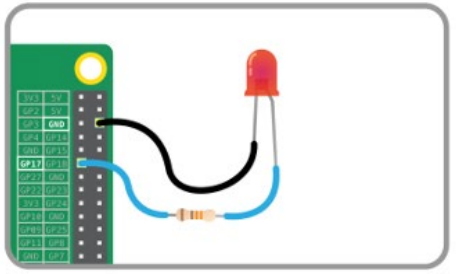
(CSE 4110)

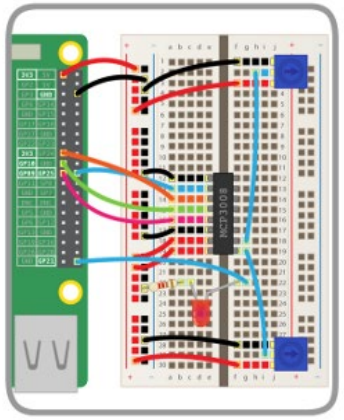
(LECTURE – 2)











Course Outcomes

Course Outcomes	
CO1	Understand general concepts of Internet of Things (IoT), the working of Raspberry Pi and its features.
CO2	Recognize various components, sensors, actuators, devices and their applications.
CO3	Analyze various python programs to interface with sensors, actuators, LED's, cloud and camera using Raspberry pi.
CO4	Measure physical parameters using sensors.
CO5	Demonstrate the ability to transmit data wirelessly between different devices to build simple IoT systems using Raspberry Pi.
CO6	Create IoT devices and systems through a variety of interfaces, including web apps and mobile apps.

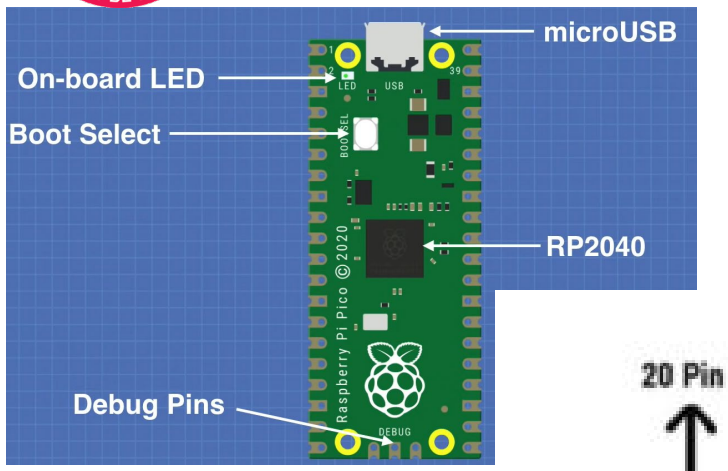
What Students will learn?

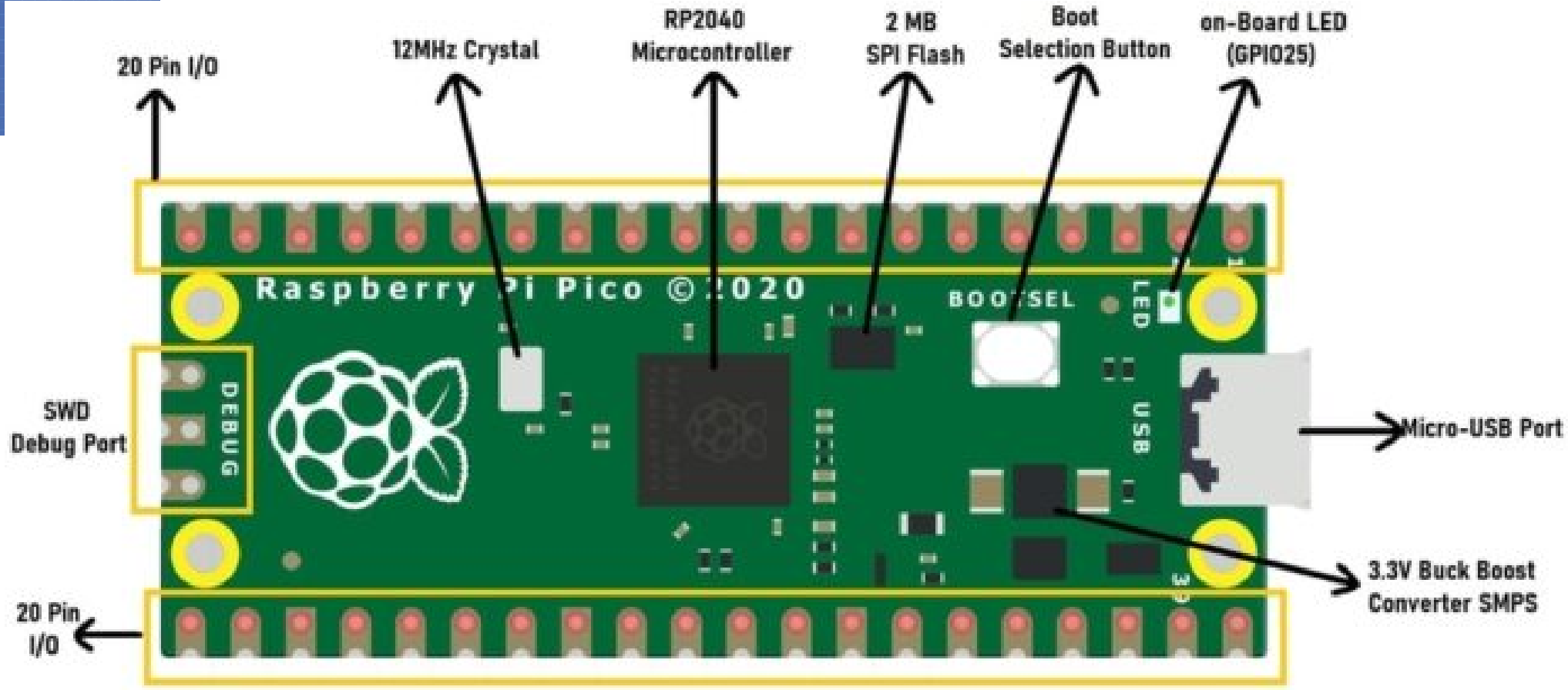
Students will revise the basics of electronics, types of electrical signal, Resistance colour code to enhance the knowledge to build an electronic circuit. Also students will experience about ability to learn coding and electronics to implement a LED flash SOS (Morse code) .

- ✓ **Familiarization with Basic Electronic concepts, their origin, impact, and different applications to improve technical results.**

- ✓ **Implementation of LED Flash SOS (Morse Code) using Raspberry Pi Pico**

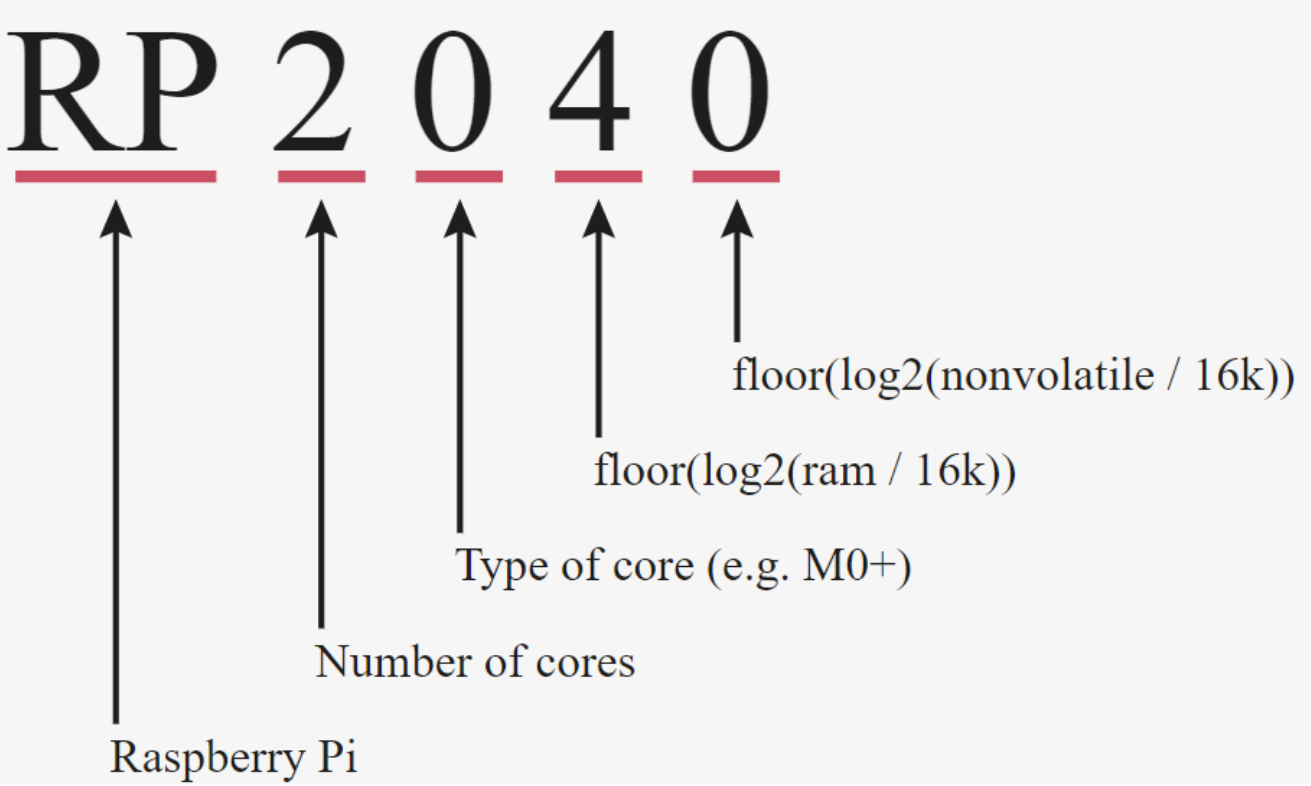
Raspberry Pi Pico – Simple Pinout





Why is the chip called RP2040?

The post-fix numeral on RP2040 comes from the following,



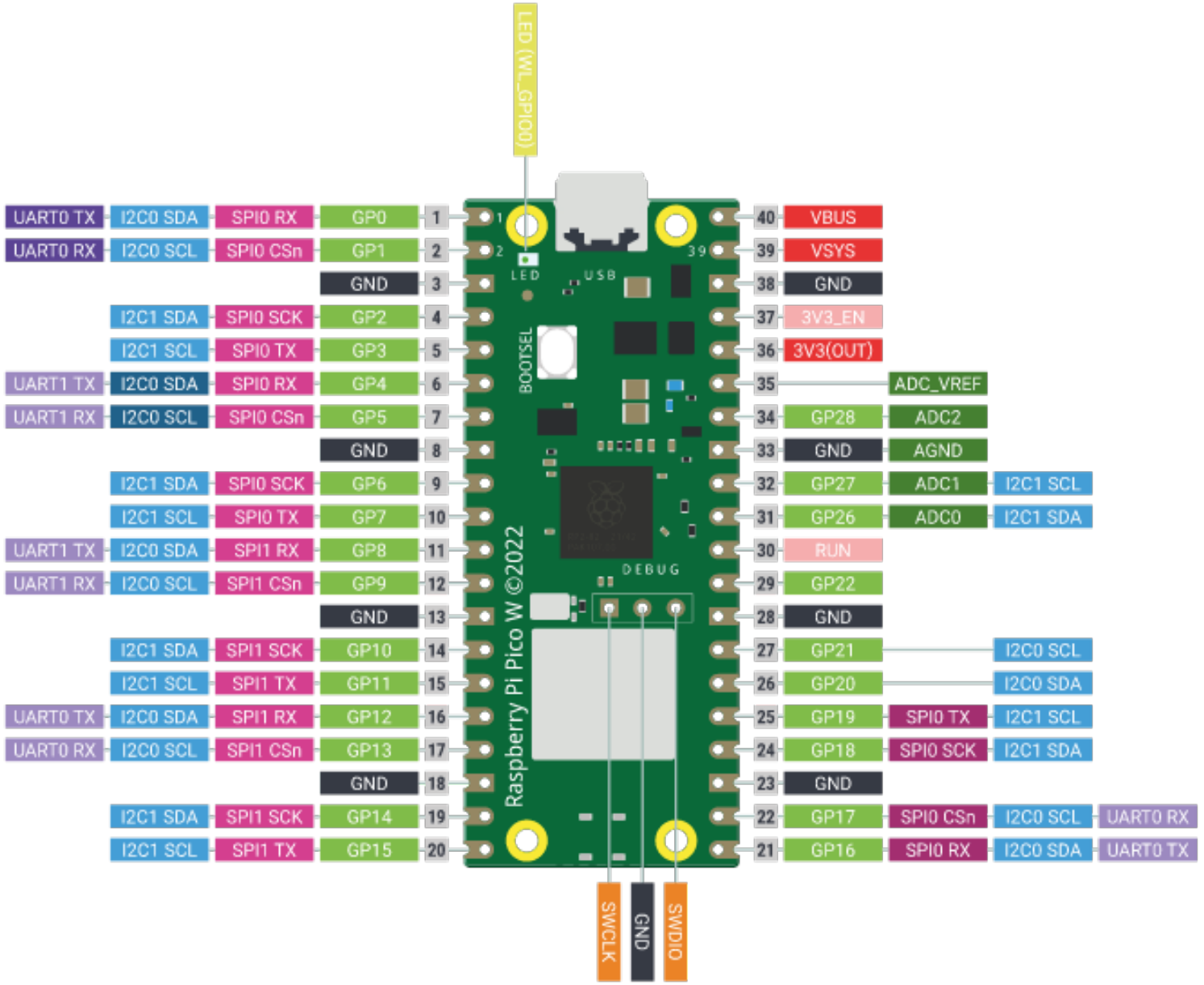
1.Number of processor cores (2)

2.Loosely which type of processor (M0+)

3.floor(log2(ram / 16k))

4.floor(log2(nonvolatile / 16k)) or 0 if no onboard nonvolatile storage

Raspberry Pi Pico W Pinout



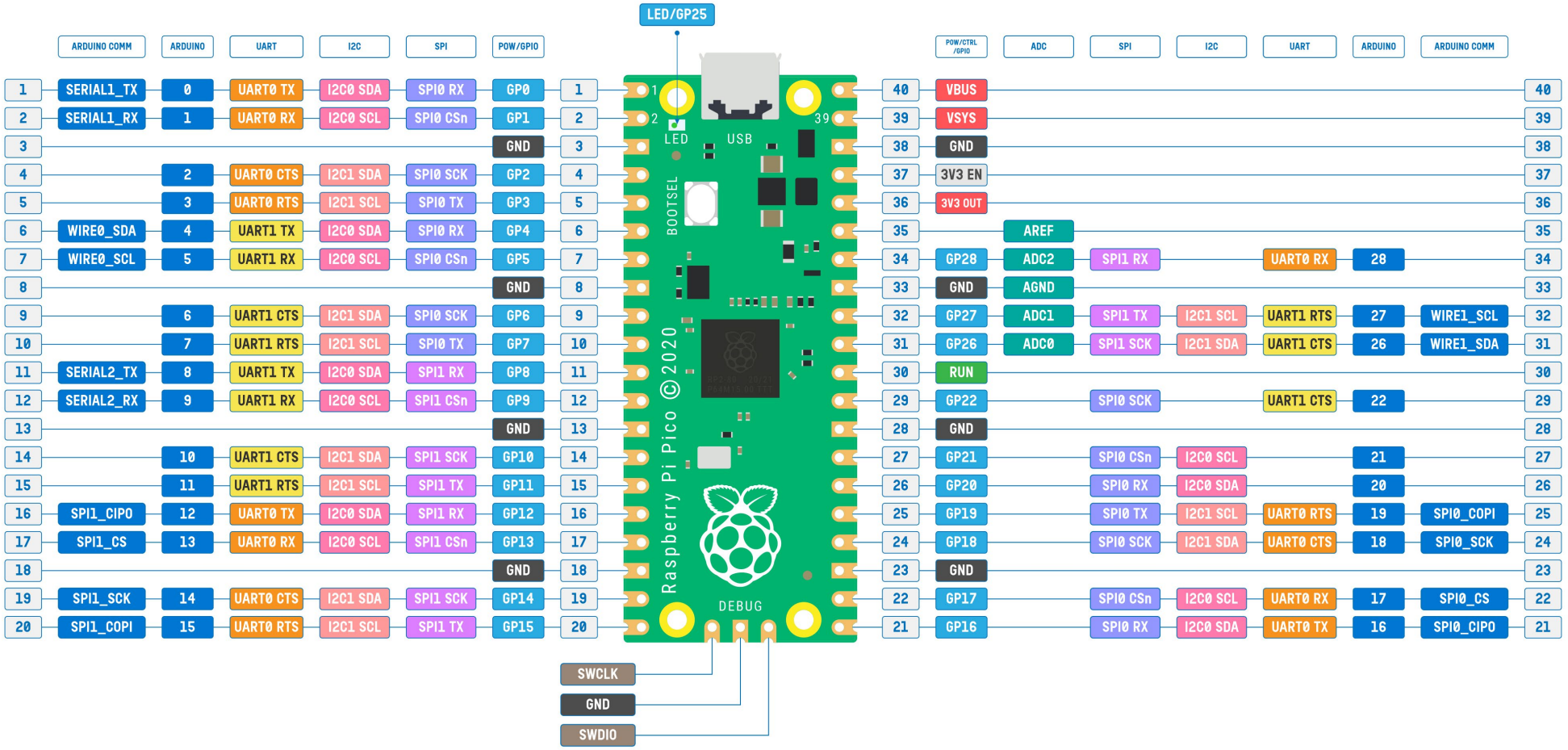
RP2040

- Power
- Ground
- UART / UART (default)
- GPIO, PIO, and PWM
- ADC
- SPI / SPI (default)
- I2C / I2C (default)
- System Control
- Debugging

Infineon 43439

- GPIO

Raspberry Pi Pico – Full Pinout



*Raspberry Pi and the Raspberry Pi logo are trademarks of Raspberry Pi Ltd.

Raspberry Pi Pico vector image is originally designed by Raspberry Pi. Please visit **raspberrypi.com** for more info.

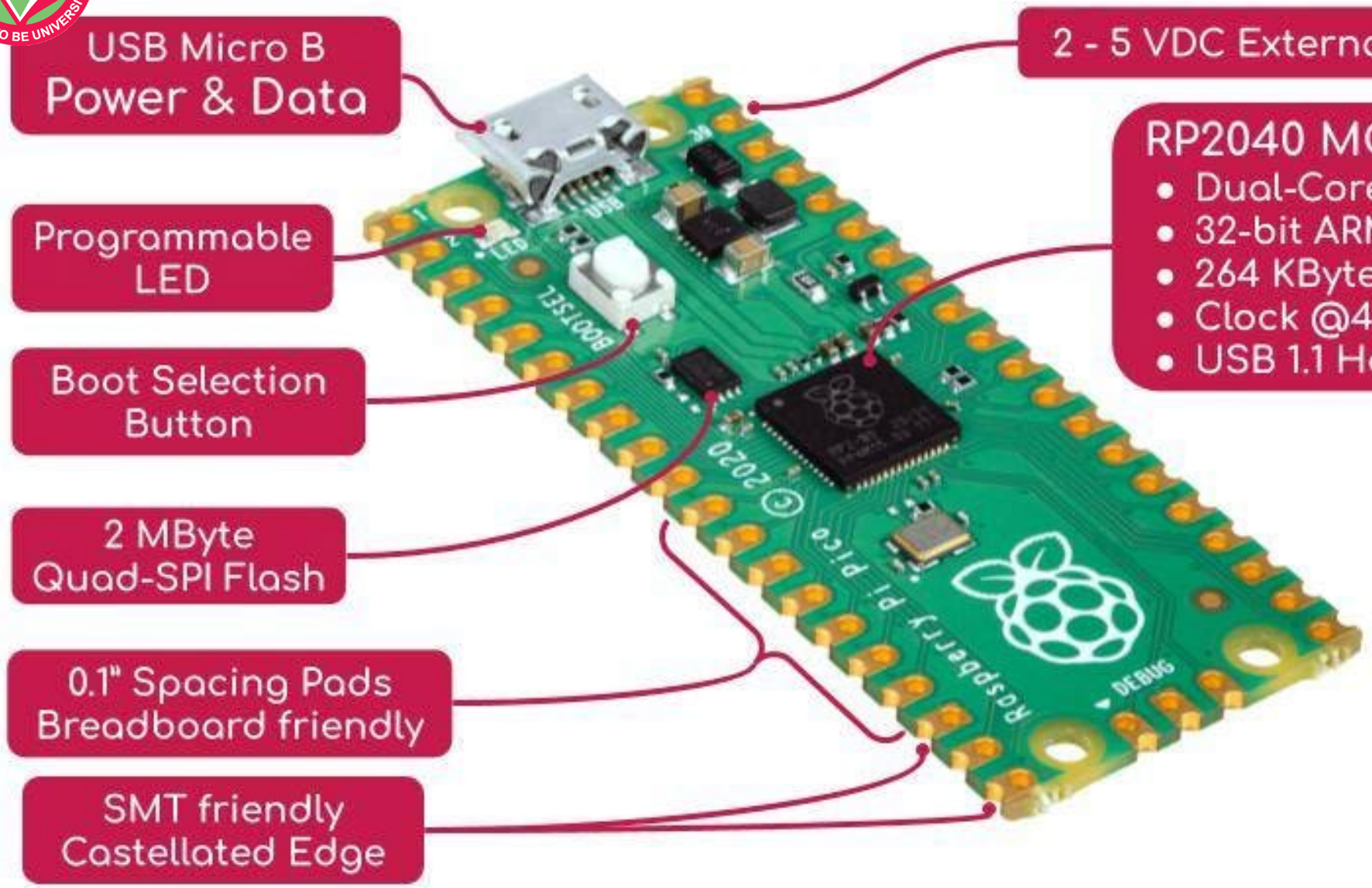
ARDUINO PINS	SWD Pins		
PHYSICAL PIN	POSITIVE SUPPLY	UART1 Pins	UART0 Pins
RESET/ENABLE	GROUND SUPPLY	I2C1 Pins	I2C0 Pins
GPIO PORT/PIN	ANALOG PIN	SPI1 Pins	SPI0 Pins

- **GP29/ADC3** is used to measure VSYS.
- **GP25** is used by debug LED.
- **GP24** is used for VBUS sense.
- **GP23** is connected to SMPS Power Save pin.
- All GPIO pins support PWM. There are total 16 PWM channels.
- All GPIO pins support level and edge interrupts.
- Arduino pins are as per **Arduino-Pico** core by *Earle F. Philhower, III* @earlephilhower
- Arduino's default **Serial** is the USB-CDC of Pico.

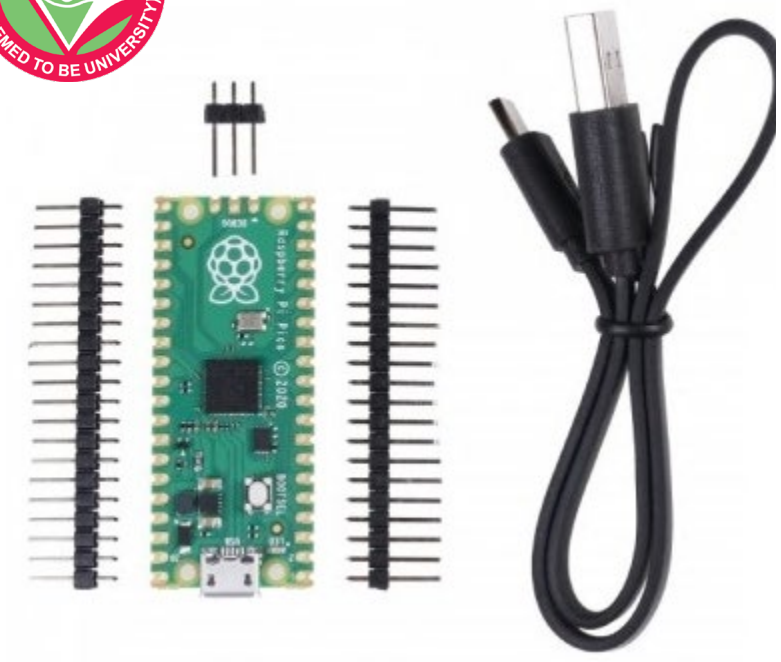
Rev. 0.2, 15-07-2022

Design: Vishnu Mohanan

Raspberry Pi Pico – In Short

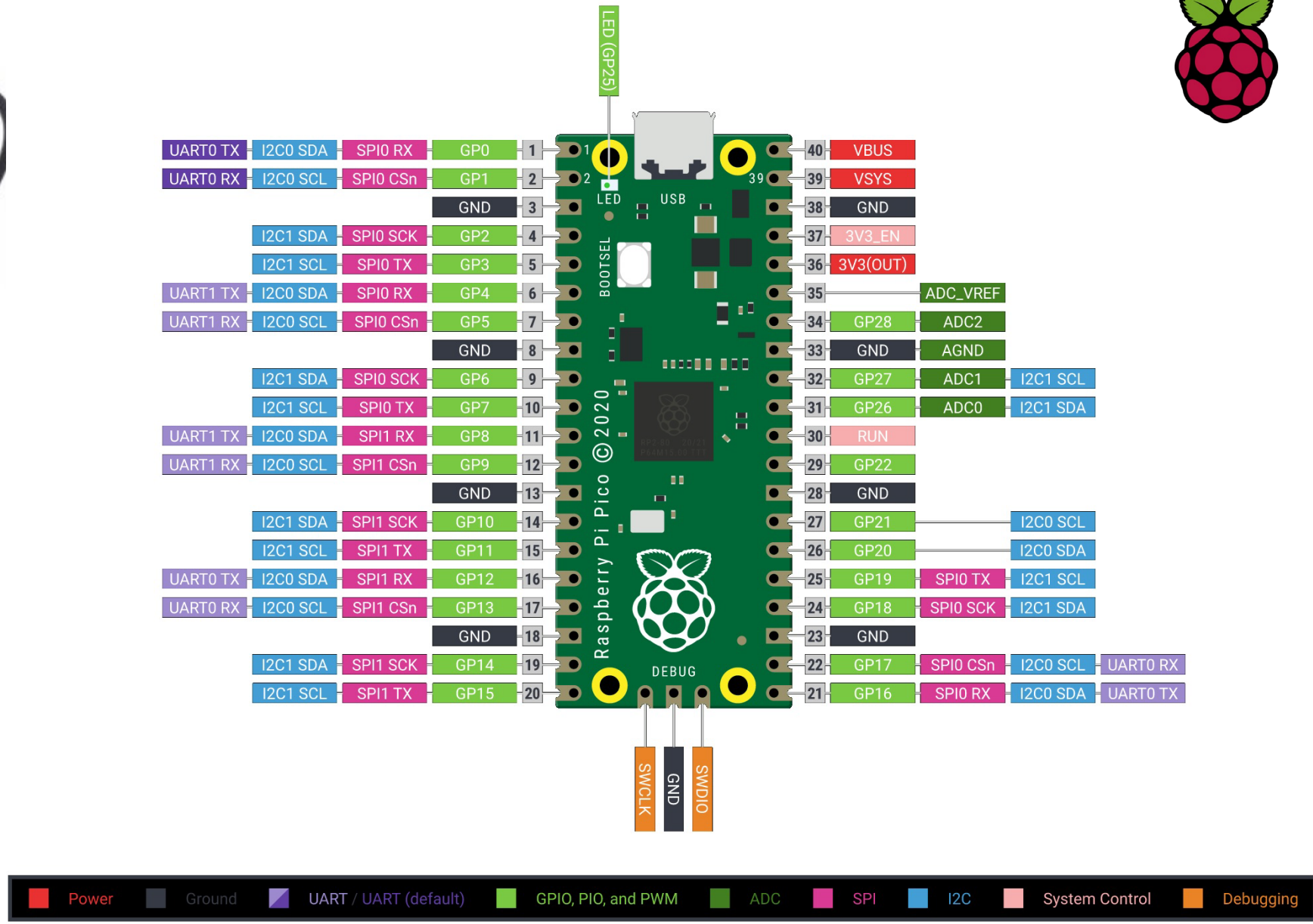


Pi Foundation has released an RP2040 Microprocessor based development board, in the same form factor as an Arduino Nano.



Package Includes:

- 1 x Raspberry Pi Pico
- 1 x Micro-USB cable
- 2 x 20 Pin Header
- 1 x 3 Pin Header



Morse Code

- ✓ Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called dots and dashes, or dits and dahs. Morse code is named after Samuel Morse, one of the inventors of the telegraph.
- ✓ Morse Code uses an alphabet made up of dots and dashes (for instance, the letter "s" is three dots and "o" is three dashes.) It is used by **tapping the combination of dots and dashes needed and pausing for the correct gap duration**. There are longer gaps between words than letters in a word.

Morse Code: “SOS”

A	•—	N	—•	1	•————	?	••——••
B	—•••	O	———	2	••———	!	—•—•——
C	—•—•	P	•——•	3	•••——	.	•—•—•—
D	—••	Q	——•—	4	••••—	,	——••——
E	•	R	•—•	5	•••••	;	—•—•—•
F	••—•	S	•••	6	—••••	:	———•••
G	——•	T	—	7	——•••	+	•—•—•
H	••••	U	••—	8	———••	-	—••••—
I	••	V	•••—	9	————•	/	—••—•
J	•———	W	•——	0	—————	=	—•••—
K	—•—	X	—••—				
L	•—••	Y	—•——				
M	——	Z	——••				

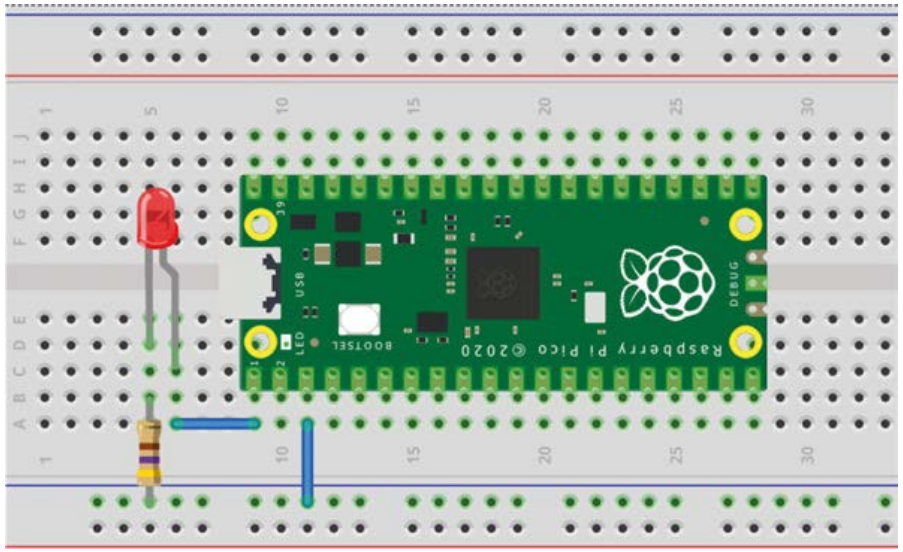
What Does "SOS" Mean?

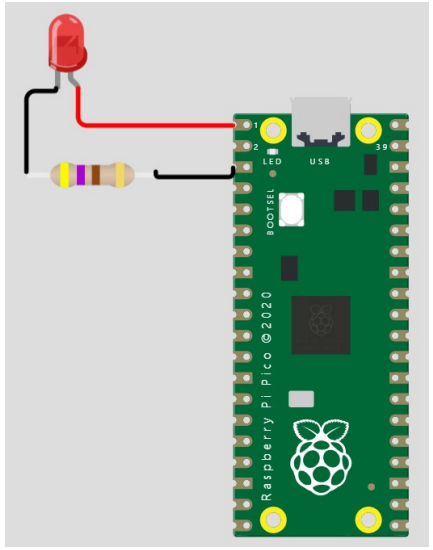
SOS Meaning

- However, meaning 'Save Our Ship' the phrase 'sos' is more widely known as another way of saying 'please help me now'. On a few occasions and depending on the lore behind each story it has also been known to mean 'Save Our Souls'.

Try to implement a Morse code system by flashing SOS with the help of Electronics Components

Flashing SOS in Morse





- ✓ An external LED flashes the SOS signal in **Morse code (three dots, followed by three dashes, followed by three dots)** continuously.
- ✓ In this experiment, a dot is represented with the LED being ON for **0.25 seconds** (**Dot time**) and a dash is represented with the LED being ON for **1 second** (**Dash time**).
- ✓ The delay between the dots and dashes is set to **0.2 second** (**GAP time**). This process is repeated continuously after **2 seconds of delay**.

LED FLASHING SOS

WOKWI SAVE SHARE Docs B

diagram.json ● main.py Library Manager

```
from machine import Pin
from utime import sleep

Dot = 0.25                # Dot time
Dash = 1.0                # Dash time
Gap = 0.2                 # Gap time
ON = 1                    # ON
OFF = 0                   # OFF
LED = Pin(0, Pin.OUT)     # LED at GP0

while True:               # DO FOREVER (INFINITE LOOP)
    for i in range(0, 3):
        LED.value(ON)     # LED ON
        sleep(Dot)        # Wait Dot time
        LED.value(OFF)    # LED OFF
        sleep(Gap)        # Wait Gap time

    sleep(0.5)            # 0.5 second delay

    for i in range(0, 3):
        LED.value(ON)     # LED ON
        sleep(Dash)       # Wait Dash time
        LED.value(OFF)    # LED OFF
        sleep(Gap)        # Wait Gap time
    sleep(2)              # Wait 2 seconds
```

Simulation