



# Kick-Start your STEM Curriculum using SparkFun XRP/XRP Beta Robots

07 March 2026  
ACSE Conference 2026

Dennis Cecic, P. Eng.  
([d.cecic@ieee.org](mailto:d.cecic@ieee.org), [info@cool-mcu.com](mailto:info@cool-mcu.com))



# Facilitator



- **Principal Technical Training Engineer, Microchip Technology Inc.**
  - Semiconductor manufacturer
  - Design & Develop technical training for engineers (our customers)
- **Subject matter expertise**
  - 8, 16 and 32-bit Microcontrollers and embedded firmware development
  - Wireless connectivity and networking applications
- **STEM Outreach Coordinator for IEEE Canada**
- **Runs a Pre-U STEM Educator LMS:**
  - [www.cool-mcu.com](http://www.cool-mcu.com)

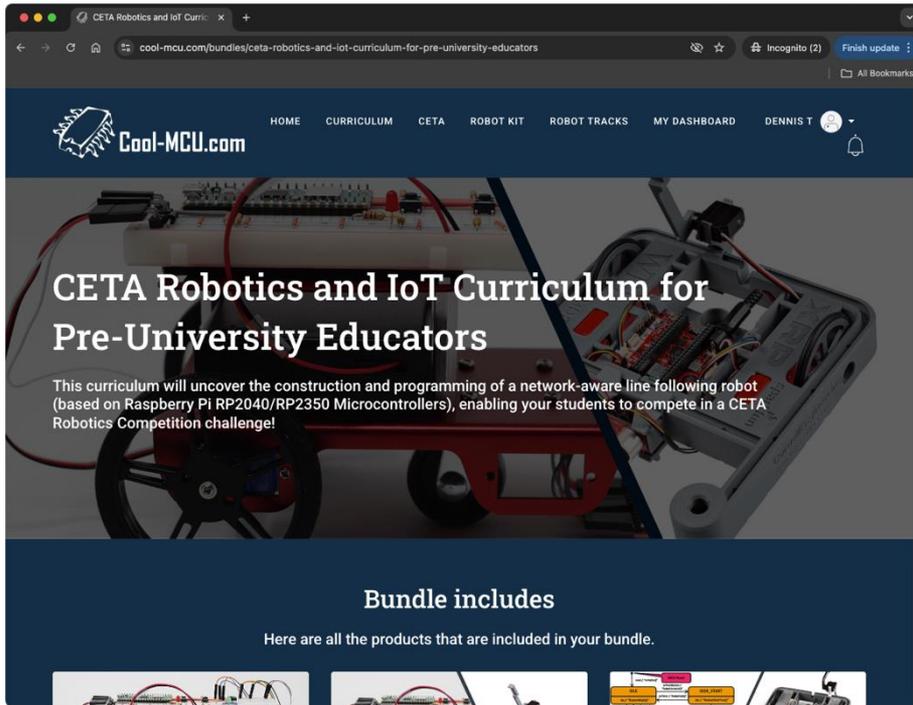
# Goal

- Use the XRP Robot to Incorporate STEM concepts into your teaching practice
  - Embedded Computing Systems
  - Sensors
  - Actuators
  - Software Design & Programming
  - Networking & IoT
- Create an engaging learning experience for your students



# The Learning Program

## “CETA Robotics & IoT Curriculum”



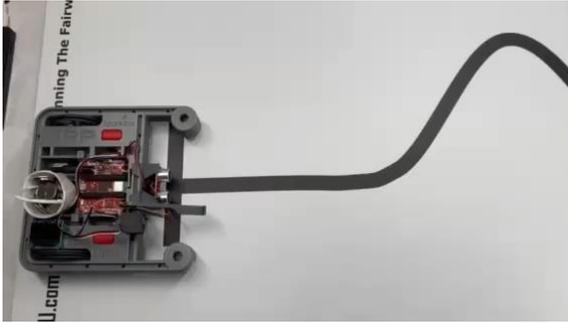
The screenshot shows a web browser displaying the Cool-MCU.com website. The page features a dark blue header with the logo and navigation links: HOME, CURRICULUM, CETA, ROBOT KIT, ROBOT TRACKS, MY DASHBOARD, and DENNIS T. The main content area has a background image of a robot and text that reads: "CETA Robotics and IoT Curriculum for Pre-University Educators". Below this, a paragraph states: "This curriculum will uncover the construction and programming of a network-aware line following robot (based on Raspberry Pi RP2040/RP2350 Microcontrollers), enabling your students to compete in a CETA Robotics Competition challenge!". A section titled "Bundle includes" is visible at the bottom, with the text "Here are all the products that are included in your bundle." and several small images of components.

- Multi-course curriculum covering a variety of STEM topics
- Targets Grade 11 & 12 Computer Engineering curriculum
- Delivered asynchronously using the [www.cool-mcu.com](http://www.cool-mcu.com) LMS
- With an Engineer in the loop for regular Q/A and technical support

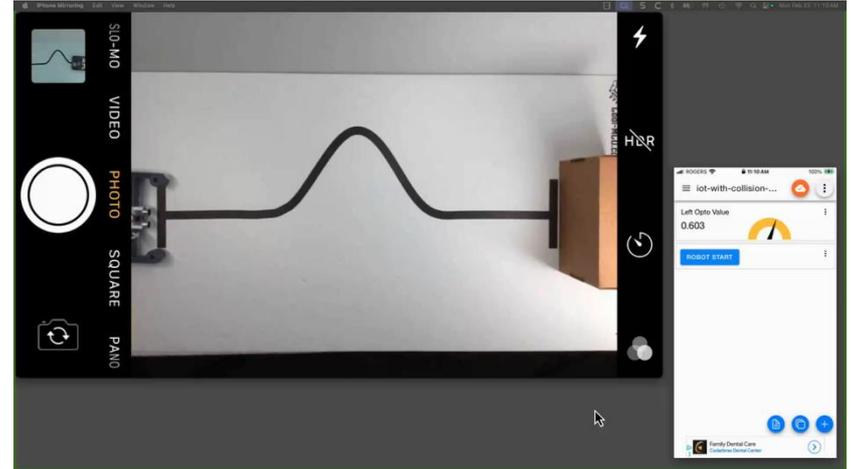
# Learning Context

View/Download the CETA 2026 Rulebook: <https://www.cool-mcu.com/pages/ceta>

## Challenge 1: "Running the Fairway"



## Challenge 2: "IoT & Collision Avoidance"

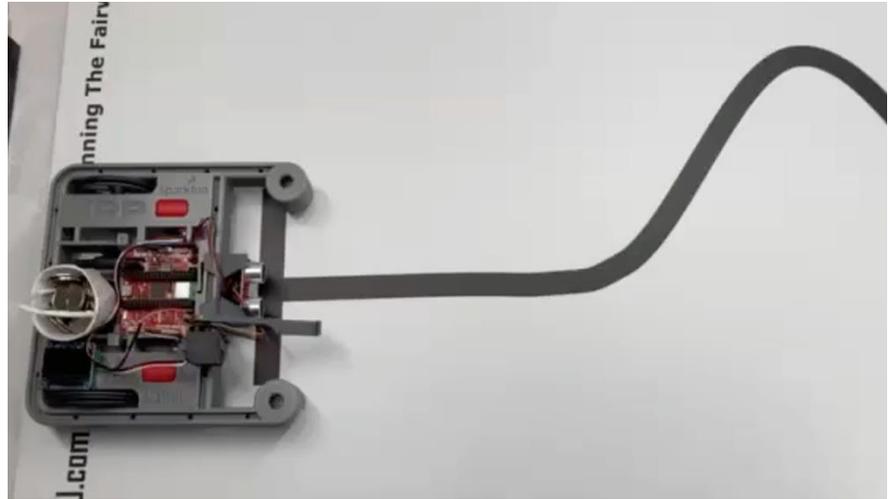


## Challenge 3: "Bucket Challenge"



# Exit Competency

Upon completion of the workshop, you will be able to **construct a robot**, then **code** a basic program that **successfully completes** the basic “Running the Fairway” CETA Robotics Challenge



# CETA Robotics & IoT Curriculum Courses

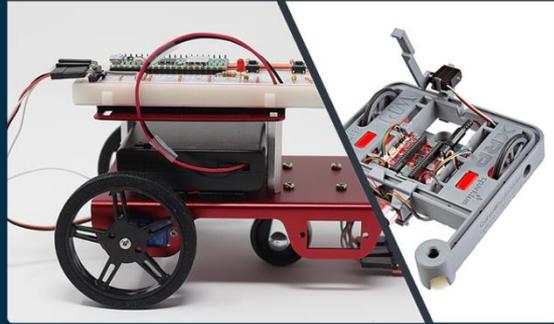
- “CETA Robotics Workshop 1 – Basic I/O & Control”
  - Introduce the 3 robot platforms
    - CETA IoT Robot
    - SFE XRP/XRP Beta Robots
  - Covers the Microcontroller, Robot Outputs, Robot Inputs, Programming in Arduino
  - Build/Test the Robot

CETA Robotics Workshop 1 -  
Basic I/O & Control (19-  
00039A)

---

0%

[See Overview](#) [Start Course](#)



# Some lessons/labs are robot specific!

## Complete the lesson/lab for your robot!

CETA Robotics Workshop 1 - Basic I/O & Control (19-00039A)

0% complete

Search by lesson title

- Welcome to the Course 0/6
- Basics of Programming 0/5
- Robot Control 0/6
- Agenda
  - CETA IoT Robot Controller Overview**
  - SFE XRP Beta Controller Overview
  - SFE XRP Controller Overview
- RP2040 & RP2350 Microcontroller Architecture
- ctalib "board" module
- Lab 1. Blink (CETA IoT Robot) 0/7

COMPLETE & CONTINUE →

CETA Robotics Workshop 1 - Basic I/O & Control (19-00039A)

0% complete

Search by lesson title

- Welcome to the Course 0/6
- Basics of Programming 0/5
- Robot Control 0/6
- Lab 1. Blink (XRP Beta Robot) 0/7**
- Lab 1. Blink (CETA IoT Robot) 0/7
- Embedded Programming Concepts 0/3
- Robot Outputs 0/8
- Lab 2. Controlling DC Motors (CETA IoT Robot) 0/8

COMPLETE & CONTINUE →

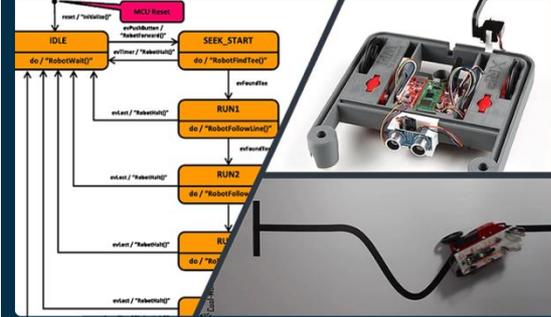
# CETA Robotics & IoT Curriculum Courses

- “CETA Robotics Workshop 2 – Embedded Software Design Concepts”
  - Learn how to capture knowledge using models
  - Learn how to translate models into C-Code in Arduino
  - Learn about PID Control and how to follow a line
  - Successfully complete the basic line follower challenge

CETA Robotics Workshop 2 -  
Coding Robot Behavior (19-  
00041A)

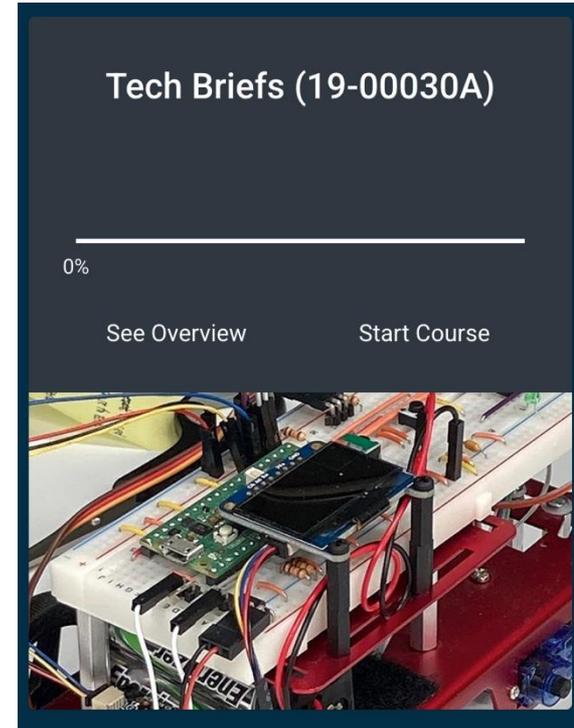
0%

[See Overview](#) [Start Course](#)



# CETA Robotics & IoT Curriculum Courses

- “Tech Briefs” Lessons provide short tech tips and tricks to improve your robot
  - **cetalib** Arduino Library
  - Ultrasonic Range Sensing
  - OLED Display
  - Servo-Arm “Lift” capability
  - Gyro/Accelerometer



# Tech Brief #19

- This reading list covers the knowledge and skills needed to complete the 3 challenges
- Use it as a guide through the courseware

Go to Dashboard

Display to the CETA IoT Robot

- TB #12: Using the cetalib "oled" module functions 0/1
- TB #13: Add a Servo Motor and Lift Arm to the CETA IoT Robot 0/7
- TB #14: Using the cetalib "servoarm" module functions 0/1
- TB #15: Completing the "Bucket Challenge" (CETA IoT Robot) 0/8
- TB #15: Completing the "Bucket Challenge" (XRP/XRP Beta Robot) 0/8
- TB #16: Using the cetalib "mqttc" module functions 0/4
- TB #17: Using a local MQTT Broker for IoT Projects 0/7
- TB #18: Completing the "IoT with Collision Avoidance" challenge 0/1
- TB #19: CETA Robotics Competition Reading List and Solution Sketches 0/4

- Purpose and Overview
- Prerequisites
- Student Files: CETA 2026
- Reading List: CETA 2026

Reading List: CETA 2026

CETA Robotics Competition 2026:

**Challenge 1: "Running the Fairway"**

Skills: Construct the robot, and use the "cetalib" library to program it to complete this basic challenge.

- [CETA Robotics Workshop 1](#)
- [CETA Robotics Workshop 2](#)

**Challenge 2: "IoT & Collision Avoidance"**

Skills: Add Internet Connectivity and Ultrasonic Range detection capability and program it to complete this challenge.

IoT Lessons:

- [RPI Pico IoT Workshop 1: "Message From the Instructor - What is the Internet of Things?"](#)
- [MQTT Basics: "TCP Network Connections"](#)
- [MQTT Basics: "What is MQTT?"](#)
- [Tech Brief 16: "Using the cetalib "mqttc" module Functions"](#)

Collision Avoidance Lessons:

- [Tech Brief 3: "Add Collision Detection to the CETA IoT Robot"](#)
- [Tech Brief 10: "Using the cetalib "rangefinder" module Functions"](#)

Running the Challenge:

- [Tech Brief 18: "Completing the 'IoT With Collision Avoidance' challenge"](#)

**Challenge 3: "Bucket Challenge"**

Skills: Add a servo motor and lift arm and program it to complete this challenge.

- [Tech Brief 13: "Add a servo motor and lift arm to the CETA IoT Robot"](#)
- [Tech Brief 14: "Using the cetalib "servoarm" module Functions"](#)

Running the Challenge:

- [Tech Brief 15: "Completing the Bucket Challenge \(CETA IoT Robot\)"](#)
- [Tech Brief 15: "Completing the Bucket Challenge \(XRP/XRP Beta Robot\)"](#)

COMPLETE & CONTINUE →

# Teaching Robot: SFE XRP/XRP Beta Robot Platform!

- A product of the Experiential Robotics Consortium
  - <https://experientialrobotics.org>
- “XRP Robot” Kits available from Sparkfun Electronics and Digikey



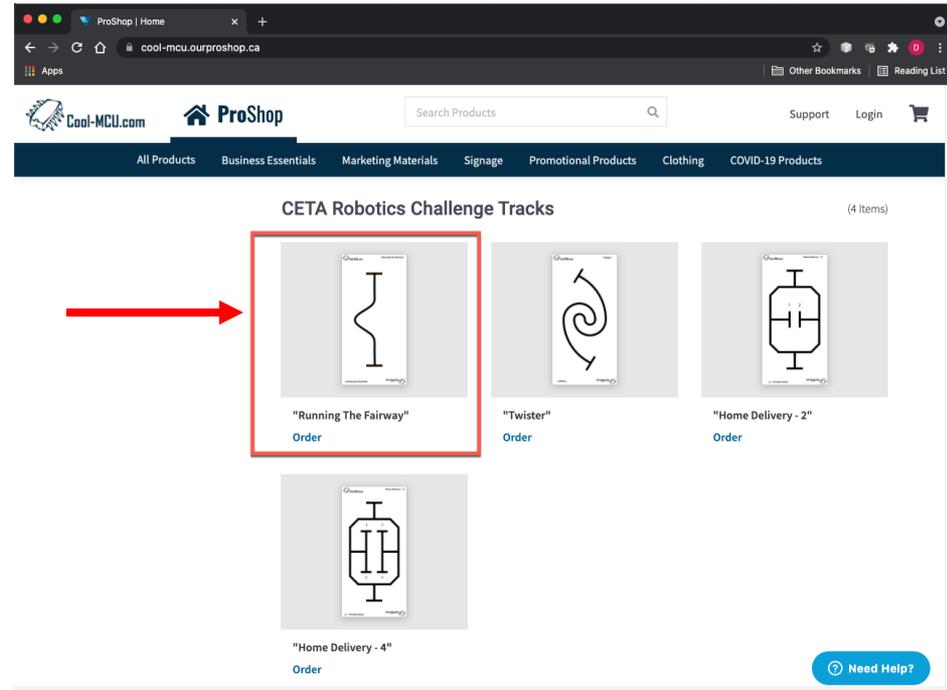
SFE XRP Robot Platform



SFE XRP Beta Robot Platform

# CETA Challenge Track

- You need the “**Running The Fairway**” track
- Available from our Vistaprint ProShop:
  - <https://cool-mcu.ourproshop.ca/>
- .pdfs of all tracks are available in the “tracks” subfolder in the student files
  - Send to your own printer



# “Guided Discovery” Learning Experience

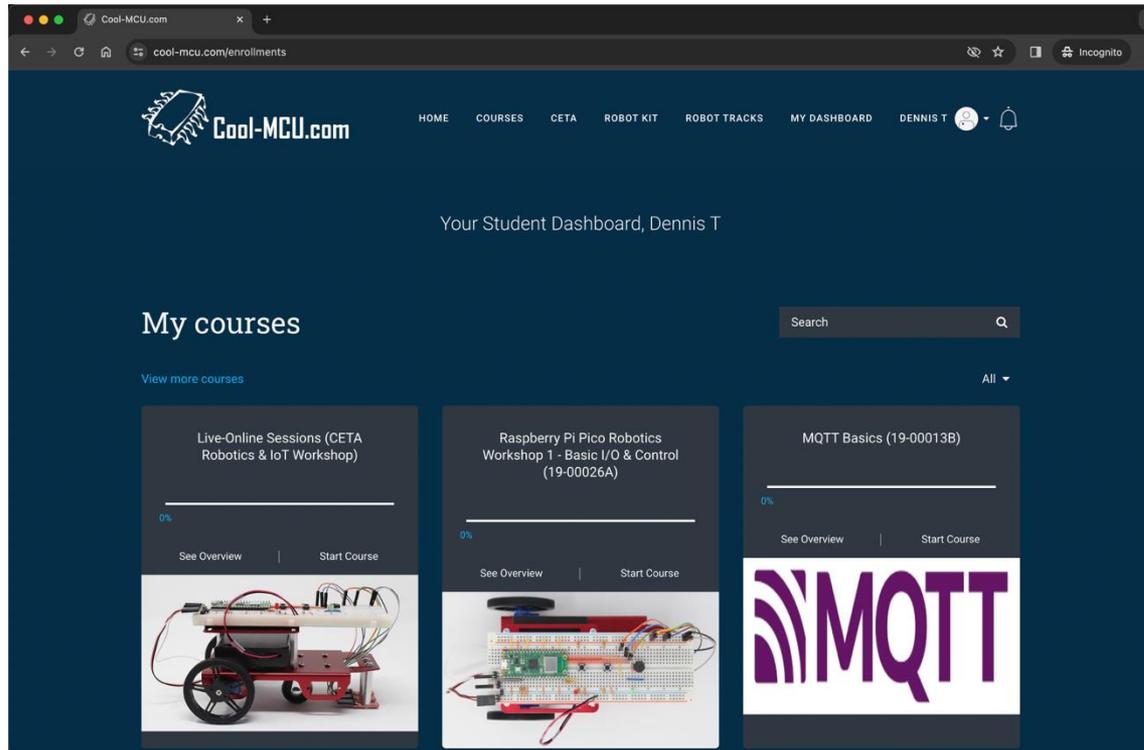
- Enroll on the [www.cool-mcu.com](http://www.cool-mcu.com) LMS
- Work at your own pace through video/text lessons and lab exercises
- Email support via [info@cool-mcu.com](mailto:info@cool-mcu.com)
- **Educator Community Chat Room**
  - Shared online learning space
- Attend weekly live Zoom sessions for troubleshooting and feedback
  - Thursdays @ 7:00pm-8:00pm EST



The screenshot shows the homepage of the Cool-MCU.com Learning Management System. At the top, there is a dark blue header with the Cool-MCU.com logo on the left and navigation links for 'CETA', 'MY DASHBOARD', and 'DENNIS C.' on the right. Below the header is a large image of people working at computers in a classroom or lab. Overlaid on this image is the text 'Welcome STEM Educators!' and 'Kick-Start Your Robotics Curriculum!'. Below the image is a dark blue section with the heading 'Learn to Code' and the subtext 'Build skills in Microcontrollers, Electronics and Programming'. At the bottom of this section is a smaller image of a robot with a microcontroller board on top. Below the robot image is a short paragraph of text: 'Targeting STEM educators and makers, our LMS is dedicated to helping you uncover the basic principles of computer architecture and programming using microcontrollers, culminating with the construction'.

# Student Dashboard

- Lists all enrolled courses and communities

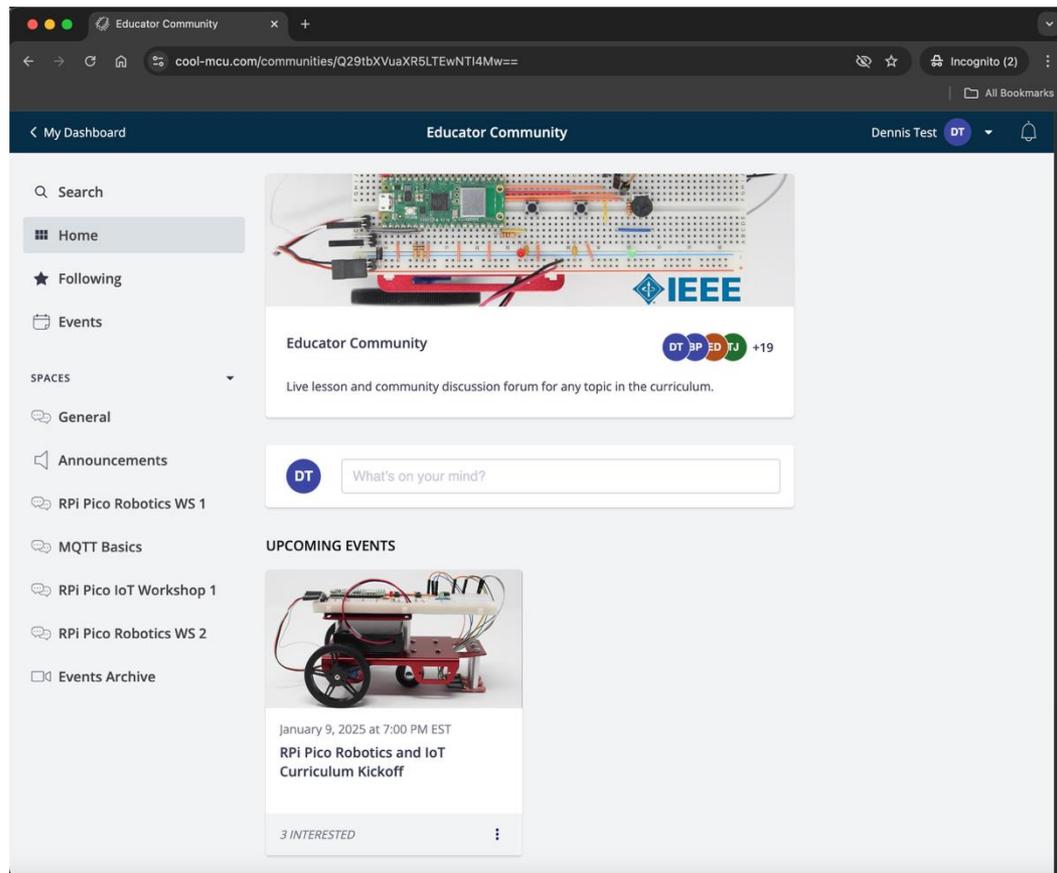


The screenshot displays the Cool-MCU.com Student Dashboard. The browser address bar shows the URL `cool-mcu.com/enrollments`. The page features a dark blue header with the Cool-MCU.com logo and a navigation menu including `HOME`, `COURSES`, `CETA`, `ROBOT KIT`, `ROBOT TRACKS`, `MY DASHBOARD`, and `DENNIS T`. Below the header, the text "Your Student Dashboard, Dennis T" is displayed. The main section is titled "My courses" and includes a search bar and a "View more courses" link. Three course cards are visible, each with a progress bar at 0% and buttons for "See Overview" and "Start Course":

- Live-Online Sessions (CETA Robotics & IoT Workshop)**: Accompanied by an image of a robot.
- Raspberry Pi Pico Robotics Workshop 1 - Basic I/O & Control (19-00026A)**: Accompanied by an image of a Raspberry Pi Pico board.
- MQTT Basics (19-00013B)**: Accompanied by the MQTT logo.

# Educator Community

- Shared online learning space
  - Q/A Chat
  - Live Events
    - Links to past event recordings
  - “Spaces” dedicated to each class
    - Use “General” for general inquiries



The screenshot displays the Educator Community website interface. At the top, the browser address bar shows the URL `cool-mcu.com/communities/Q29tbXVuaXR5LTUwNTI4Mw==`. The page header includes a navigation bar with "My Dashboard", "Educator Community", and a user profile for "Dennis Test". A search bar is located at the top left. The main content area features a large image of a breadboard with an RPi Pico and various components, with the IEEE logo. Below the image, the "Educator Community" section is highlighted, showing a description: "Live lesson and community discussion forum for any topic in the curriculum." A text input field with the placeholder "What's on your mind?" is visible. The "UPCOMING EVENTS" section lists an event: "RPI Pico Robotics and IoT Curriculum Kickoff" on January 9, 2025, at 7:00 PM EST, with 3 interested users.

# Weekly Schedule & Agenda

- 16-week schedule of assigned reading and lab exercises are found in "Guided Reading Schedule" course

Guided Reading Schedule - Cool-MCU.com

Go to Dashboard

## Guided Reading Schedule

0% complete

Search by lesson title

- Week #1 0/1
- Agenda** TEXT
- Week #2 0/1
- Week #3 0/1
- Week #4 0/1
- Week #5 0/1
- Week #6 0/1

### Agenda

#### Week #1

#### Agenda:

Welcome to the RPI Pico Robotics & IoT Curriculum for Pre-University Educators

Overview: Raspberry Pi Pico Robotics Workshop 1

Lab 1 Demo

Assigned Reading:

Raspberry Pi Pico Robotics Workshop 1

- [Module 0: Welcome to the Course](#)
- [Module 1: Basics of Programming](#)
- [Module 2: Raspberry Pi Pico W Board](#)
- [Lab 1. Blink](#)



My Dashboard - Cool-MCU.com

cool-mcu.com/enrollments

HOME

## Guided Reading Schedule

0%

See Overview Start Course

# LMS Enrollment

- **Enrolled until 07 May 2026**
- Options for continued LMS use
  - **CDN \$ 240/year for an all-access membership**
    - email [info@cool-mcu.com](mailto:info@cool-mcu.com) if interested
  - **IEEE-Sponsored Program for Jan 2027**
    - Registration beginning in Oct 2026
    - Watch for the notice in the STEM Muse Newsletter!
    - **Seating is limited**
    - \$ 75 registration fee

# Terms of Use

- .pdfs of PowerPoint and Lab Manuals may be shared with your students
- GitHub published code & YouTube videos may be used freely by your students
- Source code (sketches) for all lab exercises may also be shared with your students
- While logged into the LMS, you may play a video for your students as part of a lesson.
- You are **not allowed** to share your LMS login, or Zoom links with students or other teachers
  - The LMS is for enrolled adult learners.
  - This activity will cause you to be un-enrolled in the courses.
- You are **not allowed** to screen capture live sessions

# Any Questions?

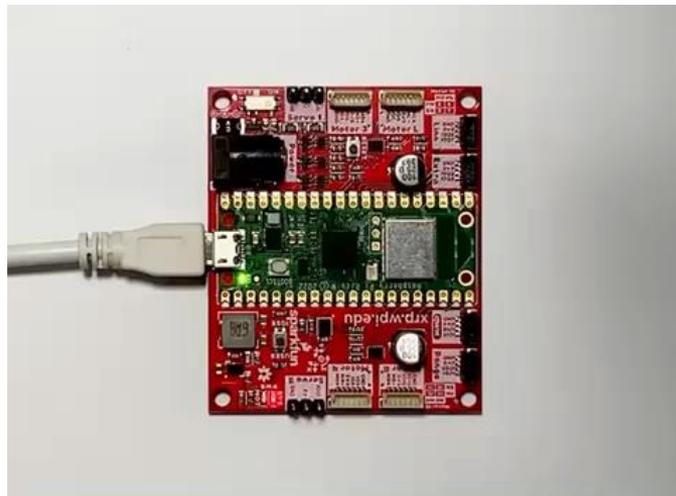
- Let get started and review today's agenda...

# Agenda (Part 1 - Today)

- **Enroll on the LMS & Download this presentation**
  - <https://www.cool-mcu.com/pages/acse-2026>
- **Log in to your Dashboard ([www.cool-mcu.com](http://www.cool-mcu.com))**
  - Review your courses
- **Receive an XRP Beta Robot**
- **Complete these lessons on your own:**
  - **CETA Robotics Workshop 1 – Chapter 1 - “Welcome to the Course”**
    - **“Required Tools, Components & Files”**
      - Install Arduino IDE v2.x
      - Install the latest “Arduino-Pico” Core
    - **“Student Files”**
      - Download/Extract the Student Files & Point Arduino to the “Sketches” sub-folder

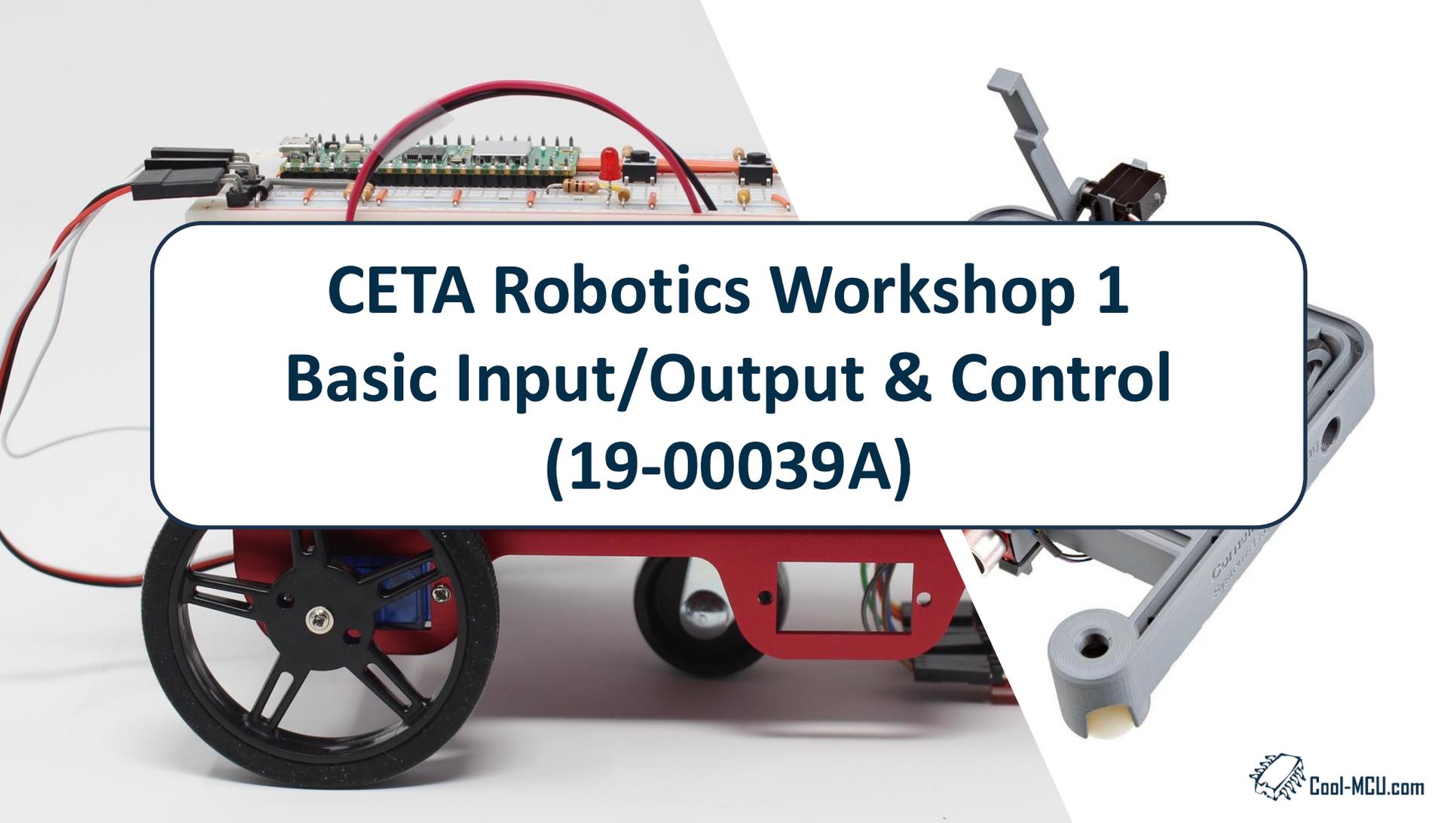
# Agenda (Part 2 - Today)

- **CETA Robotics Workshop 1 (Lectures)**
  - Basics of Programming
  - Robot Control
  - **Lab 1. Blink**
  
- **ACSE 2026 Survey:**
  - <https://bit.ly/acse2026-survey>



# Agenda (Part 3 – Next 2 Months)

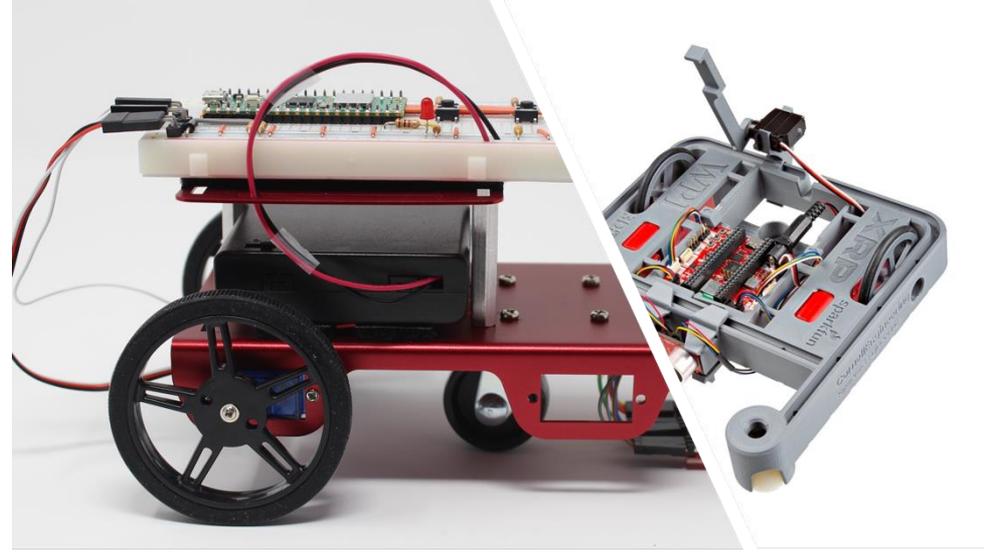
- **Homework** (Self-paced using the LMS and weekly live online check-ins)
  - Finish the **CETA Robotics Workshop 1** class
  - Obtain a challenge Track
  - Finish the **CETA Robotics Workshop 2** class
    - Successfully complete the basic challenge!



# **CETA Robotics Workshop 1**

## **Basic Input/Output & Control**

### **(19-00039A)**

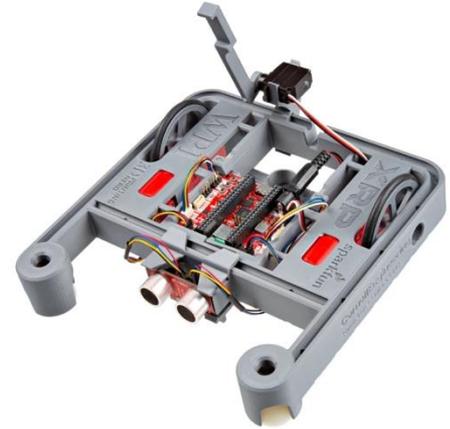
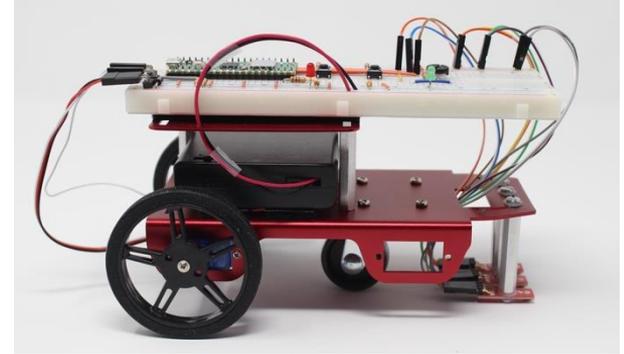


CETA Robotics Workshop 1  
Module 0: Welcome to the Course

# Job Outcome & Learning Objectives

# Job Outcome

- Given the courseware, and either a **CETA IoT Robot** or **SFE XRP/XRP Beta Robot kit**, you will be able to fully construct and verify all major robot functions
  - LEDs, Pushbutton, Potentiometer
  - Motors
  - Opto-sensors



# Learning Objectives

- Understand the fundamentals of the **Raspberry Pi Pico WH and SFE XRP/XRP Beta Controller Boards** (hardware/software)
- Understand the basics of **embedded control firmware design**
- Understand DC motor basics and be able to write a program to drive the motors (the Robot “Outputs”) using **basic Arduino code** as well as the **“cetalib” Library “diffDrive” functions.**
- Understand opto-sensor basics and be able to write a program to read the analog signals to detect a line, using **basic Arduino code** as well as the **“cetalib” Library “reflectance” functions.**

# The “SFE XRP Beta Robot” Platform

- **Autonomous, Self-contained robot. No joysticks!**
- **Inputs**
  - 2 Line sensing Opto emitter/detectors
  - Ultrasonic Range Sensor
  - 2 Motor Encoders
- **Processing**
  - **SparkFun XRP Beta Controller Board with Pico W and RP2040 32-bit MCU**
  - Programmed using industry-standard C/C++ language in **Arduino**
  - **Can also be programmed in Python**
- **Outputs**
  - 2- Brushed DC Motors
- **Connectivity & Security**
  - Integrated Internet (Wi-Fi) & Bluetooth Connectivity
- **Simple chassis**
  - 1-battery pack (4xAA)
  - No soldering required!



SparkFun XRP Beta Controller Board

# CETA Robotics Workshop 1

1

## Basics of Programming

2

## Robot Control

- **Lab 1. Blink**

3

## Embedded Programming Concepts

4

## Robot Outputs

- Lab 2. Controlling DC Motors

5

## Robot Inputs

- Lab 3. Line Detection

All Content © dBm Signal Dynamics Inc. 2026

# Basics of Programming

1

Microcontrollers & Embedded Systems

2

Solving Problems using a Computer

3

The Arduino Framework

4

The “cetalib” Arduino Library



CETA Robotics Workshop 1  
Module 1. Basics of Programming

# The Arduino Framework

# The Arduino Framework

<https://www.arduino.cc/en/Guide/Introduction>

- An open-source electronics platform based on easy-to-use hardware and software
- Aimed at students without a background in electronics and programming
- Good stepping stone on your path to professional embedded systems development

# Benefits

- Sketch source code is based on ANSI C/C++
- Simplified Program Structure
  - `setup()`, `loop()` functions
- Lots of Built-in Functions and available Libraries
- Excellent hardware abstraction
  - Same code will run on many Arduino Boards
- No Flashing Tools Needed
  - Only a USB Cable

# Limitations

- Arduino is not used for commercial application development
  - Limited Editor (Navigation, Multiple Source Code Files)
  - Restrictive Licenses
  - No Source Code Debugging Capability
  - No Hardware Simulator
  - No Direct Hardware Debugging Capability
  - Limited Control over Compilation Process
  - In a real project, you need to know how the hardware works!

# Programming Model

A sketch is required to implement two functions:

```
void setup() {  
    // called ONCE at the beginning  
}
```

```
void loop() {  
    // called repeatedly  
}
```

# Programming Model

The Arduino IDE adds additional code behind the scenes to create a proper C++ file for compilation

```
int main(void) {  
    init();           // system initialization  
    setup();         // sketch initialization  
    while (1) {  
        loop();      // sketch main loop  
    }  
    return 0;  
}
```

# “Arduino Pico” Core

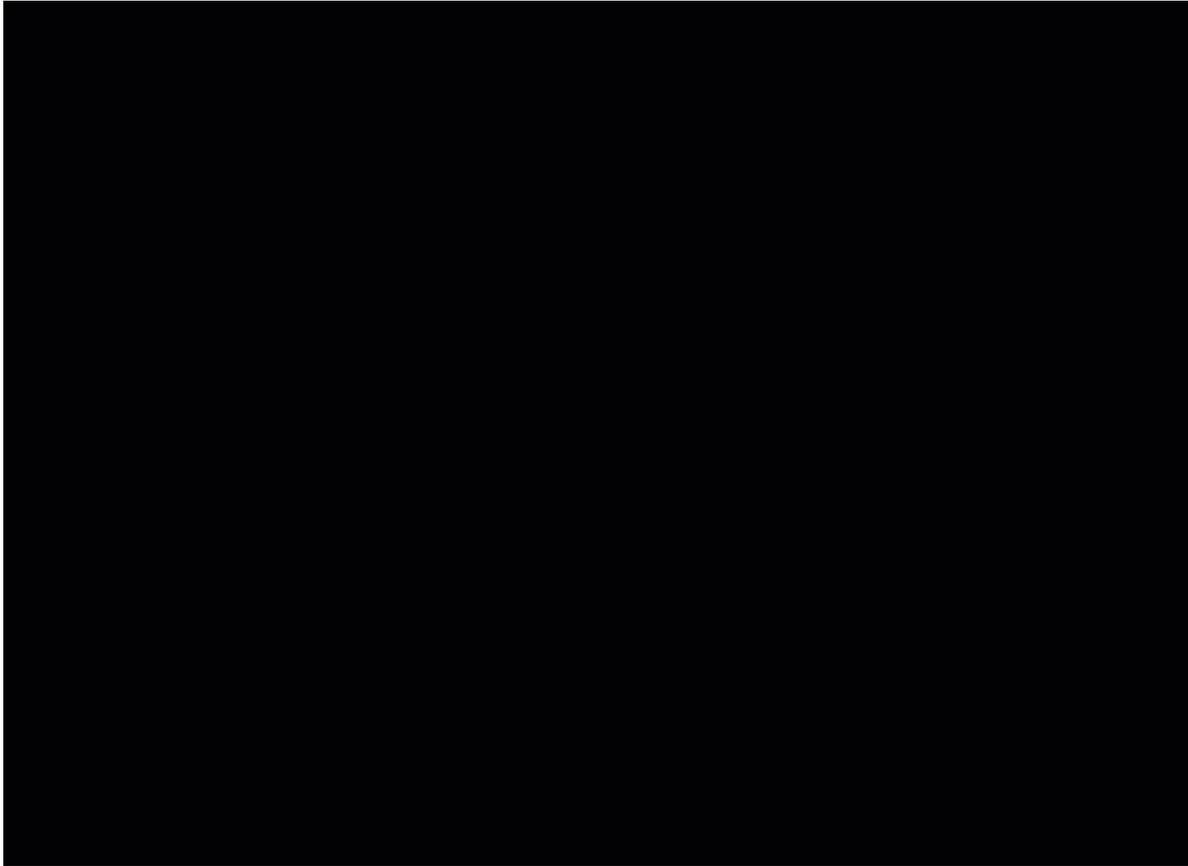
<https://github.com/earlephilhower/arduino-pico>

- **Raspberry Pi Pico Arduino Core**
  - Developed by Earl Philhower
  - Port of the RP2040 processor to the Arduino ecosystem
    - Raspberry Pi Pico SDK and GCC toolchain
    - Supports use of familiar Arduino functions, including WiFi
    - Supports **many** RP2040 boards
- **Documentation**
  - <https://arduino-pico.readthedocs.io/en/latest>

CETA Robotics Workshop 1  
Module 1. Basics of Programming

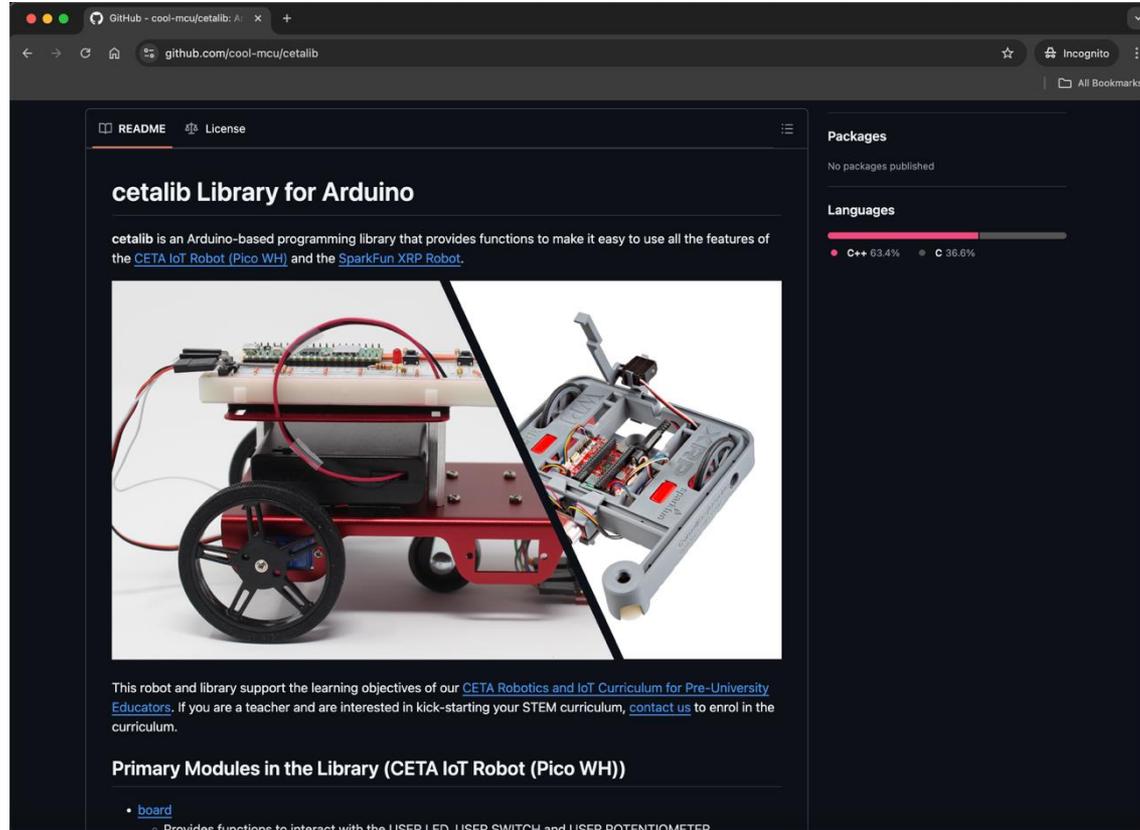
# The “cetalib” Arduino Library

# “cetalib” Arduino Library



# “cetalib” Code Repository

<https://github.com/cool-mcu/cetalib>



The screenshot shows the GitHub repository page for 'cetalib' by 'cool-mcu'. The page is viewed in a browser window with the URL 'github.com/cool-mcu/cetalib'. The repository name 'cetalib Library for Arduino' is prominently displayed. Below the title, a description states that 'cetalib' is an Arduino-based programming library designed to simplify the use of specific hardware features. A central image shows a red robot chassis with a white breadboard and various electronic components mounted on top. To the right of the main content area, there are sections for 'Packages' (indicating no packages are published) and 'Languages', which shows a bar chart with C++ at 63.4% and C at 36.6%. At the bottom, there is a section for 'Primary Modules in the Library (CETA IoT Robot (Pico WH))' with a link to a 'board' module.

GitHub - cool-mcu/cetalib: A

github.com/cool-mcu/cetalib

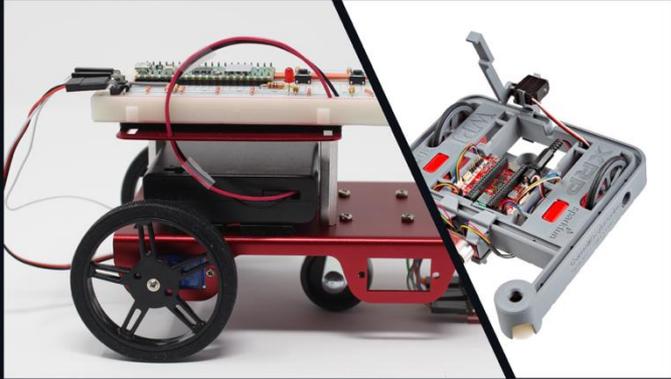
Incognito

All Bookmarks

README License

## cetalib Library for Arduino

cetalib is an Arduino-based programming library that provides functions to make it easy to use all the features of the [CETA IoT Robot \(Pico WH\)](#) and the [SparkFun XRP Robot](#).



This robot and library support the learning objectives of our [CETA Robotics and IoT Curriculum for Pre-University Educators](#). If you are a teacher and are interested in kick-starting your STEM curriculum, [contact us](#) to enrol in the curriculum.

### Primary Modules in the Library (CETA IoT Robot (Pico WH))

- [board](#)
  - Provides functions to interact with the USER\_LED, USER\_SWITCH and USER\_POTENTIOMETER

# Installing the Library into Arduino as an Enrolled Student on the LMS

- Download/extract the student .zip file for the course
- Point the Arduino IDE to the “sketches” sub-folder

The screenshot shows the Cool-MCU.com LMS interface. The top navigation bar includes a 'Go to Dashboard' link and the Cool-MCU.com logo. The main content area is titled 'Student Files' and contains a 'Welcome to the Course' message. Below the message, there is a section for 'Student Files' with instructions to download/extract files. A red box highlights a file named 'rpi-pico-robotics-ws-2-19-00028a-v2025-04-16.zip' with a 'DOWNLOAD' button.

# Manually Installing the Library into Arduino

- Navigate to the repository on Github
  - <https://github.com/cool-mcu/cetalib>
- Manually install the dependent Core and libraries
- Clone/Download/Extract a .zip of this repository
- Follow the steps outlined in this YouTube video:
  - <https://youtu.be/OTbfWM7dy2E>

# Using the Library in a Sketch

```
// Blink the USER LED once every second.  
  
#include <cetalib.h> ←  
  
const struct CETALIB_INTERFACE *myRobot = &CETALIB; ←  
  
void setup() {  
    myRobot->board->initialize(); ←  
}  
  
void loop() {  
    myRobot->board->led_on(); ←  
    delay(100);  
    myRobot->board->led_off(); ←  
    delay(1000);  
}
```

# Primary Documentation

- Github Repository
  - <https://github.com/cool-mcu/cetalib>
- Module Examples
- [YouTube Playlist](#)

# Robot Control

1

CETA IoT Robot Controller Overview

2

**SFE XRP Beta Controller Overview**

3

SFE XRP Controller Overview

4

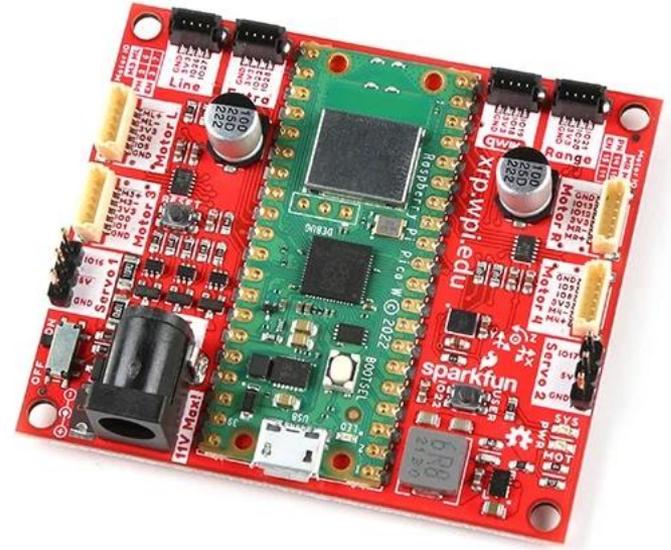
**RP2040 & RP2350 Microcontroller  
Architecture**

5

**cetalib “board” module**

6

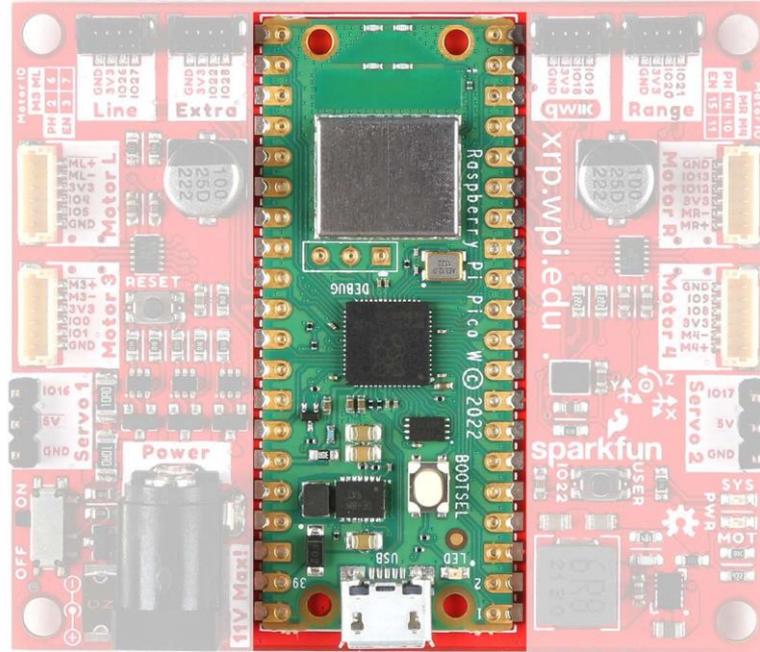
**Lab 1. Blink**



CETA Robotics Workshop 1  
Module 2: Robot Control

# SFE XRP Beta Robot Controller Overview

# Raspberry Pi Pico W

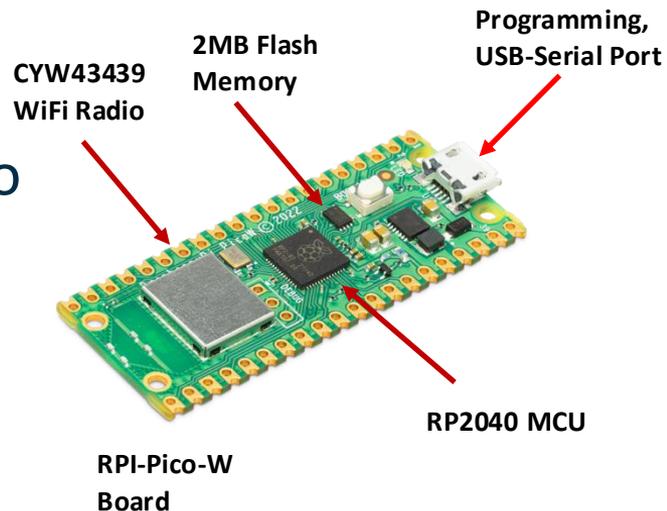


[Raspberry Pi Pico W](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# Raspberry Pi Pico W

PN: SC0918

- 32-Bit, IoT-Capable board.
- RP2040 Dual-Core Cortex M0+ MCU
  - 2MB External FLASH, 264kB SRAM
- Infineon CYW43439 WiFi/Bluetooth radio
  - Built-in Antenna
- Powered from USB, or external DC input.
- **‘WH’** version available with soldered pin headers
- <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>

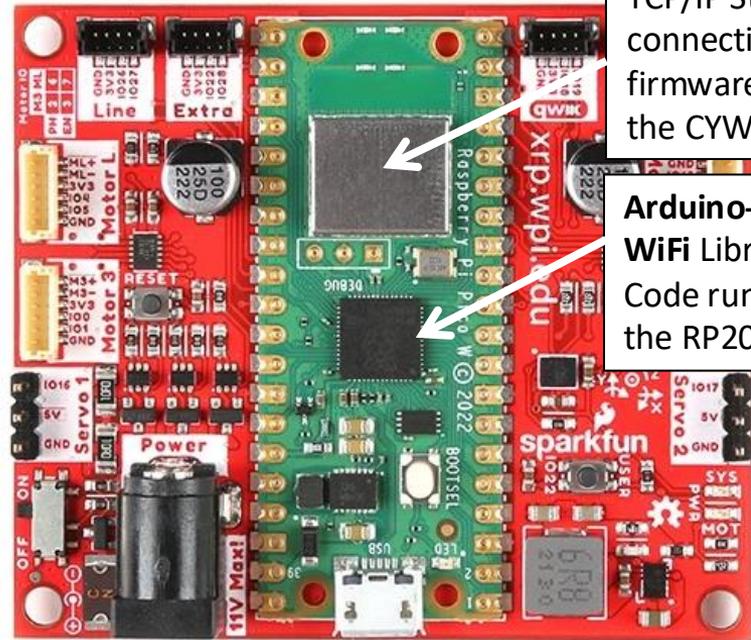


# XRP Beta (Pico W) vs. UNO

Feature	XRP Beta (RPI Pico-W)	Arduino UNO
MCU	RP2040 Cortex M0+ (32-bit)	ATmega328P-PU (8-bit)
Operating Voltage	3.3 V	5 V
Memory	2MB Flash, 264kB SRAM	32kB Flash, 2kB SRAM
Clock Speed	133 MHz (PLL)	16 MHz (Crystal)
Input Voltage	1.8-5.5 V	7-20 V
Digital I/O Pins	30	14
PWM Pins	16	6
Analog Input Pins	4	6
Board Selection	“SparkFun XRP Controller (Beta)”	“Arduino/Genuino UNO”
Wireless Interface	802.11 b/g/n & BT 4.2	---
USB Interface	USB 1.1 Host/Device	---
Cost	CDN \$11	CDN \$40

# Networking Configuration

- **CYW43439** Firmware runs the **TCP/IP Stack** and **WiFi** connections.
- **Arduino-Pico WiFi** Library functions run on the **RP2040 MCU**.
  - Communicate with the CYW43439 firmware via SPI interface
- CYW43439 Firmware binary is bundled with the RP2040 binary and programmed into the radio.
- <https://arduino-pico.readthedocs.io/en/latest/wifi.html>

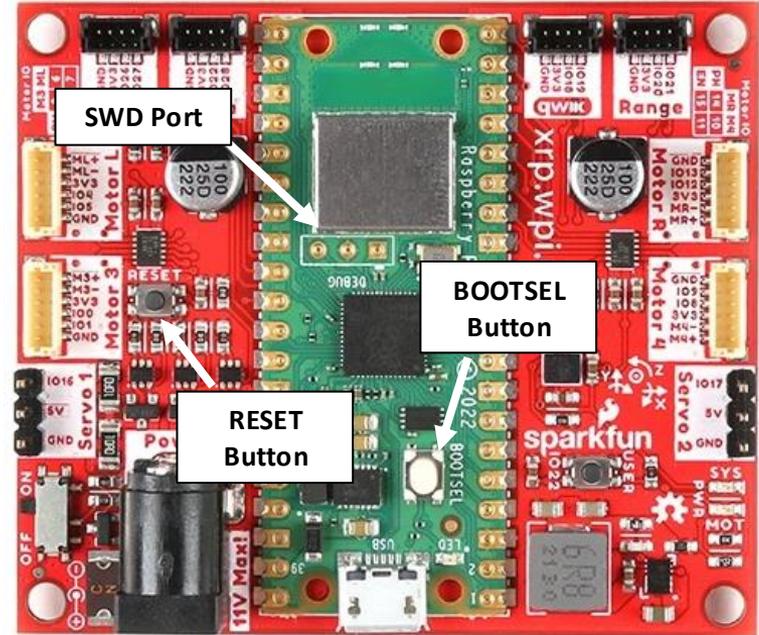


TCP/IP Stack, WiFi connectivity firmware runs on the CYW43439

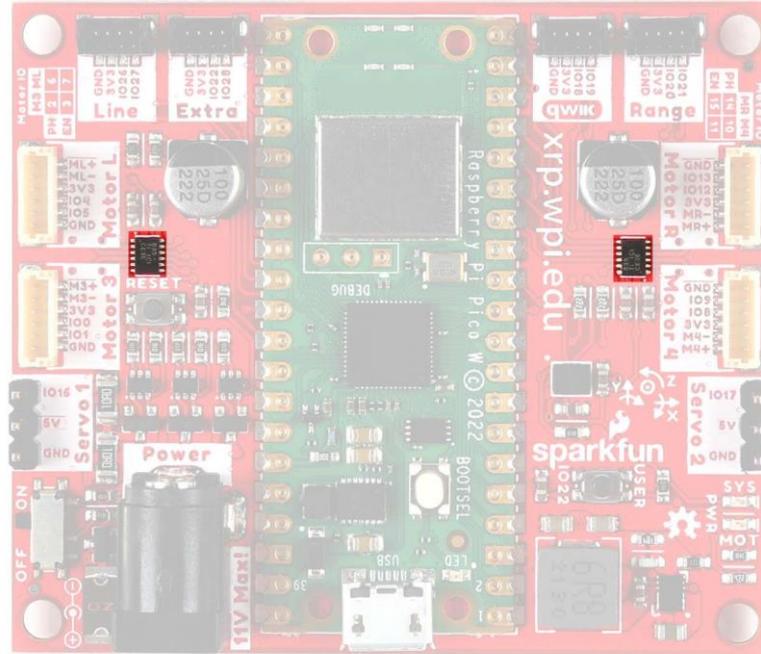
Arduino-Pico WiFi Library Code runs on the RP2040

# Programming the XRP Beta Controller

- **Serial Wire Debug (SWD) port**
  - Requires a dedicated debug probe
- **USB Mass Storage Mode**
  - Press/Hold Reset Button
  - Press/Hold BOOTSEL
  - Release Reset Button
  - Release BOOTSEL
  - Drag .uf2 file onto the “disk”

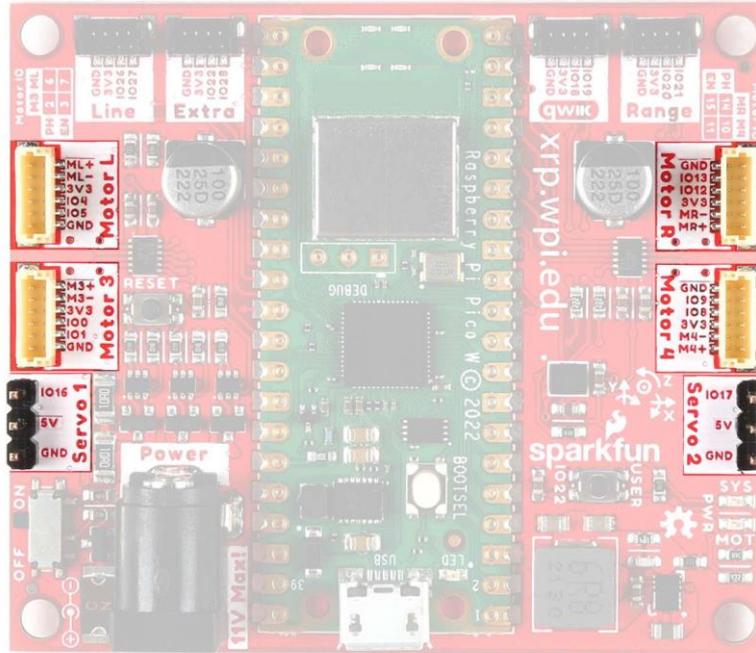


# DRV8835 Motor Drivers



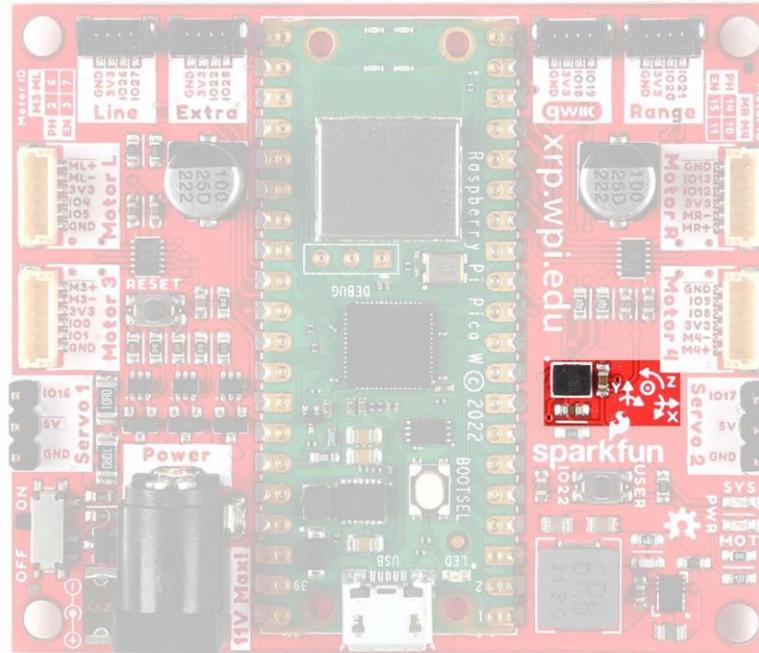
[XRP Controller Beta Motor Drivers](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# Motor Connectors



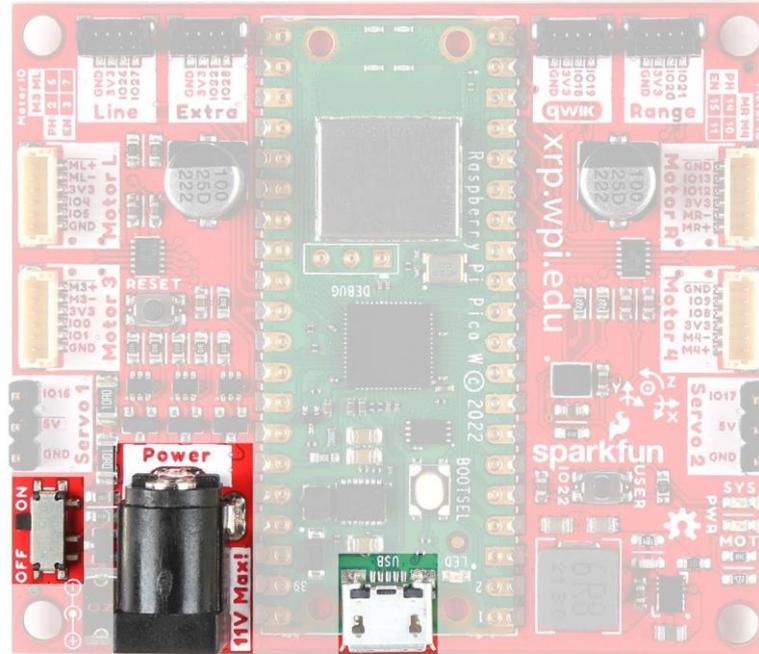
[XRP Controller Beta Motor Connectors](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# LSM6DSO 6-Dof IMU



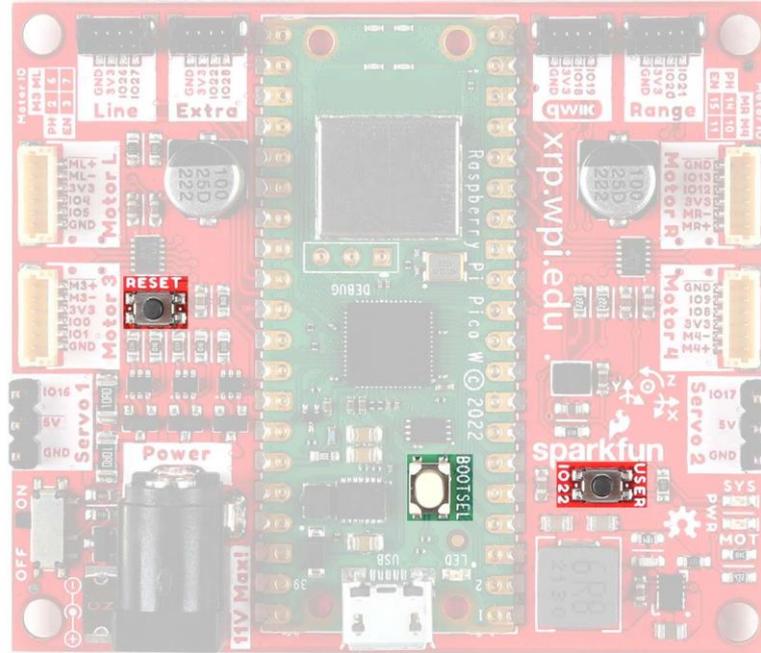
[XRP Controller Beta IMU](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# Powering the Board



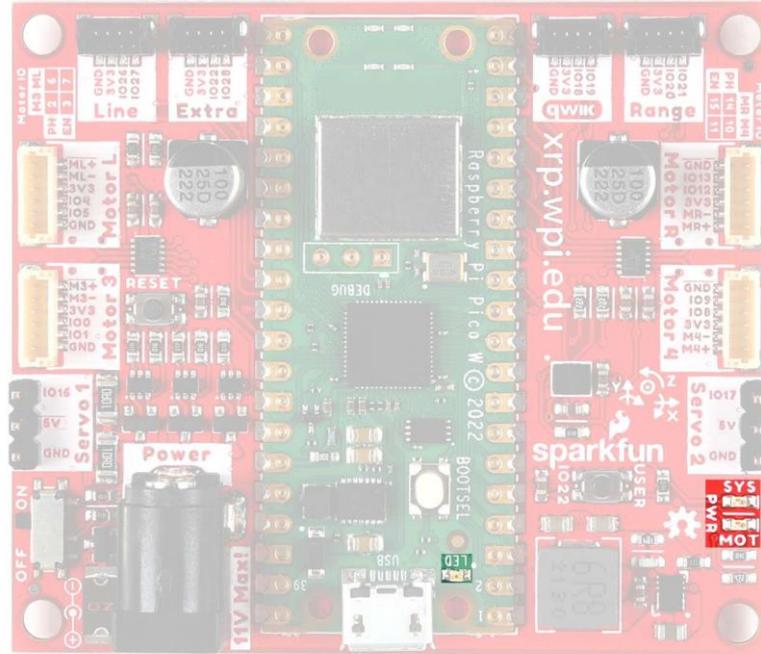
[XRP Controller Beta Power](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# Buttons



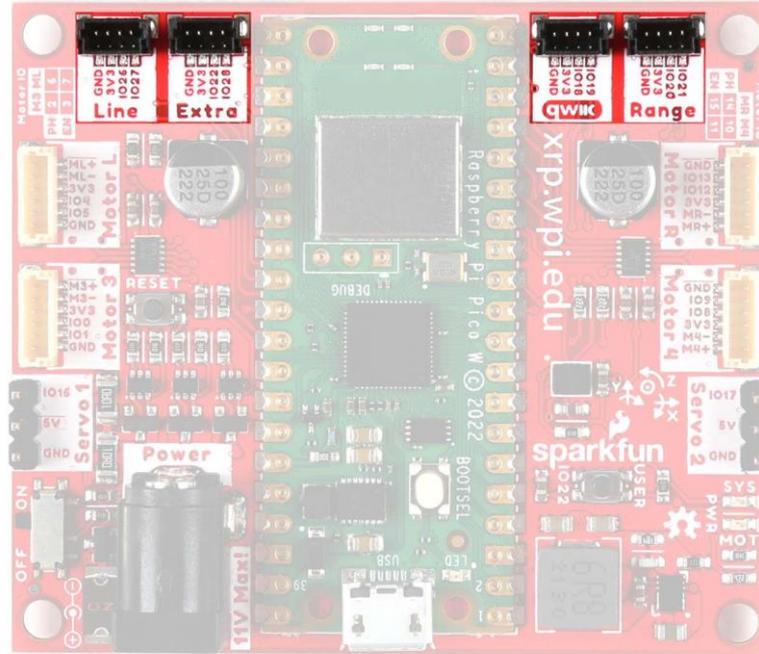
[XRP Controller Beta Buttons](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# LEDs



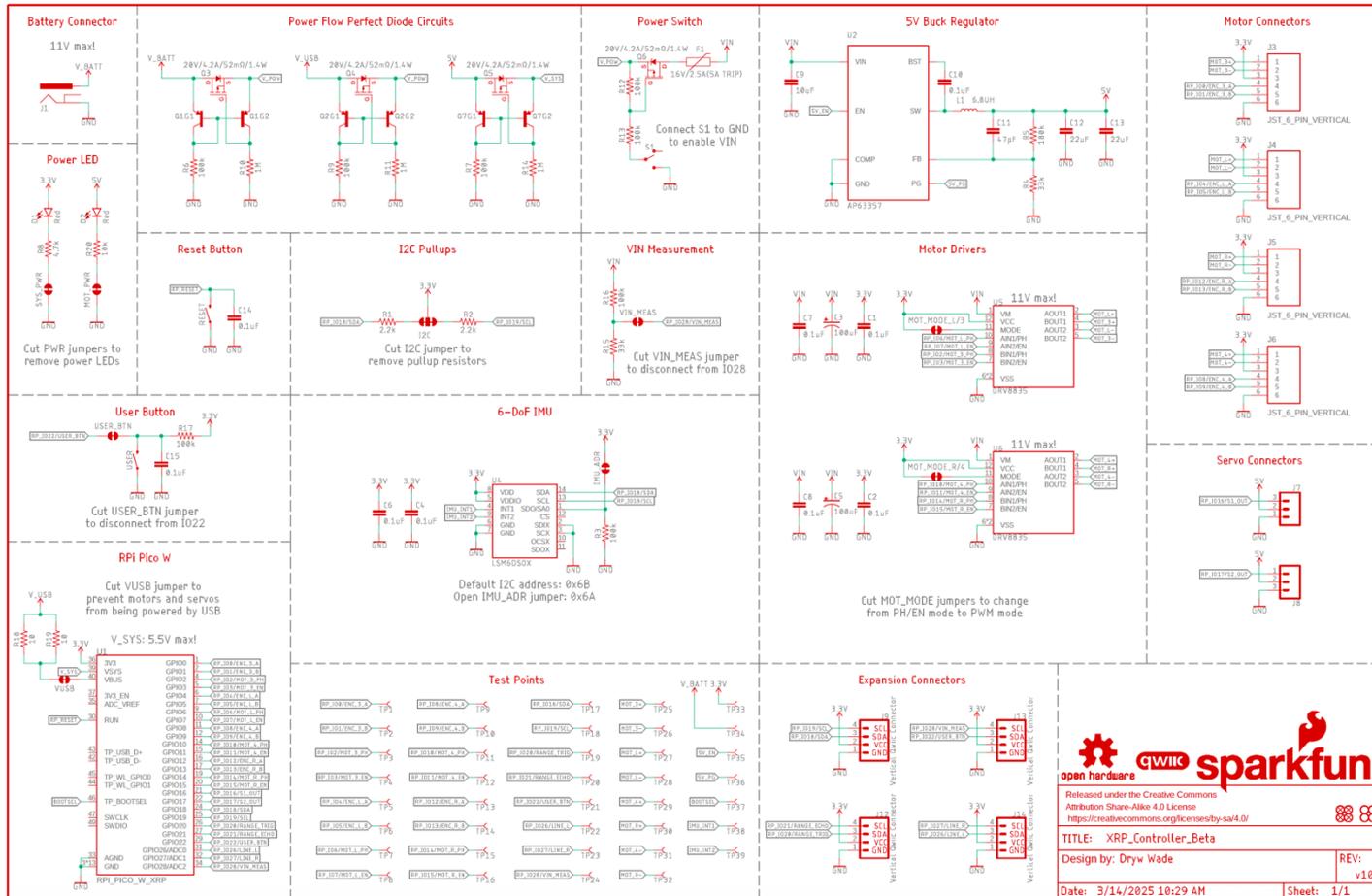
[XRP Controller Beta LEDs](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# Sensor & Qwiic Connectors



[XRP Controller Beta Qwiic](#) (c) SparkFun Electronics, [CC BY-SA 3.0](#)

# Schematic Diagram



**open hardware** **QWIK** **sparkfun**

Released under the Creative Commons Attribution-ShareAlike 4.0 License <https://creativecommons.org/licenses/by-sa/4.0/>

TITLE: XRP\_Controller\_Beta  
Design by: Dryw Wade  
Date: 3/14/2025 10:29 AM

REV: v1.0  
Sheet: 1/1



# Hardware Documentation Page

SparkFun XRP Controller Hardware Overview

## XRP Beta Controller Overview

In this document, we'll take a close look at the heart of the Experiential Robotics Platform (XRP) Kit - Beta, the XRP Controller - Beta. This document outlines all of the parts on this board you'll interact with while building and using the XRP Kit - Beta.

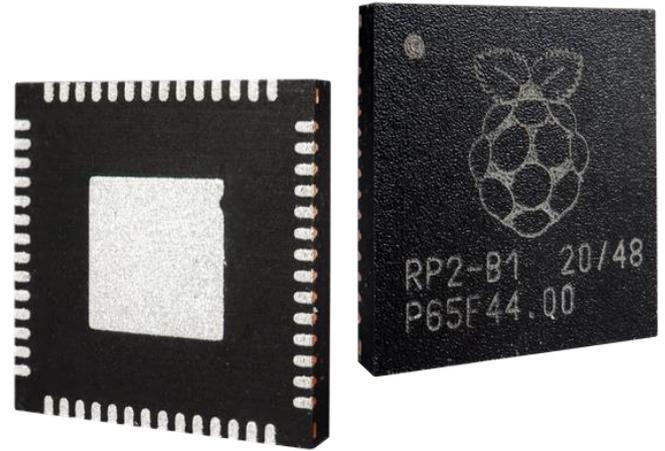
### XRP Controller - Beta Overview

Let's take a broad look at the major components on the XRP Controller - Beta. The photo below points them out along with their names:

- Motor Driver
- Expansion Headers: Line Follower & Color
- Pico W
- Expansion Headers: Range Finder & Oled
- Motor Driver
- Motor Connectors
- Reset Button
- Servo 1 Connector
- On/Off Switch
- Pico W Status LEDs
- SparkFun
- 6 DoF IMU
- Servo 2 Connector
- Power LEDs

Table of contents

- XRP Controller - Beta Overview
- Raspberry Pi Pico W
- DRV8835 Motor Drivers
- LSM6DSO 6-DoF IMU
- Power Components
  - Barrel Jack Connector
  - Pico W USB-Connector
  - Power Switch
- Motor Connectors
  - DC Motor Connectors
  - Servo Motor Connectors
- Expansion Connectors
  - Line Follower Connector
  - Range Sensor Connector
  - Qwiic Connector
  - Extra Connector
- Buttons
- LEDs
- Solder Jumpers
- Pinout Reference Table



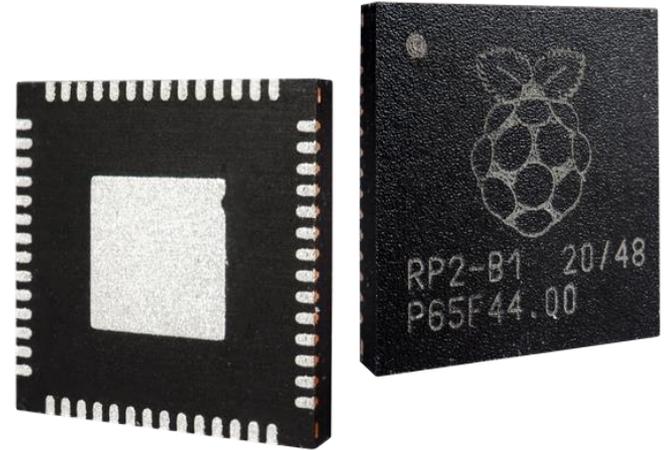
CETA Robotics Workshop 1

Module 2: Robot Control

# RP2040 & RP2350 Microcontroller Architecture

# Introduction

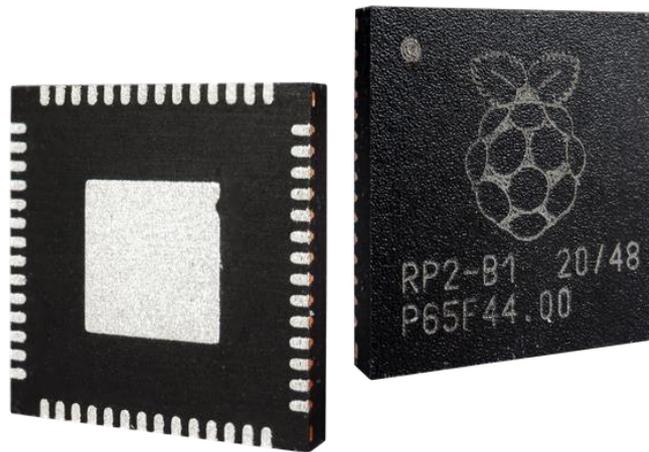
- **RP2040 & RP2350**
  - Debut Microcontrollers from the Raspberry Pi Foundation
  - High performance
  - Low cost
  - Ease of use



<https://www.raspberrypi.com/documentation/microcontrollers/silicon.html>

# RP2040 Key Features

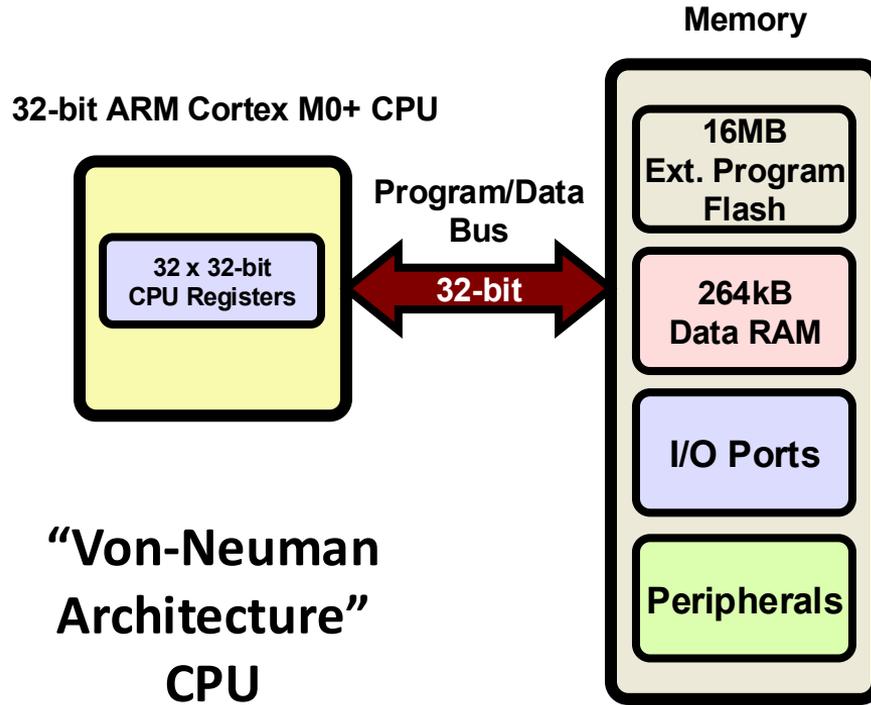
- Dual-Core ARM Cortex-M0+ @ 133MHz
- 264kB SRAM
- Support for up to 16MB ext. Flash
- 30 GPIO Pins
- Peripherals
  - 2 UARTS
  - 2 SPI Controllers
  - 2 I2C Controllers
  - 16 PWM Channels
  - 4 Ch. 12-Bit ADC
  - USB 1.1 Interface



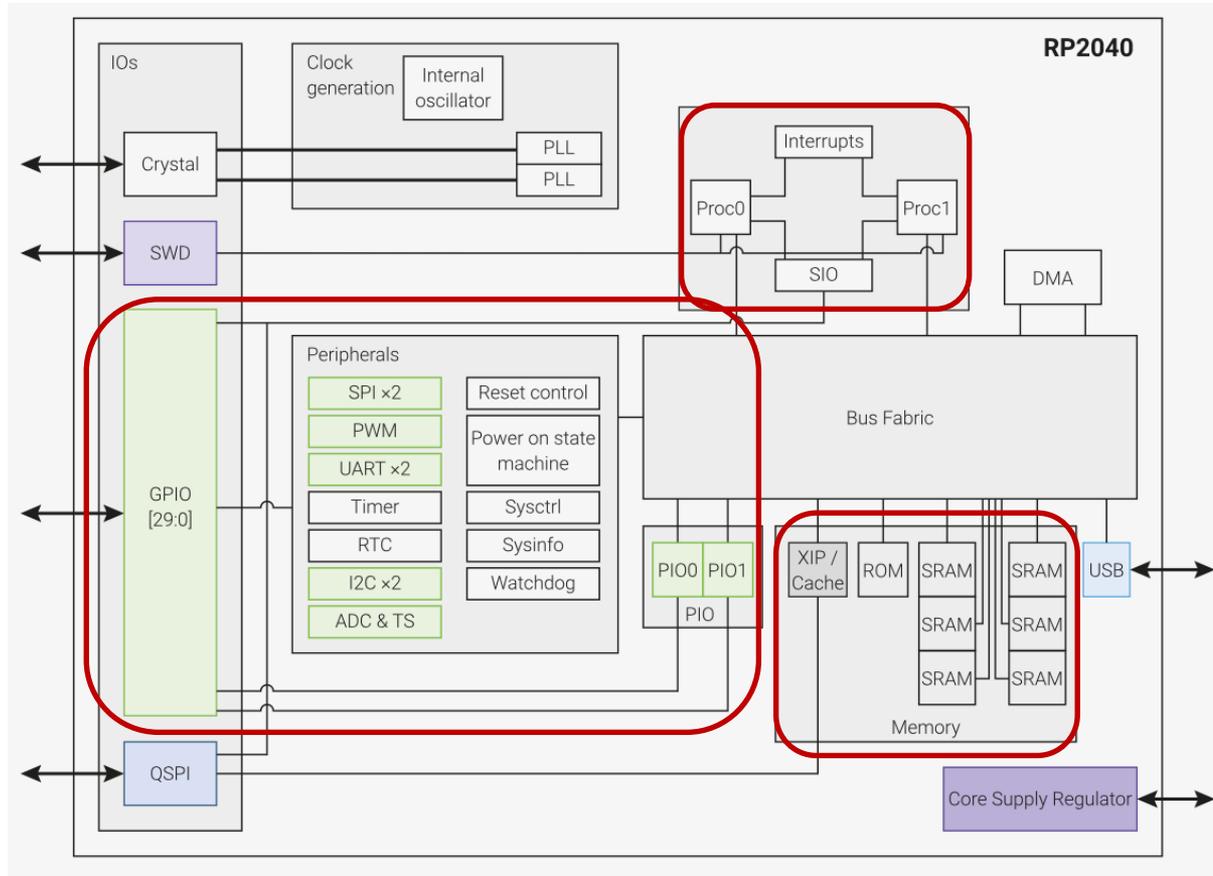
<https://www.raspberrypi.com/documentation/microcontrollers/silicon.html#rp2040>

# Simplified Block Diagram

## RP2040 MCU Architecture



# Detailed RP2040 Block Diagram

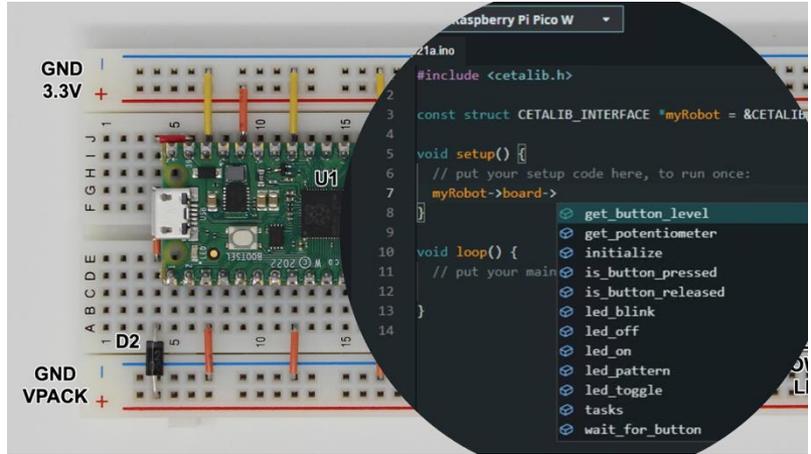


# How big is my Sketch?

```
Blink.ino
1 // 2022-08-09
2 //
3 // Modified Basic Example (Blink) for RPI-Pico-WH-based CETA IoT Robot
4 // Refer to Schematic Diagram (#14-00069A)
5 //
6 // LED ("USER LED") connected to digital pin 14 (GP14)
7 //
8 //
9 /*
10  Blink
11  Turns on an LED on for one second, then off for one second, repeatedly.
12
13  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
14  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
15  the correct LED pin independent of which board is used.
16  If you want to know what pin the on-board LED is connected to on your Arduino model, check
17  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
```

```
Output
/Users/dennis/Library/Arduino15/packages/rp2040/tools/pqt-gcc/4.0.1-8ec9d6f/bin/arm-none-eabi-gcc -Werror-return-type -Wno-psabi -DUSB_PID=0xf0
/Users/dennis/Library/Arduino15/packages/rp2040/tools/pqt-gcc/4.0.1-8ec9d6f/bin/arm-none-eabi-g++ -L/Users/dennis/Library/Caches/arduino/sketch
/Users/dennis/Library/Arduino15/packages/rp2040/tools/pqt-gcc/4.0.1-8ec9d6f/bin/arm-none-eabi-objcopy -Obinary /Users/dennis/Library/Caches/ard
/Users/dennis/Library/Arduino15/packages/rp2040/tools/pqt-python3/1.0.1-base-3a57aed-1/python3 -I /Users/dennis/Library/Arduino15/packages/rp204
/Users/dennis/Library/Arduino15/packages/rp2040/tools/pqt-picotool/4.0.1-8ec9d6f/picotool uf2 convert /Users/dennis/Library/Caches/arduino/sketch
/Users/dennis/Library/Arduino15/packages/rp2040/tools/pqt-gcc/4.0.1-8ec9d6f/bin/arm-none-eabi-size -A /Users/dennis/Library/Caches/arduino/sketch
Sketch uses 321680 bytes (15%) of program storage space. Maximum is 2093056 bytes.
Global variables use 70236 bytes (26%) of dynamic memory, leaving 191908 bytes for local variables. Maximum is 262144 bytes.
```

- Enable “verbose output” in preferences to determine memory usage

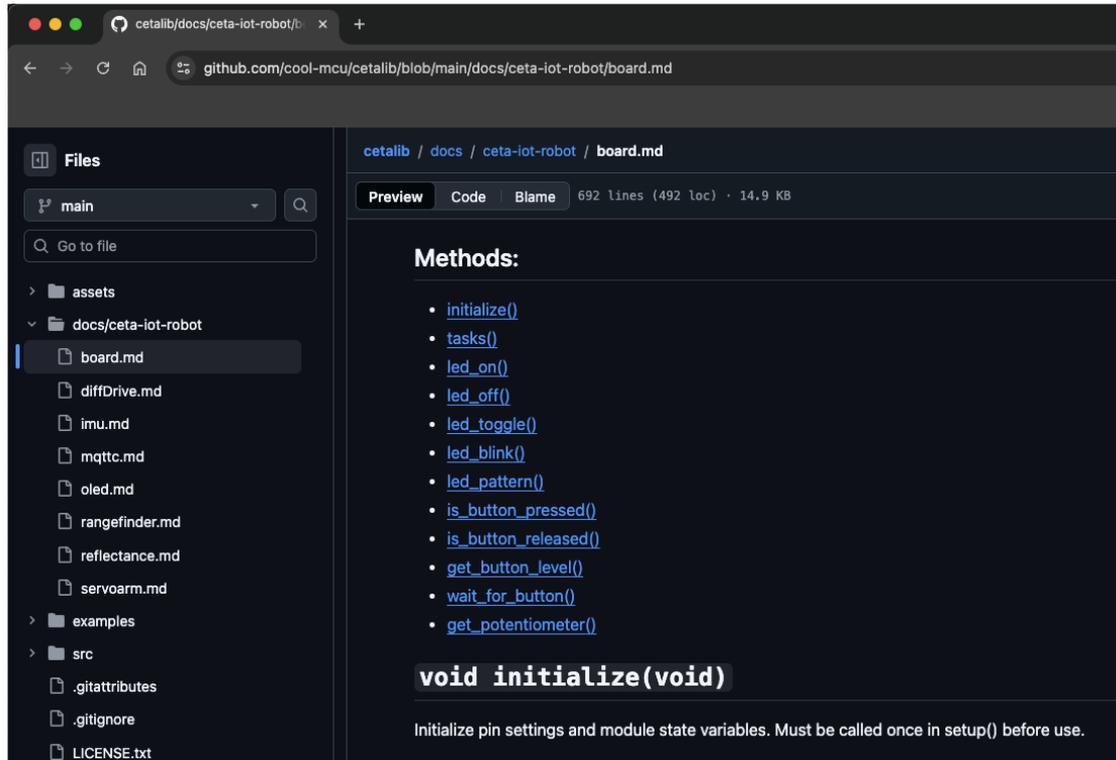


## CETA Robotics Workshop 1

### Module 2: Robot Control

# cetalib “board” module

# Board Module Functions



The screenshot shows a GitHub repository page for the file `board.md` located at `github.com/cool-mcu/cetalib/blob/main/docs/ceta-iot-robot/board.md`. The file is 692 lines long, has 492 loc, and is 14.9 KB in size. The page displays a list of methods and the signature for the `initialize()` function.

**Methods:**

- `initialize()`
- `tasks()`
- `led_on()`
- `led_off()`
- `led_toggle()`
- `led_blink()`
- `led_pattern()`
- `is_button_pressed()`
- `is_button_released()`
- `get_button_level()`
- `wait_for_button()`
- `get_potentiometer()`

**`void initialize(void)`**

Initialize pin settings and module state variables. Must be called once in `setup()` before use.

<https://youtu.be/MFxs2nmxrUc>



CETA Robotics Workshop 1

# Lab 1. Blink (XRP Beta Robot)

# Lab 1 (XRP Beta Robot)

## Blink

- **PURPOSE**

- Test the basic components of the XRP Beta control board (the USER LED, the USER SWITCH, and the Serial Monitor).
- To become comfortable working with the Pico W Controller Board in Arduino
- Learn how to use the `cetalib` "board" functions to interact with the USER LED, USER SWITCH and the USER Potentiometer.

- **PROCEDURE**

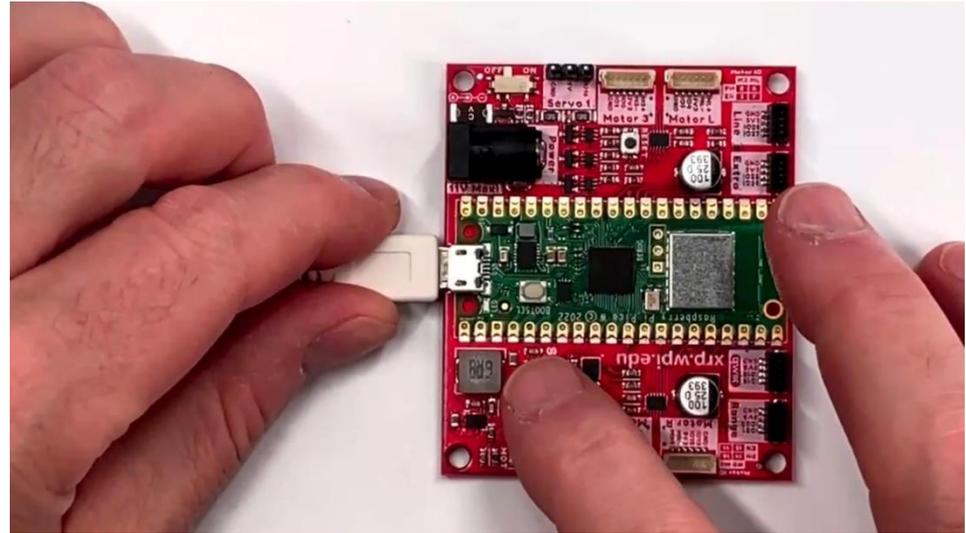
- Connect the XRP Beta Controller Board to you PC
  - Run the **`cetalib`** "board" module example sketches to test the hardware
- Follow the instructions in the lesson pages

# Lab 1 (XRP Beta Robot)

## Blink

- **RESULTS**

- Arduino environment is functional
- XRP Beta Robot control board is functional.
- USER LED, USER SWITCH, USER POT and Serial Monitor on the Robot control board are working



# Lab 1 (XRP Beta Robot)

## Blink

- **CONCLUSIONS**

- **Pico W is a WiFi-based Raspberry Pi Board**

- Can be programmed in Arduino using the **Arduino-Pico** core package

- **Can be powered through the USB connection**

- Motors & Motor Controller will be powered via Battery Pack (VPACK)

- **Programming and Serial Monitor Connection**

- Initially, the Pico W must be in “USB Mass Storage Mode” for programming
- Press/Hold BOOTSEL, Press/Hold RESET Button, Release RESET, Release BOOTSEL
- Subsequent uploads are possible through the virtual COM port which is established by the Pico

- **The cetalib Arduino library contains several modules and methods to simplify robotic application programming on the XRP Beta Robot**

# Thank You!

**Please complete the ACSE Survey:**

**<https://bit.ly/acse2026-survey>**