Developing a Flipped, Interactive Mobile Robotics Course for Computer Science Students

Anthony J. Clark

Pomona College

Claremont, California, United States

https://cs.pomona.edu/classes/cs181r/book/book.html

Quick Acknowledgement

Ella Zhu

Jack Chin

Students helped develop the web-based textbook, robot software, and robot hardware.

They were supported by the Hahn Teaching with Technology Grant at Pomona College

Who are we?

Me

- Assistant Professor
- I am very happy to be here
- This is my first time at RiE
- Research in modeling small, adaptive locomotion robots

Pomona College

- About 1700 students
- Liberal arts college
- Southern California

My Department

- Computer Science
- About 40 majors per year

CS Students

- Complete only 12/32 classes in computer science
- (For reference, 36/43 of my courses were in engineering)
- Roughly 1:1 ratio of male to non-male
- Few will continue with robotics
- Most are interested in multiple majors and broader impacts

CS Courses

- 15 weeks of instruction
- About 12 hours per week

What did we do?

Created a robotics course

Interactive web-based textbook

Flipped course design

Low cost wheeled mobile robot using off-the-shelf parts

Related

Robotics & Robots. Selvaggio. RiE 2026

Unibotics. Pascual-Hernández. RiE 2026

Dive into Deep Learning. Zhang. 2023

Introduction to Robotics. Niku. ZyBooks

EduRob. Heß. RiE 2026

Pico4Drive. Gonçalves. RiE 206

PYTORCH MXNET JAX TENSORFLOW

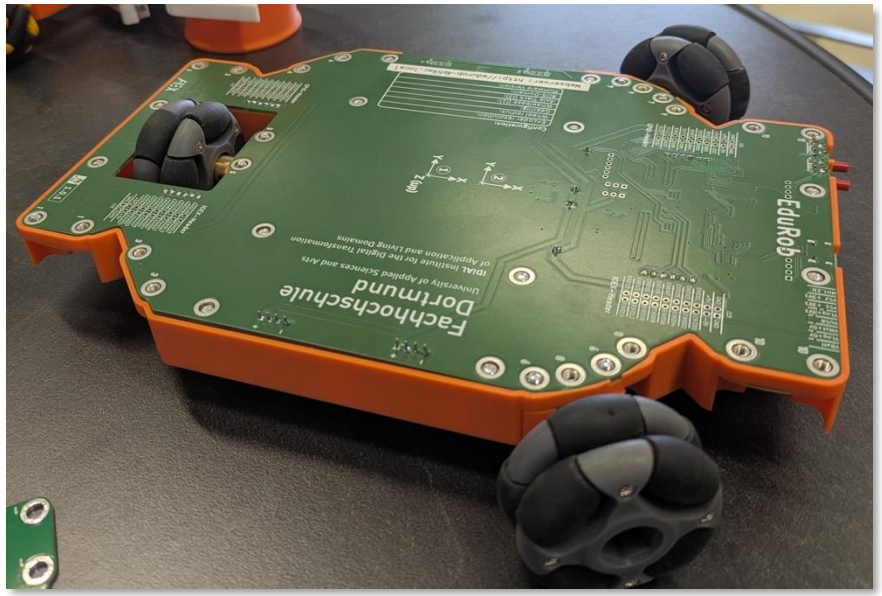
```
n = 10000
a = torch.ones(n)
b = torch.ones(n)
```

Now we can benchmark the workloads. First, we add them, one coordinate at a time, using a for-loop.

PYTORCH MXNET JAX TENSORFLOW

```
c = torch.zeros(n)
t = time.time()
for i in range(n):
    c[i] = a[i] + b[i]
f'{time.time() - t:.5f} sec'
```

```
'0.17802 sec'
```



Motivation

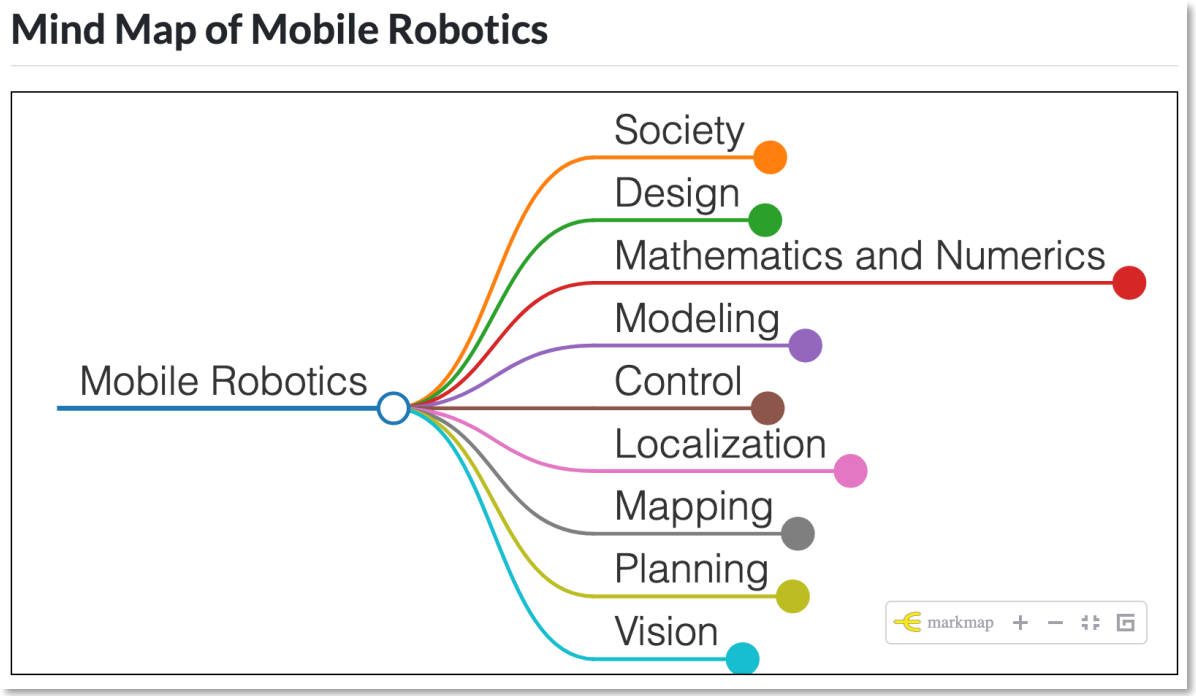
- Many (great) teaching materials already exist
- But none were well-suited to my students (we all tell ourselves)
- I wanted something for
  - Computer science students (not engineering students)
  - Liberal arts students (content should be tied to broader impacts)
  - An active classroom (materials designed for a flipped classroom)
- Something others can take, modify, and contribute back
- Something independent learners can use

Limitations and what we didn't do

- Small scale (number of students, number of offerings)
- No tracking, no login; all client-side; just a git repository
- Relying on results from your studies for best practices
- Open source, but **not** very contributor friendly at the moment
- Less polished than much of what I've seen this week
- A lot will change in future iterations (partly due to LLMs)
  - Maybe less time spent on coding
  - Maybe more time spent on testing, debugging, breaking behaviors
- Does not teach ROS (many things had to be cut)

Outline

- Introduction (already done)
- The web-based textbook
- Design, fabrication, and safety
- Hardware considerations
- Software considerations
- Discussion of tradeoffs and takeaways



The web-based textbook

- Designed for my specific computer science students
- Should be useable for many university instructors
- Should be useable for many secondary schools
- I would love this work to be useful for Vex, e-Yanta, Eurobot, etc.
- Includes videos, slides, interactive widgets, and lab exercises

Category	Concepts
Design	Requirements, Materials, CAD, Fabrication, Mechatronics, Electronics, Safety
Programming	Embedded Systems, Communication, Real-Time, Debugging, Security, IoT, DSP,
Math	Linear Algebra, Calculus, Probability, Statistics, Discretization, Optimization
Modeling	Diagramming, Numerics, Linearization, Kinematics, Dynamics, Simulation, Motion Capture
Control	Feedback, Motion, Robustness, Adaptivity, Reactive, Behavioral
Planning	Search, Mapping, Motion, Trajectory
Vision	Conventional, Learning, Optical Flow
Uncertainty	Estimation, Localization, Fuzzy Logic, Filtering, Identification, SLAM
Society	History, Laws, Ethics, Morals, Jobs, Pop-Culture, HRI

In the following, some examples of internal kinematics for wheeled mobile robots (WMR) will be given. The robot pose in a plane is defined by its state vector

$$\boldsymbol{q}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \varphi(t) \end{bmatrix}$$

in a global coordinate frame (X_g, Y_g), as illustrated in Fig. 2.2. A moving frame (X_m, Y_m) is attached to the robot. The relation between the global and moving frame (external kinematics) is defined by the translation vector $[x, y]^T$ and rotation matrix:

$$\boldsymbol{R}(\varphi) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

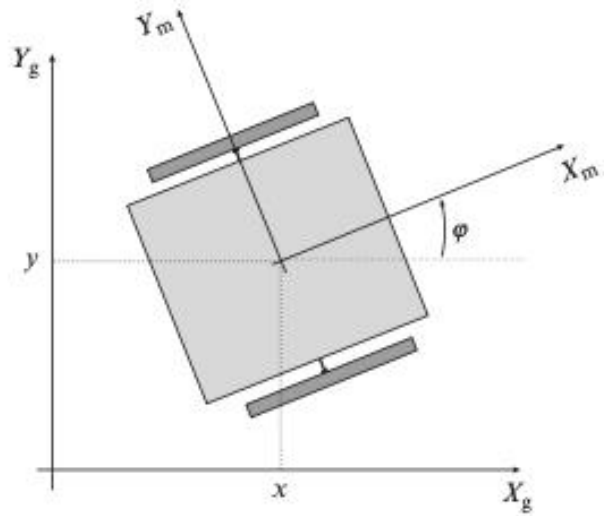
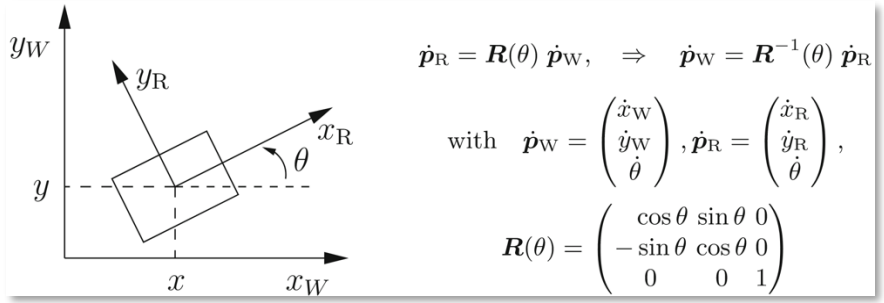
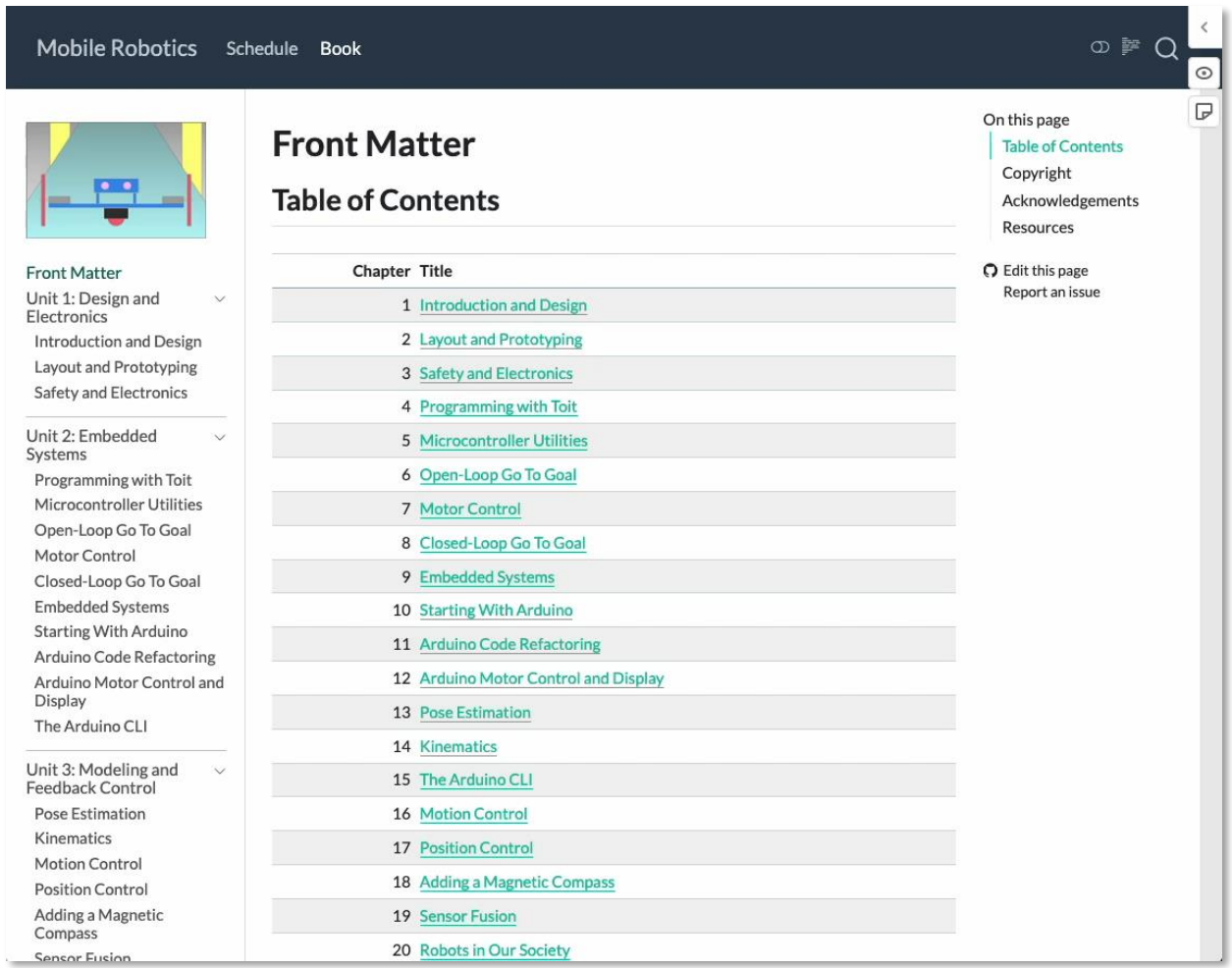


Fig. 2.2 Robot in the plane.

Klančar. Wheeled Mobile Robotics. 2017

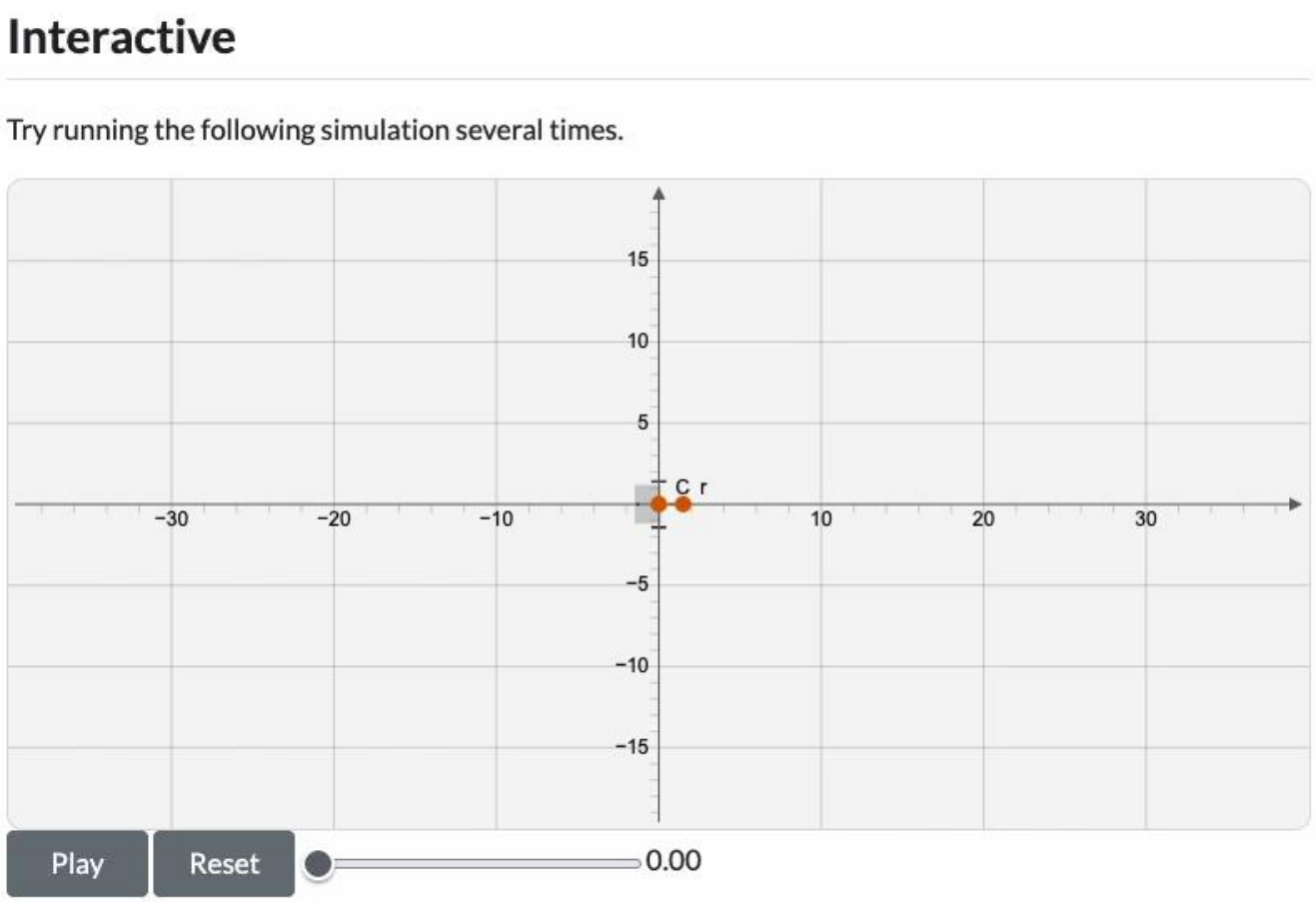


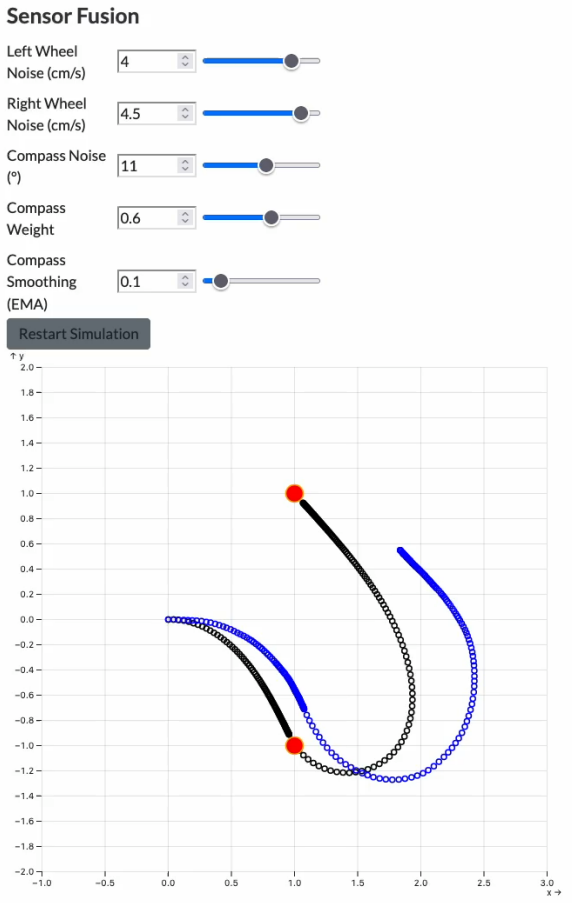
Krüger. EduRob… RiE 2024



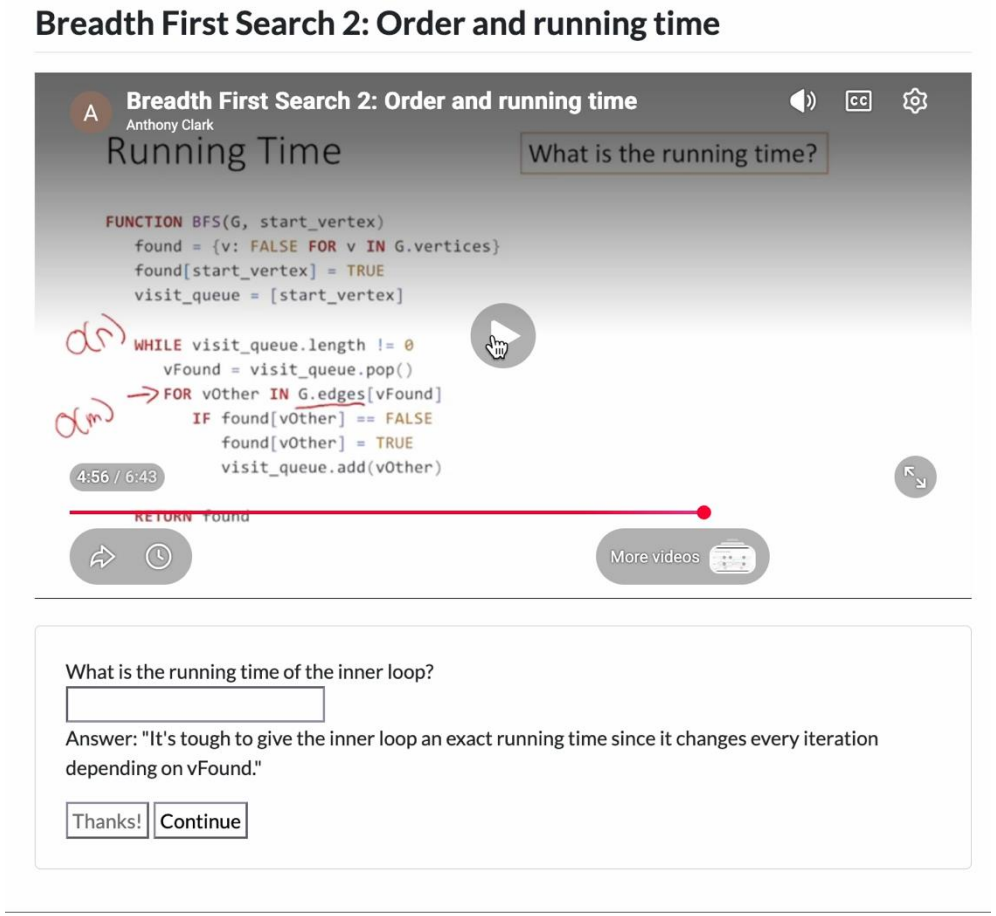
CS 181R: Mobile Robotics

Simulations and widgets





Videos with quizzes



Design, fabrication, and safety

We started with basic requirements

Our robot should:

- Automatically halt when communication is lost
- Autonomously navigate around a course
- Avoid obstacles (static and dynamic)
- Recognize and react to signs

Design, fabrication, and safety

These were broken into

Functional requirements and constraints

- How fast does the robot need to move?
- How maneuverable does the robot need to be?
- How small or large (footprint and mass) can the robot be?
- Will the robot interact closely with people?
- How costly is the robot to manufacture?
- What is the robot's environment?

Design, fabrication, and safety

And

Nonfunctional requirements

- How easy is it to build the robot?
- How easy is it to repair the robot?
- Can the robot be reused in future semesters?

Design, fabrication, and safety

Which were then refined

- What kind of processing speed is needed (MCU or SBC or …)?
- How much memory and storage are needed?
- What is the appropriate transmission (motors, drive system, etc.)?
- What form of internal communication is needed (SPI, UART, etc.)?
- What battery systems (type, capacity, C-rating, voltage)?
- …

Design, fabrication, and safety

And then specified

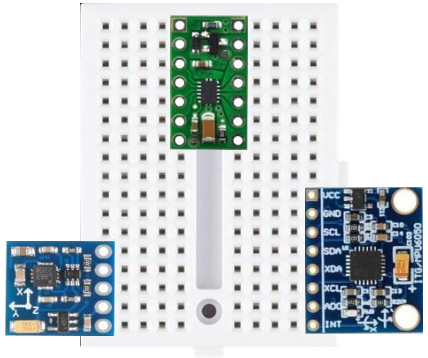
- Decide on a drive system (direct drive).
- Decide roughly on a linear robot velocity (about 50 cm/s).
- Compute the needed motor RPMs (about 140 rpm).
- Select motor driver (select 0-11 V, 1.2 A per channel).
- Find compute devices (select ESP32-S3).
- Design power system (1S 3.7 V battery; estimate 190 rpm).
- …

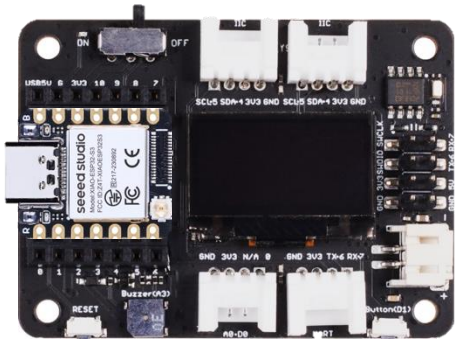
	Voltage	Current	Power
Battery	3.6 V	10 A (max)	36 W
USB	5 V	500 mA (USB 2.0)	2.5 W

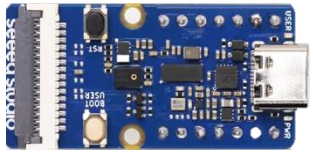
	Voltage	Current	Power
Microcontroller	3.3 V (or 5 V USB)	150 mA (with WiFi)	495 mW
Expansion Board	3.7 V (battery)	15 mA (with display)	56 mW
Motor Driver Control	3.7 V (battery) (up to 11 V)	<1.2 A> (per channel max)	<8.88 mW> (max)
Motor Driver Logic	3.3 V (2 to 7 V)	negligible	0
Left Motor	3.7 V (battery) (up to 6 V)	150 mA (full speed)	560 mW
Right Motor	3.7 V (battery) (up to 6 V)	150 mA (full speed)	560 mW
Left Encoder	3.3 V (2.7 V to 18 V)	negligible	0
Right Encoder	3.3 V (2.7 V to 18 V)	negligible	0
Total		**~500 mA**	**~1.8 W**

You always want to give yourself a safety margin for “spikes”

Fabrication



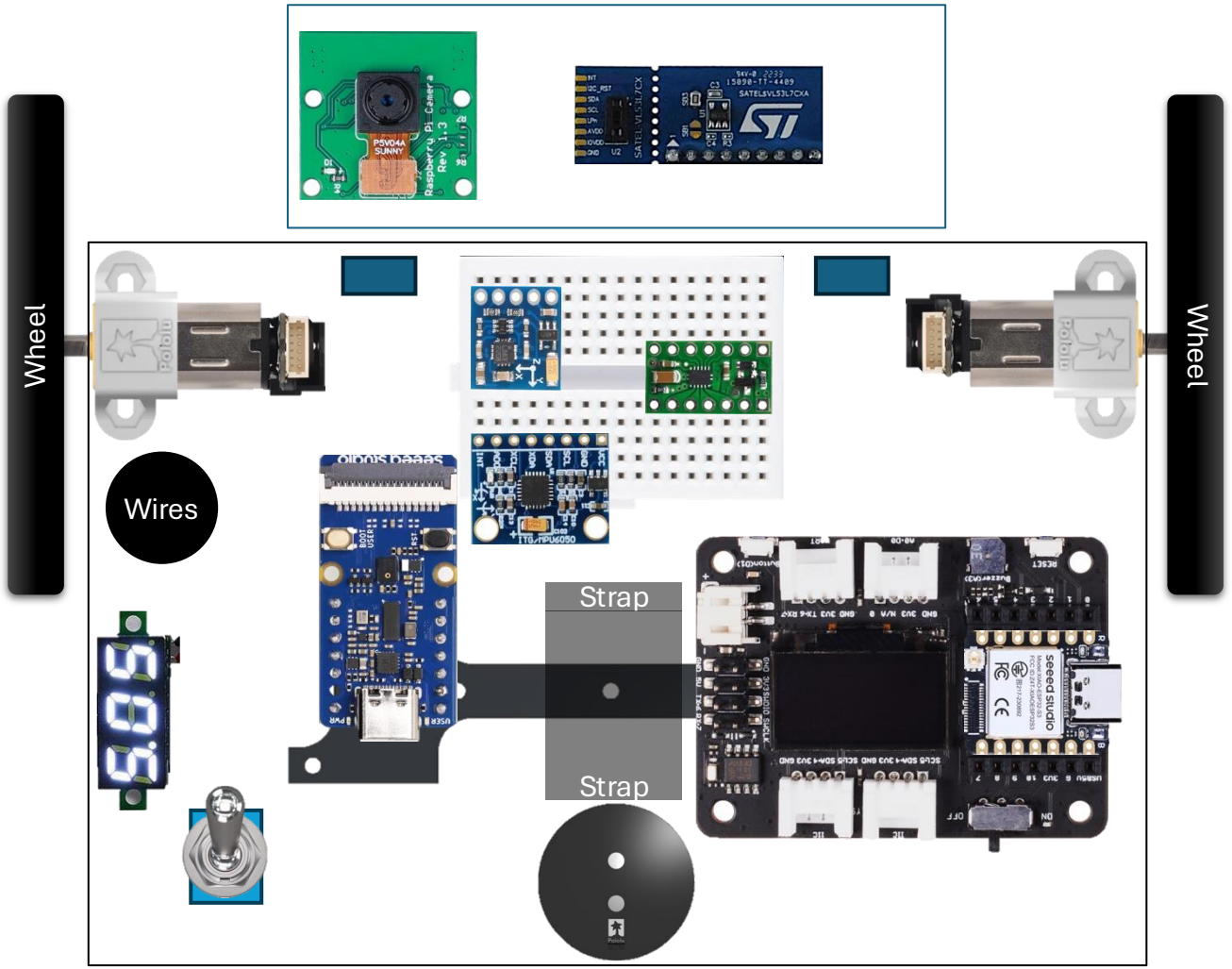




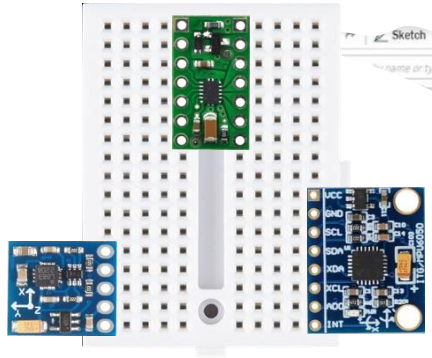
Initial prototypes

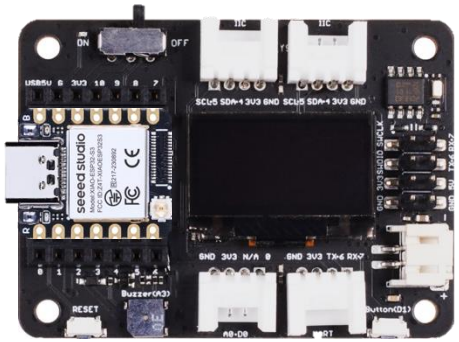
- Cardboard
- Hot glue
- Sticky dots
- Tac

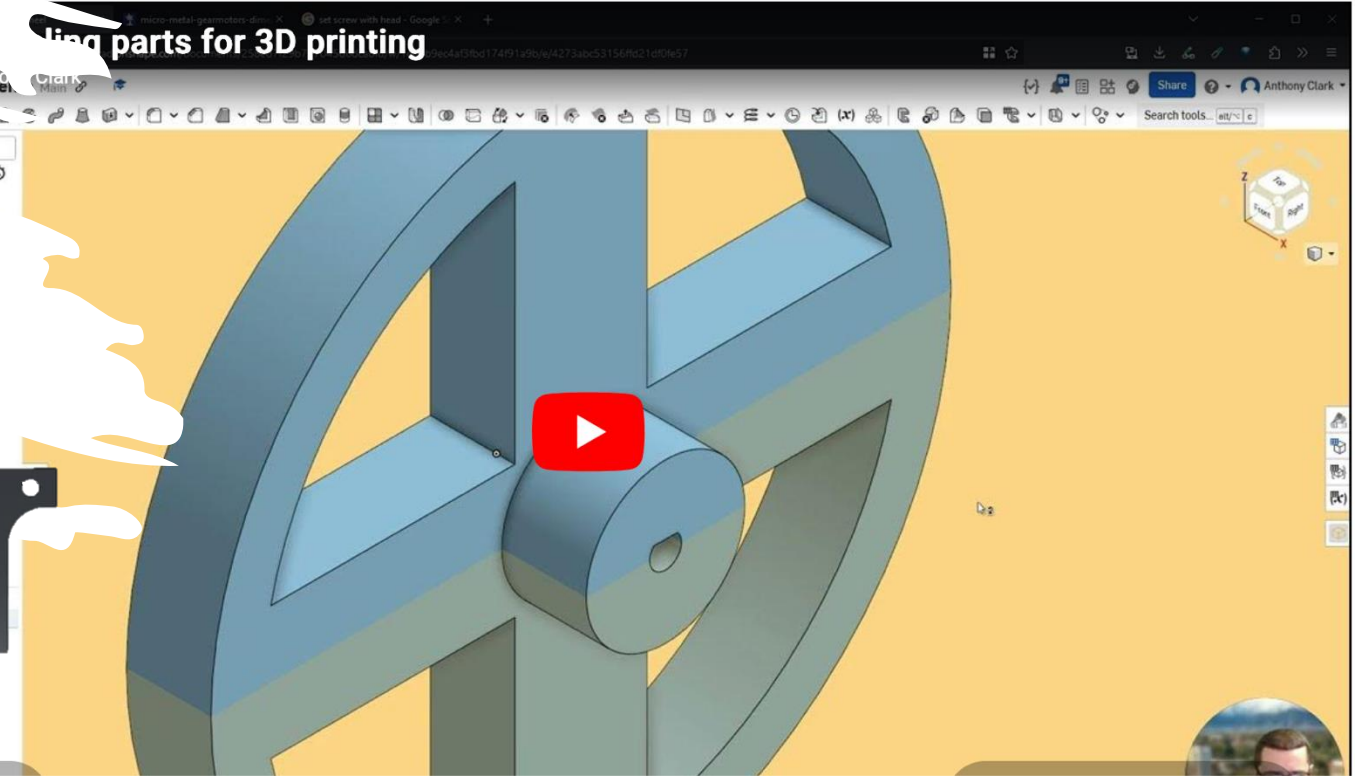
Fabrication

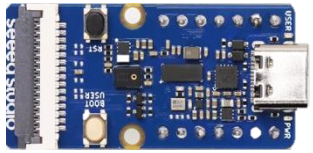


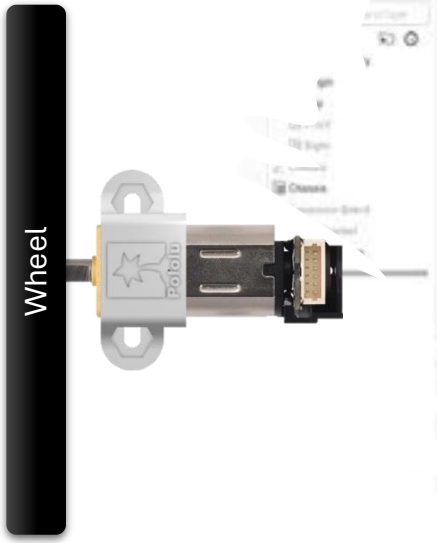
Fabrication

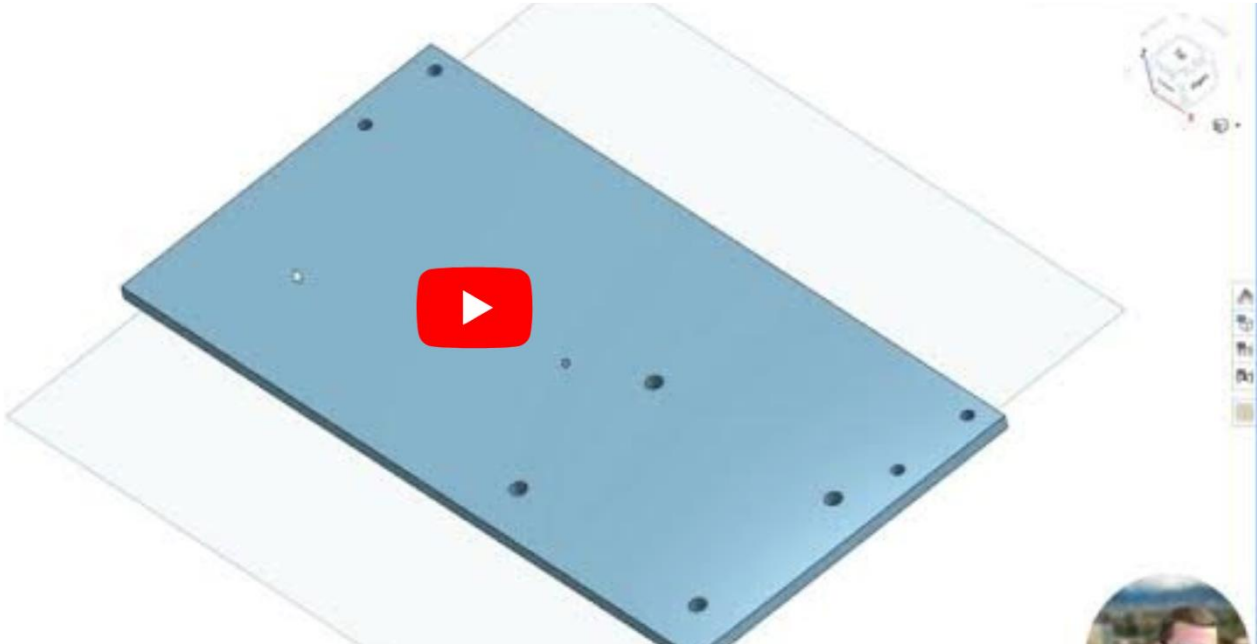












Makerspace

A hidden nonfunctional requirement

CS Department

Microcontroller

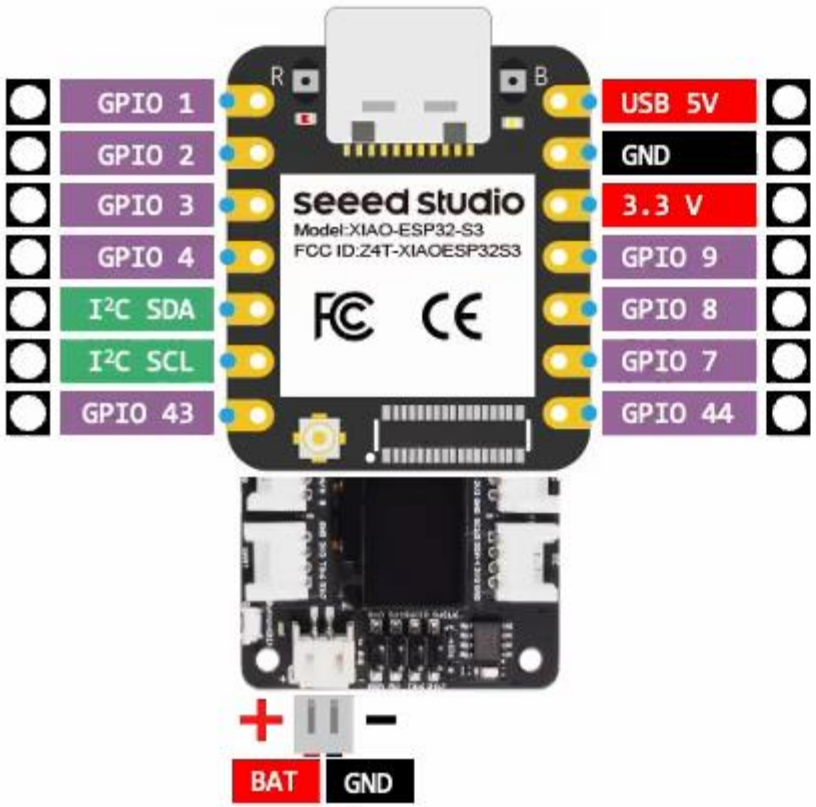
- ESP32-S3 (240 MHz, 512 KB RAM) with Wi-Fi and Bluetooth
  - EduRob. Heß. RiE 2026
  - Several RISC-V microcontrollers in their newest lineup
  - Many ways to program (Arduino, VSCode, Web)

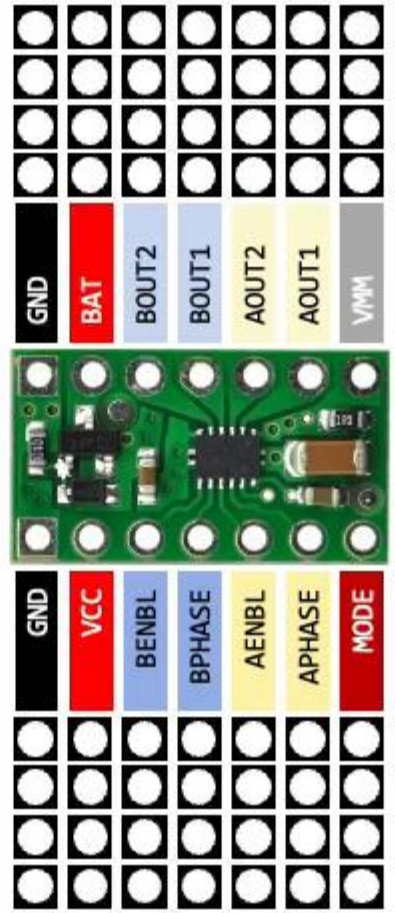
- STM32 microcontroller (168 MHz, 192 KB RAM)
  - Rusty Flying Robots. Hönig. RiE 2026

- *It would be a surprise to see a robot based on an 8-bit microcontroller*


M1
GND
C1
C2
VCC
M2


M2
VCC
C2
C1
GND
M1


GPIO 1
GPIO 2
GPIO 3
GPIO 4
I²C SDA
I²C SCL
GPIO 43
USB 5V
GND
3.3 V
GPIO 9
GPIO 8
GPIO 7
GPIO 44
seeed studio
Model:XIAO-ESP32-S3
FCC ID:Z4T-XIAOESP32S3
BAT
GND


GND
BAT
BOUT2
BOUT1
AOUT2
AOUT1
VMM
GND
VCC
BENBL
BPHASE
AENBL
APHASE
MODE


BAT
BAT

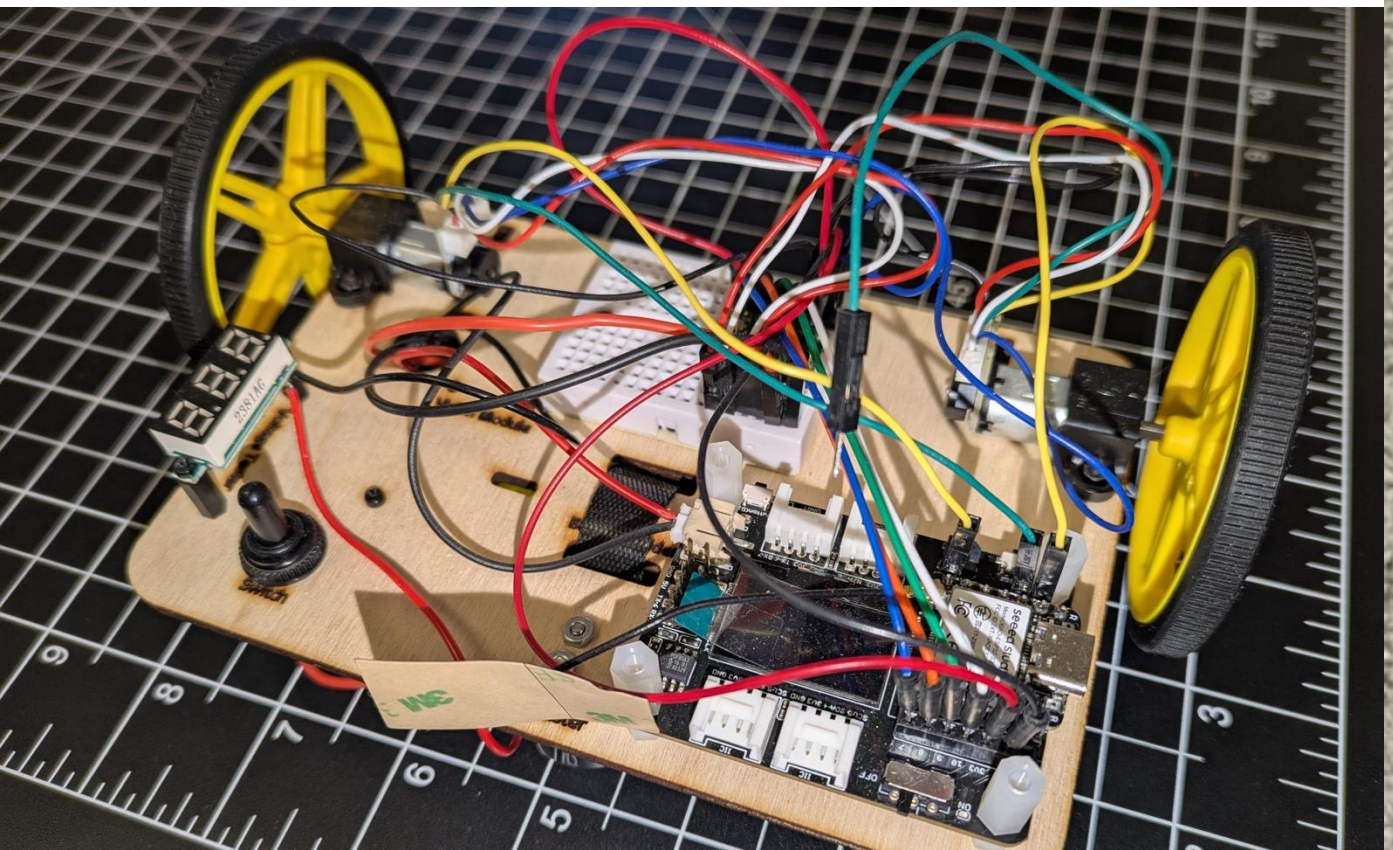

BAT
18650 3500mAh
3.6V 12.6Wh
GND


BAT
GND

Open-Ended Design

Some students stuck with their cardboard prototypes for the entire semester

- Software can compensate for poor hardware
- Better hardware → easier software



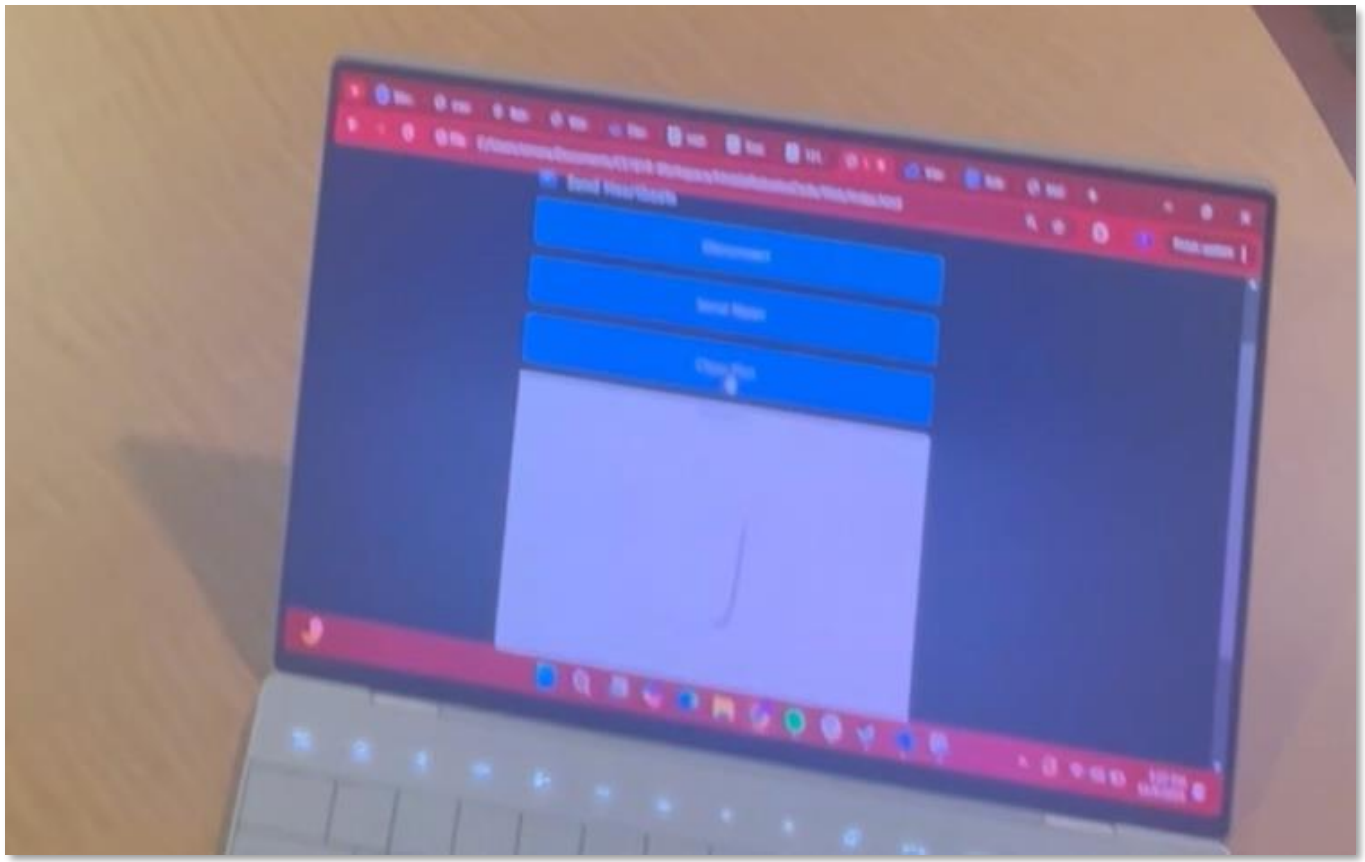
Software

Programming languages

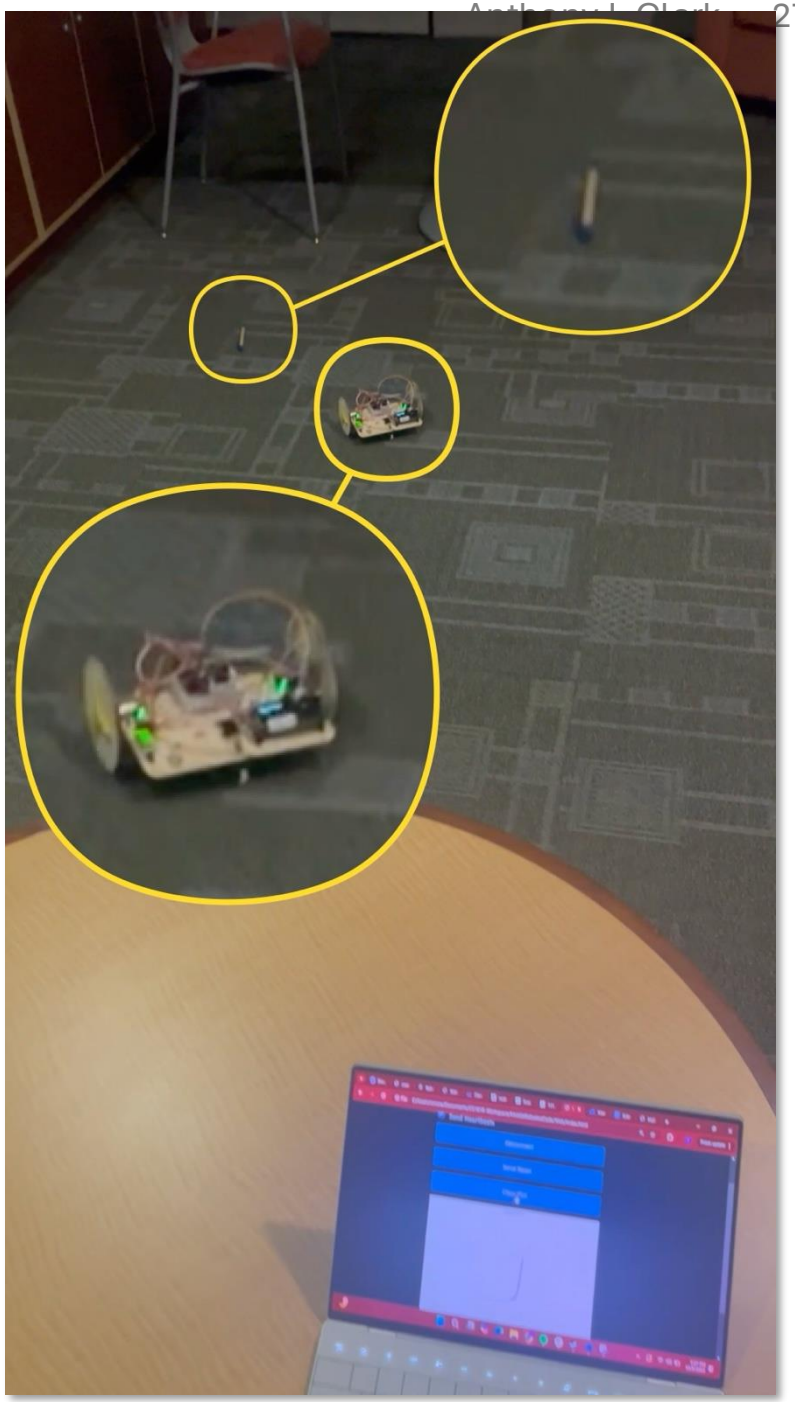
- Arduino C/C++
- ESP-IDF C/C++
- Toit (OTA)
- JS
- WASM (WARDuino. Lauwaerts)
- Rust
- Many other

Wireless communication

- Wi-Fi with WebSockets
  - Teaching TCP
  - Create wireless console (REPL)
- Web Bluetooth
  - Lower energy usage
  - I'd have to provide more code



Hosted on the course website



Toit

```
import gpio

LED1 ::= gpio.Pin.out 17
LED2 ::= gpio.Pin.out 18

main:
  task:: led-500ms
  task:: led-123ms

led-500ms:
  while true:
    sleep --ms=500
    LED1.set 1
    sleep --ms=500
    LED1.set 0

led-123ms:
  while true:
    sleep --ms=123
    LED2.set 1
    sleep --ms=123
    LED2.set 0
```

Arduino

```
IntervalTimer led1Timer(500);
const int led1Pin = 17;
int led1State = LOW;
IntervalTimer led2Timer(500);
const int led2Pin = 18;
int led2State = LOW;

void setup() {
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
}

void loop() {
  if (led1Timer) {
    led1State = led1State == LOW ? HIGH : LOW;
    digitalWrite(led1Pin, led1State);
  }
  if (led2Timer) {
    led2State = led2State == LOW ? HIGH : LOW;
    digitalWrite(led2Pin, led2State);
  }
}
```

```
import gpio

LED1 ::= gpio.Pin.out 17
LED2 ::= gpio.Pin.out 18

main:
  task:: led-500ms
  task:: led-123ms

led-500ms:
  while true:
    sleep --ms=500
    LED1.set 1
    sleep --ms=500
    LED1.set 0

led-123ms:
  while true:
    sleep --ms=123
    LED2.set 1
    sleep --ms=123
    LED2.set 0
```

ESP-IDF

```
#define LED1_PIN GPIO_NUM_17
static int led1_state = 0;
static IntervalTimer led1_timer;

#define LED2_PIN GPIO_NUM_18
static int led2_state = 0;
static IntervalTimer led2_timer;

void app_main(void)
{
  gpio_config_t io_conf = {
    .pin_bit_mask = (1ULL << LED1_PIN) | (1ULL << LED2_PIN),
    .mode = GPIO_MODE_OUTPUT,
    .pull_up_en = GPIO_PULLUP_DISABLE,
    .pull_down_en = GPIO_PULLDOWN_DISABLE,
    .intr_type = GPIO_INTR_DISABLE,
  };
  gpio_config(&io_conf);

  interval_timer_init(&led1_timer, 500);
  interval_timer_init(&led2_timer, 500);

  while (1) {
    if (interval_timer_ready(&led1_timer)) {
      led1_state = !led1_state;
      gpio_set_level(LED1_PIN, led1_state);
    }

    if (interval_timer_ready(&led2_timer)) {
      led2_state = !led2_state;
      gpio_set_level(LED2_PIN, led2_state);
    }

    vTaskDelay(1);
  }
}
```

Ethics Case Studies (Flipped)

Background (1-3 paragraphs)
Scenario (1-3 paragraphs)
Ethical issues (2-5 issues)
Discussion issues (3-5 questions)
Acknowledgements
References

An EV self-driving car company, Y, becomes incredibly successful, expands rapidly, and establishes itself as the car company with the most EVs in City X.

Its fleet includes 10,000 cars, some privately owned and others used for car-sharing purposes by corporations like Uber and Lyft.

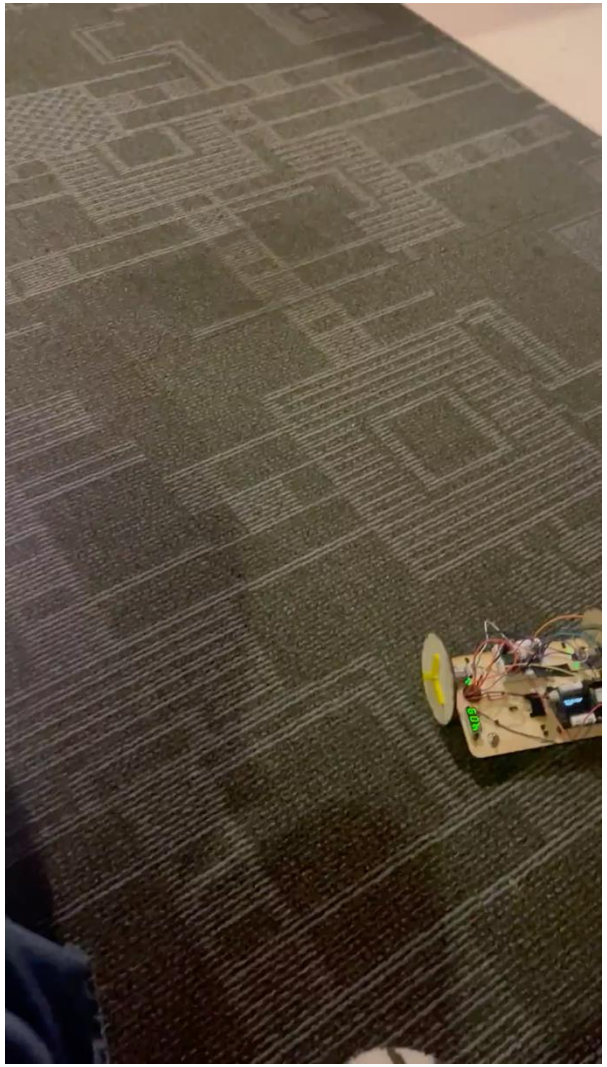
Behind the scenes, Company Y makes an illegal agreement with a major advertising firm, Company A, selling three years' worth of face-identity data for $2 billion.

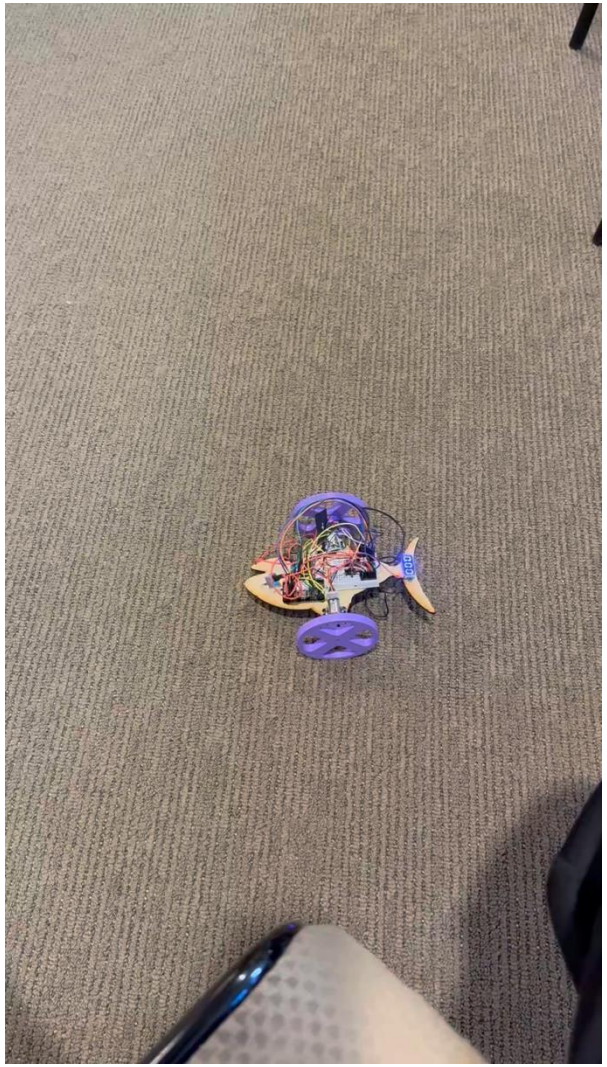
Discussion and takeaways

- I had to convince my students that a flipped-style was effective
- Including fabrication leads to fewer (less advanced) algorithms
- It was a good tradeoff for my students
- The interactive components of the textbook were appreciated

- Future changes
  - Add a competition in place of the final assignment (or two)
  - Include ROS 2
  - Include more peer review

Thank you!





https://cs.pomona.edu/classes/cs181r/book/book.html