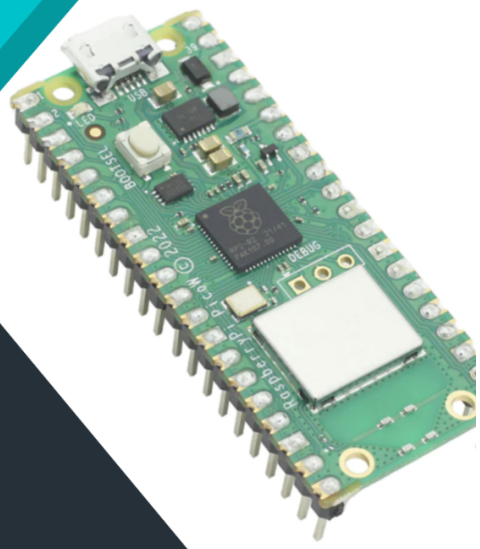


PICO W
STUDENT KIT

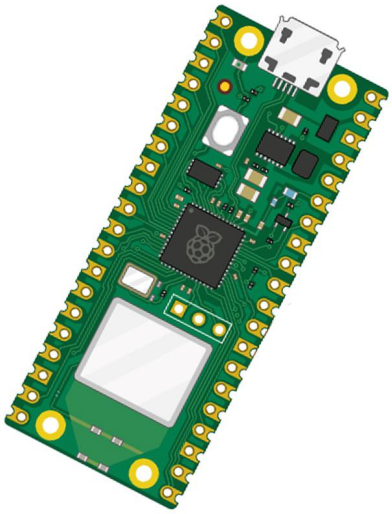
CircuitPython for Beginner

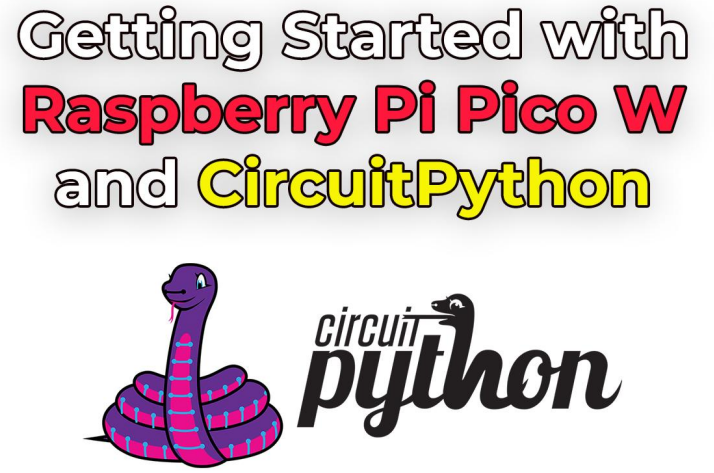
Cytron Technologies

Getting Started with Raspberry Pi Pico W and CircuitPython

This Student Kit is perfect for beginners, even if you have no prior knowledge of **Raspberry Pi Pico W** or **CircuitPython**. To get started, simply unbox your Pico W and follow the getting started video below. This video will also cover **Lesson 1: Blink Built-in LED**, which is included in this manual guide. Let's dive in and start learning!

Getting Started with Raspberry Pi Pico W and CircuitPython



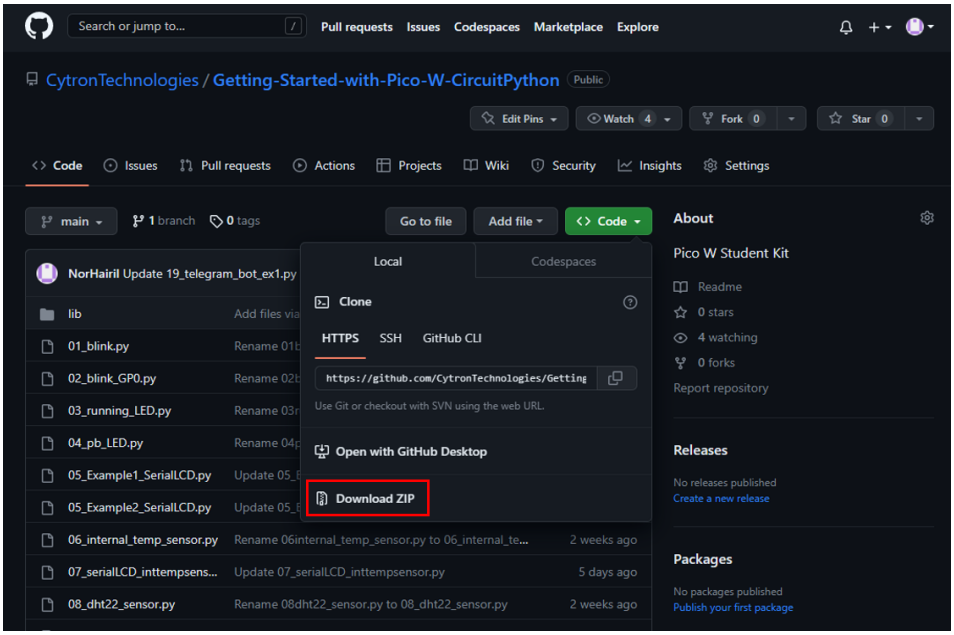


You may also refer to the **written tutorial** in the description of the video.

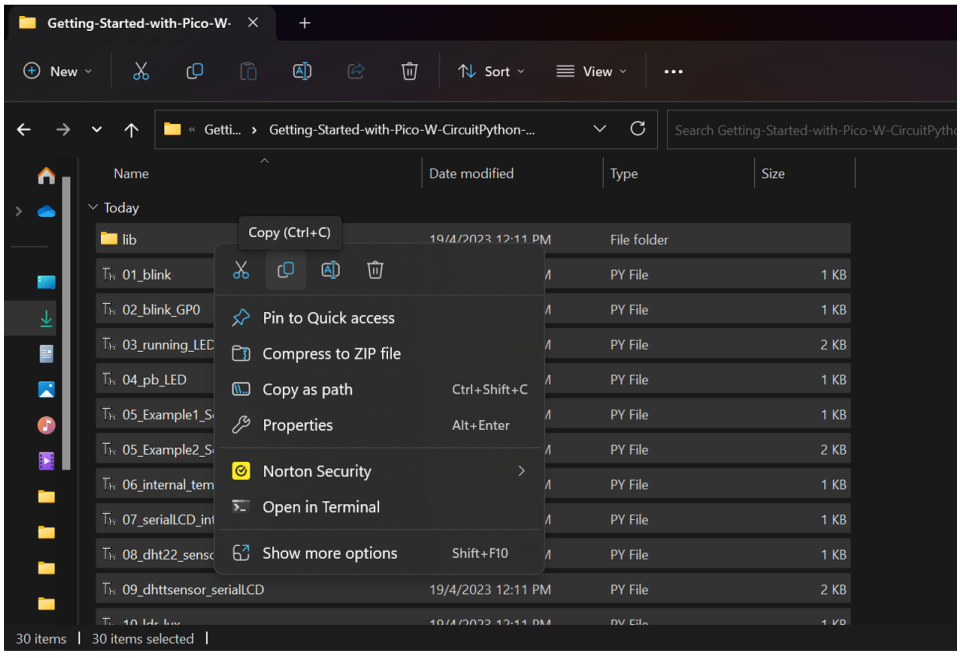
GitHub Page

All the codes and libraries for each lesson are stored in GitHub. Kindly visit our GitHub page and **download the ZIP file** for this kit.

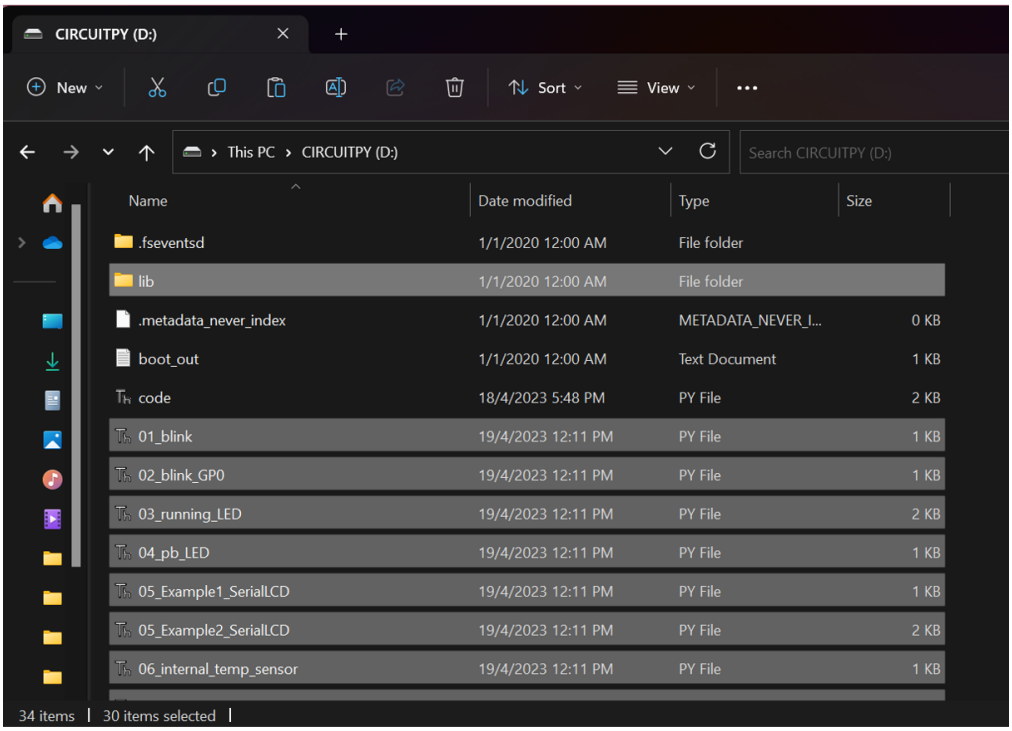
- [GitHub - CytronTechnologies/Getting-Started-with-Pico-W-CircuitPython]



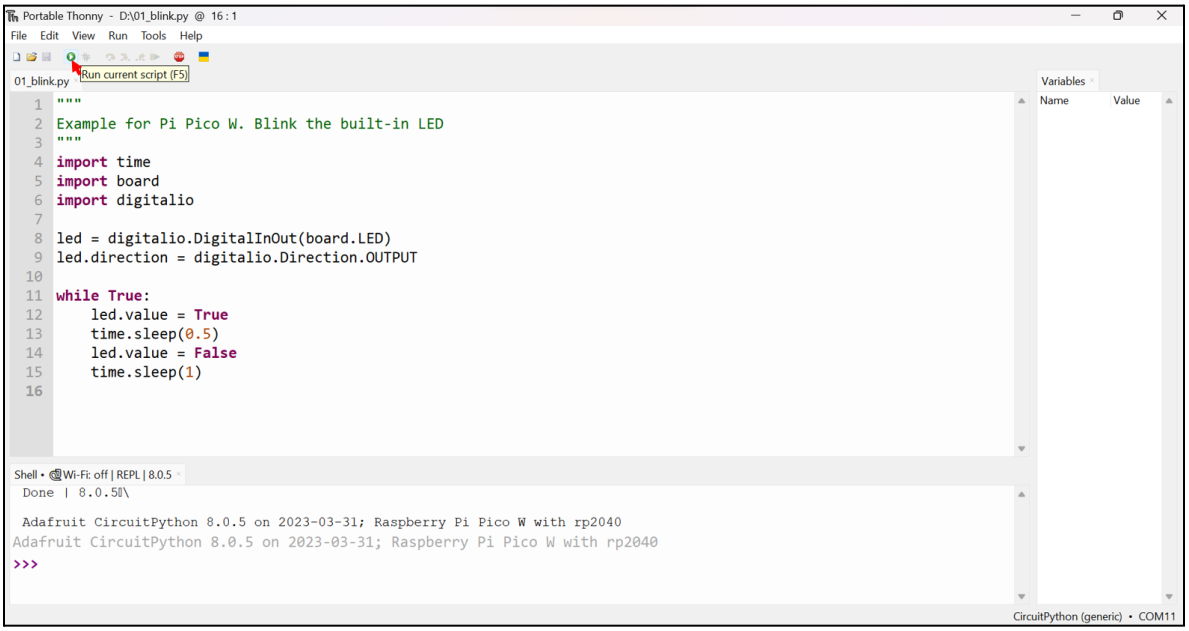
Extract the ZIP file and **copy** all the files in the Getting-Started-with-Pico-W-CircuitPython folder. There should be **30 items** selected in that folder.



Then, paste all the files into the CIRCUITPY drive. You should see something like this.



Now, you can directly open any code using the Thonny IDE by following all 24 lessons provided.



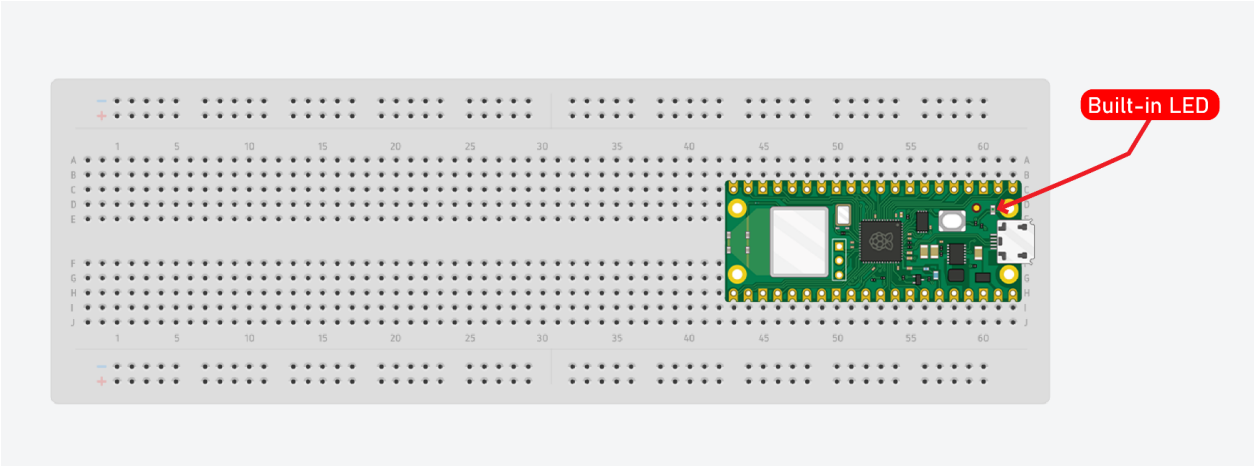
Lesson 1: Blink Built-in LED

In this lesson, you will learn how to blink the built-in LED of Raspberry Pi Pico W with CircuitPython. Blinking an LED is the most fundamental operation that every beginner should know. It is a simple way to test whether your board and code work properly. By the end of this lesson, you will be able to write a basic code to control the LED and make it blink.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard

Circuit Diagram



Code

- 01_blink.py

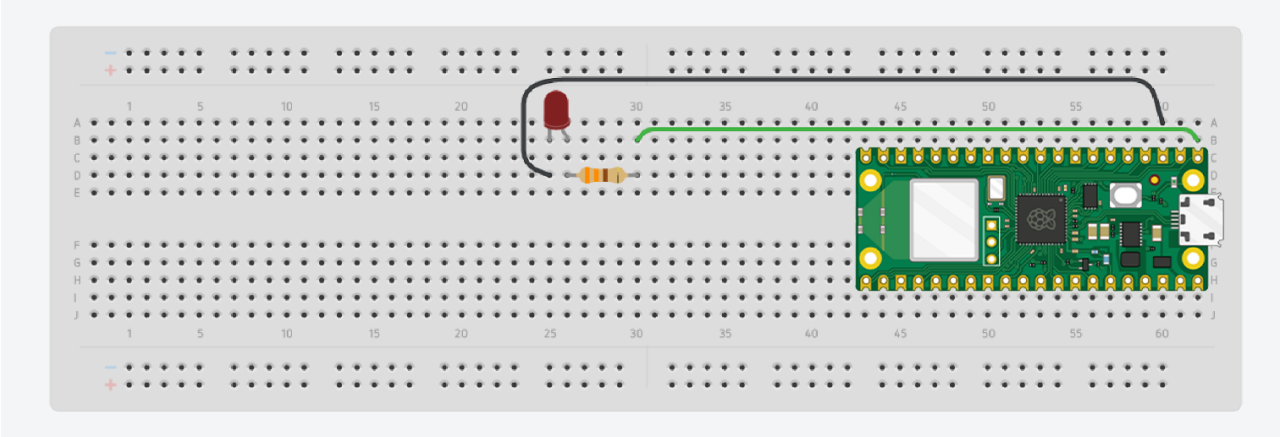
Lesson 2: Blink An External LED

In this lesson, you will learn how to blink an external LED using Raspberry Pi Pico W. You will be introduced to the GPIO pins of the board, which allow you to control external components. By the end of this lesson, you will be able to connect an LED to the GPIO pins and write code to make it blink.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- 1 x LED
- 1 x 330Ω Resistor

Circuit Diagram



Code

- 02_blink_GP0.py

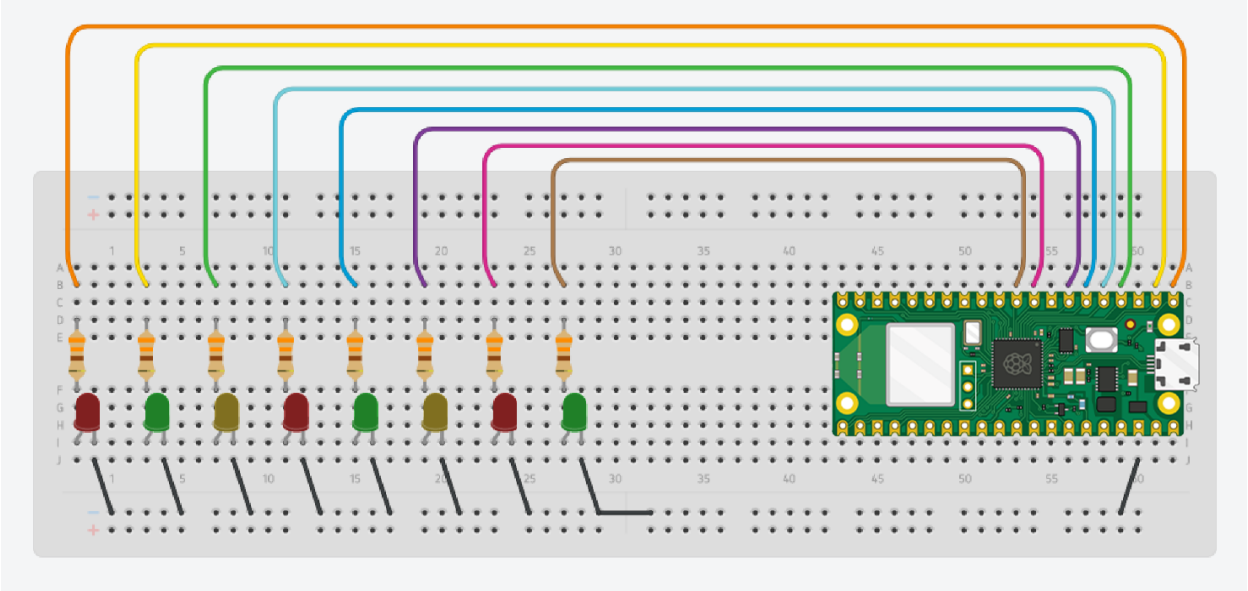
Lesson 3: Running & Blinking LEDs

In this lesson, you will learn how to control multiple LEDs using Raspberry Pi Pico W. You will learn how to make them run and blink in different patterns. By the end of this lesson, you will be able to write a code to control multiple LEDs and make them run and blink in various patterns. This will help you to create more complex LED projects in the future.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- 8 x LEDs
- 8 x 330Ω Resistor

Circuit Diagram



Code

- 03_running_LED.py

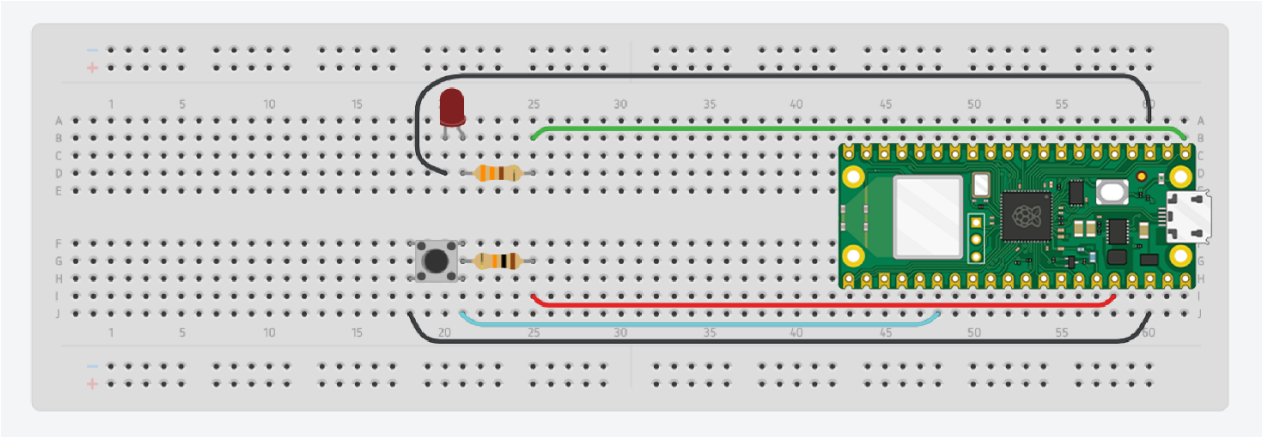
Lesson 4: Active Low Push Button

In this lesson, you will learn how to create communication between two different electronic components using Raspberry Pi Pico W. You will use an active low-push button to control the blinking of an LED. When the button is pressed, the LED will turn on, and when it is released, the LED will turn off. By the end of this lesson, you will be able to read the state of a push button and control the LED accordingly.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- 1 x LED
- 1 x Push Button
- 1 x 330Ω Resistor
- 1 x 10kΩ Resistor

Circuit Diagram



Code

- 04_pb_LED.py

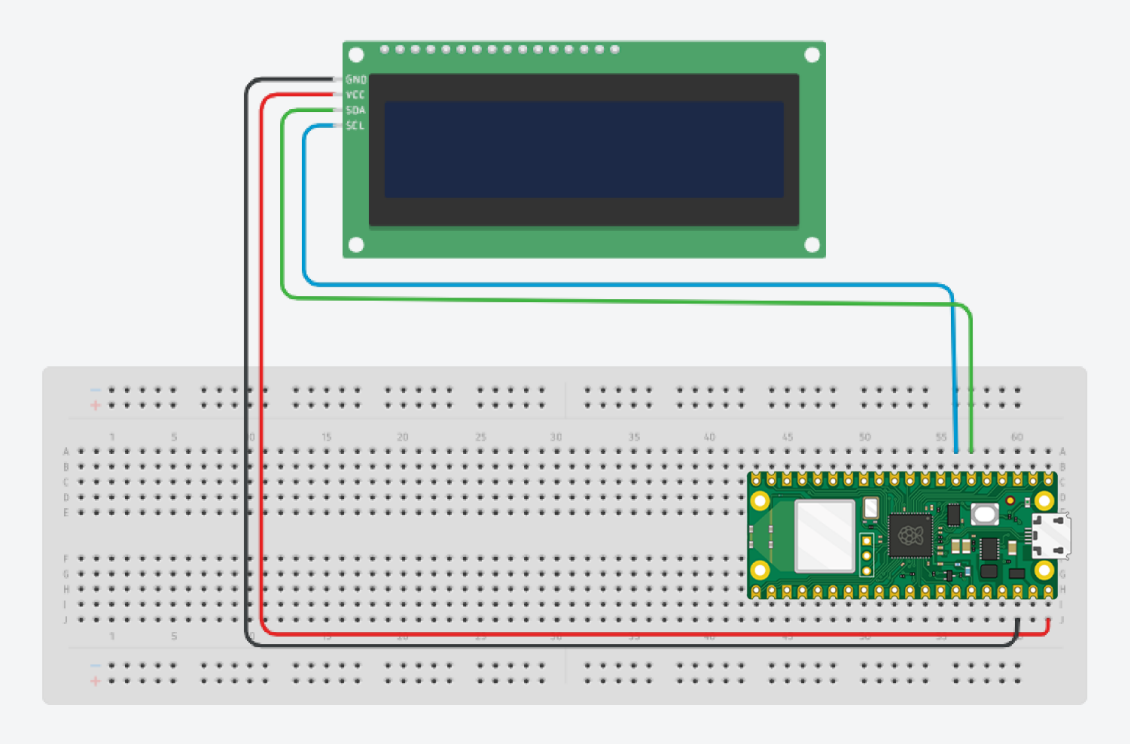
Lesson 5: Serial LCD

In this lesson, you will learn how to use an LCD with Raspberry Pi Pico W. You will be introduced to the concept of serial communication, which is used to transmit data between the board and the LCD. You will learn how to display text on the LCD using the first example, and how to shift the text to the right and left using the second example. By the end of this lesson, you will be able to interface an LCD with Raspberry Pico W and display text on it.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- 1 x LCD

Circuit Diagram



Code

- 05_Example1_SerialLCD.py
- 05_Example2_SerialLCD.py

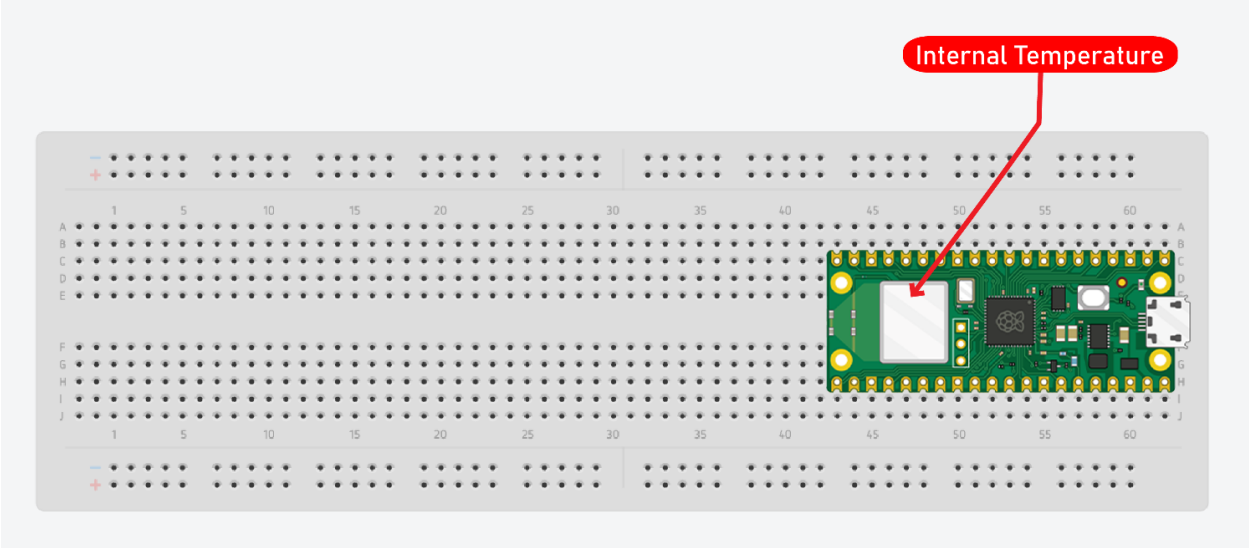
Lesson 6: Internal Temperature Sensor

In this lesson, you will learn how to read the internal temperature sensor of Raspberry Pi Pico W using CircuitPython. The board's CPU has a temperature sensor built into it, which allows you to monitor the temperature of the board. You will use the **microcontroller module** to read the data from the sensor and print it to the console.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard

Circuit Diagram



Code

- 06_internal_temp_sensor.py

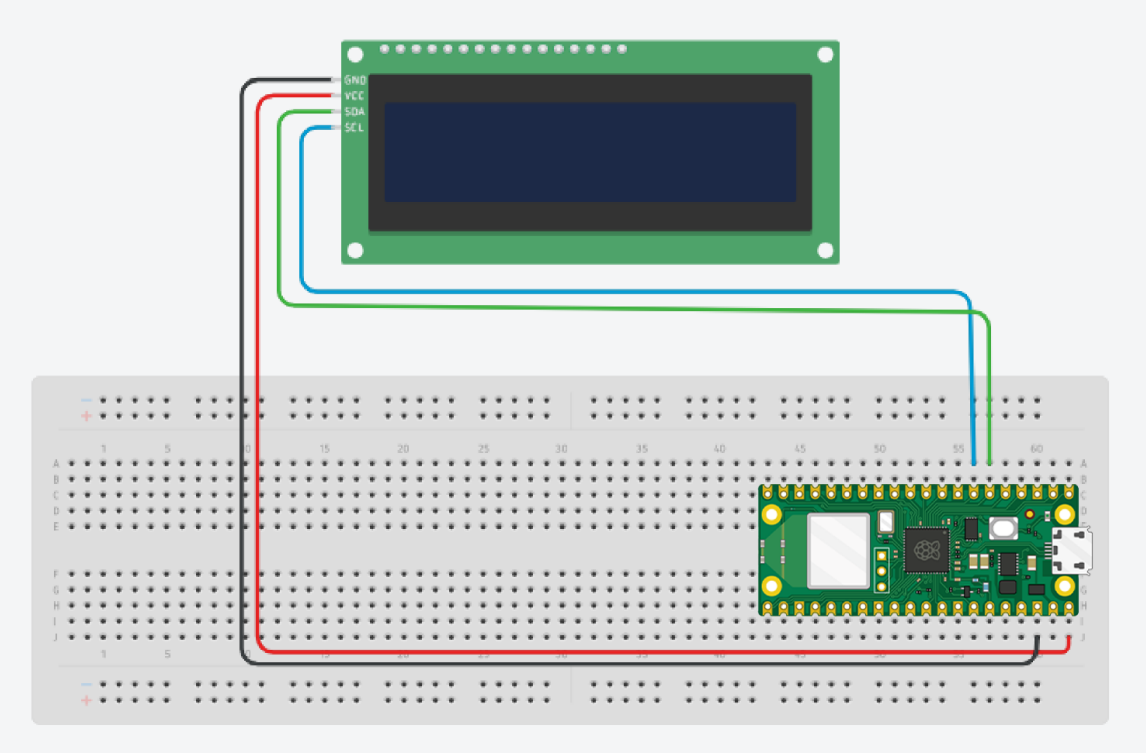
Lesson 7: Internal Temperature Sensor & Serial LCD

In this lesson, you will learn how to display the temperature data on an LCD display using Raspberry Pi Pico W. By the end of this lesson, you will be able to interface an LCD display with Pico W and display the internal temperature data on it.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- 1 x LCD

Circuit Diagram



Code

- 07_serialLCD_inttempsensor.py

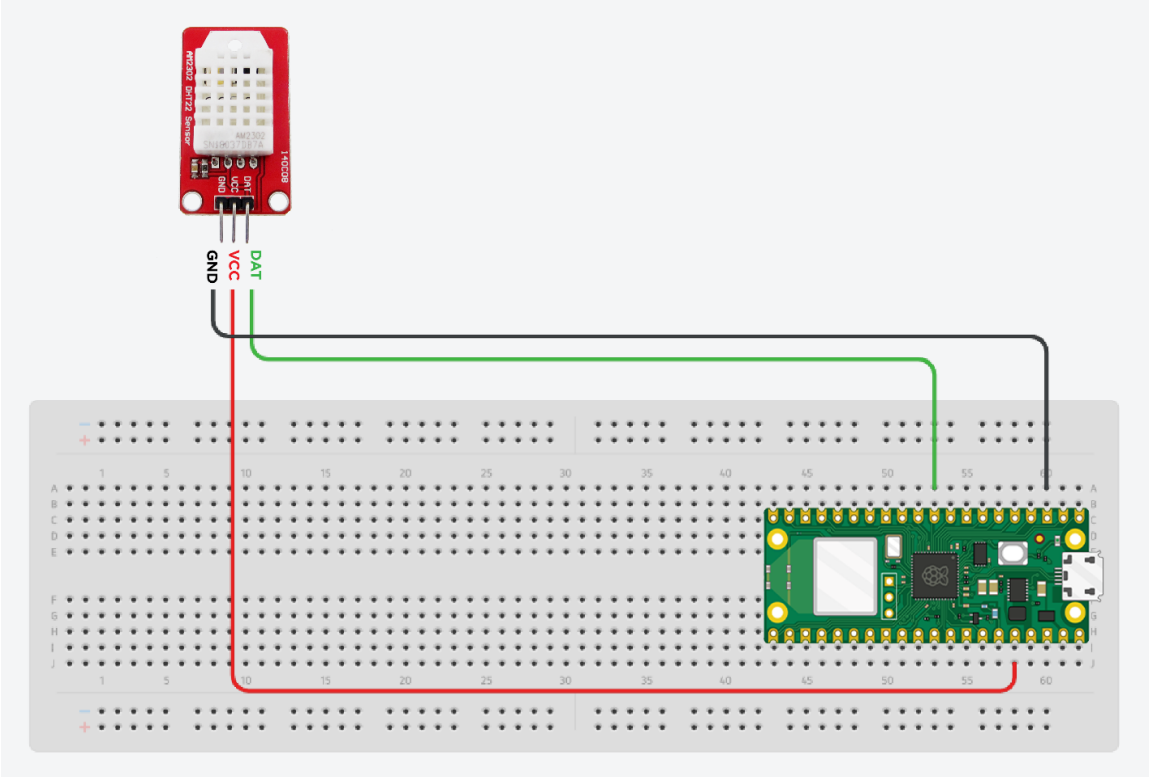
Lesson 8: DHT22 Sensor

In this lesson, you will learn how to read temperature and humidity data using the DHT22 sensor and Raspberry Pi Pico W. The DHT22 sensor is a popular sensor that can measure both temperature and humidity with high accuracy. You will use the Adafruit CircuitPython library to read the data from the sensor and print it to the console. By the end of this lesson, you will be able to read temperature and humidity data using the DHT22 sensor and use it in your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- DHT22

Circuit Diagram



Code

- 08_dht22_sensor.py

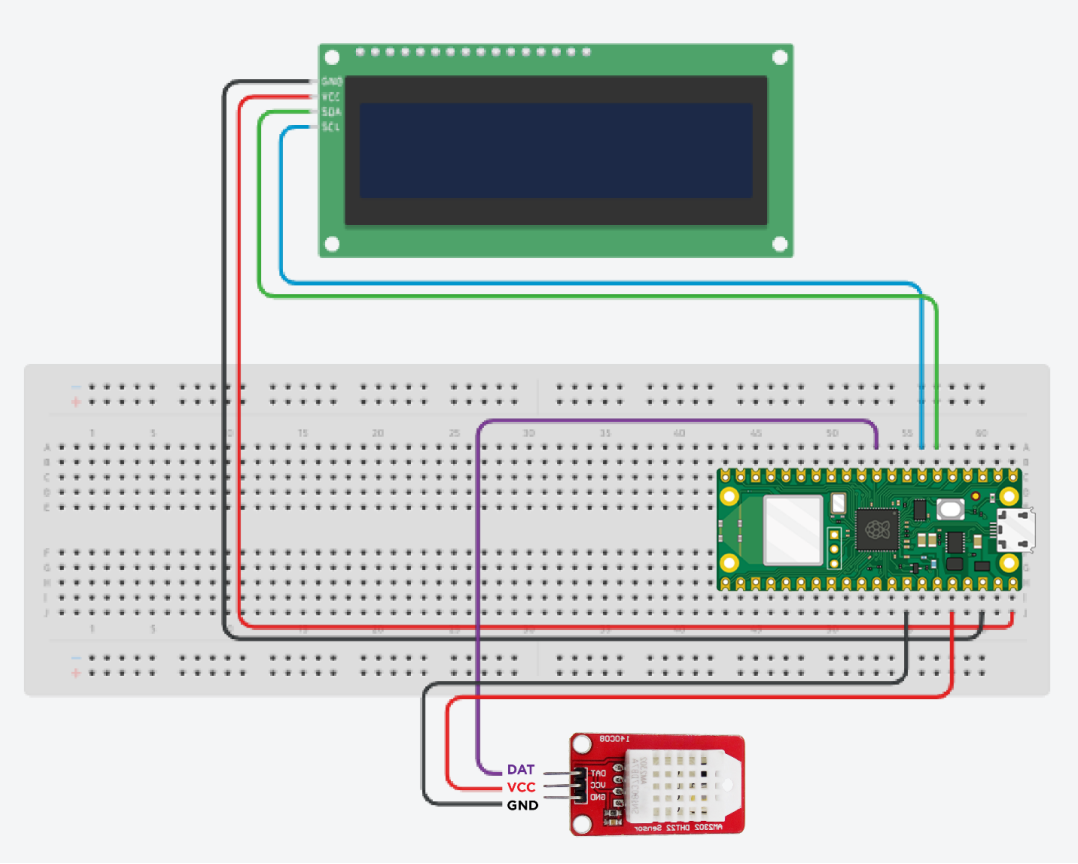
Lesson 9: DHT22 Sensor & Serial LCD

In this lesson, you will learn how to display temperature and humidity data on an LCD screen using the DHT22 sensor and Raspberry Pi Pico W. You will use the Adafruit CircuitPython library to read the data from the sensor and display it on the I2C LCD screen. You will also need two libraries for the I2C LCD to work. By the end of this lesson, you will be able to interface an I2C LCD screen with Raspberry Pico W and display temperature and humidity data on it.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- LCD
- DHT22

Circuit Diagram



Code

- 09_dhttsensor_serialLCD.py

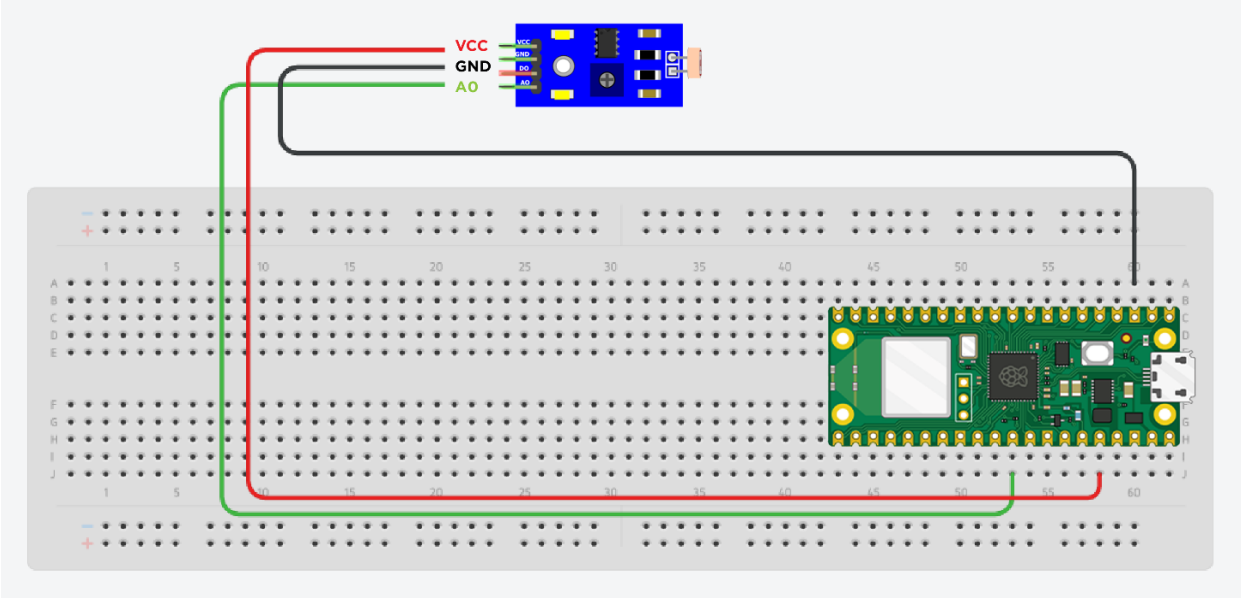
Lesson 10: Light Dependent Resistor (LDR)

In this lesson, you will learn how to collect light intensity data using a Light Dependent Resistor (LDR) and Raspberry Pi Pico W. The LDR is a sensor that changes its resistance based on the amount of light it detects. You will use the ADC (Analog to Digital Converter) of the board to read the data from the LDR and convert it to a Lux value. By the end of this lesson, you will be able to collect light intensity data using an LDR and convert the raw data to a Lux value.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- LDR

Circuit Diagram



Code

- 10_ldr_lux.py

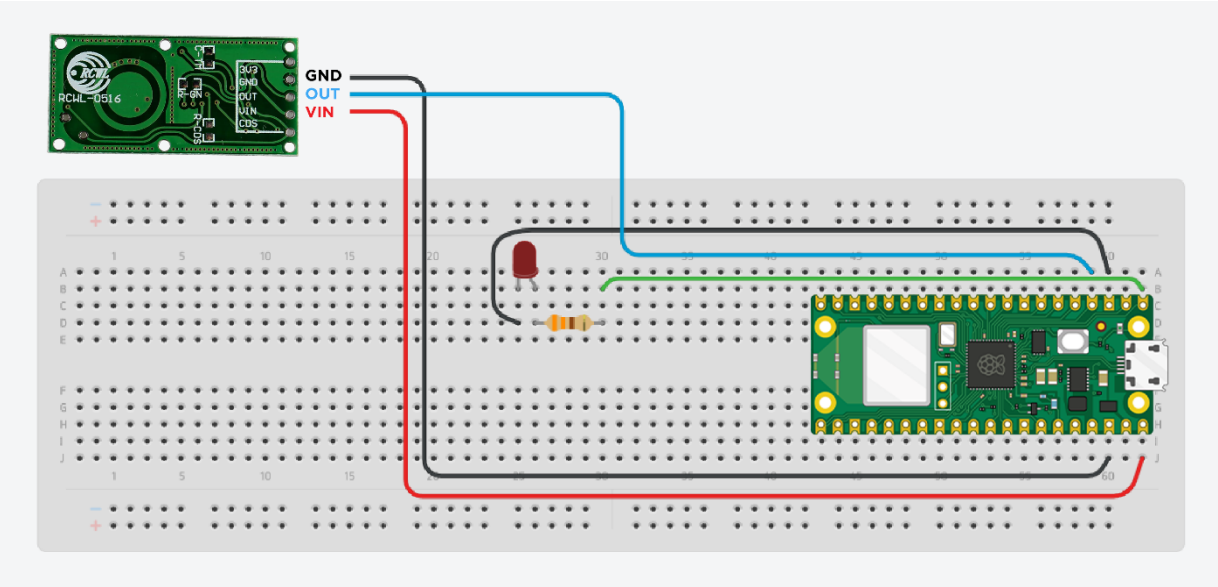
Lesson 11: Microwave Radar

In this lesson, you will learn how to use a Microwave Radar sensor with Raspberry Pi Pico W. The Microwave Radar sensor is a motion detection sensor that can detect human body motion. You will use the GPIO pins of the board to connect the sensor and control an external LED. When the sensor detects human body motion, the LED will blink. By the end of this lesson, you will be able to use the Microwave Radar sensor to detect motion and give output through the LED.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- RCWL-0516 Radar Sensor
- LED
- 1 x 330Ω Resistor

Circuit Diagram



Code

- 11_microwave_radar.py

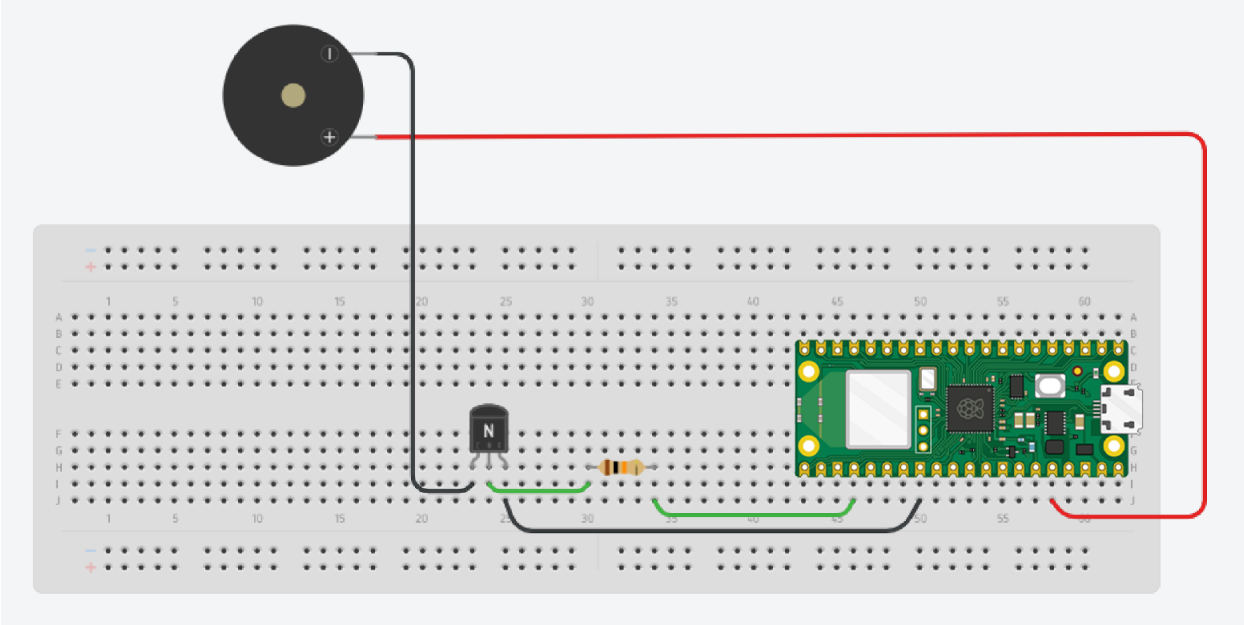
Lesson 12: Buzzer

In this lesson, you will learn how to use a piezo buzzer with Raspberry Pi Pico W. A piezo buzzer is a simple device that can generate sound by vibrating a piezoelectric crystal. You will use the GPIO pins of the board to connect the buzzer and generate different tones to create a simple song. By the end of this lesson, you will be able to use a piezo buzzer to generate sound and create simple melodies using Raspberry Pico W.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Buzzer
- 1 x 2N2222 Transistor
- 1 x 10kΩ Resistor

Circuit Diagram



Code

- 12_Buzzer.py

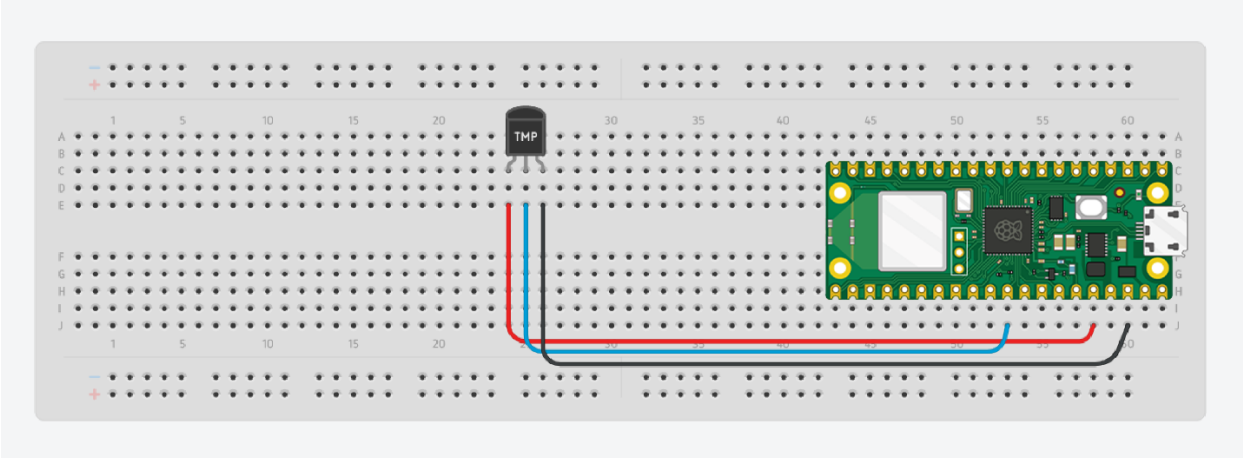
Lesson 13: MCP9700 Temperature Sensor

In this lesson, you will learn how to collect temperature data using an MCP9700 sensor and Raspberry Pi Pico W. The MCP9700 sensor is a temperature sensor that generates a voltage proportional to the temperature. You will use the ADC (Analog to Digital Converter) of the board to read the data from the sensor and convert it to a temperature value. Unlike the DHT22 sensor, the MCP9700 does not require any library to get the temperature data. By the end of this lesson, you will be able to collect temperature data using an MCP9700 sensor.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- MCP9700

Circuit Diagram



Code

- 13_mcp9700.py

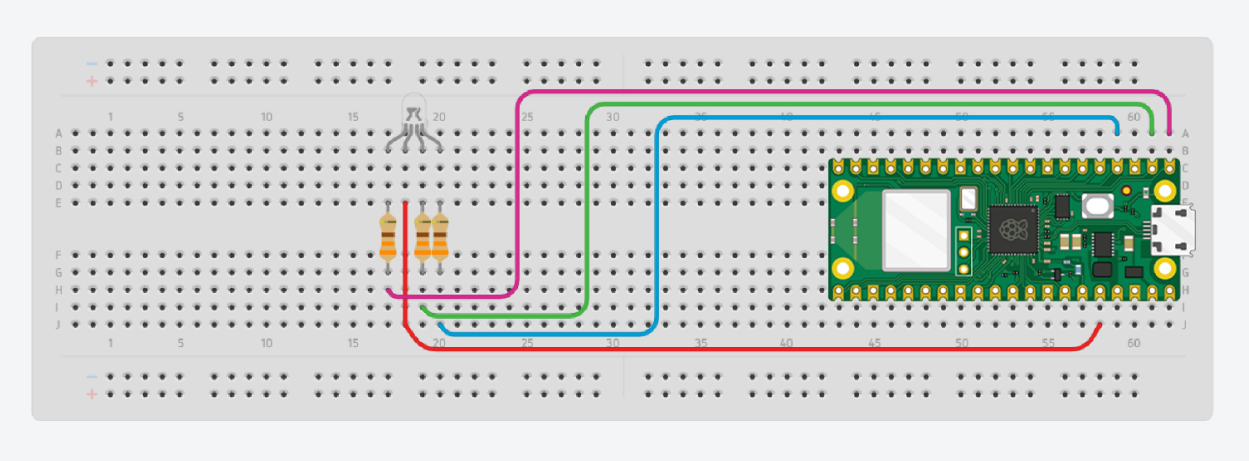
Lesson 14: RGB LEDs

In this lesson, you will learn how to use an RGB LED with Raspberry Pi Pico W. An RGB LED is a type of LED that can produce different colors by mixing red, green, and blue light. You will use the GPIO pins of the board to control the individual colors of the LED and create different colors. By the end of this lesson, you will be able to use an RGB LED and control it with three different GPIO pins using Raspberry Pico W.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- 1 x Common Anode RGB LED
- 3 x 330Ω Resistor

Circuit Diagram



Code

- 14_rgb_leds.py
- 14_rgb_leds_v2.py

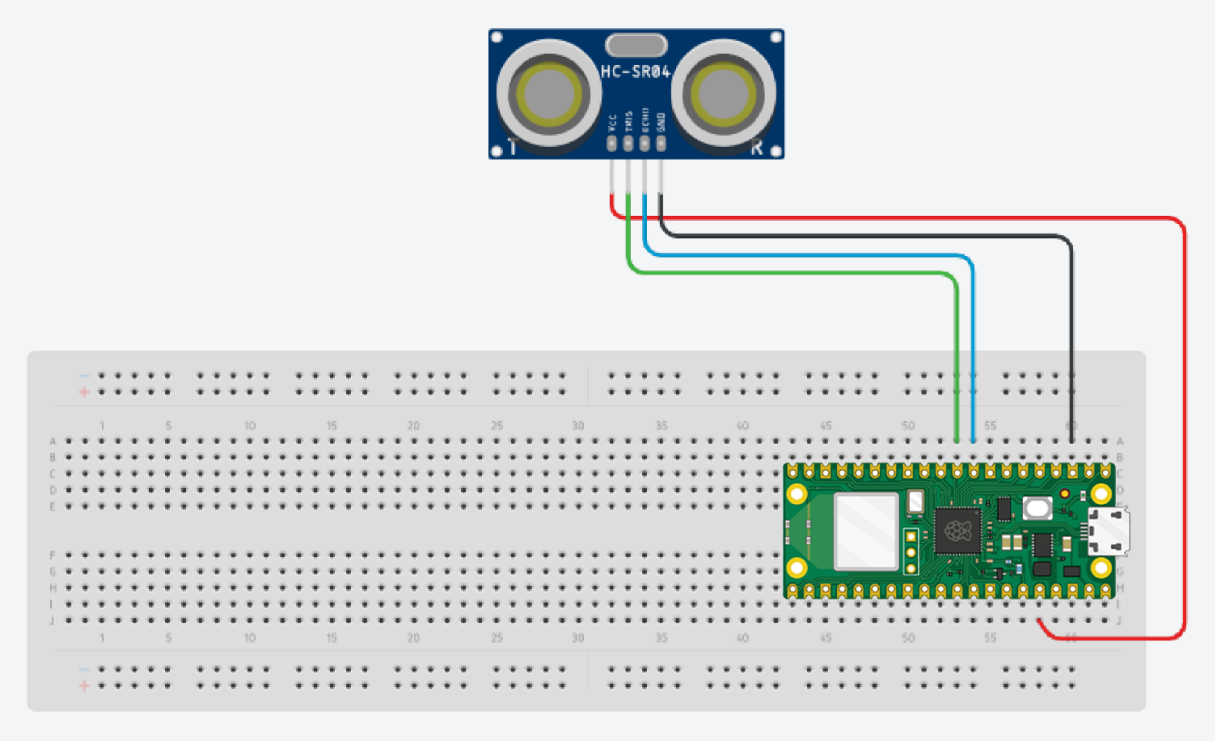
Lesson 15: Ultrasonic Sensor

In this lesson, you will learn how to use an ultrasonic sensor with Raspberry Pi Pico W to measure distance. An ultrasonic sensor sends out sound waves and measures the time it takes for the waves to bounce back, which can be used to calculate distance. You will use the GPIO pins of the board to trigger the sensor and receive the echo signal, and use this information to calculate the distance. By the end of this lesson, you will be able to use an ultrasonic sensor with Raspberry Pico W to measure distance.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- HC-SR04 Ultrasonic Sensor

Circuit Diagram



Code

- 15_ultrasonic_hcsr04.py

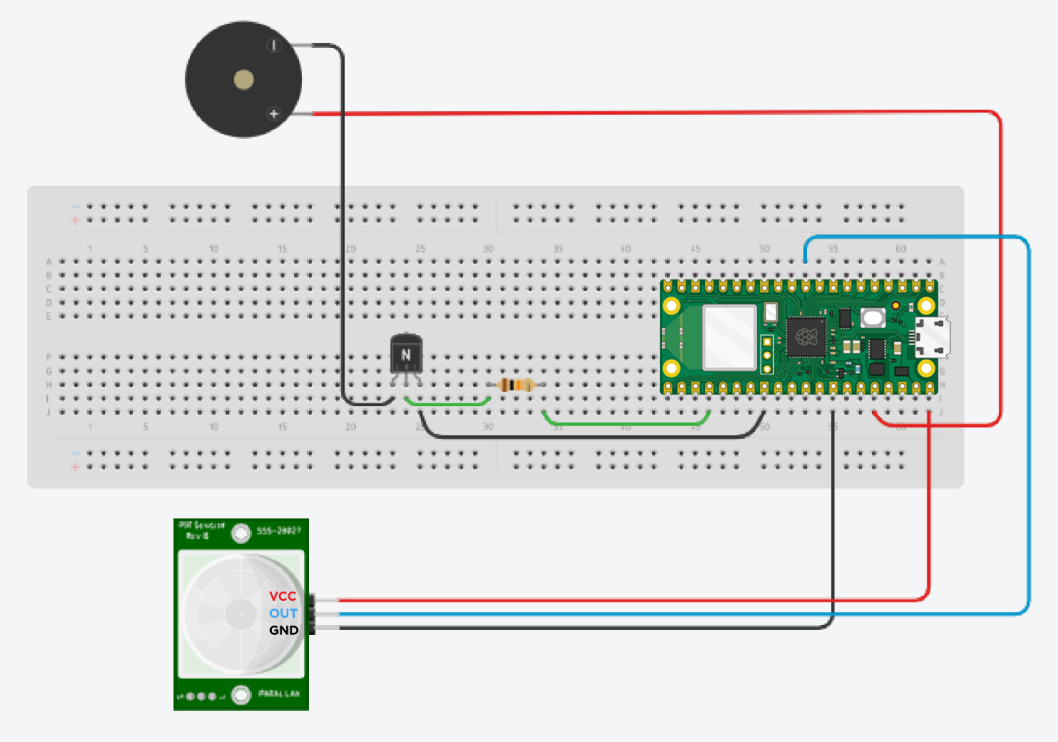
Lesson 16: PIR Sensor Module

In this lesson, you will learn how to use a PIR (Passive Infrared) sensor module with Raspberry Pi Pico W to detect motion. A PIR sensor measures infrared light radiating from objects in its field of view and can detect changes in the amount of infrared radiation, which can be used to detect motion. You will use the GPIO pins of the board to read the data from the sensor and trigger a buzzer when motion is detected. By the end of this lesson, you will be able to use a PIR sensor with Raspberry Pico W to detect motion and trigger an action in your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Buzzer
- PIR Sensor
- 1 x 2N2222 Transistor
- 1 x Resistor 10kΩ Resistor

Circuit Diagram



Code

- 16_PIRsensor.py

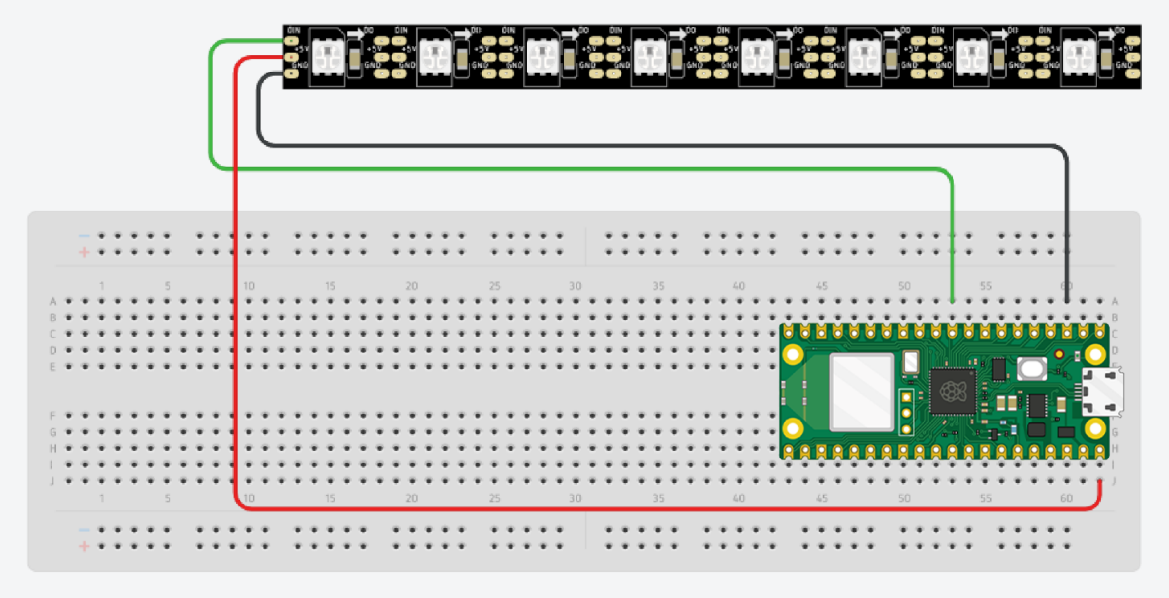
Lesson 17: Neopixel RGB LED Stick

In this lesson, you will learn how to use a Neopixel RGB LED stick with Raspberry Pi Pico W. Neopixels are a type of RGB LED that can be individually controlled and daisy-chained together, allowing for a wide range of lighting effects. You will use the GPIO pins of the board to control the Neopixel stick and experiment with different lighting patterns and colors. By the end of this lesson, you will be able to use a Neopixel RGB LED stick with Raspberry Pico W to create custom lighting effects in your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Neopixel RGB LED Stick

Circuit Diagram



Code

- 17_neopixel_ex1.py

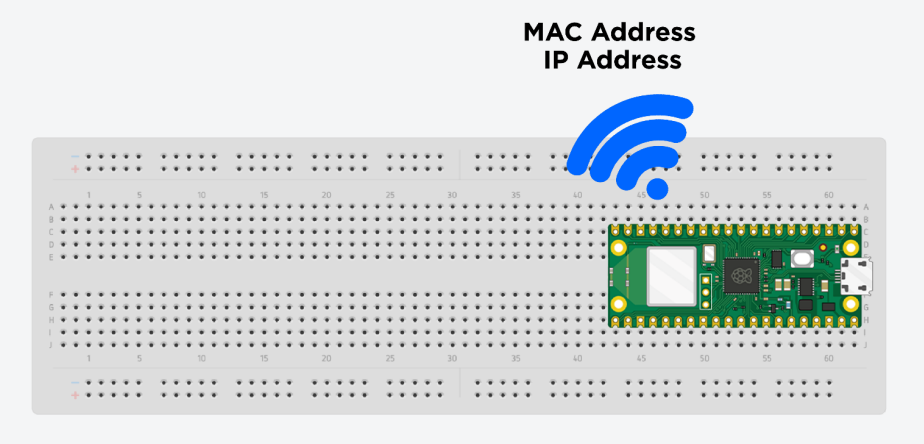
Lesson 18: WiFi Connect

In this lesson, you will learn how to connect Raspberry Pi Pico W to a WiFi network and obtain the IP and MAC address of the board. WiFi connectivity is a key feature of IoT projects, and by connecting your board to a network, you can enable remote control and monitoring of your projects. You will use the built-in WiFi module of Raspberry Pico W and connect to a local WiFi network based on **settings.toml** file. This file is used to setup WiFi credentials and IoT projects authentication key, so you need to update it. By the end of this lesson, you will be able to connect Raspberry Pico W to a WiFi network and obtain the IP address and MAC address.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard

Circuit Diagram



Code

- 18_wifi_connect.py

Attention: Please update your **settings.toml** file in the **CIRCUITPY** drive. Put in **YOUR WiFi SSID** and **password**.

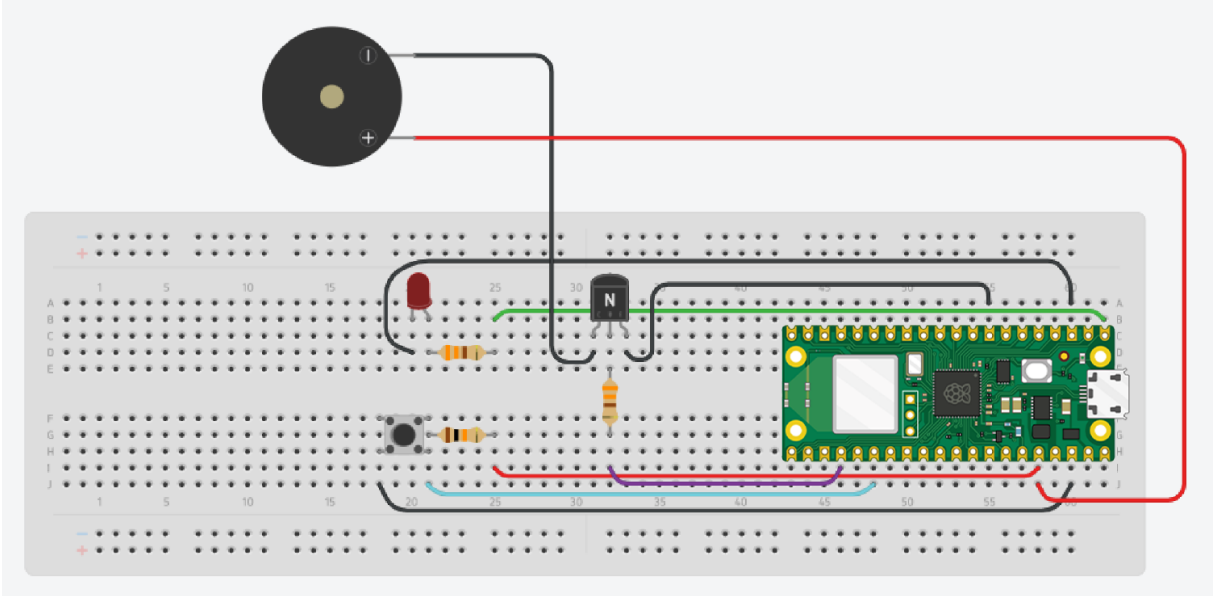
Lesson 19: Telegram Bot

In this lesson, you will learn how to use a Telegram bot to collect data and control the LED connected to Raspberry Pi Pico W. Telegram is a popular messaging platform that can be used to remotely control and monitor IoT projects. You will use CircuitPython libraries to create a simple bot that can receive commands and control the state of the LEDs on your board. By the end of this lesson, you will be able to create a Telegram bot and use it to collect data and control the LEDs connected to Raspberry Pico W, opening up a wide range of possibilities for remote control and monitoring of your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Buzzer
- Push Button
- 1 x LED
- 1 x 2N2222 Transistor
- 1 x 10kΩ Resistor
- 2 x 330Ω Resistor

Circuit Diagram



Code

- 19_telegram_bot_ex1.py

Attention:

- Please update the **botToken** in settings.toml file.
- You can refer here on how to generate a botToken: https://www.siteguarding.com/en/how-to-get-telegram-bot-api-token

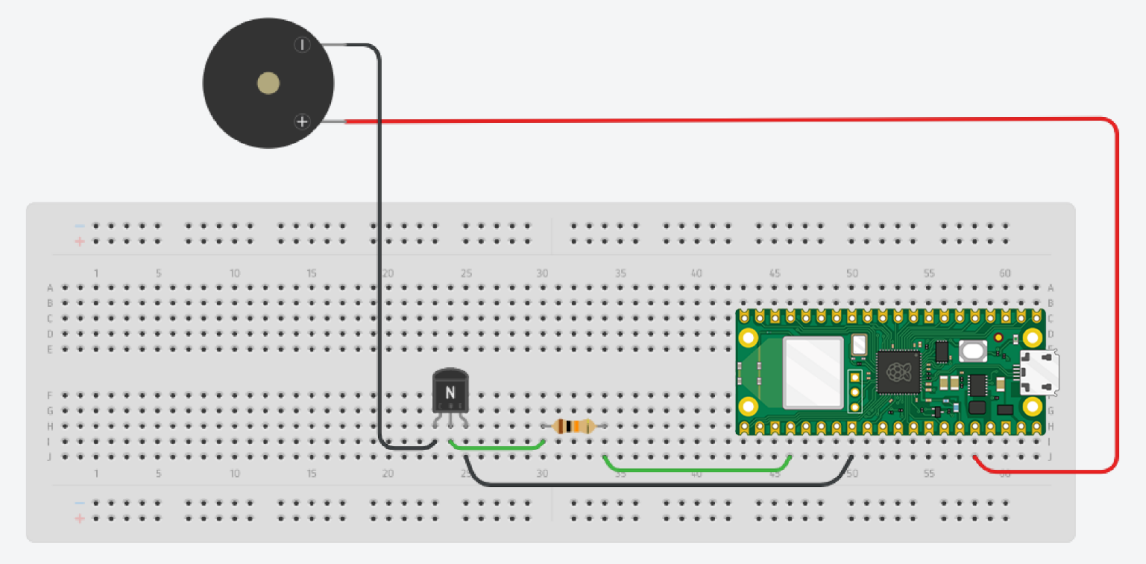
Lesson 20: Thingspeak

In this lesson, you will learn how to use the Thingspeak IoT platform to log data from your Raspberry Pi Pico W. Thingspeak is a popular platform that allows you to store, visualize and analyze data from IoT devices. The code will generate 3 random numbers and send them to Thingspeak via the Internet. By the end of this lesson, you will be able to set up a simple data logging system using Thingspeak and Raspberry Pico W, which can be expanded to monitor and analyze various parameters in your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Buzzer
- 1 x 2N2222 Transistor
- 1 x 10kΩ Resistor

Circuit Diagram



Code

- 20_thingspeak.py

Attention:

- Create an account or login to Thingspeak here.
- Create **a channel** with **3 charts**.
- Copy both **write and read API keys** of the channel and paste them into the **settings.toml** file.

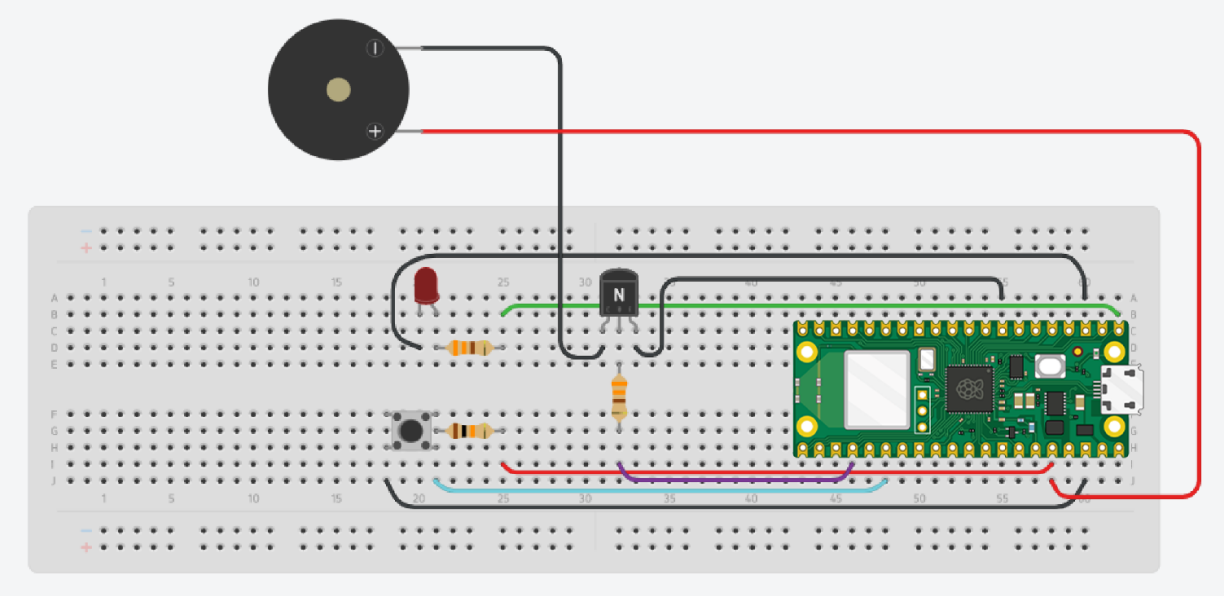
Lesson 21: Blynk IoT

In this lesson, you will learn how to use the Blynk IoT platform to control your Raspberry Pi Pico W and monitor its status remotely. Blynk is a popular platform that allows you to create mobile and web apps to interact with IoT devices. You will use CircuitPython libraries to generate random data and send it to Blynk via the Internet. You will also use Blynk's app to control an LED connected to your board and receive status updates from it. By the end of this lesson, you will be able to create a simple IoT control system using Blynk and Raspberry Pico W, which can be expanded to include various sensors and actuators in your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Buzzer
- LED
- Push Button
- 1 x 2N2222 Transistor
- 1 x 10kΩ Resistor
- 2 x 330Ω Resistor

Circuit Diagram

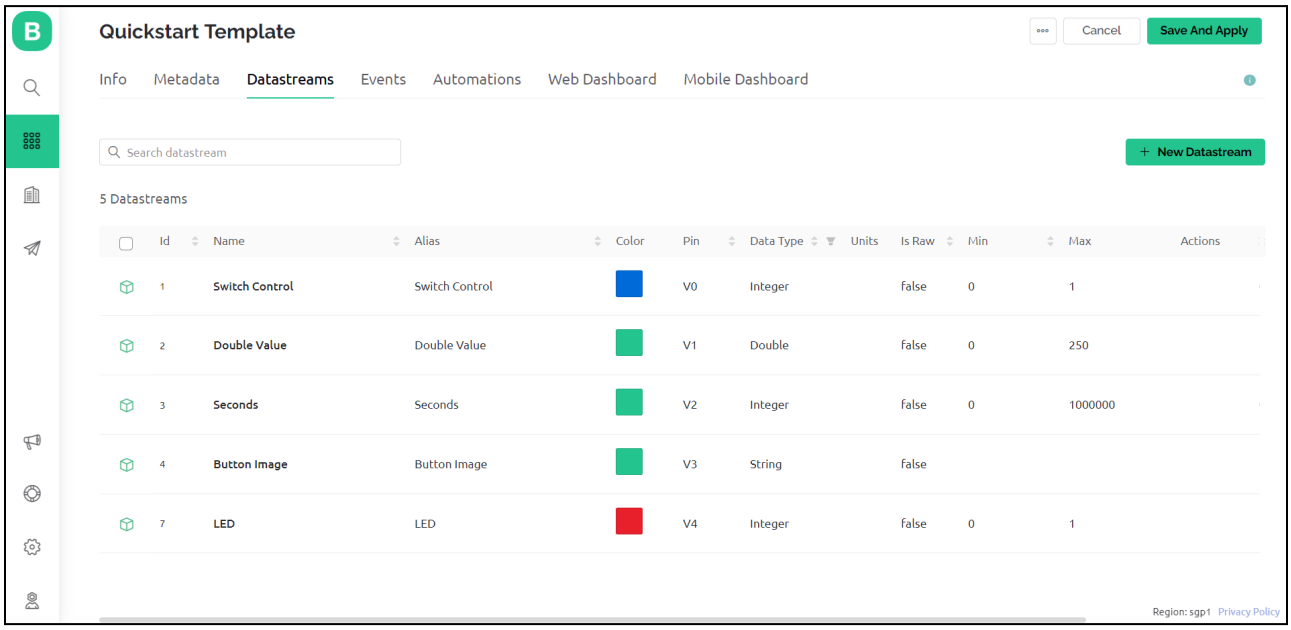


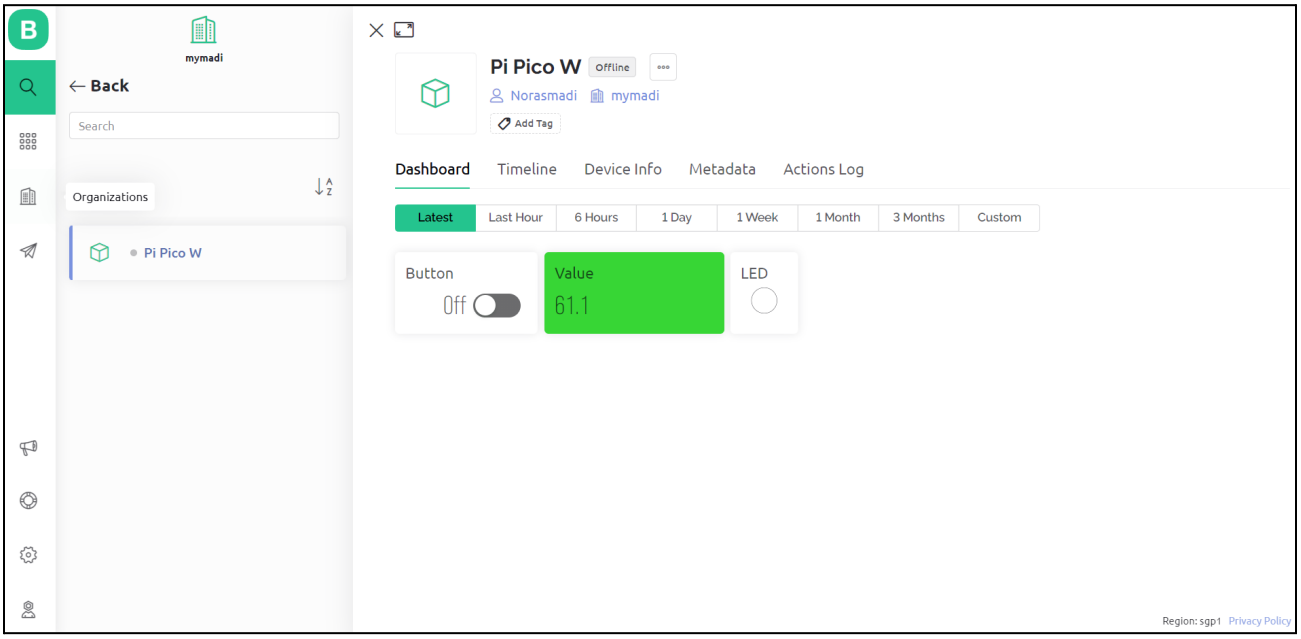
Code

- 21_blynk.py

Attention:

- Please update the Blynk **Auth Token** in **settings.toml** file.
- Follow this tutorial on how to setup Blynk account and get the **Auth Token**.
  -How to get Auth Token from Blynk (cytron.io)
- Create the datastreams and dashboard widgets based on the pictures below.





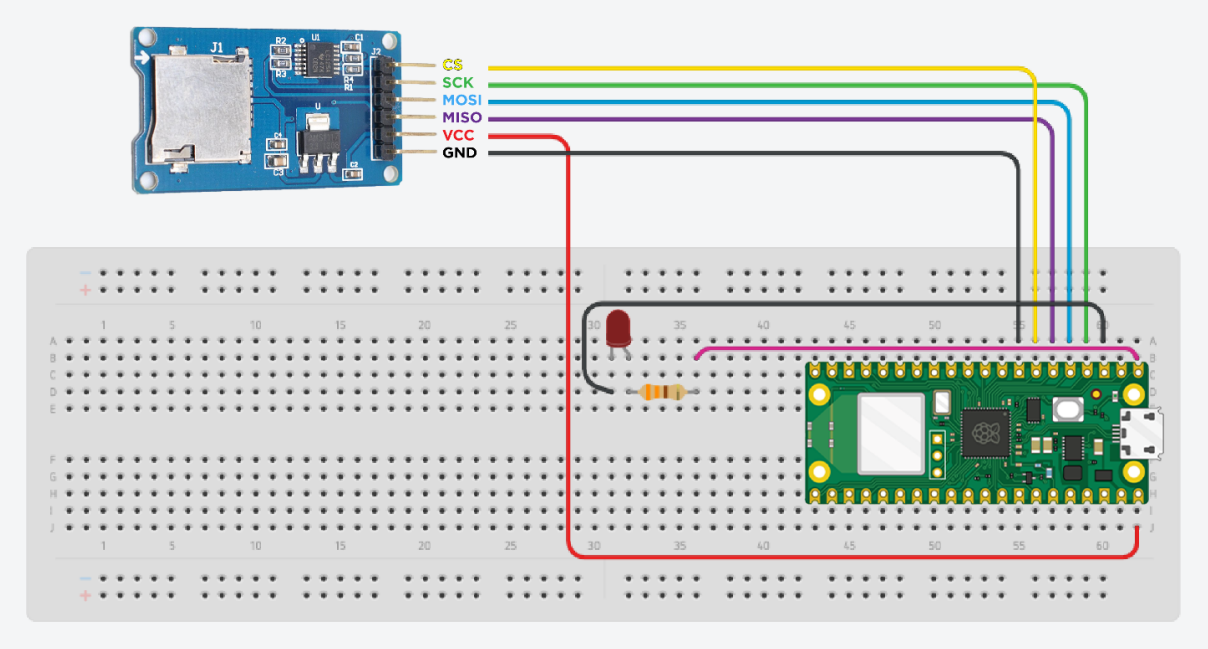
Lesson 22: SD Card

In this lesson, you will learn how to read the Raspberry Pi Pico W CPU temperature and store it in a micro SD card. The LED will light up to indicate that the data has been successfully written to the SD card.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Micro SD Card Adapter

Circuit Diagram



Code

- 22_sdcard_temperature.py

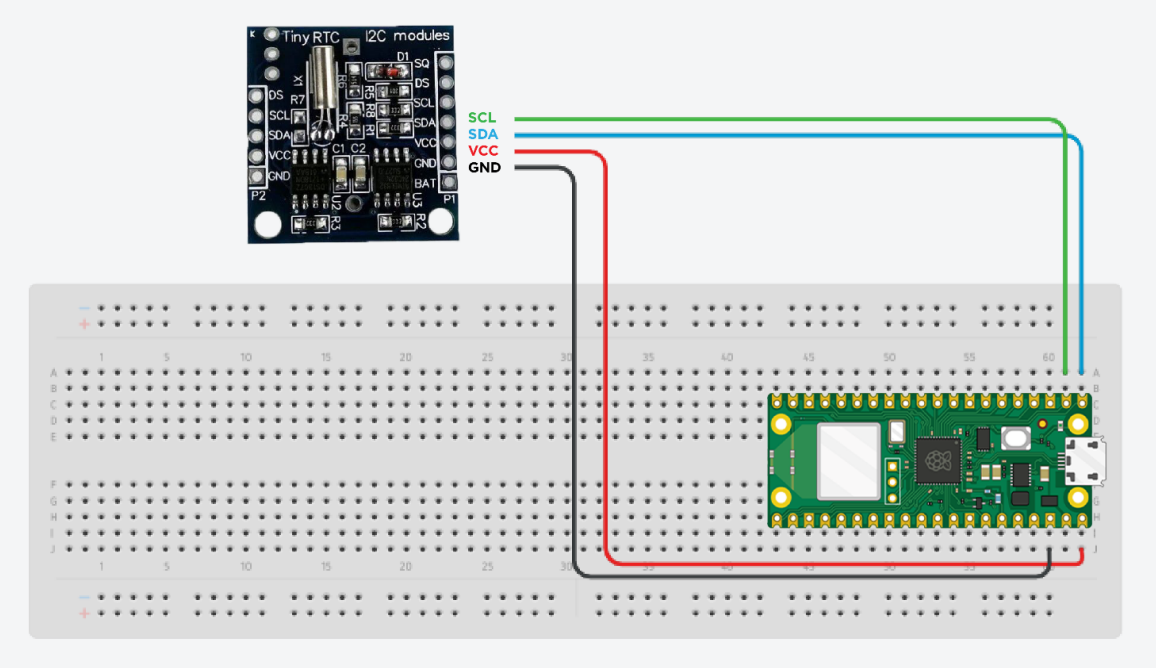
Lesson 23: Real-Time Clock (RTC)

In this lesson, you will learn how to interface the DS1307 real-time clock (RTC) with Raspberry Pi Pico W. RTCs are commonly used in embedded systems to keep track of time and date even when the system is powered off. You will use the CircuitPython library to communicate with the DS1307 and read or write the current time and date. You will also learn how to format and display the time on a serial console. By the end of this lesson, you will have a basic understanding of how to use RTCs in your projects and keep accurate time records.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- RTC module

Circuit Diagram



Code

- 23_rtc_ds1307.py

Atttention:

- Change the **if False** to **if True** in the code to set the time, otherwise it will just print the current date and time every second.

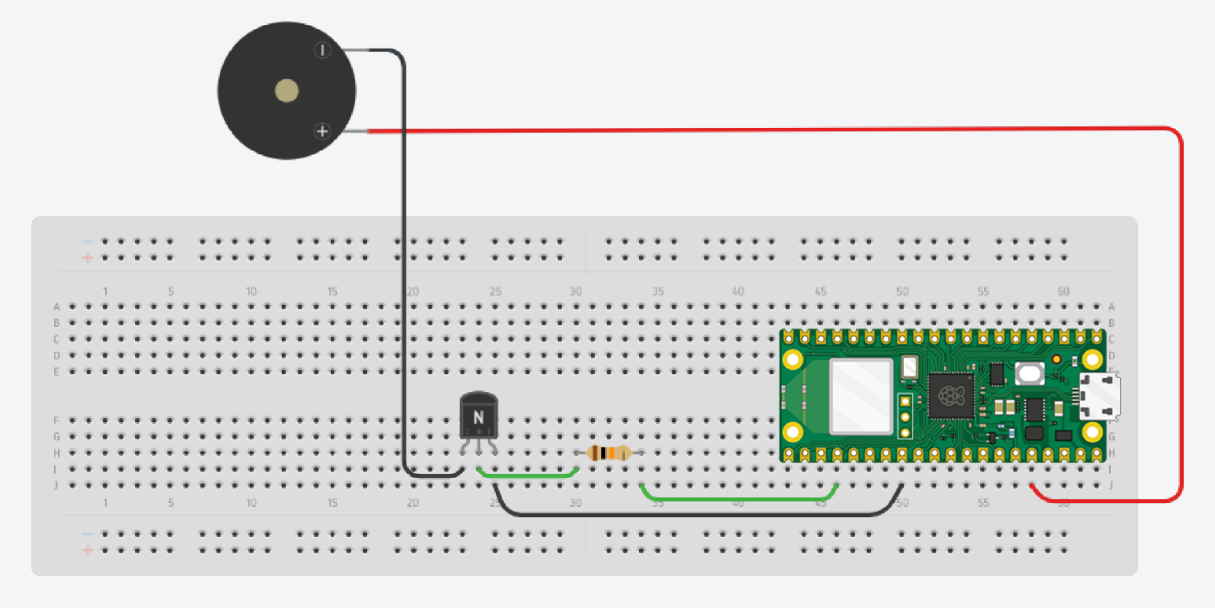
Lesson 24: Adafruit IO

In this lesson, you will learn how to use the Adafruit IO cloud service to send data from your Raspberry Pi Pico W. Adafruit IO is a powerful platform that allows you to create IoT projects, store data and control devices over the Internet. You will use CircuitPython libraries to generate random data and send it to Adafruit IO via the Internet. The buzzer beeps when the board is connected to the WiFi. You will also use Adafruit IO's dashboard to visualize and monitor the data in real time. By the end of this lesson, you will be able to set up a simple data sending system using Adafruit IO and Raspberry Pico W, which can be expanded to control and monitor various devices in your projects.

Required Components

- Raspberry Pi Pico W
- Micro USB Cable
- Breadboard
- Jumper Wires
- Buzzer
- 1 x 2N2222 Transistor
- 1 x 10kΩ Resistor

Circuit Diagram

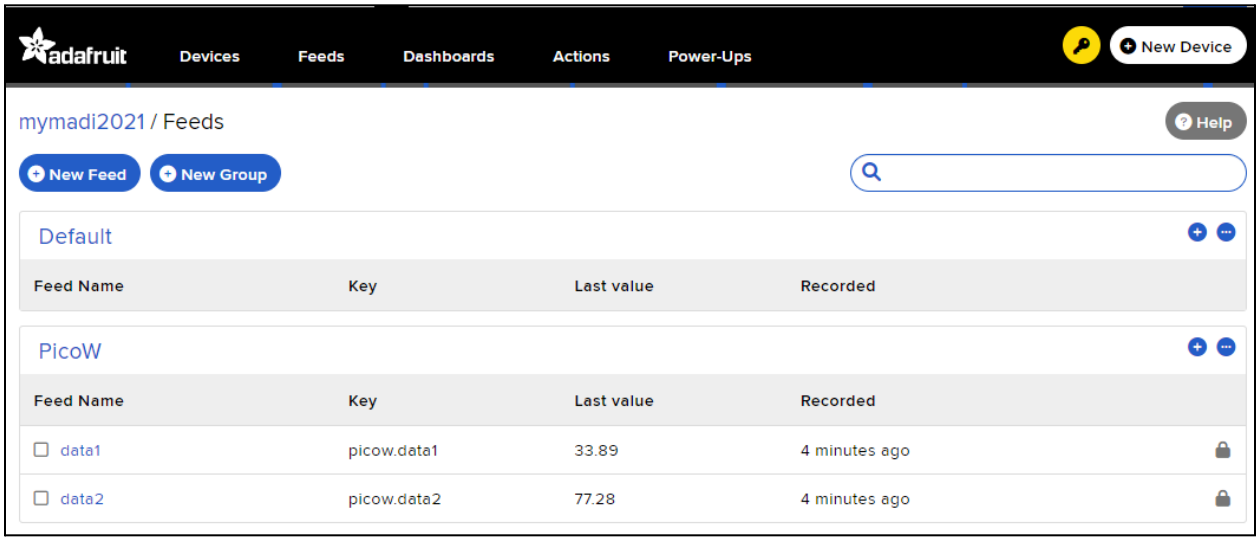


Code

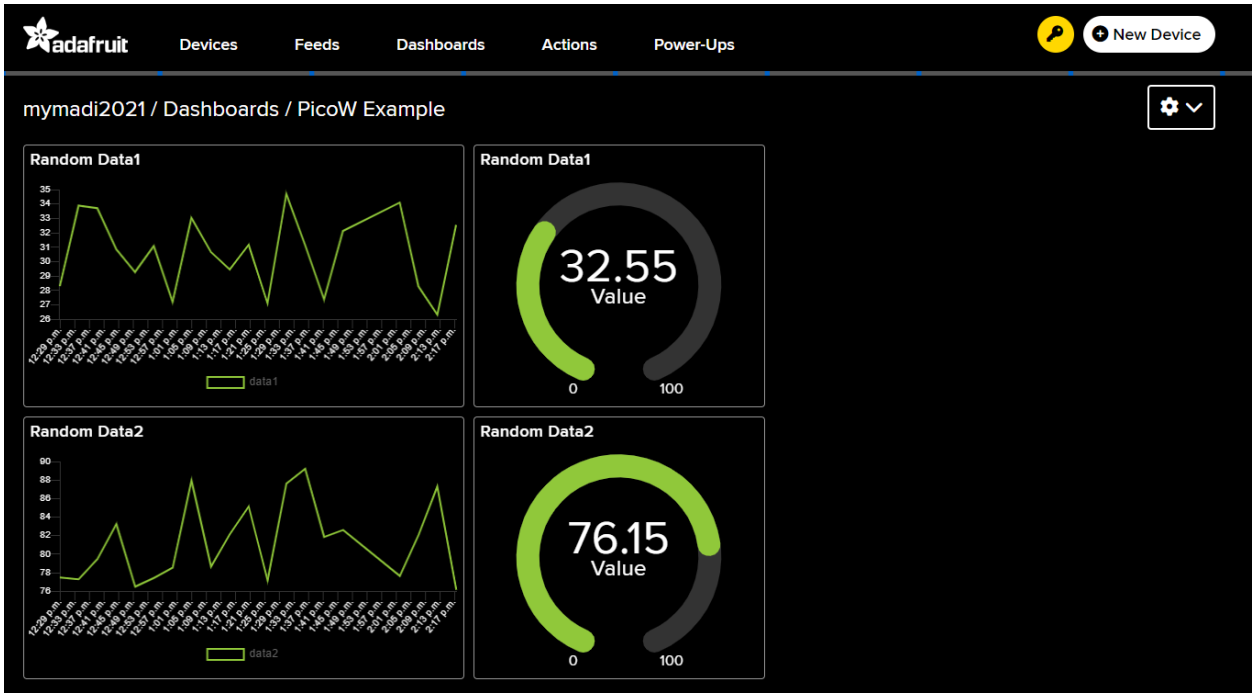
- 24_adafruitio.py

Attention:

- Create an account or login to Adafruit IO here.
- Copy both **Username** and **Active Key** and paste them into the **settings.toml** file.
- Create a **New Group** with **two New Feeds** as shown below. Please make sure that you follow the exact name of the group and feeds to sync with the code.



- In **Dashboard**, create **two Line Charts** and **Gauges** for both random data as shown below.



Feedbacks & Questions

If there are any questions / corrections / suggestions / improvements, please email support@cytron.io or you can call **04-548 0668** for technical assistance. Thank you.