lamaPLC: Raspberryes

The RP2040 is a small, specially engineered silicon chip developed by Raspberry Pi to serve as the central controller for electronic devices.

The Raspberry Pi Pico is a compact green circuit board centered on the RP2040 chip, featuring a USB port, storage memory, and pin connections. This design allows for easy connection to a computer, enabling programming to control electronics such as sensors, lights, and motors.

Feature	RP2040 (The Chip)	Pi Pico (Standard)	Pi Pico H	Pi Pico W	Pi Pico WH	RP2040-Zero	RP2040-ETH
Manufacturer	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi	Waveshare	Waveshare
Dimensions	7 × 7 mm	51 × 21 mm	51 × 21 mm	51 × 21 mm	51 × 21 mm	23.5 × 18 mm	51 × 21 mm
Flash Storage	0 MB	2 MB	2 MB	2 MB	2 MB	2 MB	4 MB
USB Connector	None	Micro-USB	Micro-USB	Micro-USB	Micro-USB	USB Type-C	USB Type-C
Wireless (Wi-Fi/BT)	None	None	None	2.4GHz Wi-Fi & BT 5.2	2.4GHz Wi-Fi & BT 5.2	None	None
Wired Network	None	None	None	None	None	None	RJ45 Ethernet Port
Header Pins	None	Unsoldered (Bare pads)	Pre-soldered	Unsoldered (Bare pads)	Pre-soldered	Unsoldered (Castellated)	Unsoldered
Debug Connector	None	None	3-pin JST-SH	None	3-pin JST-SH	None	None
Onboard Light	None	Green LED	Green LED	Green LED	Green LED	RGB NeoPixel	None
Usable GPIO Pins	30	26	26	26	26	29	14
Logic Voltage (GPIO)	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V
Input Voltage (VBUS/VIN)	N/A (Requires exact 1.1V & 3.3V)	1.8V to 5.5V	1.8V to 5.5V	1.8V to 5.5V	1.8V to 5.5V	5.0V (via USB-C or 5V pin)	5.0V (via USB-C or 5V pin)
Software Support	C/C++, MicroPython, CircuitPython	C/C++, MicroPython, CircuitPython	C/C++, MicroPython, CircuitPython	C/C++, MicroPython, CircuitPython	C/C++, MicroPython, CircuitPython	C/C++, MicroPython, CircuitPython	C/C++, MicroPython (with special build)
Network Libraries	None	None	None	network (Wi-Fi), bluetooth	network (Wi-Fi), bluetooth	None	wiznet / custom CH9120 drivers
Primary Usage	Custom circuit design & commercial products	Learning code, DIY hobbies, basic automation	Prototyping without a soldering iron	Smart home, wireless IoT, web servers	Wireless IoT without a soldering iron	Tiny gadgets, wearable tech, macro pads	Hardwired network devices, industrial IoT

RP2040-Zero

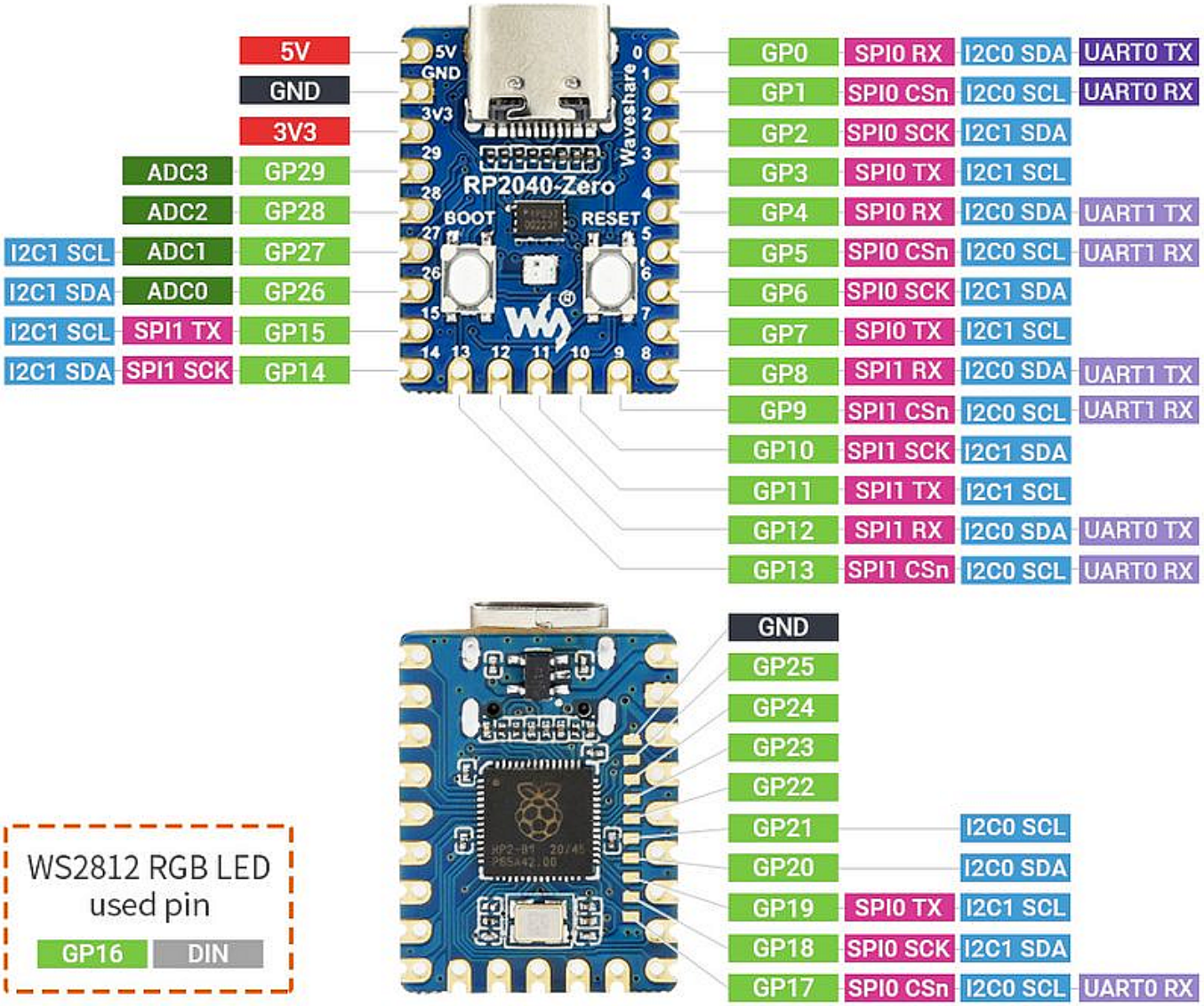
The RP2040-Zero is an ultra-compact, low-cost microcontroller development board designed by Waveshare. It is built around the Raspberry Pi RP2040 silicon chip, squeezing the processing power of a standard Raspberry Pi Pico into a tiny form factor roughly the size of a postage stamp.

More information from the board: https://www.waveshare.com/wiki/RP2040-Zero

RP2040-Zero Core Hardware Specifications

- **Processor:** Dual-core ARM Cortex-M0+ clocked at 133 MHz.
- **Memory:** 264 KB of internal SRAM and 2 MB of onboard Flash storage for code.
- **USB Interface:** Upgraded USB Type-C port for programming and power.
- **Onboard LED:** Features a single WS2812 RGB LED (NeoPixel) that can be programmed to any color.
- **Pins:** Breaks out 29 GPIO pins (20 via outer pin headers, and 9 via solder pads on the back).
- **Programmable I/O (PIO):** 8 PIO state machines for custom peripheral support
- **Built-in temperature sensor**, See: RP2040-Zero/Eth Built-in temperature sensor
- **Integrated control for WS2812 RGB LED:** See: See RP2040-Zero/Eth WS2812 RGB LED

RP2040-Zero Pinout



RP2040-Zero/Eth I²C communication

Primary Hardware I²C Pin Pairs on RP2040-Zero

The main pins for hardware I²C on the Waveshare RP2040-Zero are **GP0 (SDA) / GP1 (SCL)** or GP4 (SDA) / GP5 (SCL):

```
i2c = machine.I2C(0, scl=machine.Pin(1), sda=machine.Pin(0))
```

Since the RP2040 chip has a versatile digital network matrix (pin multiplexing), nearly any GPIO pin can be configured for I²C. Nevertheless, the typical hardware blocks (I²C_0 and I²C_1) are most straightforwardly accessed with the configurations below.

Hardware Block	SDA (Data)	SCL (Clock)	Location on RP2040-Zero
I²C_0 (Standard)	GP0	GP1	Top-left edge pins.
I²C_1	GP4	GP5	Left edge, mid-bottom pins.
I²C_1 (Alternative)	GP26	GP27	Bottom edge pins (Shared with ADC0/ADC1).

Key Electrical Requirements

- Pull-up Resistors: The RP2040-Zero does not include onboard pull-up resistors for the I²C lines. You must add external 4.7 kΩ resistors connected to 3.3V unless your peripheral module (sensor/display) already includes them.
- Voltage Limits: The RP2040 is strictly 3.3V logic compatible. Connecting a 5V I²C device directly to these pins without a logic level shifter can permanently damage the controller.

RP2040-Zero/Eth UART communication

To enable UART (serial communication) on the Waveshare RP2040-Zero/Eth, you can use its two dedicated hardware UARTs: UART0 and UART1. The RP2040's flexible pin multiplexing allows these UART peripherals to be assigned to various GPIO pins on your board.

Default Hardware UART Pin Mapping

While you can map UART to multiple pins, the standard layout for the RP2040-Zero relies on these primary pins:

Peripheral	TX Pin	RX Pin
UART0	GP0	GP1
UART1	GP4	GP5

Here is how to initialize and use UART0 at a standard baud rate of 9600 to send and receive data:

```
import machine
import utime

# Initialize UART0 on GP0 (TX) and GP1 (RX)
# Common baud rates: 9600, 115200
uart = machine.UART(0, baudrate=9600, tx=machine.Pin(0), rx=machine.Pin(1))

print("UART0 Initialized. Waiting for data...")

while True:
    # 1. Sending Data
    uart.write("Hello from RP2040-Zero!\n")

    # 2. Receiving Data (Check if any bytes are waiting in the buffer)
    if uart.any():
        # Read the incoming data bytes
        incoming_bytes = uart.read()

        # Convert bytes to a readable string
        incoming_text = incoming_bytes.decode('utf-8', errors='ignore')

        print(f"Received: {incoming_text.strip()}")

    utime.sleep(1)
```

Note: The RP2040 works only with 3.3V logic. When connecting it to a 5V device, such as an older Arduino Uno, always use a logic level shifter to protect the RP2040 pins.

RP2040-Zero/Eth Built-in temperature sensor

The Waveshare RP2040 features a built-in temperature sensor integrated directly inside the RP2040 microcontroller. It works by measuring the internal temperature of the silicon chip rather than the ambient room temperature.

Sensor Characteristics

- **ADC Channel:** Connected internally to ADC Channel 4.
- **Voltage at 27°C:** Usually outputs 0.706V.
- **Temperature Slope:** Decreases by 1.721mV per degree Celsius with rising temperature.

MicroPython Code Example

```
import machine
import utime

# Configure ADC channel 4 for the internal sensor
sensor_temp = machine.ADC(4)
conversion_factor = 3.3 / (65535)

while True:
    # Read the raw 16-bit ADC value
    reading = sensor_temp.read_u16() * conversion_factor

    # Calculate temperature in Celsius based on the RP2040 datasheet
    temperature = 27 - (reading - 0.706) / 0.001721

    print(f"Chip Temperature: {temperature:.2f} °C")
    utime.sleep(2)
```

RP2040-Zero/Eth WS2812 RGB LED

The WS2812 RGB LED is operated through a single data line connected to **pin GP16**, enabling programming of unique colors and blinking effects.

To control the onboard WS2812 RGB LED on the RP2040-Zero, use the neopixel library and connect it to GP16, which is the correct pin for this board:

```
import machine
import neopixel
import time
```

```
# The RP2040-Zero onboard RGB LED is connected to GP16
RGB_PIN = 16
NUM_PIXELS = 1

# Initialize the NeoPixel object
pixels = neopixel.NeoPixel(machine.Pin(RGB_PIN), NUM_PIXELS)

def set_color(r, g, b):
    """Sets the RGB LED color (values from 0 to 255)"""
    pixels[0] = (r, g, b)
    pixels.write()

# Cycle through Red, Green, Blue, and White
while True:
    set_color(255, 0, 0)    # Red
    time.sleep(1)

    set_color(0, 255, 0)    # Green
    time.sleep(1)

    set_color(0, 0, 255)    # Blue
    time.sleep(1)

    set_color(50, 50, 50)   # Dim White
    time.sleep(1)
```

RP2040-Eth

The RP2040-eth is an ultra-compact, low-cost microcontroller development board designed by Waveshare. It is built around the Raspberry Pi RP2040 silicon chip, squeezing the processing power of a standard Raspberry Pi Pico into a tiny form factor roughly the size of a postage stamp.

More information from the board: https://www.waveshare.com/rp2040-eth.htm

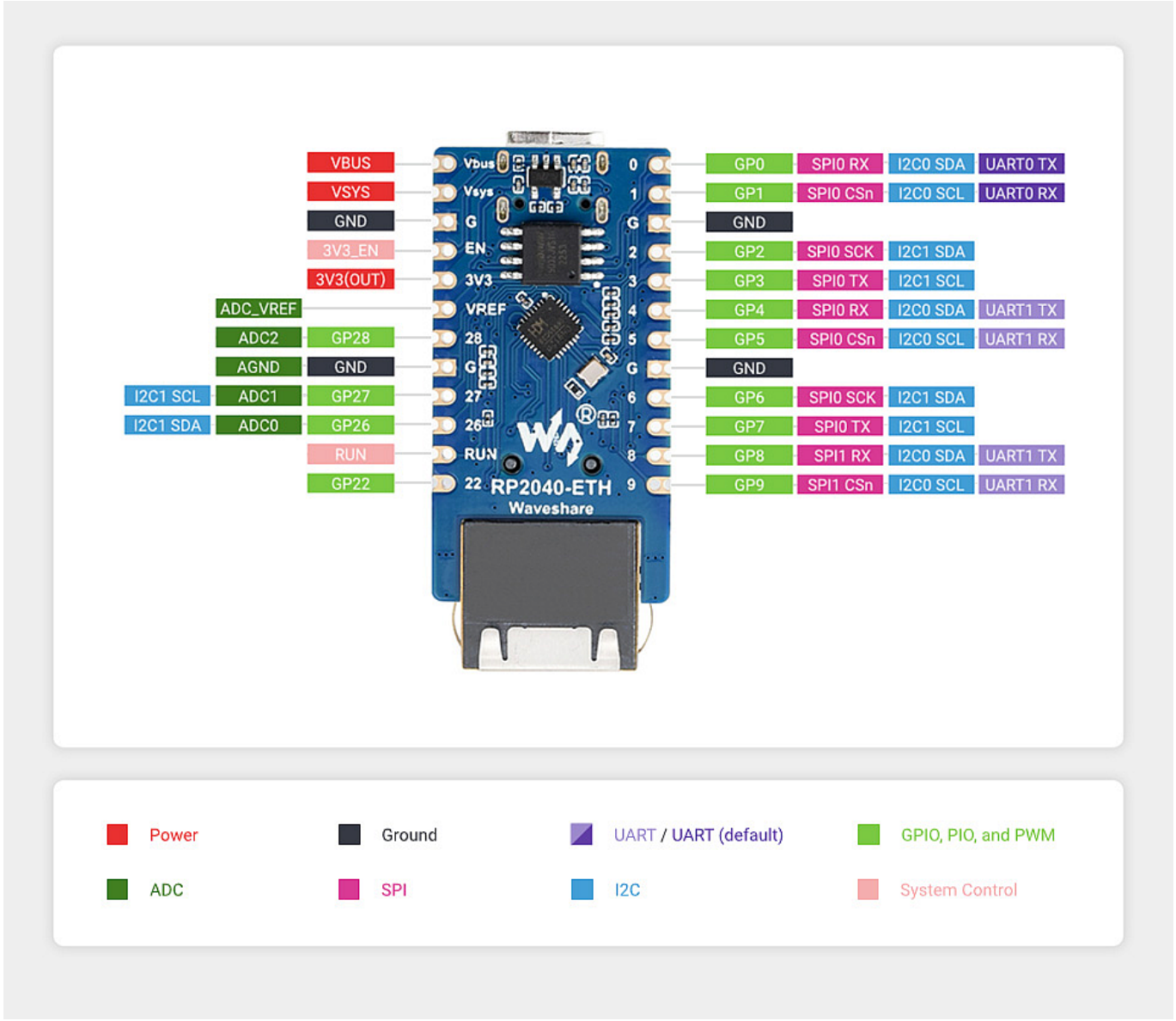
RP2040-Eth Core Hardware Specifications

- **Processor:** Dual-core ARM Cortex-M0+ clocked at 133 MHz.
- **Memory:** 264 KB of internal SRAM and 2 MB of onboard Flash storage for code.
- **USB Interface:** Upgraded USB Type-C port for programming and power.
- **Onboard LED:** Features a single WS2812 RGB LED (NeoPixel) that can be programmed to any color.
- **Pins:** Breaks out 29 GPIO pins (20 via outer pin headers, and 9 via solder pads on the back).
- **Programmable I/O (PIO):** 8 PIO state machines for custom peripheral support
- **Built-in temperature sensor**, See: RP2040-Zero/Eth Built-in temperature sensor
- **Integrated control for WS2812 RGB LED:** See: See RP2040-Zero/Eth WS2812 RGB LED

RP2040-Eth Features

- **High-performance Processing:** The dual-core ARM Cortex MO+ processor, running at up to 133MHz, delivers robust processing power for a range of applications.
- **Ample Memory:** Equipped with 264KB of SRAM and 4MB of onboard Flash, providing enough space for data storage and program execution. Integrated TCP/IP Protocol Stack: The CH9120's integration facilitates seamless networking, allowing straightforward internet communication.
- **Versatile GPIO:** The board provides 14 GPIO pins, enabling users to connect and manage external devices, which enhances its flexibility for various project needs.
- **Simple Programming and Debugging:** When connected through USB, the board appears as a high-capacity storage device, making program transfer easier and debugging more straightforward.
- **More hardware peripherals:**
  - 1x SPI (SPI0)
  - 2x I²C
  - 2x UART
- 13x Controllable PWM channels
- Built-in temperature sensor.
- 8 × Programmable I/O (PIO) state machines for custom peripheral support.

RP2040-eth Pinout



RP2040-Eth I²C communication

See: RP2040-Zero/Eth I²C communication

RP2040-Eth UART communication

See: RP2040-Zero/Eth UART communication

RP2040-Eth Ethernet communication

The RP2040 communicates internally with the CH9120 Ethernet chip using a dedicated UART serial interface and control pins:

CH9120 Pin	RP2040 Pin	Function
RXD	GP21	Serial Data Input
TXD	GP20	Serial Data Output

CH9120 Pin	RP2040 Pin	Function
TCPCS	GP17	TCP Client Connection Status
CFG0	GP18	Serial Debug / Configuration Enable
RSTI	GP19	Hardware Reset (Active Low)

The integrated CH9120 chip supports four selectable communication protocols:

- TCP Client
- TCP Server
- UDP Client
- UDP Server

Network Configuration

The code for configuring the operating mode, IP, gateway, subnet mask, port number, and serial port baud rate is located in the Python/RP2040-ETH-Demo/RP2040-ETH-Demo.py file. You can modify it according to your specific requirements.

```
MODE = 1  #0:TCP Server 1:TCP Client 2:UDP Server 3:UDP Client
GATEWAY = (192, 168, 1, 1)    # GATEWAY
TARGET_IP = (192, 168, 1, 10) # TARGET_IP
LOCAL_IP = (192, 168, 1, 200) # LOCAL_IP
SUBNET_MASK = (255,255,255,0) # SUBNET_MASK
LOCAL_PORT1 = 1000            # LOCAL_PORT1
TARGET_PORT = 2000            # TARGET_PORT
BAUD_RATE = 115200            # BAUD_RATE
```

Raspberry topics on lamaPLC

Page	Date	Tags
• lamaPLC: Raspberryes	2026/07/07 19:15	raspberry, rp2040, rp2350, rp2040 zero, rp2040 eth, pi pico h, pi pico w, pi pico wh, raspberry pi pico, pi pico 2, i2c, microcontroller

raspberry, RP2040, RP2040 zero, RP2040 eth, Pi Pico H, Pi Pico W, Pi Pico WH, raspberry Pi Pico, i2c, microcontroller

This page has been accessed for: Today: 1, Until now: 67