ENR355 Robotics and Sensors

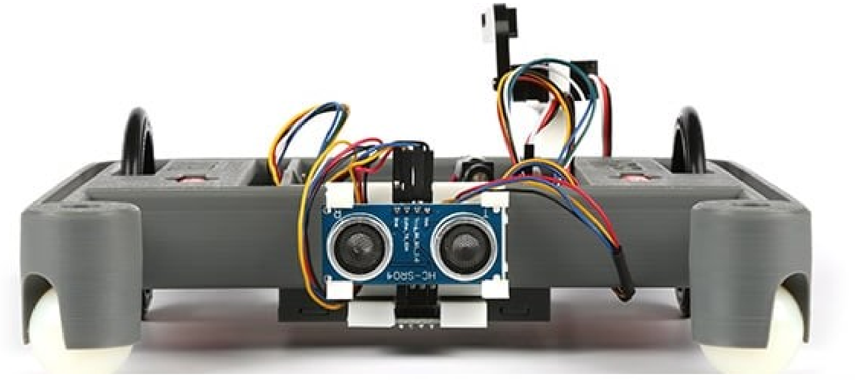
Xiang Li

Spring 2026

Coe College

Let’s start to work on the “brain” of robotics

Experiential Robotics Platform



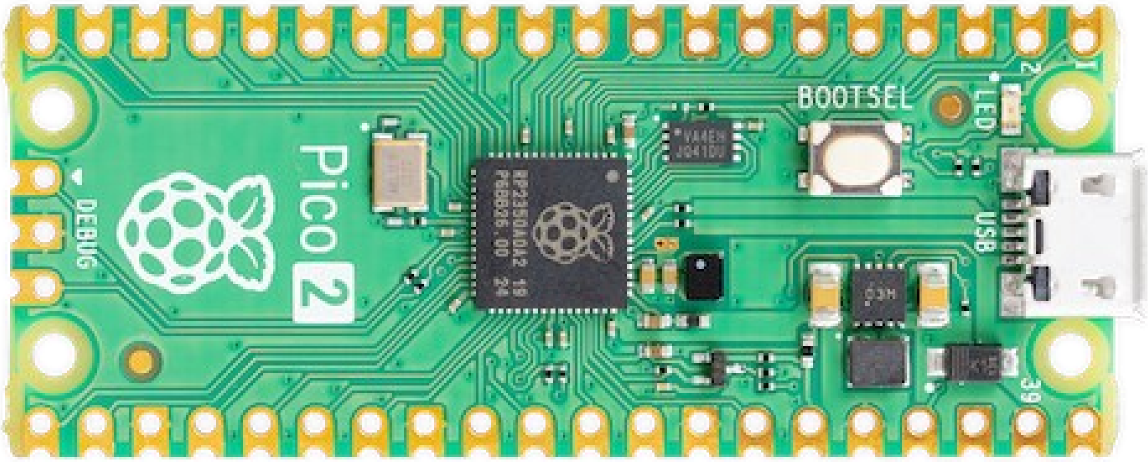
www.raspberrypi.com

Identify the brain for me, please.

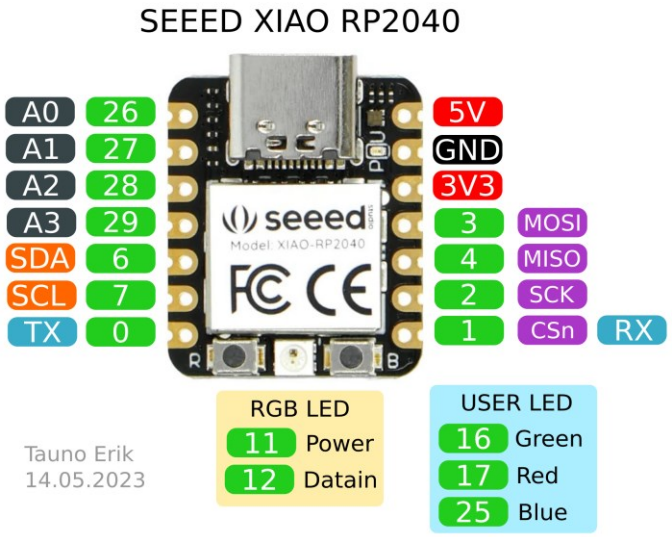
COE COLLEGE

Introducing microcontroller (MSC)

This brain is based on a special type of Systems on a Chip (SoC).

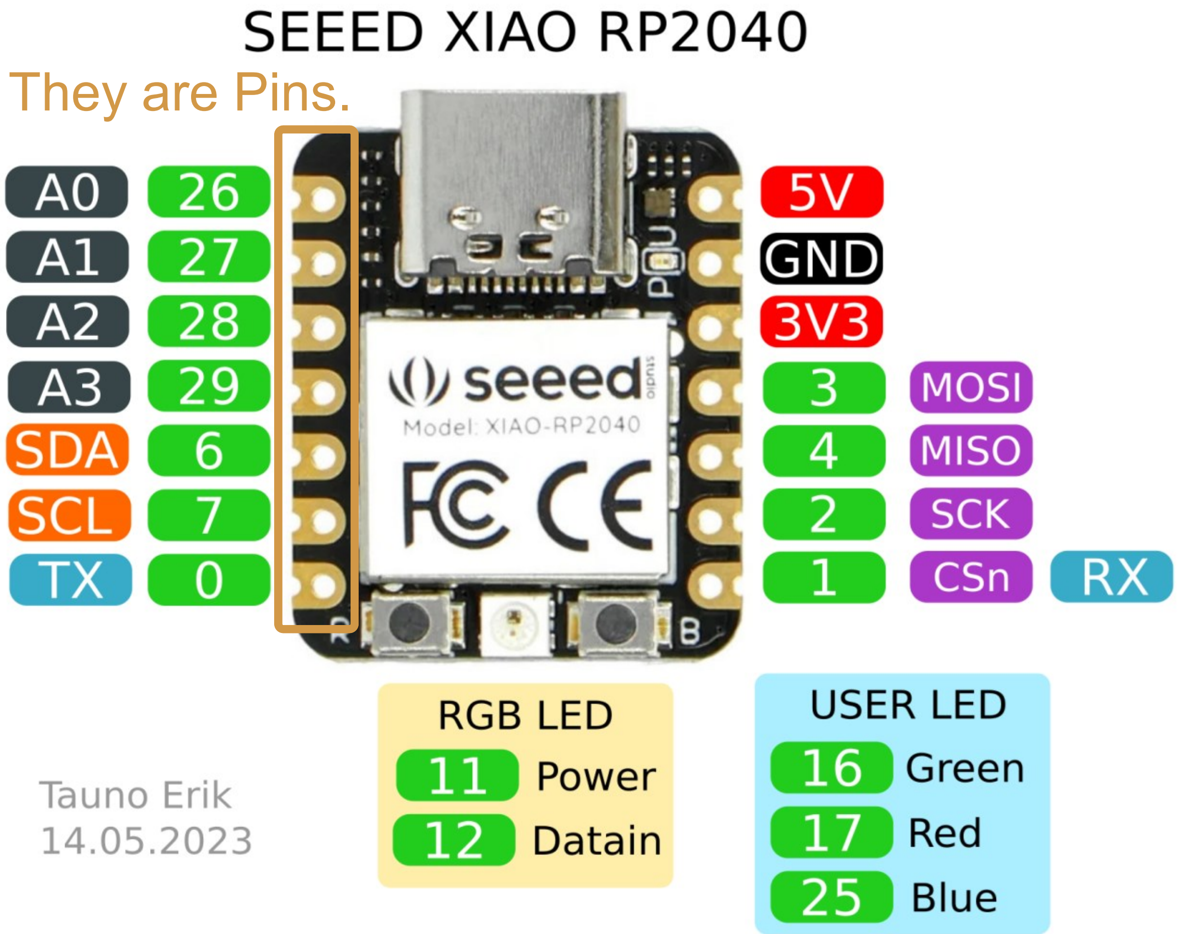


This is technically a development board (dev board).



This is another dev board, same SoC as Pico.

Anatomy of a MCU dev board



Not all pins are the same:

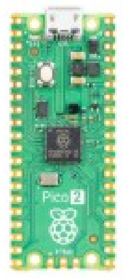
- Power?
- Data?
- Input/output?
- General purpose?
- General purpose input/output? (GPIO)
- Some additional bells and whistles.

Coe College

Step 1: install the “OS”

https://micropython.org/

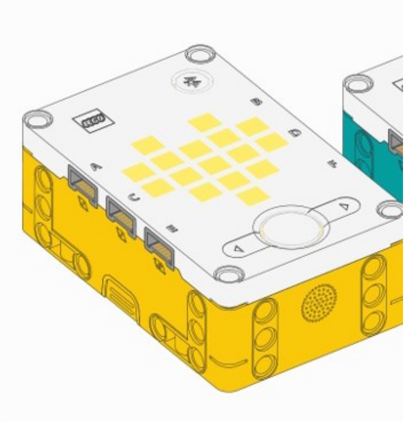
MicroPython DOWNLOAD



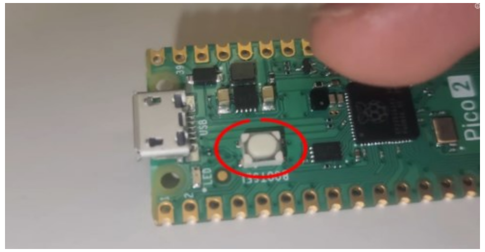
Pico 2
Raspberry Pi

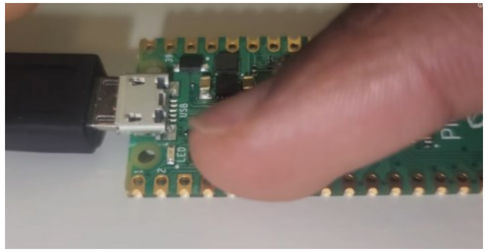
XIAO RP2350
Seeed Studio

micro:bit v1
BBC



https://www.youtube.com/watch?v=2d21MBF4i6Q





1. A removable drive
2. Drag and drop .uf2 file
3. Flash done!

Installation instructions

Flashing via UF2 bootloader

To get the board in bootloader mode ready for the firmware update, execute `machine.bootloader()` at the MicroPython REPL. Alternatively, hold down the BOOTSEL button while plugging the board into USB. The uf2 file below should then be copied to the USB mass storage device that appears. Once programming of the new firmware is complete the device will automatically reset and be ready for use.

Firmware

Releases

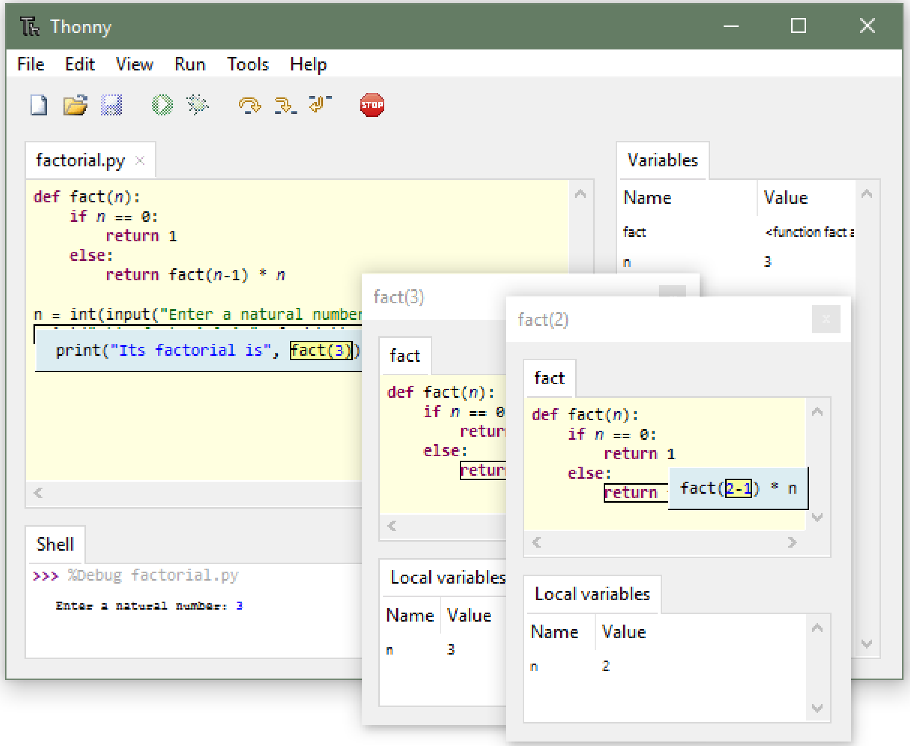
v1.27.0 (2025-12-09) .uf2 / [Release notes] (latest)
v1.26.1 (2025-09-11) .uf2 / [Release notes]
v1.26.0 (2025-08-09) .uf2 / [Release notes]
v1.25.0 (2025-04-15) .uf2 / [Release notes]
v1.24.1 (2024-11-29) .uf2 / [Release notes]
v1.24.0 (2024-10-25) .uf2 / [Release notes]

Step 2: use IDE, any IDE

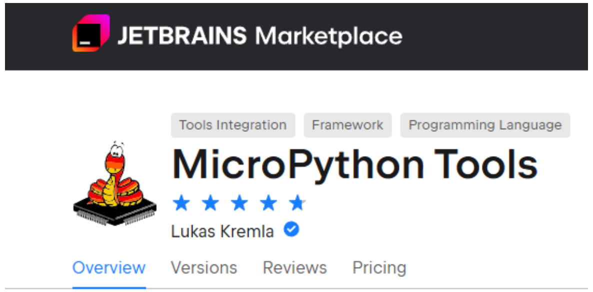
Option 1: Thonny: https://thonny.org/

Thonny
Python IDE for beginners

Download version 4.1.7 for
Windows • Mac • Linux



Option 2: any Python IDE



https://github.com/lukaskremla/micropython-tools-jetbrains/blob/main/DOCUMENTATION.md#getting-started

Coe College

Step 3: bare-metal coding… just kidding!

https://github.com/AndreaFavero71/pio_stepper_control

README MIT license

Accurate stepper motor control, based on micropython and PIO

This code is for RP2040 or RP2350 microprocessors, as it leverages on the PIO feature; Boards with these micros are Raspberry Pi Pico, Pico2, RP2040-Zero, RP2350-Zero, and many others.

I'm working on a project with stepper motors and RP2040 microprocessor and I realized this stepper motor's control (Class) might be useful to others.

The overall concept is to predefine the speed and the steps for the stepper, and let it running in open loop ... by trusting it stops once those steps are made!

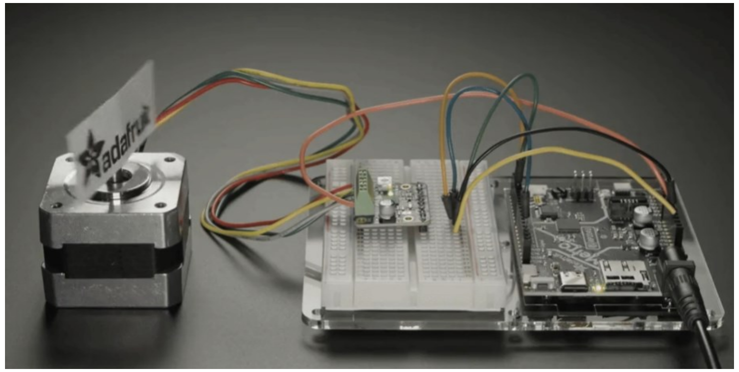
The precise spinning time is calculated, upfront the stepper activation, allowing to be ready with a new set of instructions for the following run.

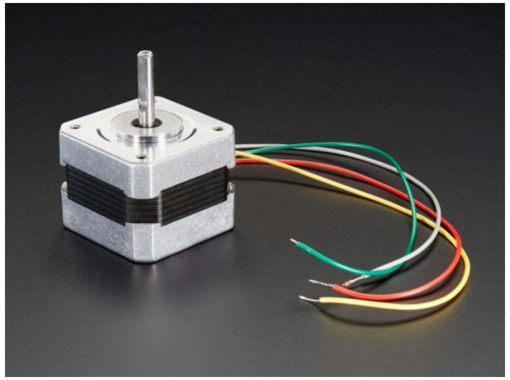
Depending on the microprocess used, this implementation is accurate for pulses frequency between 50Hz and 5KHz (RP2040) or between 10Hz and 15KHz (RP2350). The common Nema 17 steppers, with 200 pulses per revolution, controlled with 1/8 microsteps, very likely do not require more than 5KHz in most of the applications.

COE COLLEGE

Step 4.1: wiring the motor

- If you are having trouble with assignment #3, talk to me!
- If you are having trouble finding all the information on this page online, talk to me!





- 200 steps per revolution, 1.8 degrees
- Coil #1: Red & Yellow wire pair. Coil #2 Green & Brown/Gray wire pair.
- Bipolar stepper, requires 2 full H-bridges!
- 4-wire, 8 inch leads
- 42mm/1.65" square body
- 31mm/1.22" square mounting holes, 3mm metric screws (M3)
- 5mm diameter drive shaft, 24mm long, with a machined flat
- 12V rated voltage (you can drive it at a lower voltage, but the torque will drop) at 350mA max current
- 28 oz*in, 20 N*cm, 2 Kg*cm holding torque per phase
- 35 ohms per winding

Image source from https://learn.adafruit.com/

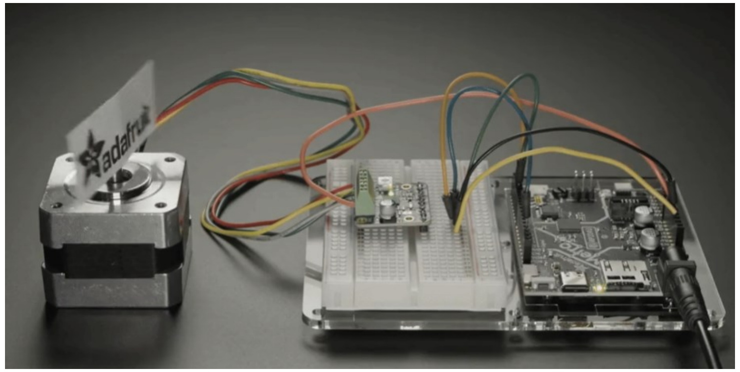
Pinout source https://learn.adafruit.com/adafruit-tmc2209-stepper-motor-driver-breakout-board/pinouts

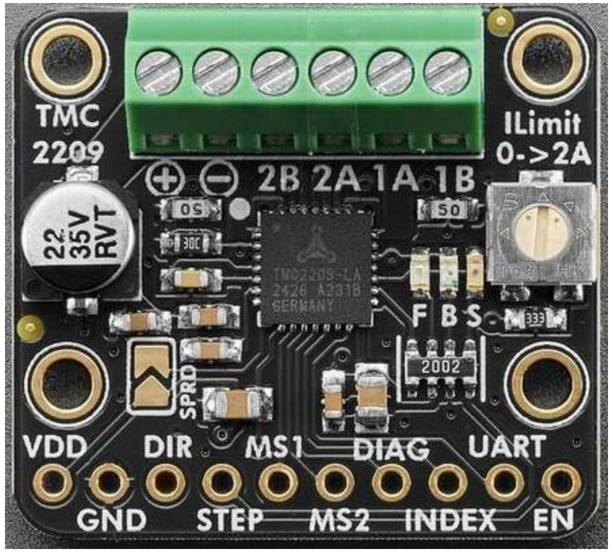
COE COLLEGE

Step 4.2: wiring the driver board

- If you are having trouble finding all the information on this page online, talk to me!

1. How many pins we need from the Pico2?
2. Where to connect?





Scan QR code to the pinouts page.

Stepper Control Pins

- **DIR** - This is the direction control pin. You can set this pin high or low to set the spin orientation. Pulling it high turns the motor clockwise. Pulling it low turns it counterclockwise.
- **STEP** - This is the microstep control pin. Toggle this pin to take one step or microstep at a time. By default, the driver is set to 1/8 microstep mode.

Image source from https://learn.adafruit.com/

Pinout source https://learn.adafruit.com/adafruit-tmc2209-stepper-motor-driver-breakout-board/pinouts

Step 4.3: wiring the MCU

The question is NOT *to be…*
(which pin to use)

but *to not be*!
(which pin NOT to use)

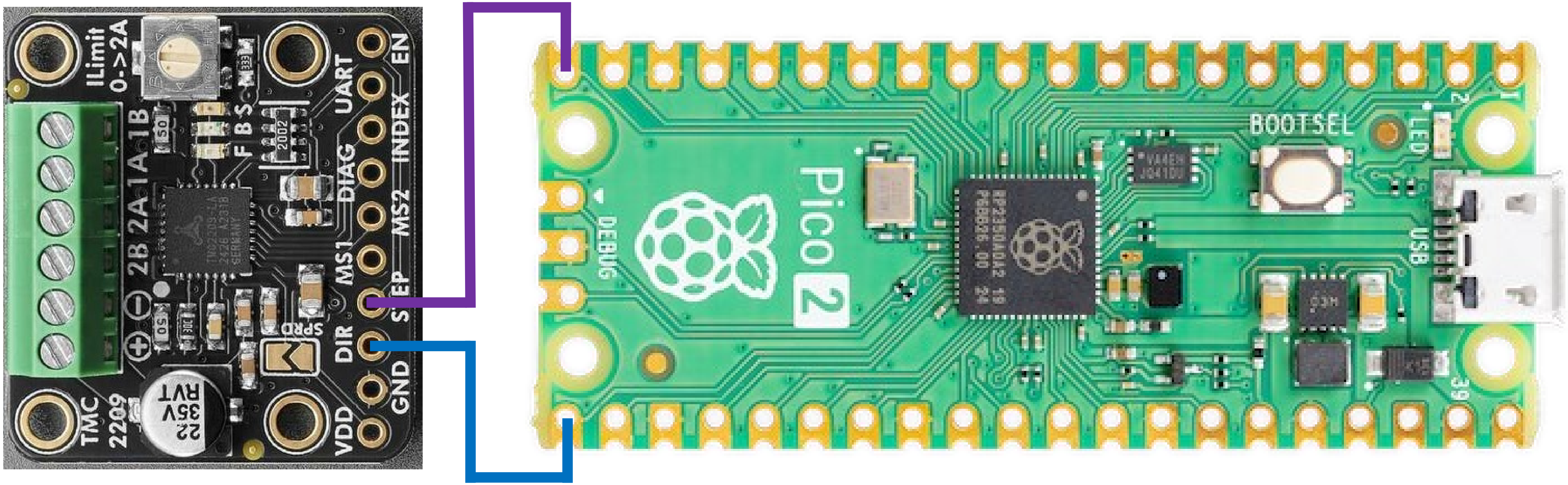


Image source from https://learn.adafruit.com/ and www.raspberrypi.com

Pinout source https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html

Coe College