Software

Overview

Welcome to the Software Onboarding for the 2025-2026 school year!

Goals

- Learning
  - How do we write software in SCR?
  - How does a microcontroller control hardware?
- practical
  - Writing Arduino/ C++ code
  - Managing software dependencies
  - Reading software documentation

Downloads

Arduino:

Go here and download the version of the arduino IDE for your operating system (Windows, OSX, linux).

Pico board support:

We'll be using the Raspberry Pi Pico 2 for onboarding. To develop on the pico, you need to add an additional board manager URL to the Arduino IDE.

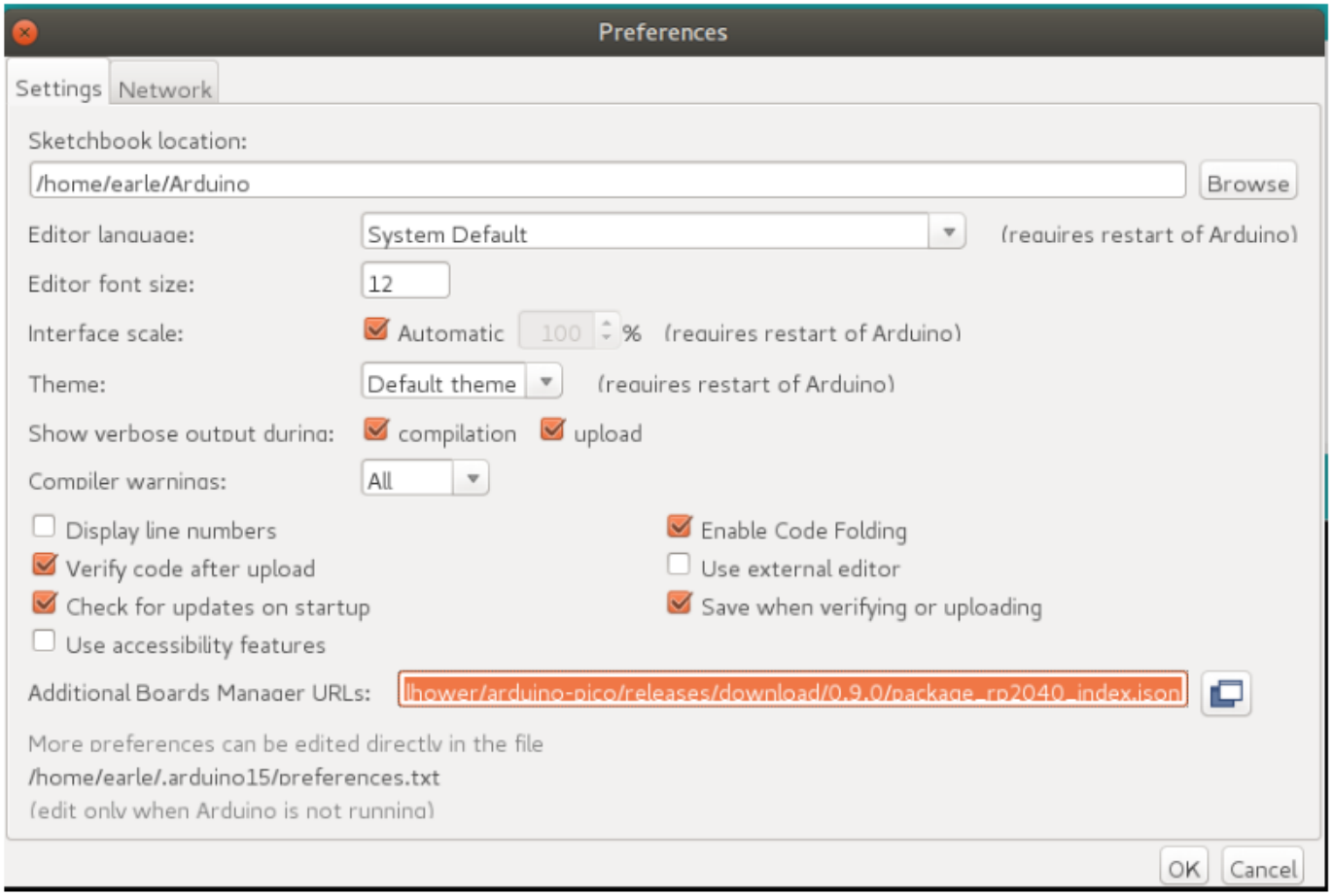
Go here and scroll to the **installation** section. Follow the short directions there.

Installation

Open up the Arduino IDE and go to File->Preferences.

In the dialog that pops up, enter the following URL in the "Additional Boards Manager URLs" field:

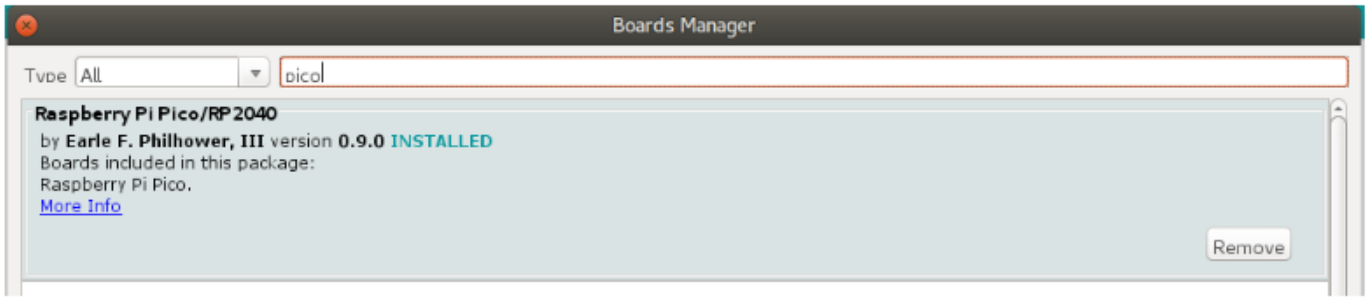
https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json



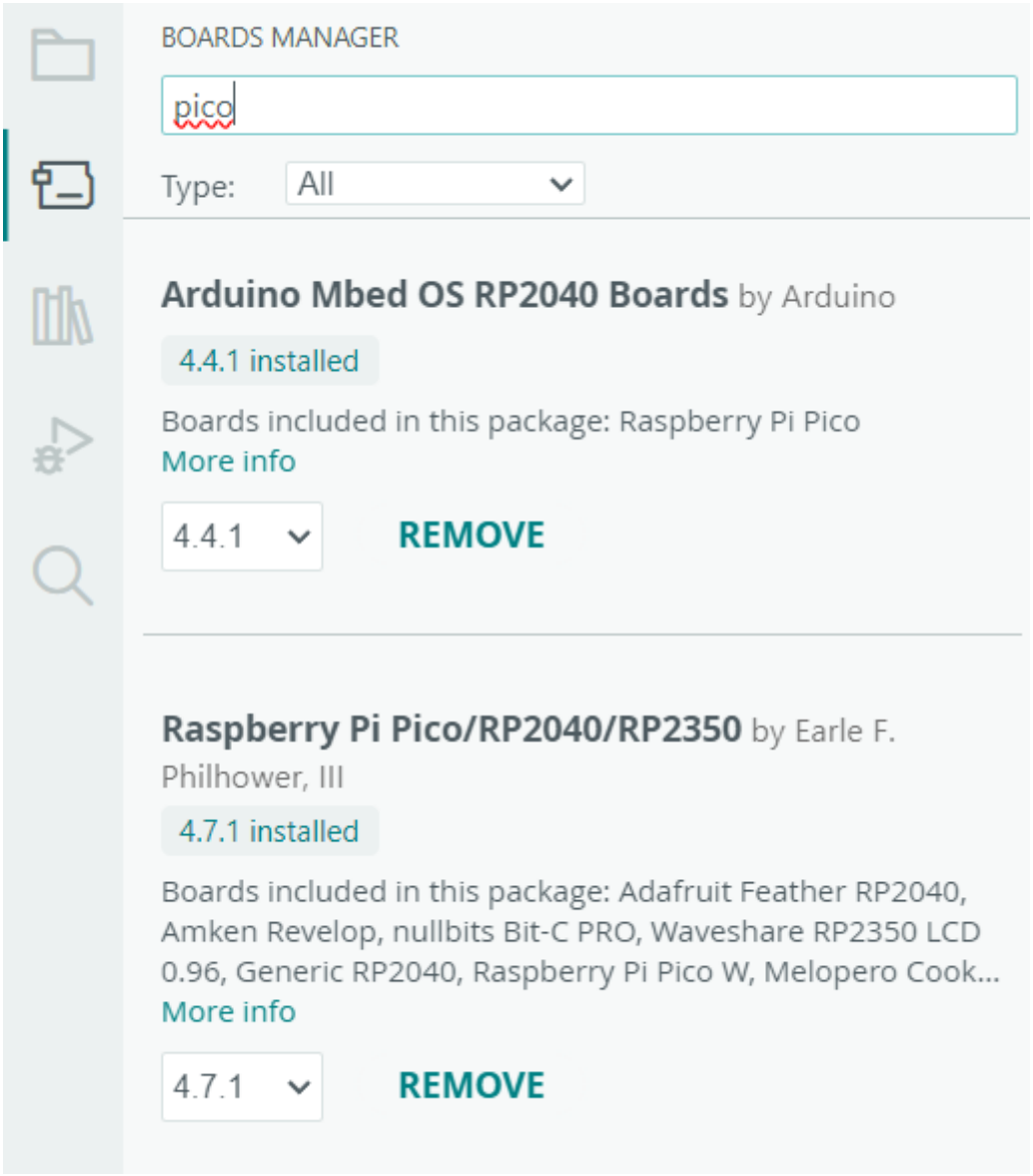
Hit OK to close the dialog.

Go to Tools->Boards->Board Manager in the IDE

Type "pico" in the search box and select "Add":



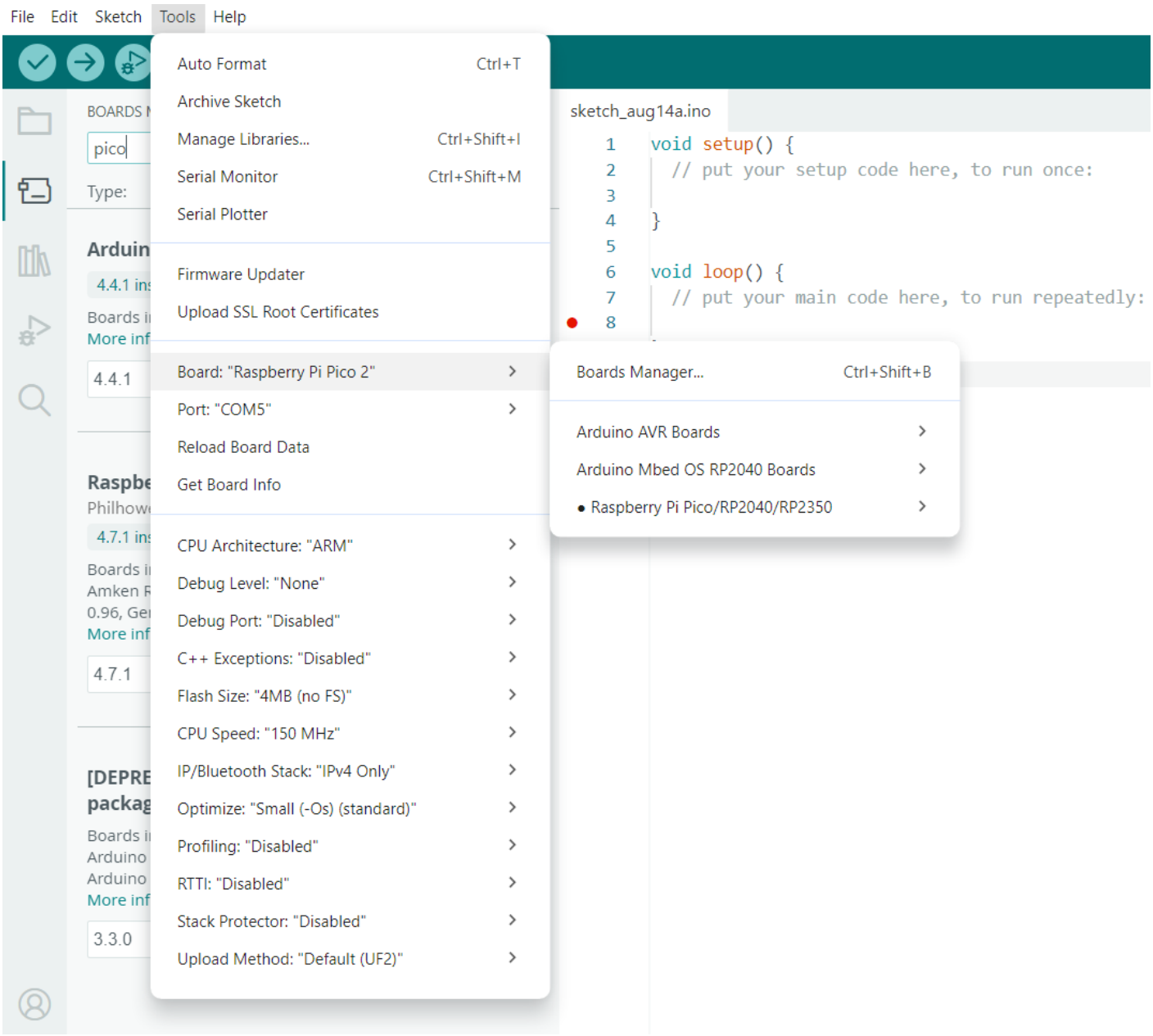
In the boards manager tab, search for these two libraries and **install** them



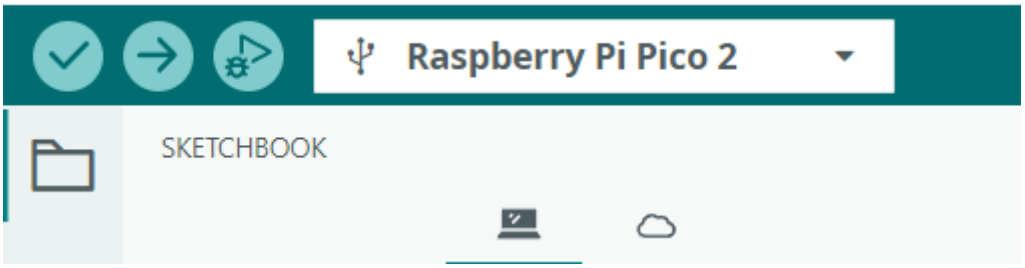
Compilation

Select the Pico 2:

Use the Arduino IDE to select the Pico 2 as the compilation target.



Connect the Pico 2 to your computer using a micro-usb cable. You should see this drop down turn **bold**.



Verifying Source Code:

Copy this skeleton code into an arduino sketch file.

```
#include <Servo.h>

#define LED1_PIN 13
#define LED2_PIN 14
#define LED3_PIN 15

#define IR1_PIN 0
#define IR2_PIN 1
#define IR3_PIN 2
#define IR4_PIN 3
#define IR5_PIN 4

#define MOTOR2_IN3 18
#define MOTOR2_IN4 19

#define MOTOR1_IN1 20
#define MOTOR1_IN2 21

#define SERVO_PIN 22

Servo payloadServo;

void driveForward(uint8_t speed) {
  // right motor
  analogWrite(MOTOR1_IN1, speed);

  // left motor
  analogWrite(MOTOR2_IN3, speed);
}

void dumpPayload(Servo& servo) {
  servo.write(0);
}

void raisePayload(Servo& servo) {
  servo.write(180);
}

bool is_detecting_obstacle(int pin)
{
```

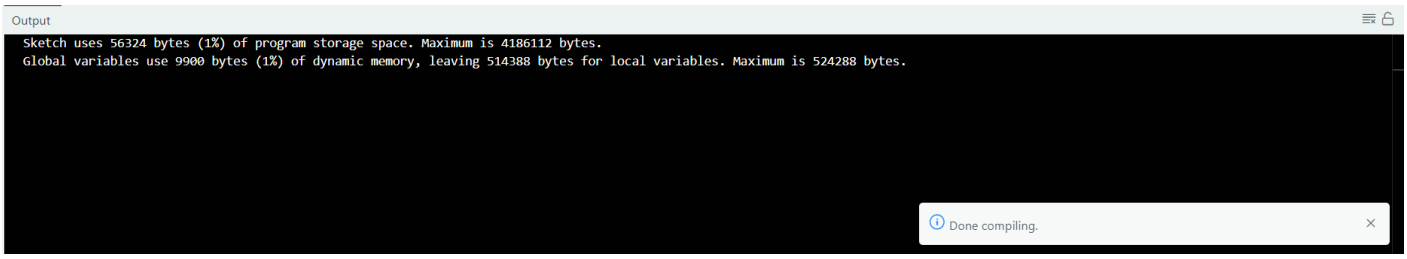
```
  int sensorStatus = digitalRead(pin);
  return sensorStatus == LOW;
}

void setup() {
  // IR Pin Setup
  pinMode(IR1_PIN, INPUT);
  pinMode(IR2_PIN, INPUT);
  pinMode(IR3_PIN, INPUT);
  pinMode(IR4_PIN, INPUT);
  pinMode(IR5_PIN, INPUT);

  // LED Setup
  pinMode(LED1_PIN, OUTPUT);
  pinMode(LED2_PIN, OUTPUT);
  pinMode(LED3_PIN, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  // Motor setup
  pinMode(MOTOR1_IN1, OUTPUT);
  pinMode(MOTOR1_IN2, OUTPUT);

  pinMode(MOTOR2_IN3, OUTPUT);
  pinMode(MOTOR2_IN4, OUTPUT);

  // Servo setup
  payloadServo.attach(SERVO_PIN, 600, 2400);
}

void loop() {
  // put your main code here, to run repeatedly:
  driveForward(200);
  delay(500);
  dumpPayload(payloadServo);
  delay(500);
  raisePayload(payloadServo);
  delay(500);
}
```

This file contains `#define`s for the Pico 2's inputs and outputs as well as a function to drive both motors forward.

Edit the `setup()` function and add the line

`digitalWrite(LED1_PIN, HIGH);`

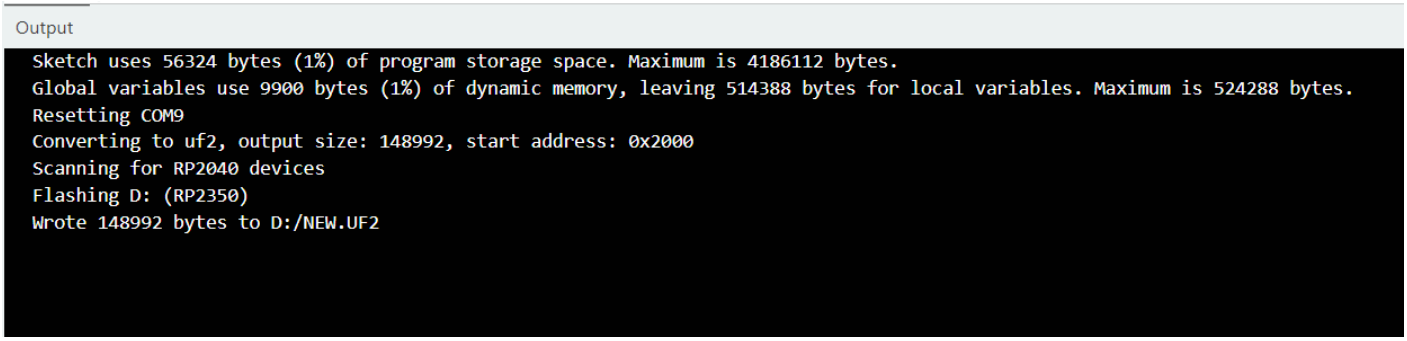
Click the check mark at the top left of the IDE to check for compilation errors. If your program compiles correctly you should see this output.

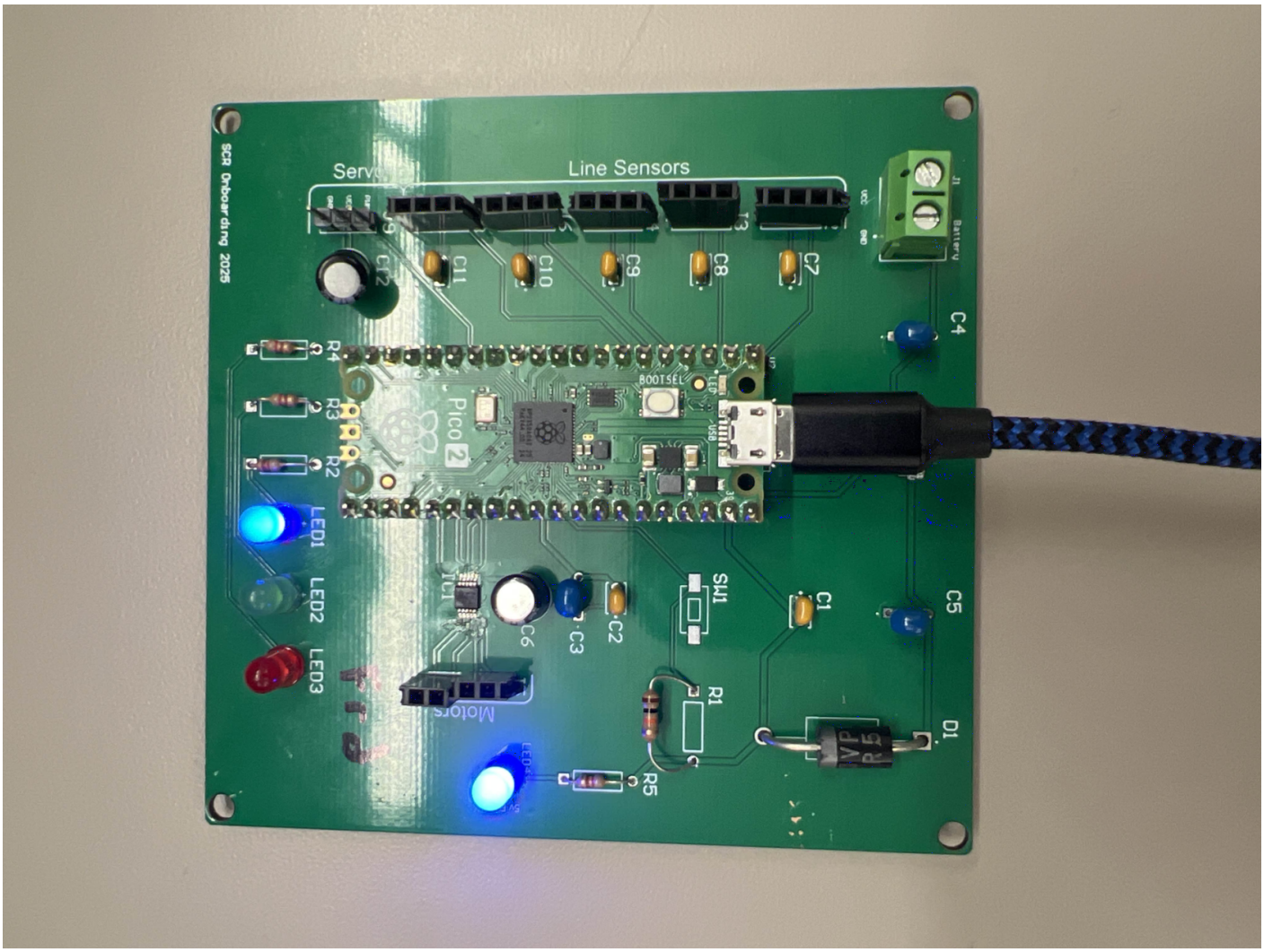


Flashing Pico 2:

Click the arrow at the top left of the IDE to flash the Pico 2. If flashing doesn't work, you may need to unplug the Pico 2 and hold the BOOTSEL button while reconnecting the micro-usb cable.

If your flash is successful, you should see this output, and LED1 should be turned on.





Useful Documentation

Motor Driver:

You shouldn't need more than this truth table, but if you'd like to read the full documentation you can do that here.

7.3.2 Truth Tables

DRV8421A: 4-wire input version shows the logic for the inputs IN1, IN2, IN3, and IN4. DRV8421B: 2-wire input version shows the logic for the inputs IN1 and IN2.

Table 7-1. DRV8421A: 4-wire input version

<table>
<tr><th colspan="4">Inputs</th><th colspan="4">Outputs</th><th colspan="2" rowspan="2">Function</th></tr>
<tr><th>IN1</th><th>IN2</th><th>IN3</th><th>IN4</th><th>OUT1</th><th>OUT2</th><th>OUT3</th><th>OUT4</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Off</td><td>Off</td><td>Off</td><td>Off</td><td colspan="2">Standby (Hi-Z)</td></tr>
<tr><td>0</td><td>0</td><td colspan="2" rowspan="3">-</td><td>Off</td><td>Off</td><td colspan="2" rowspan="3">-</td><td rowspan="4">Channel 1</td><td>Standby (Hi-Z)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Forward</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Reverse</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Brake</td></tr>
<tr><td colspan="2" rowspan="4">-</td><td>0</td><td>0</td><td colspan="2" rowspan="4">-</td><td>Off</td><td>Off</td><td rowspan="4">Channel 2</td><td>Standby (Hi-Z)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Forward</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Reverse</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Brake</td></tr>
</table>

Arduino:

Arduino API, Servos, Search

Revision #30
Created 14 August 2025 14:27:04 by Dylan Zemlin
Updated 23 September 2025 23:23:15 by Dylan Zemlin