

Microcontroller

Introduction HackLab

<https://github.com/KABK-HackLab/microcontroller-intro>



What is a MicroController

MCU (MicroController Unit)

A small self-contained computer on a single chip. The microcontroller is the core of an embedded system.

Embedded systems are specialized, computer-based systems combining hardware and software with the purpose of interacting with the physical world using IO Ports

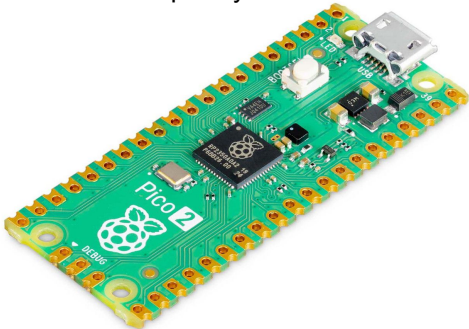
The Microcontroller contains:

- Processor (that executes our code)
- Flash Memory (for program storage)
- Ram memory (for processing, variables)
- IO (Input / Output Ports) via 'pins'



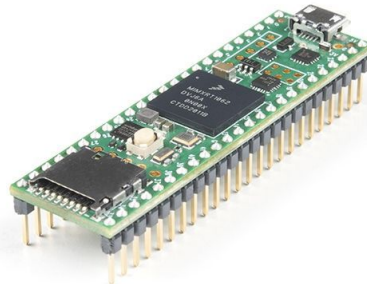
Arduino Well known, large community, open source software library. Various boards ranging in price and performance. Arduino is both a microcontroller board and a software development environment (IDE)

Raspberry Pico Cheap, Not the fastest board, but has interesting set of advanced features (e.g. dual core). Not to be confused with the Raspberry Pi.



ESP32 Includes built-in Wi-Fi and Bluetooth. Compatible with Arduino IDE.

Teensy Powerful platform that extends on the Arduino platform with advanced features, performance, and software libraries. Expensive



There are various brands and development environments.

While a MCU is only a single chip, most of them come on a microcontroller-board that makes it easier to work with. For example, it features a USB connector to upload code.

The microcontroller-board is often just called microcontroller.

What microcontroller is best depends on your project and personal preference.

Each one has its own advantages and disadvantages.

DEMO TIME

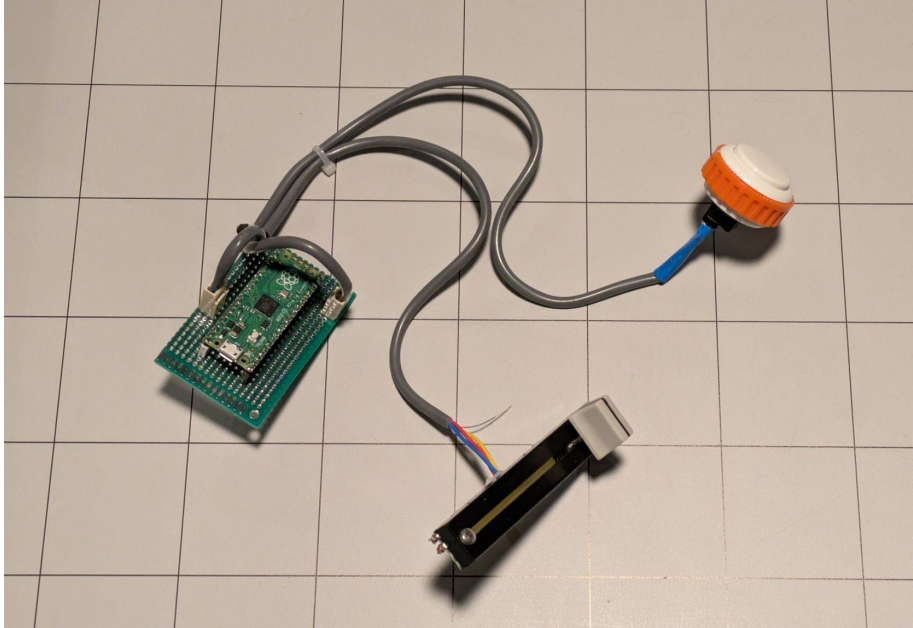
Create a basic music instrument

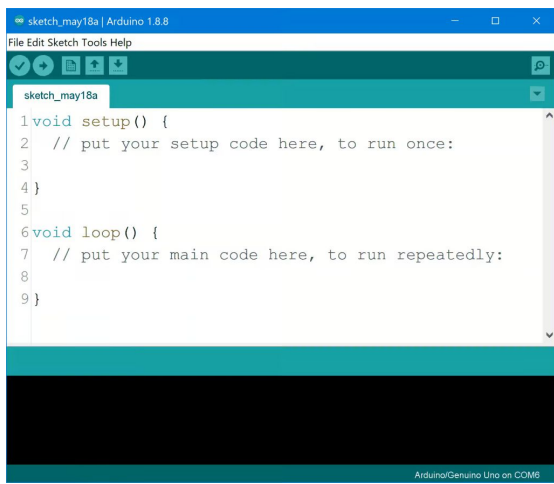
Microcontroller functions as the interface between

software: audio synthesis software, running laptop

and

hardware: buttons / faders





Arduino IDE

For use with arduino and compatible boards. Used the the c/c++ programming language with specialised software libraries for writing code for microcontrollers.

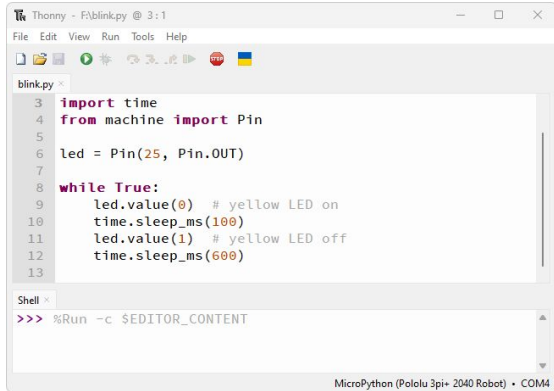
The arduino ecosystem features an extensive range of libraries to connect a wide range of hardware. Sensors, displays, shields (extension boards), etc

There are various programming development environments.

Similar to the microcontroller boards, each one has there one advantages and disadvantages.

VSCODE + Platform IO

Similar to Arduino IDE in that it uses C/C++ based language for microcontrollers. It has many advanced Features such as autocomplete, realtime debugging (stepping through code) and AI integration. Somewhat intimidating for beginners.

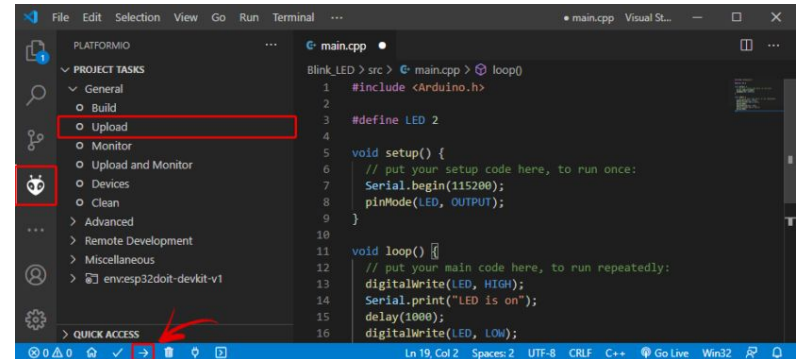


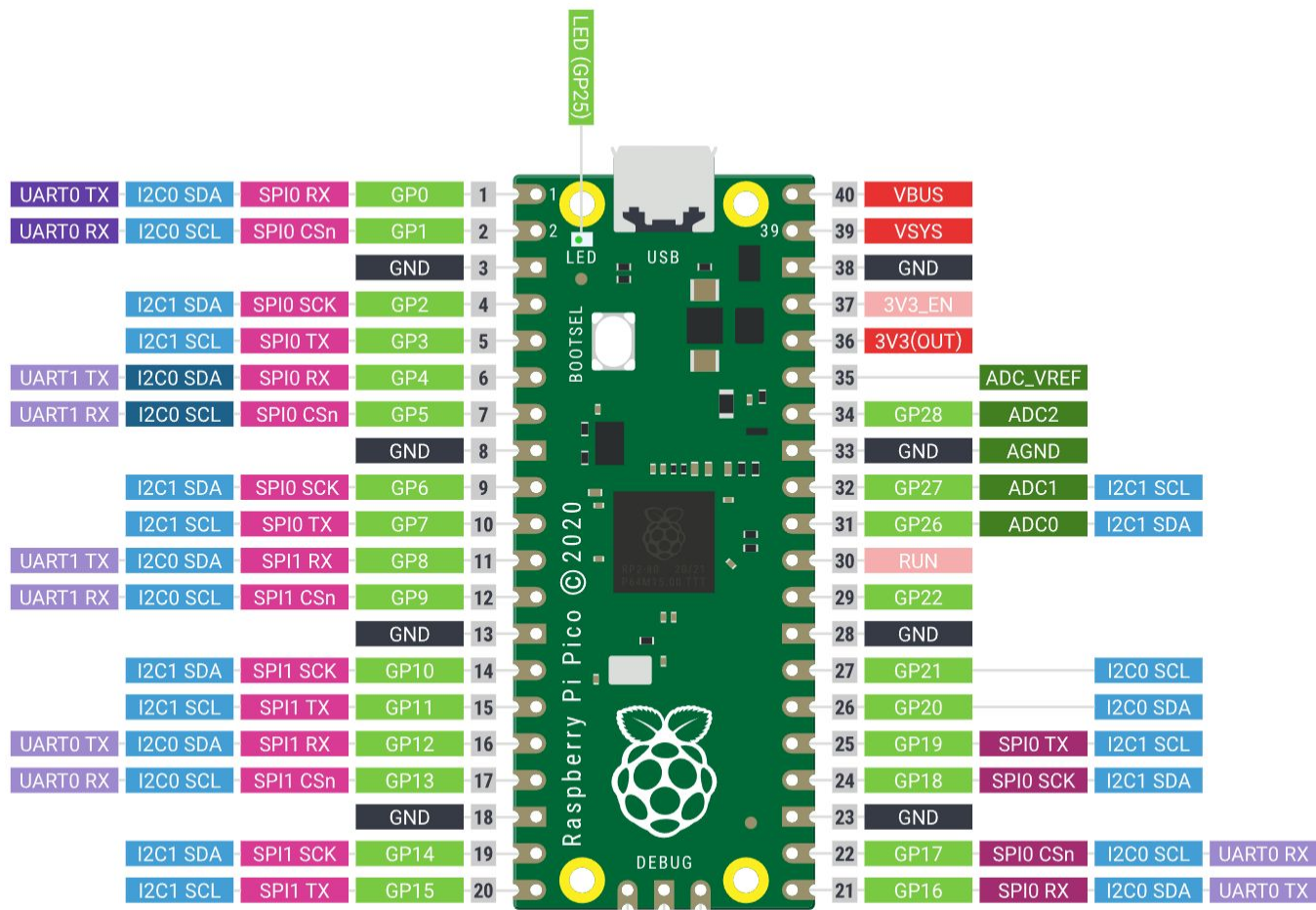
Thonny

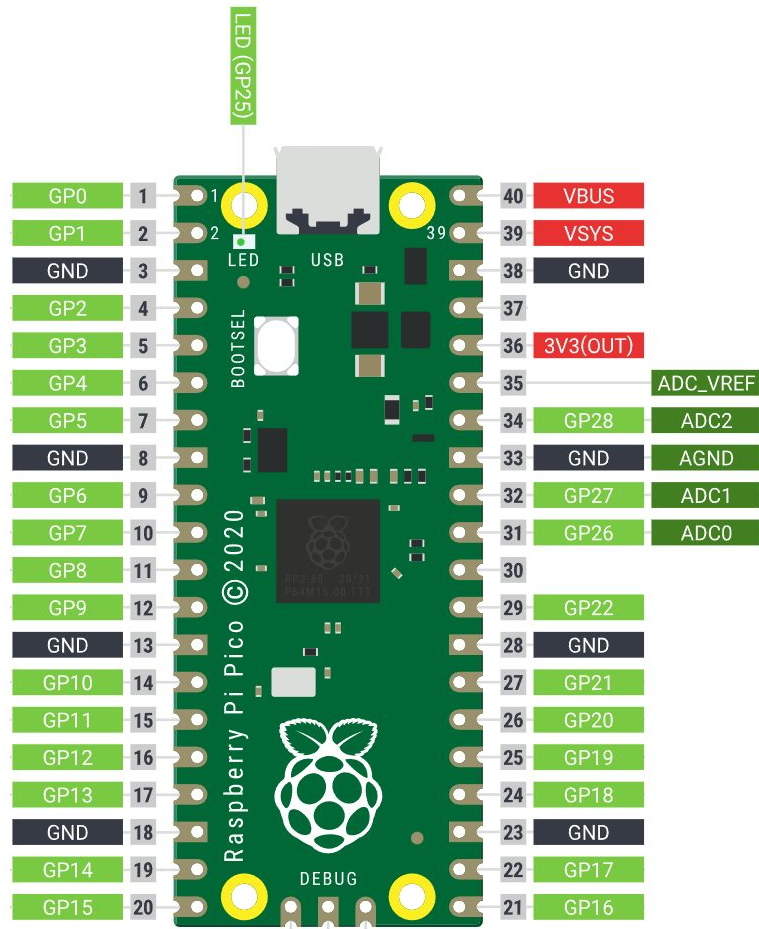
Focussed on the python programming language.

Can be used to write python code for PC's as well as microcontrollers.

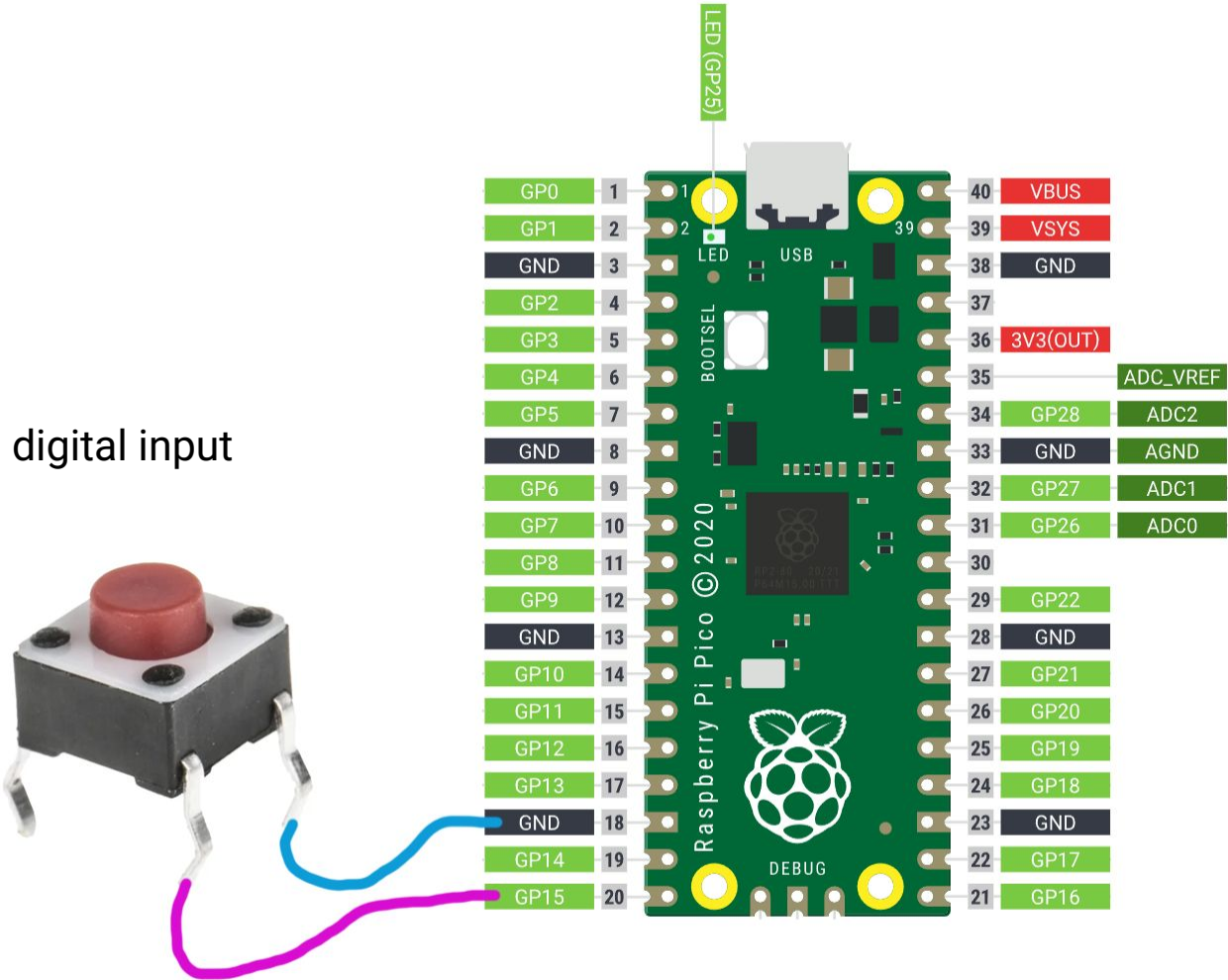
Python is arguably easier for beginners, but less performant than the C/C++ language

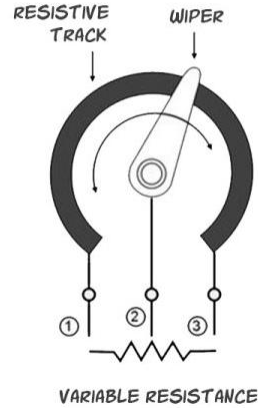
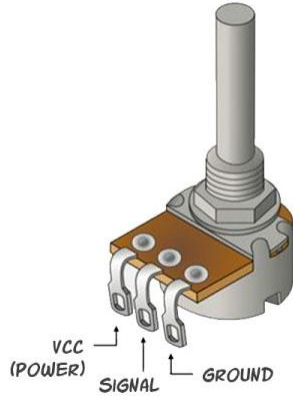






digital input





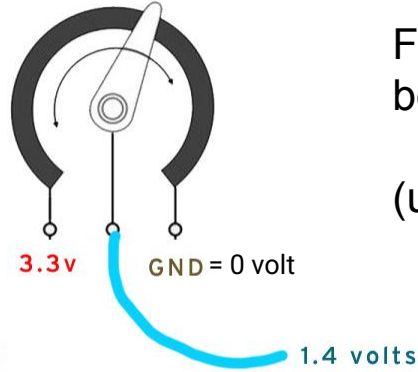
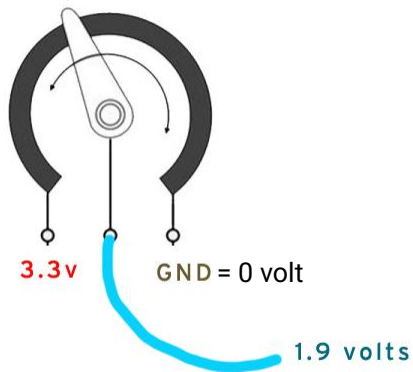
Potmeter overview

The Fader works exactly the same, different style / housing.

Potmeter: output is the middle pin (pin 2)

Fader: output position can vary between designs.

(usually labeled: pin 2)

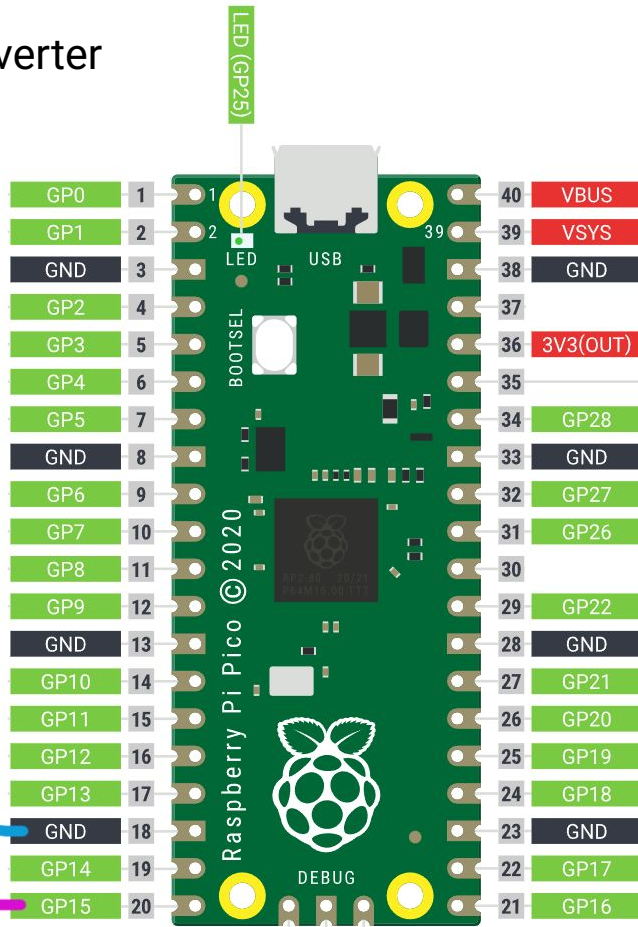


ADC = Analog to Digital Converter

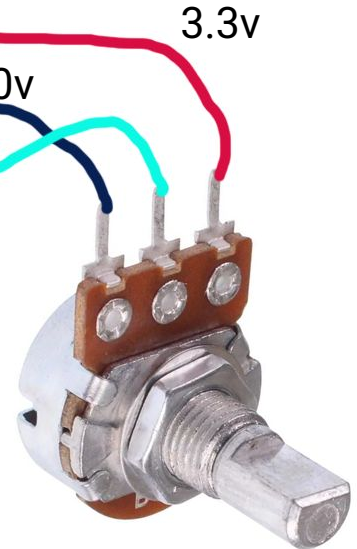
ADC can read a voltage

between 0 and 3.3volt

digital input



analog input



[Coding Time]

Testing the hardware.

Install Thonny:

<https://thonny.org/>

Install circuitpython library:

<https://circuitpython.org/libraries> or

https://github.com/adafruit/Adafruit_CircuitPython_Bundle