Making STEM fun

XRP Robots and resources

Agenda

- Talk about the XRP robot and available materials
- Discuss robot programming: Blockly vs Python
- Build your robot
- Go through the curriculum modules and do some of the exercises
- Discuss resources for teachers to help ensure success in the classroom
- Discuss the next steps and schedule
- Future opportunities

What is the XRP

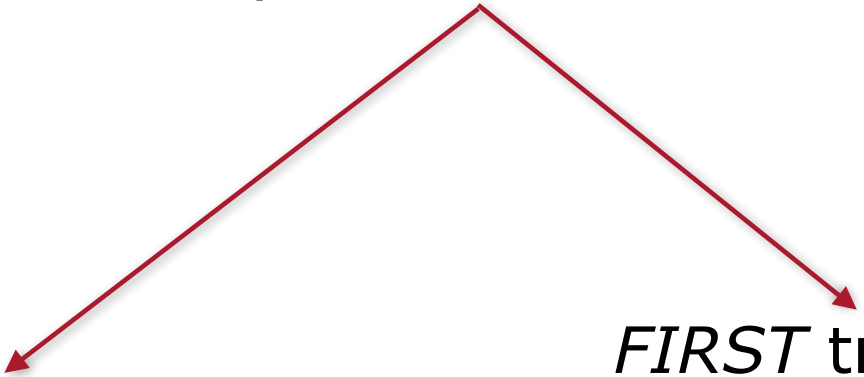
Entry-level open-source robotics ecosystem intended to expand global participation in robotics, inspire future engineers, and serve as a springboard for STEM education for middle/high school, electronics beginners, and robotics enthusiasts of any age.

Demystifies advanced technology, The XRP ecosystem equips teachers with the confidence and tools they need to inspire and prepare students for today's thriving STEM careers

5500 kits being locally manufactured by NH CTEs and Community Colleges and given (FREE) to HS and MS. Funded by a grant from Governor Sununu and NH Department of Education

What is the XRP

It is actually two robots in one!



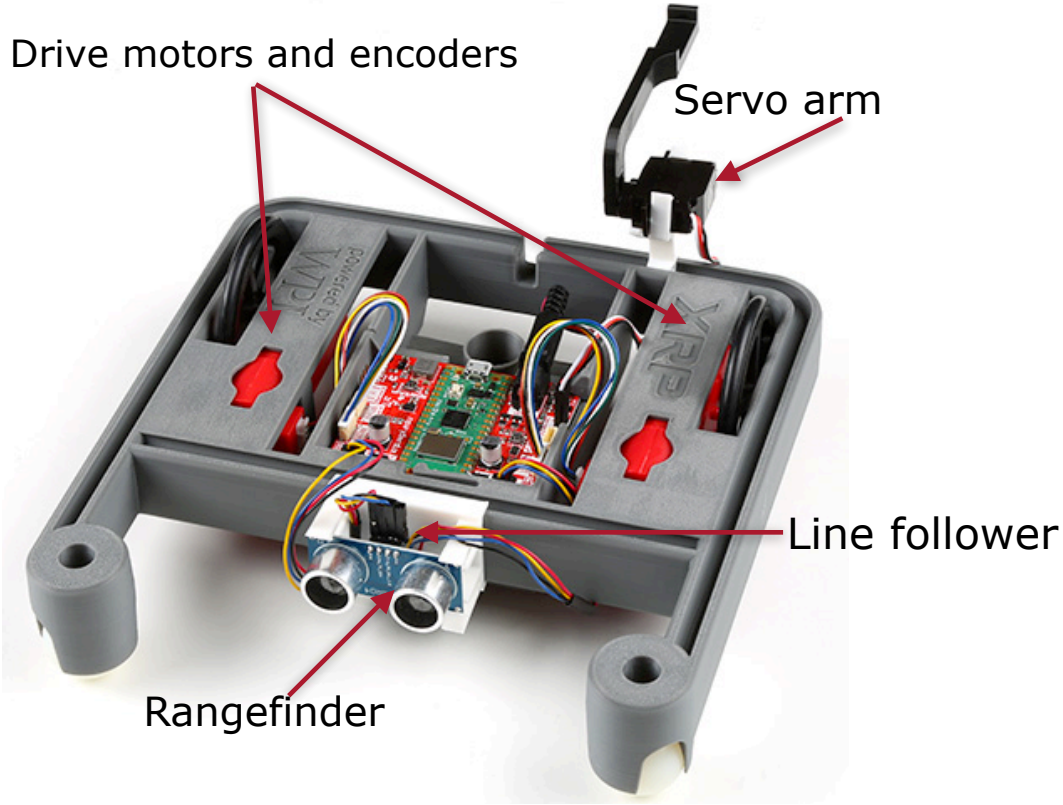
Educational robot with Python and Blockly

FIRST training robot that can use the full suite of FRC programming tools

What is the XRP

Consortium of companies

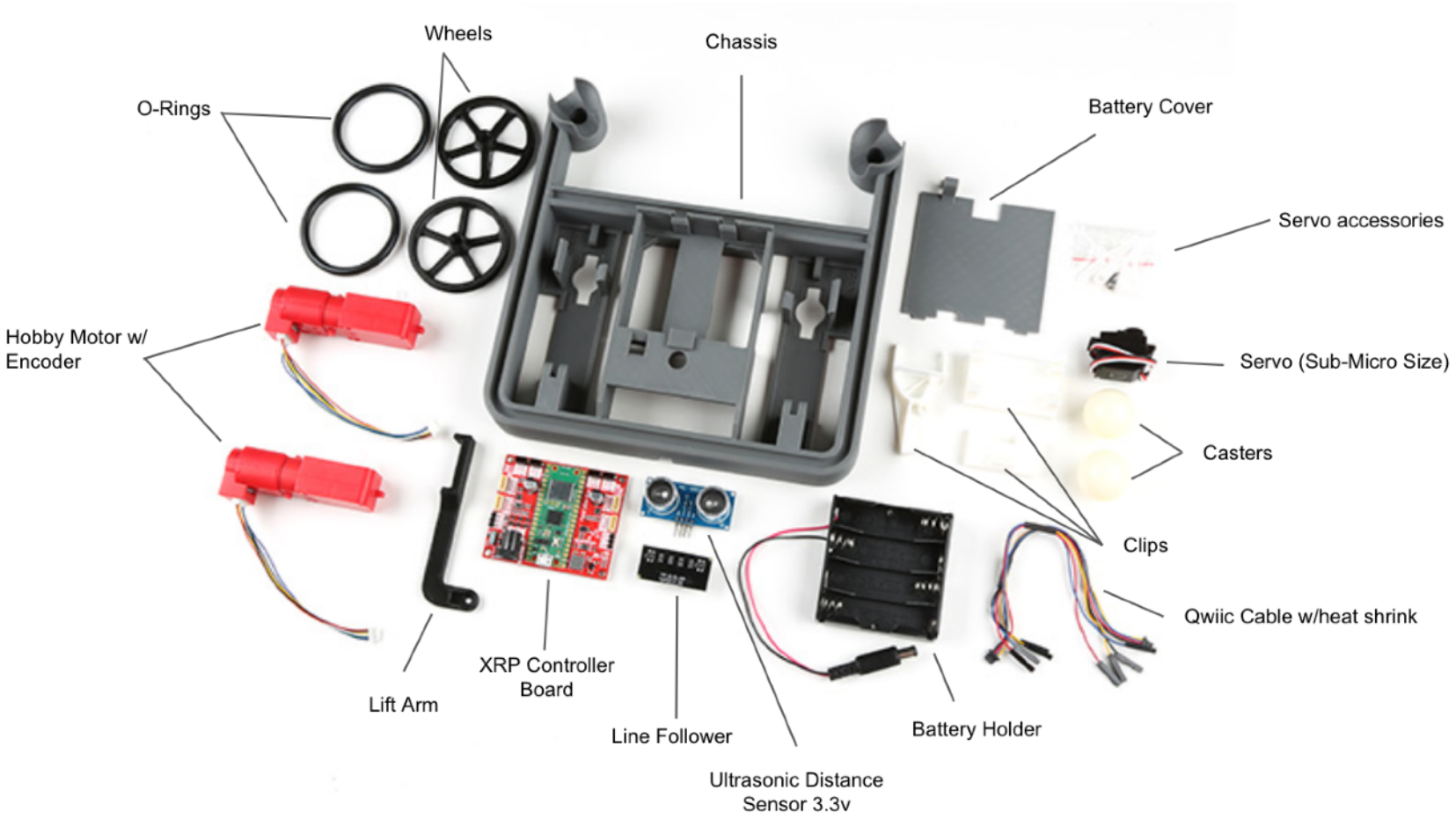
XRP Robot



- Circuit board has a
  - Raspberry Pi Pico
  - 4 Motor controllers
  - 2 Servo controllers
  - IMU for navigation
  - Power regulator
  - Qwiic connector for expansion

Additional purchase: batteries and cables

XRP Hardware



Curriculum Resources

- Introduction to Robotics
  - Multi-module that teaches beginning Python programming and basic robotics
- XRP User Guide
  - Assembly guide for XRP kit, guide to the software tools, and use of all the motors and sensors, including examples
- XRP API Reference
  - A reference guide to all the complete Python API (Application Programming Interface) documenting all the software library classes and functions

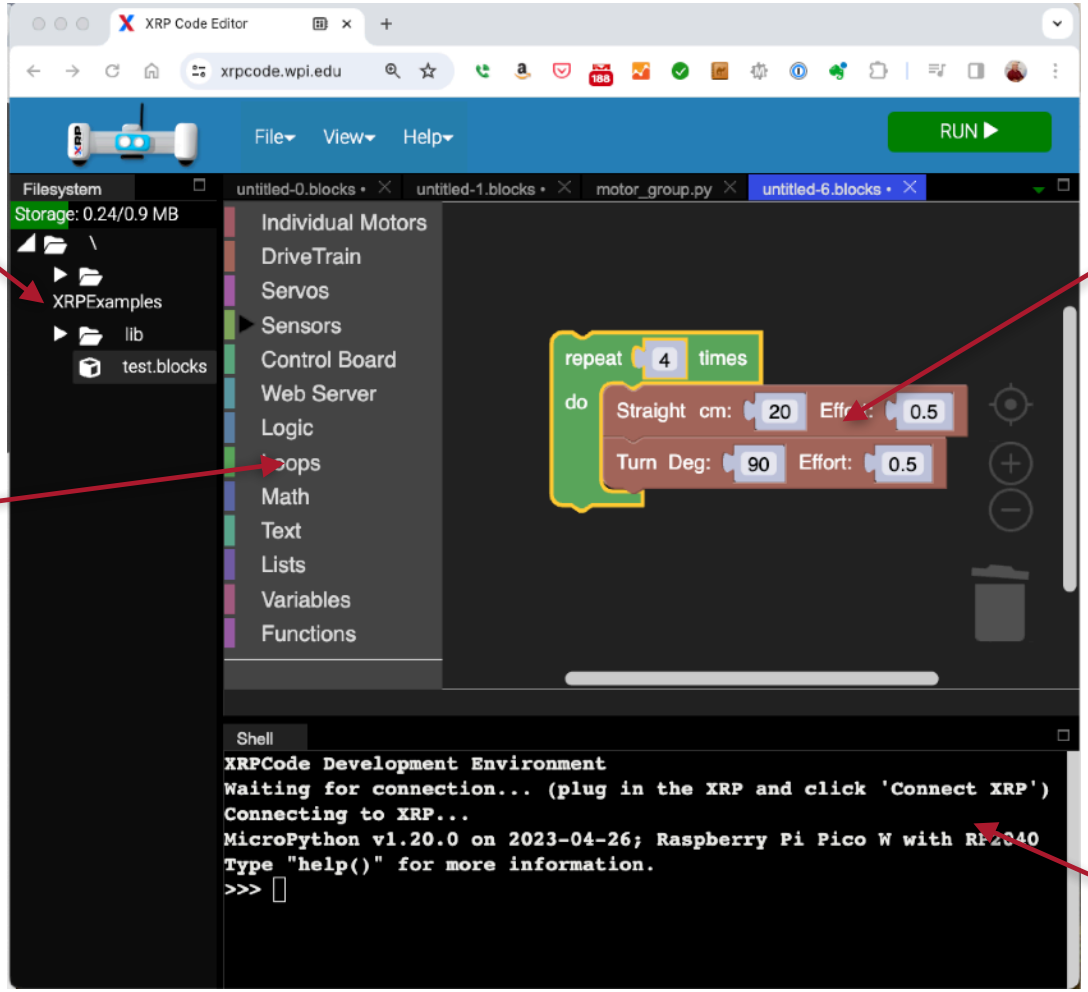
XRPCode IDE
Integrated Development Environment

File storage

Programming elements

IDE runs completely in a web browser

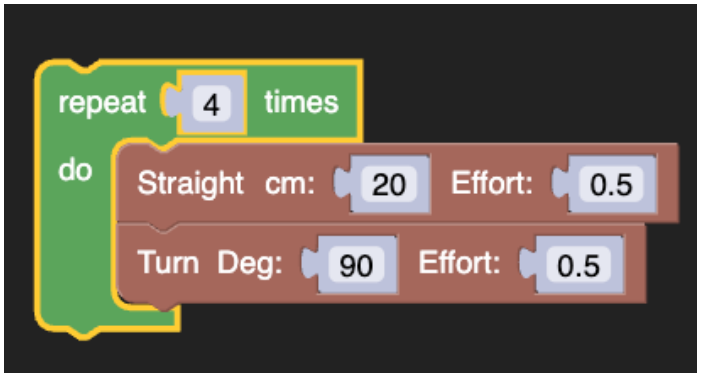
Works on Chrome, Edge, or a ChromeBook



Student program in either Blockly or Python

Messages and program output

Blockly or Python?



Blockly is excellent for non-programmers - students can get their first program going in minutes

Python is used throughout industry and is taught in college

```
from XRPLib.differential_drive import DifferentialDrive

differentialDrive = DifferentialDrive.get_default_differential_drive()


for count in range(4):
  differentialDrive.straight(20, 0.5)
  differentialDrive.turn(90, 0.5)
```

XRPCode helps student make the transition

Course modules

- Introduction to the XRP
- Robot driving
- Measuring distances
- Controls
- Following lines
- Actuation
- Final project

Introduction to the XRP

Assembly and quick tour of the software

Introduction to the XRP

- In this module you will:
  - Learn about the field of robotics
  - Assemble your robot
  - Install the software required for programming the robot
  - Write a short program to familiarize themselves with programming the XRP robot.
- At the end of this module, you will be able to:
  - Recognize the characteristics of robots
  - Talk about the core disciplines that make up the field of robotics
  - Understand the components of the robot and how it is assembled
  - Build familiarity with the tools for programming the XRP

Building your robot

- Robot assembly can be completed in 15-30 minutes
- No tools required (except option small screwdriver)
- Video and written instructions are provided (see the Users Guide)

- We'll build your robots now

https://xrpusersguide.readthedocs.io

Install the XRP software

- XRPCode will automatically install the programming software on the robots
- It can also detect out of date robots and update software as necessary
- Simply plug in the XRP via the USB cable and refresh the XRPCode.wpi.edu web page

Robot driving

Getting predictable movement from the drivetrain

Robot Driving

- In this module you will:
  - Learn how to make the robot drive at different speeds
  - Understand the relationship between the wheel speeds and the robot's motion
  - Be introduced to motor encoders and use them to measure the robot's movements
  - Learn about the built-in driving functions that help make complicated motions easy
- At the end of this module, students will be able to…
  - Make the robot drive in straight lines, turn corners, and turn in place
  - Write and use basic functions in the Python programming language
  - Use different types of loops in Python
  - Break down repetitive movement sequences using functions

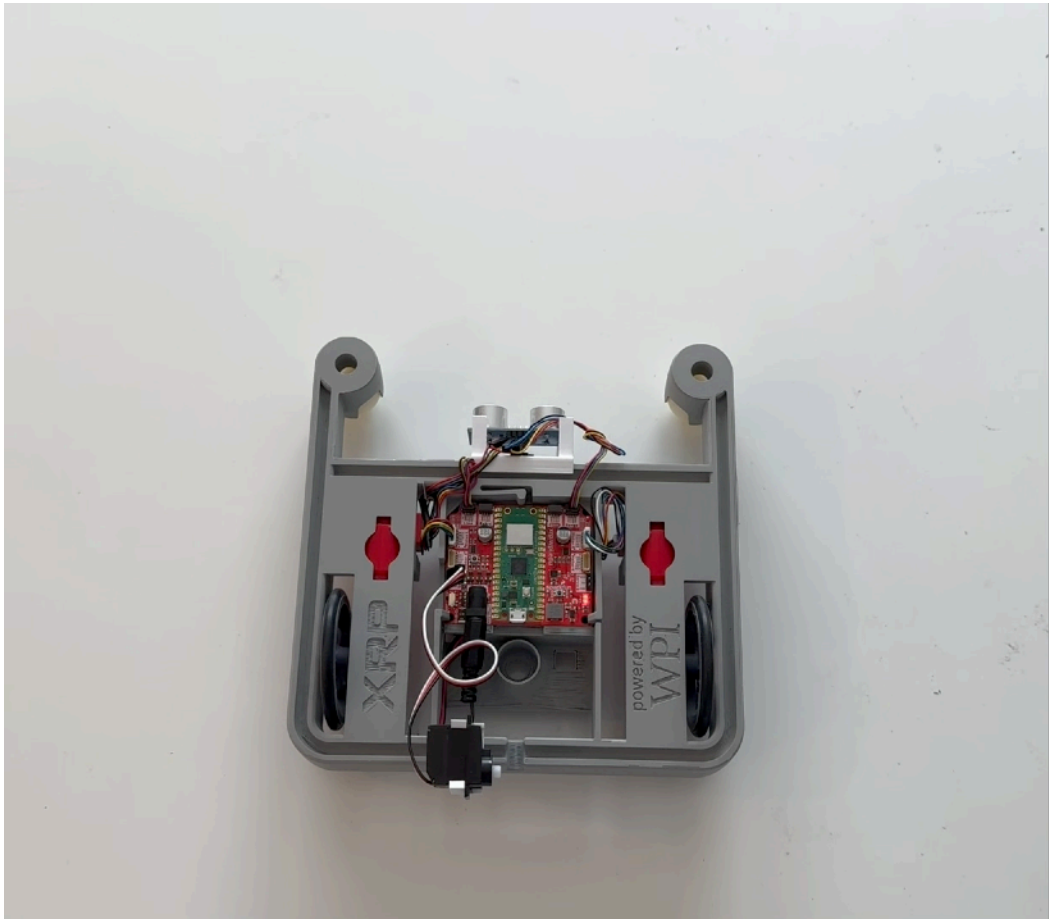
Your first programs

- Make the robot drive forward 20 cm with 50% effort

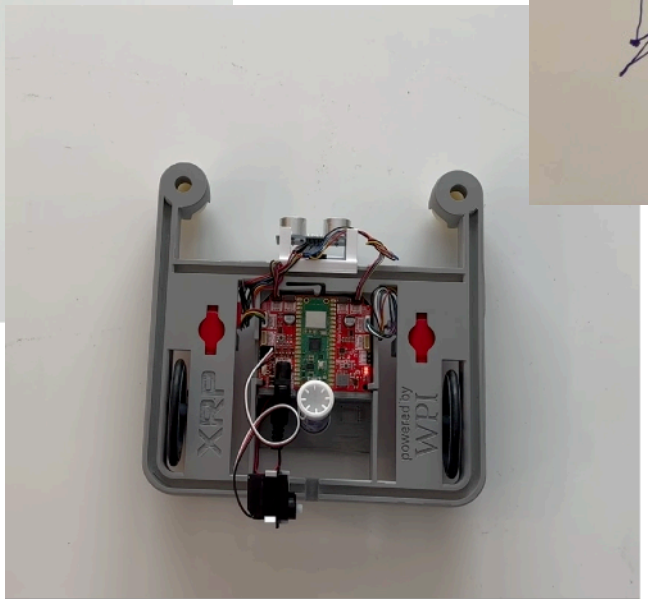
- Now make the robot turn 90 degrees after going straight

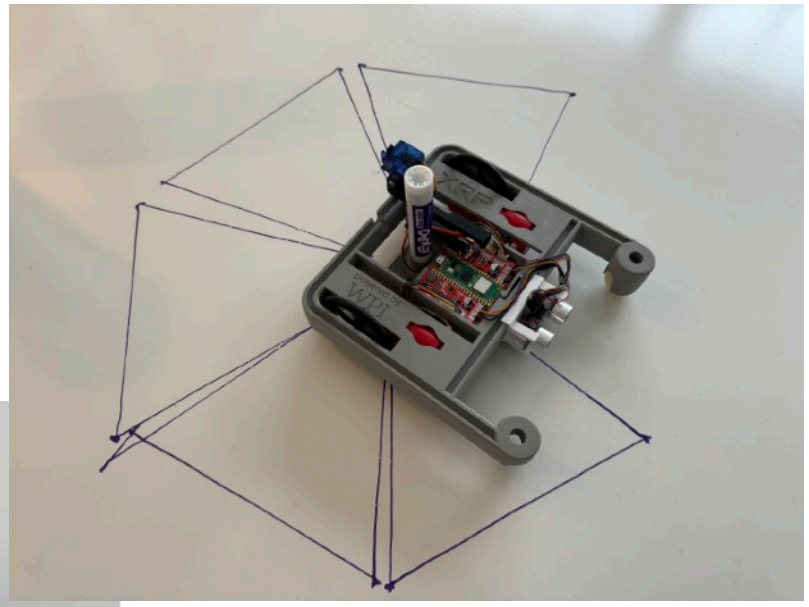
- Make it do that 4 times

Making a square



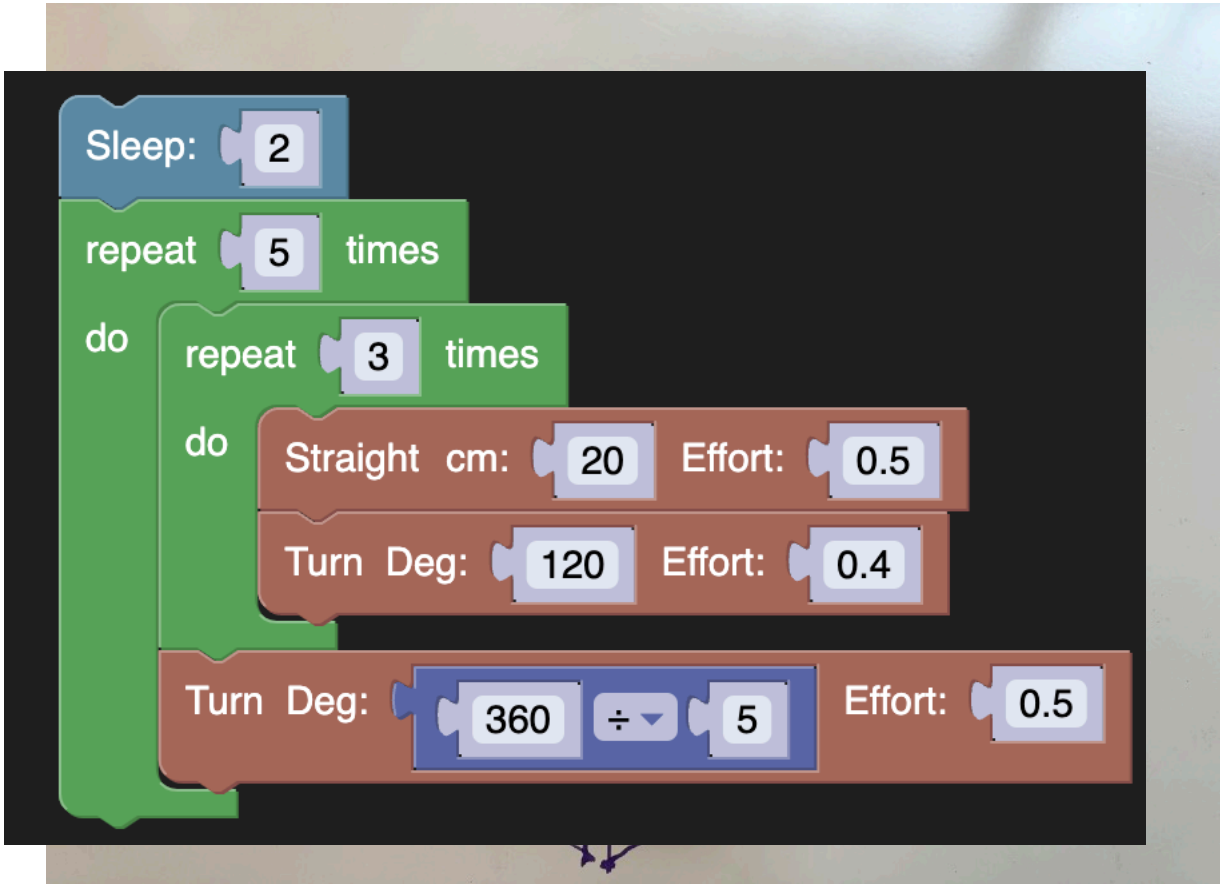
Making shapes more engaging





What else can you draw?

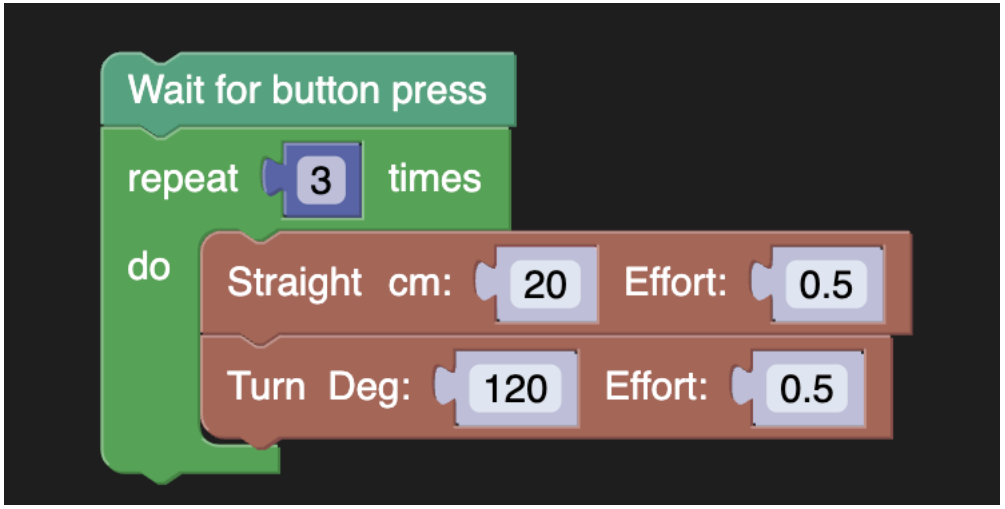
By the way, can you guess how to draw the last shape?



Drawing a triangle

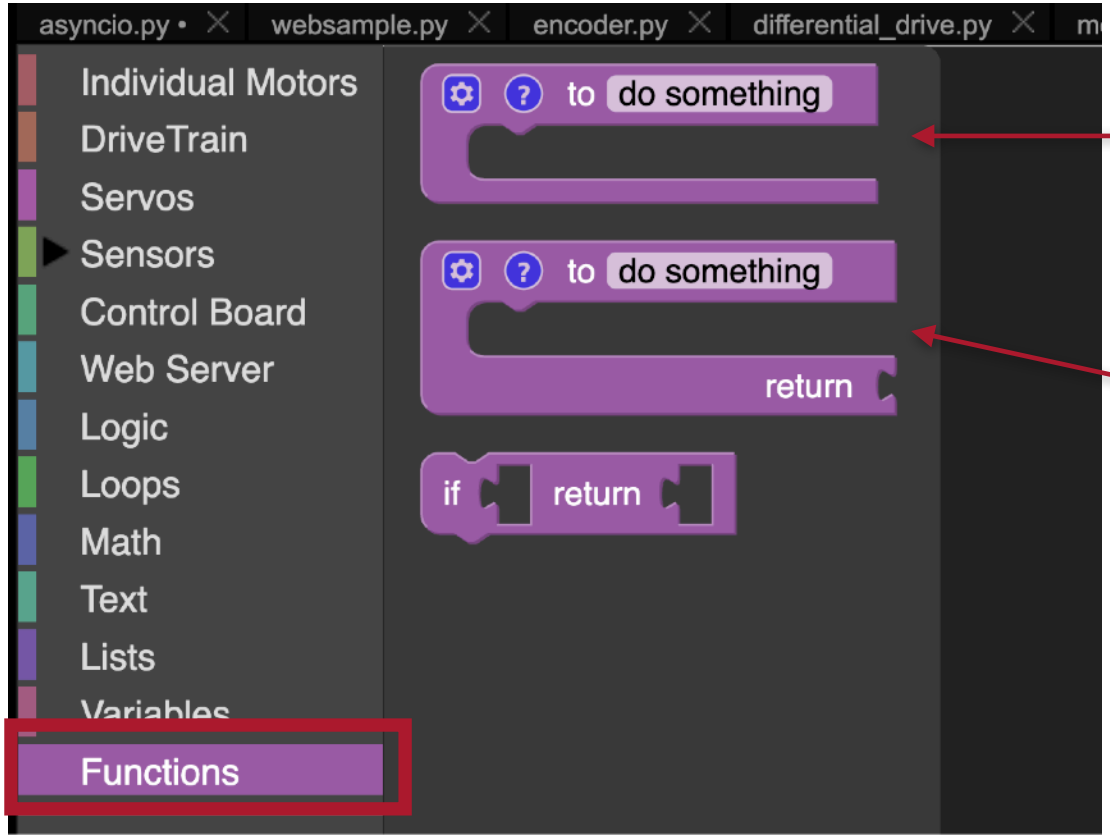
How many sides, and what is the turn angle?

Give it a try



Can you generalize the turn angle for any polygon?

Making a polygon block (functions)



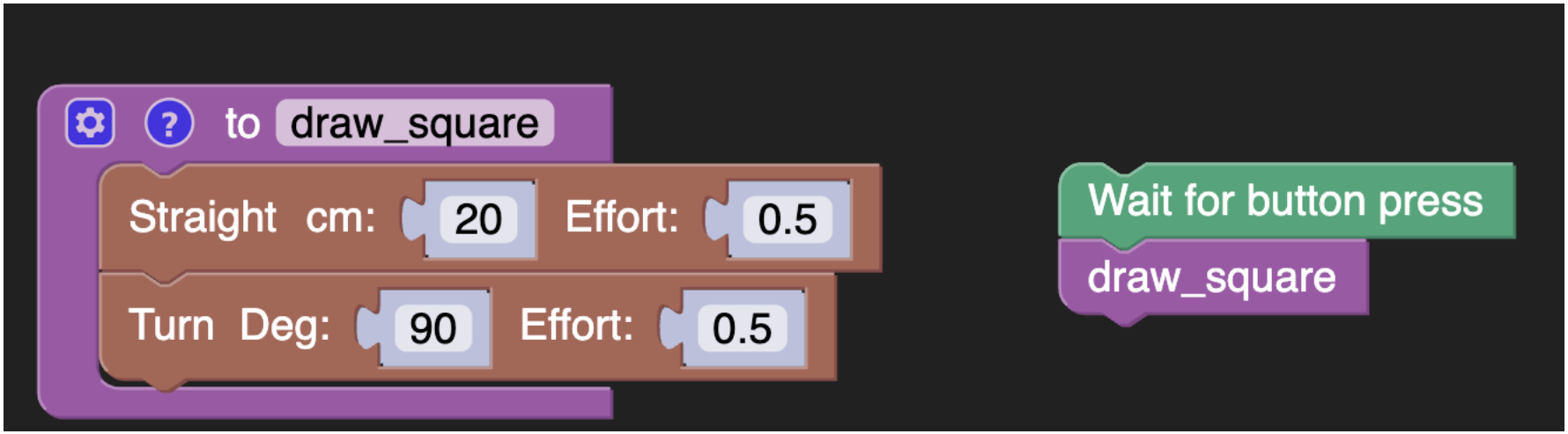
Blocks that do something

Blocks that do something and return a value

Creating your own functions teaches code reuse

Start with a square

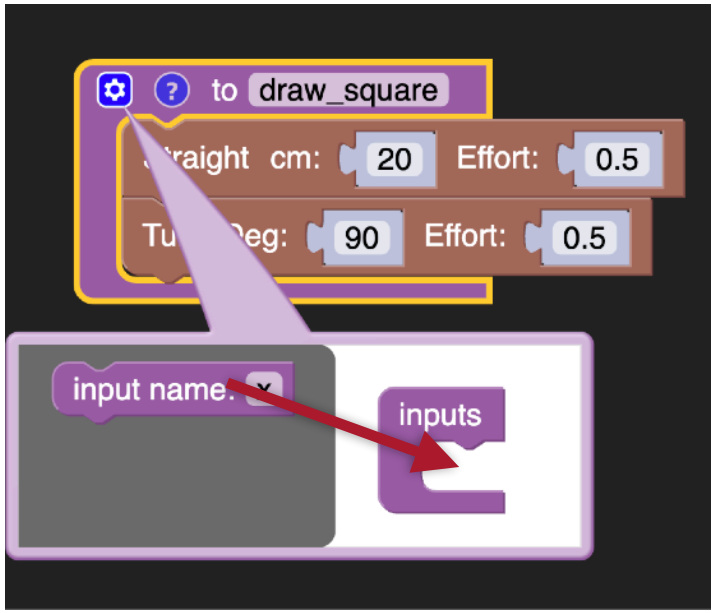
Give it try

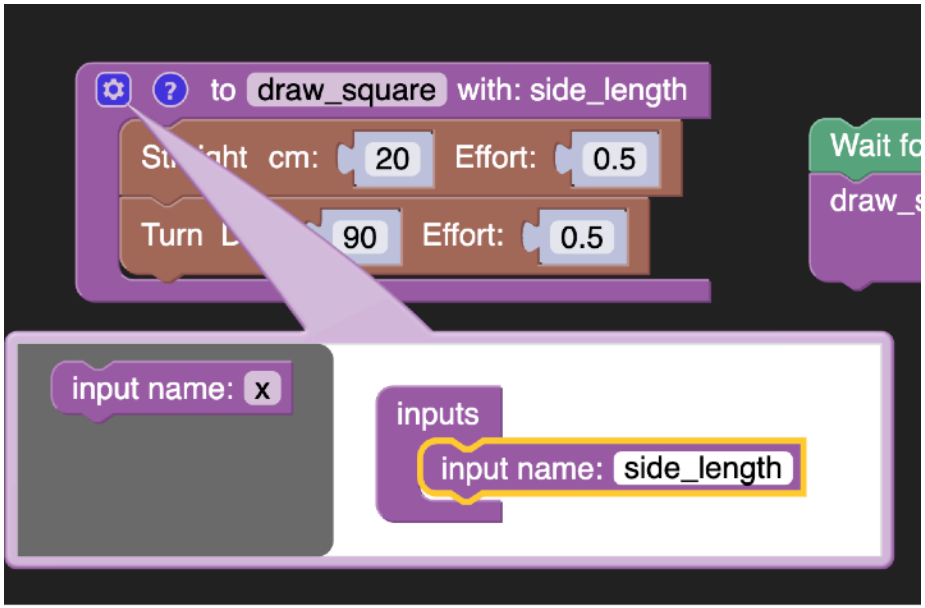


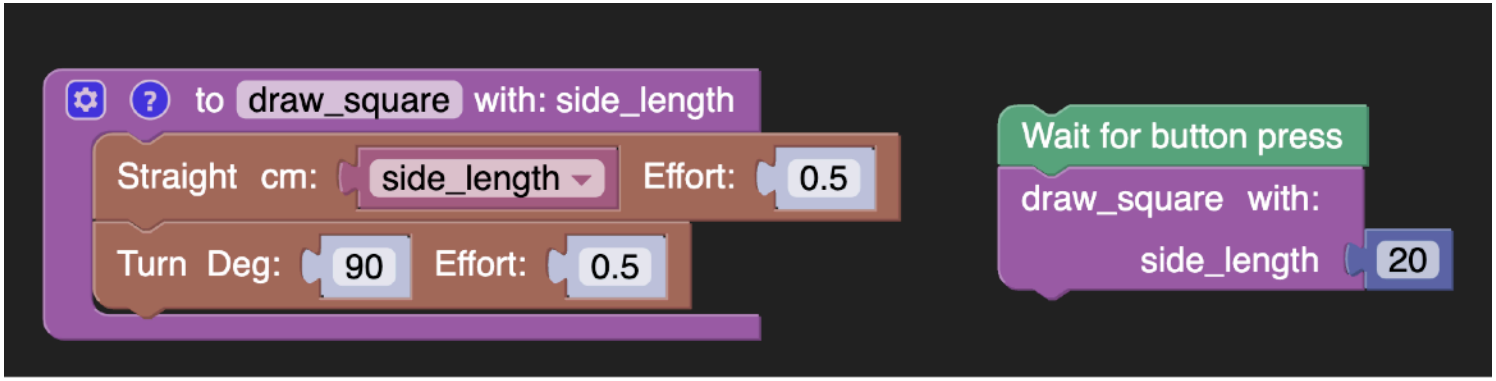
Define the draw_square block

Use the draw_square block

Add side length as an input

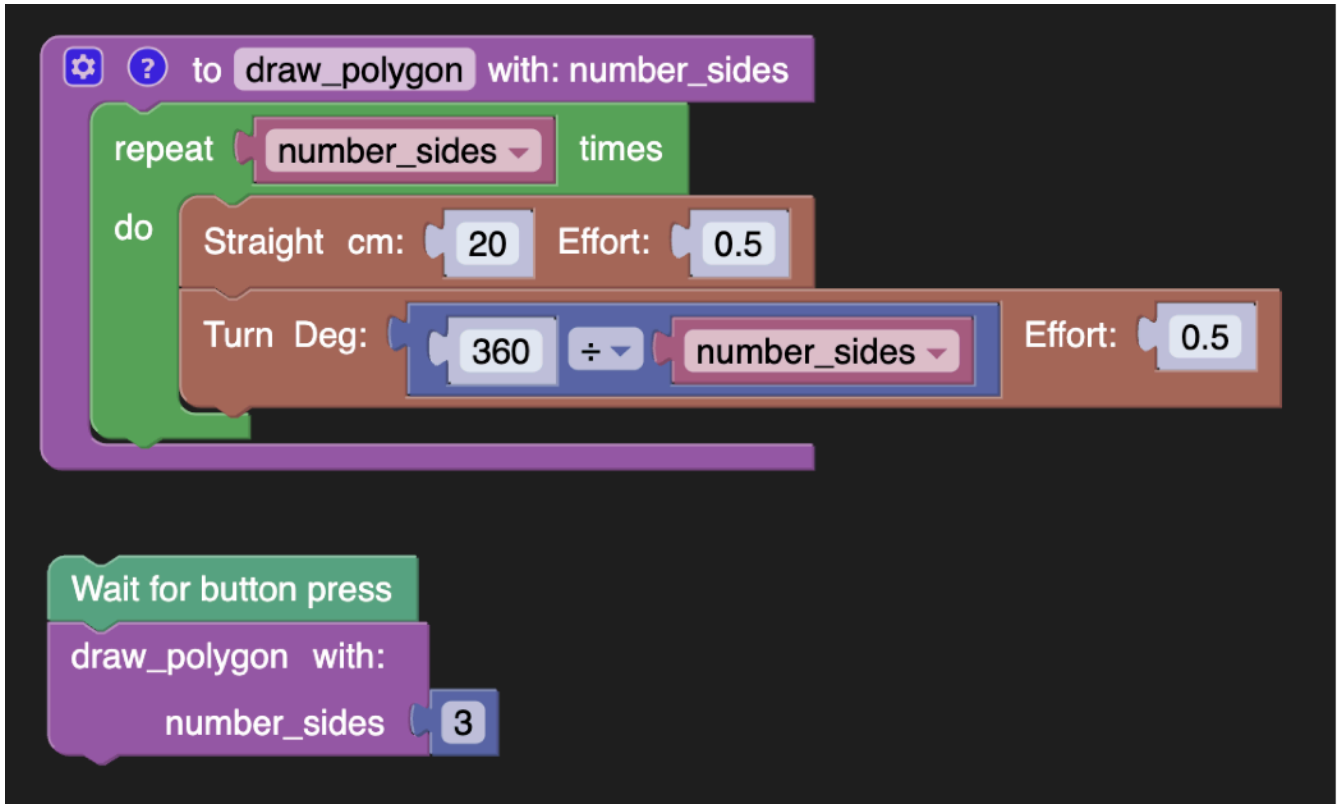




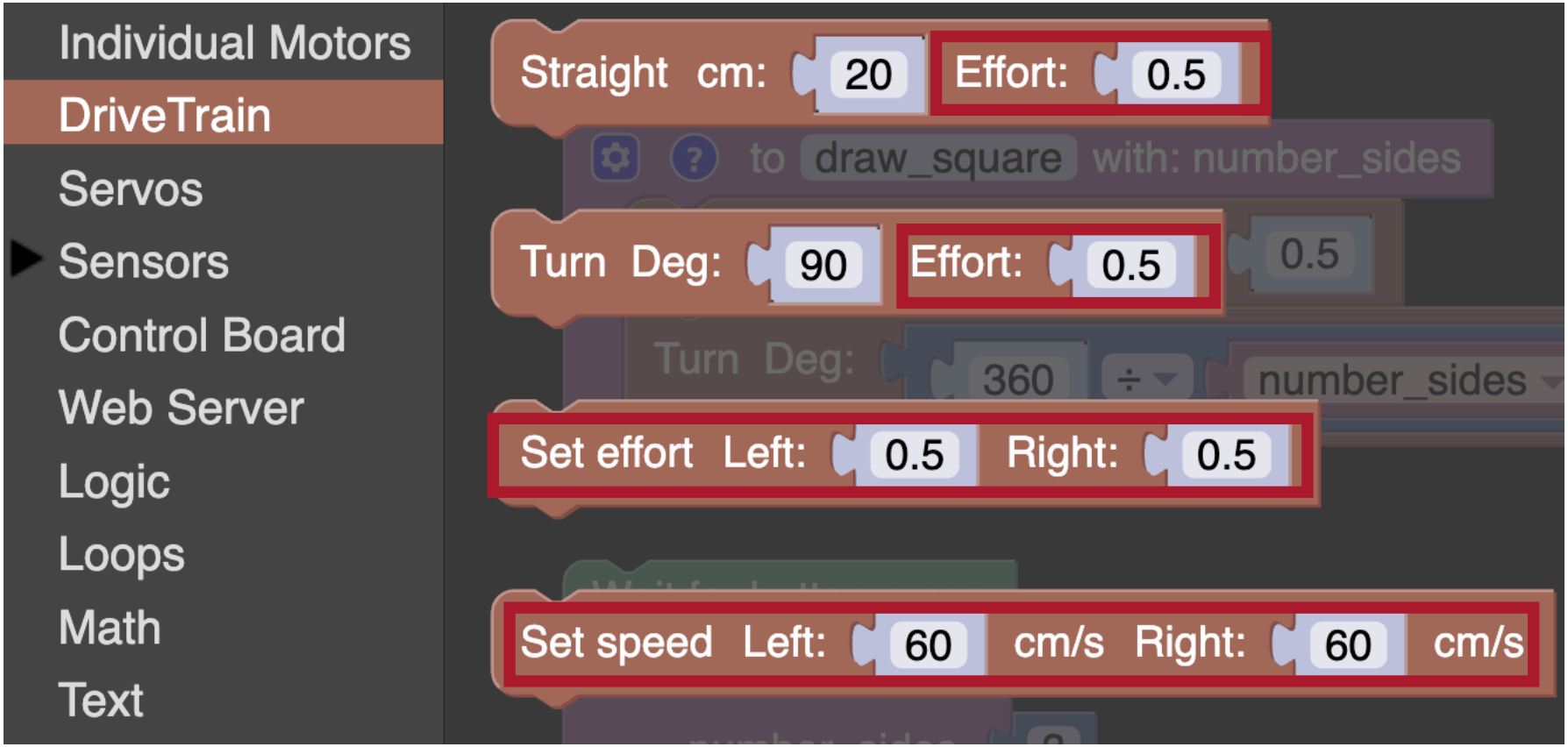


Now try polygons

See if you can make a polygon function with a parameter for the number of sides



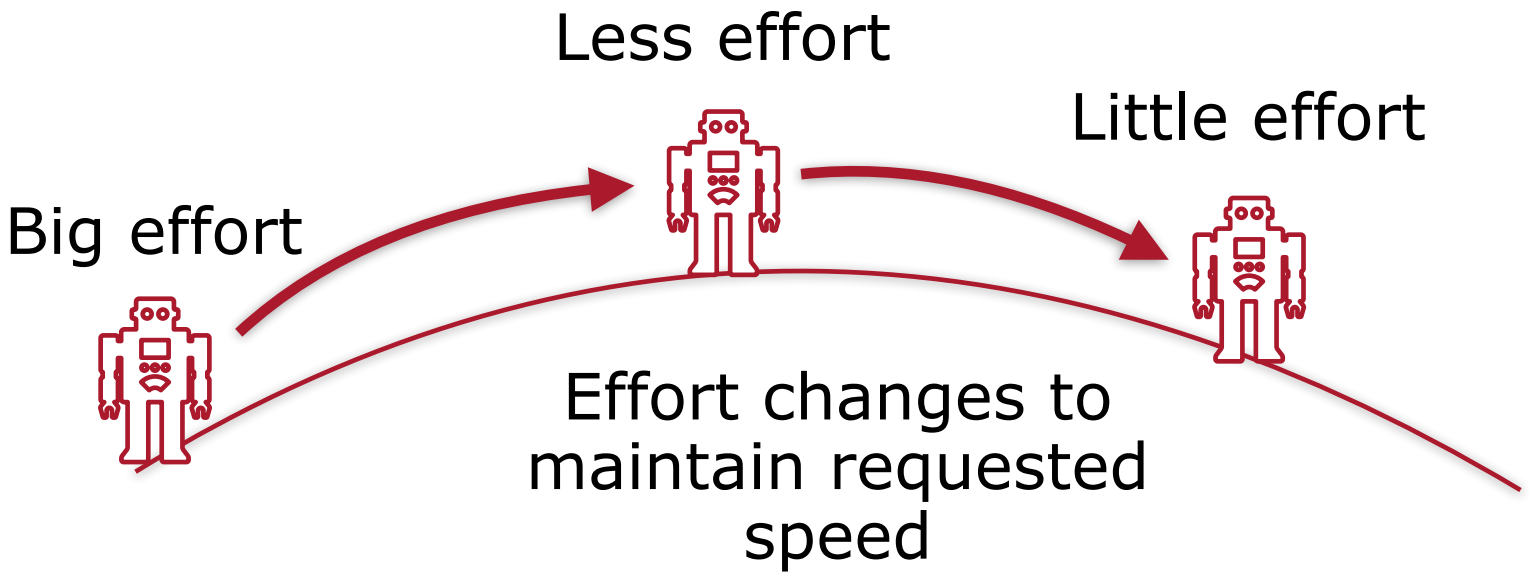
Effort vs Speed



Effort vs Speed

Effort - how hard the motors should try, effectively, the motor voltage

Speed - how fast to turn the motors, using more or less effort as necessary

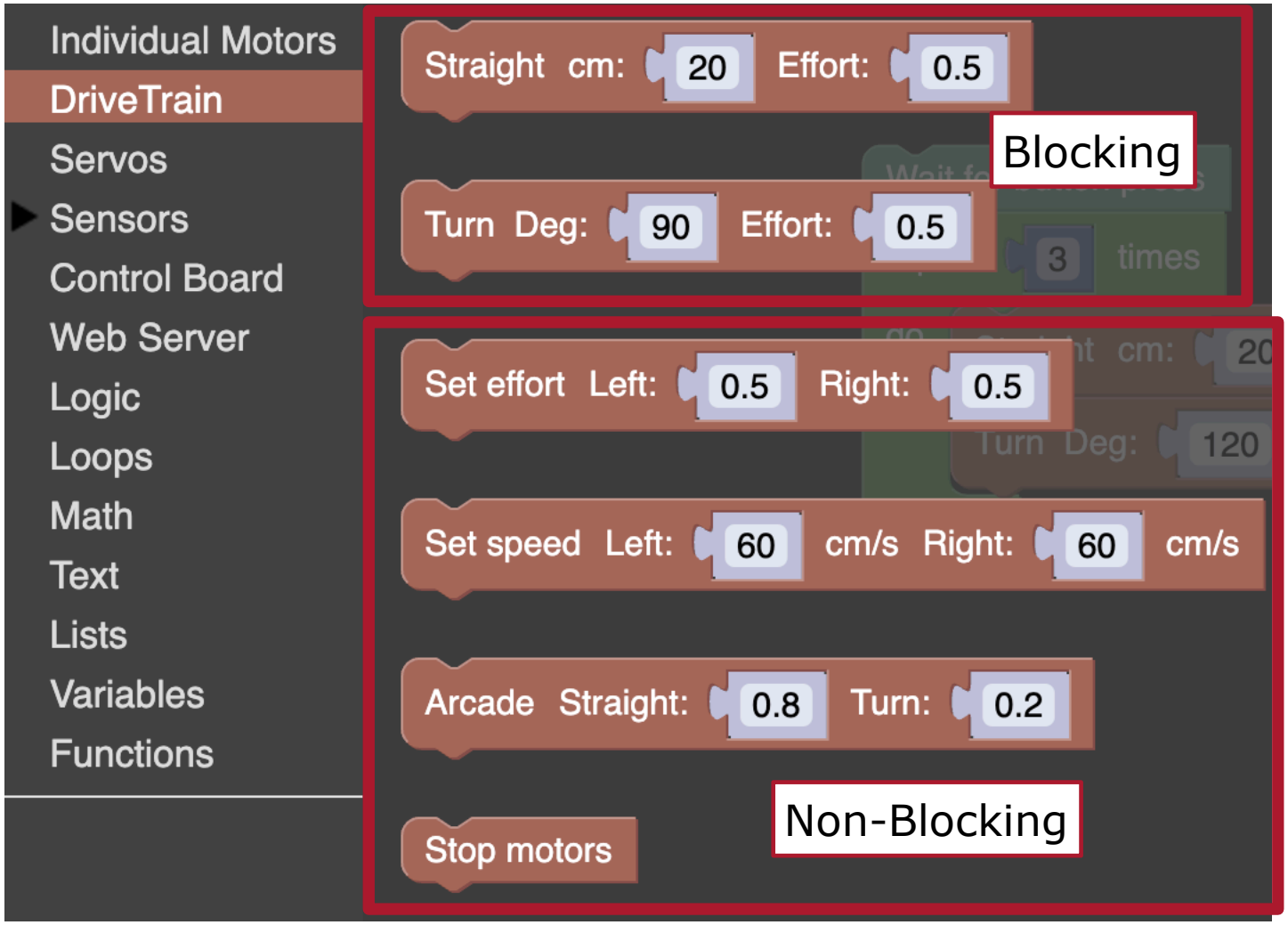


Blocking vs non-blocking

Blocking operations wait until the block completes

Non-blocking starts an operation and immediately continues

Blocking vs non-blocking



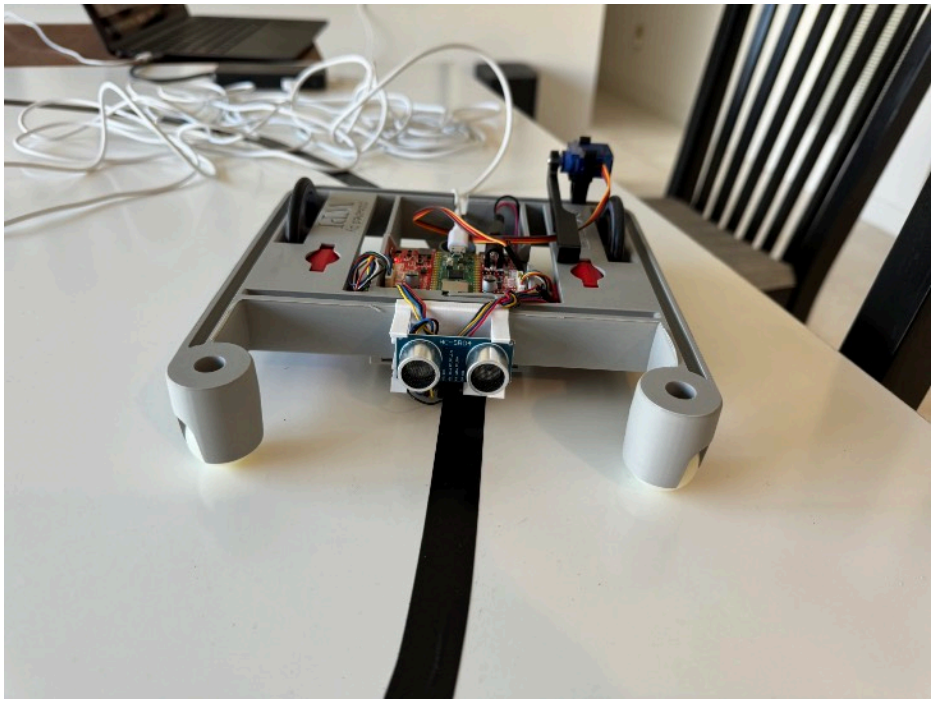
Blocking vs non-blocking

Why does it matter if the function blocks?

Typically a program is continuously reading a sensor value and adjusting the motor speeds, or moving an actuator.

If the motor function blocks then the program isn't looking at the sensor during the motor command.

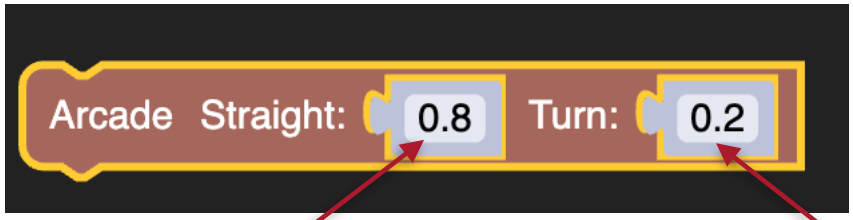
Blocking vs non-blocking



If the program tells the robot to stay on the line by moving 0.25” towards the line, then it’s not reading the sensor during that move

This is a very common student mistake!

Arcade block



Driving effort

Amount of turn

Arcade sets a forward (or backward) speed and adds the turn amount to one wheel and subtracts it from the other. Positive values turn right and negative values turn left.

```
scale = max(abs(straight), abs(turn))/(abs(straight) + abs(turn))
left_speed = (straight - turn)*scale
right_speed = (straight + turn)*scale
self.set_effort(left_speed, right_speed)
```

As you'll see, arcade is really good for sensor operations

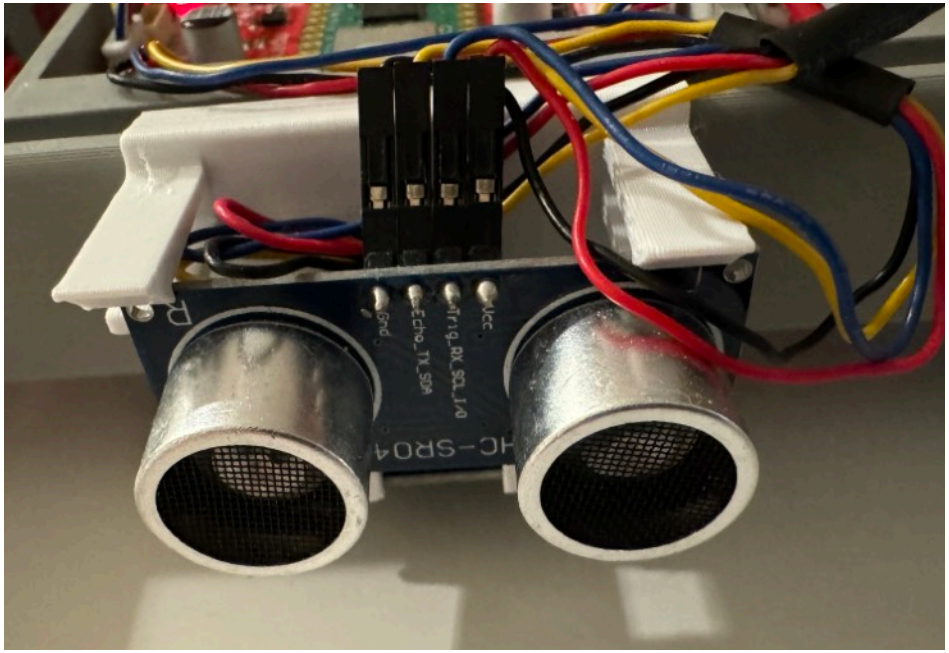
Measuring distances

Using the ultrasonic rangefinder

Measuring Distances

- In this module students will:
  - Understand the basics of measuring distance and its application
  - Learn to use distance sensors to determine the robot's location with respect to the target.
  - Use reading from the sensor to avoid obstacles or stop before the object
  - Locate and align to objects using the distance sensor

Rangefinder sensor

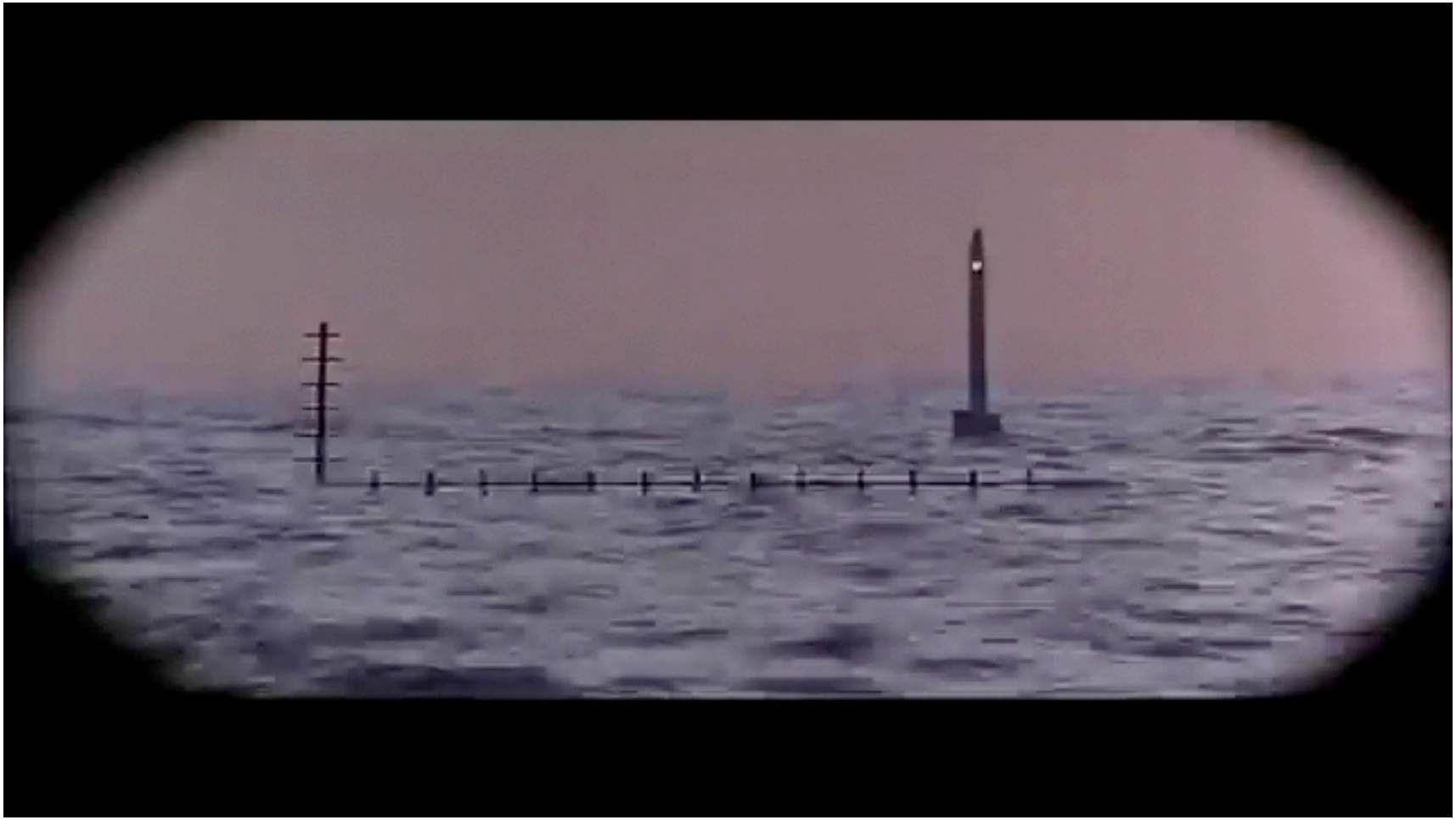


Rangefinder has a speaker and a microphone

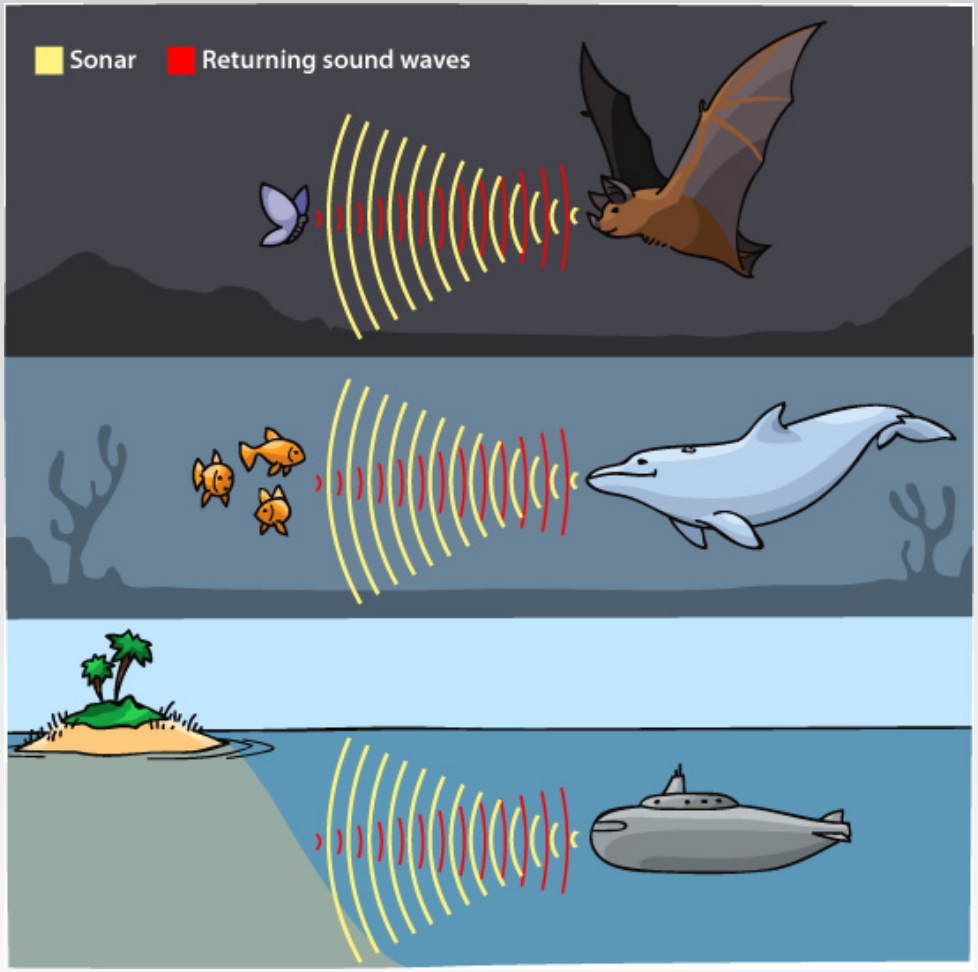
The speaker sends an ultrasonic short tone that reflects off objects and is then received by the microphone.

The sensor is then able to calculate the distance to the nearest object by measuring the round trip time of the sound - this is called **echolocation**

One ping Only - echolocation



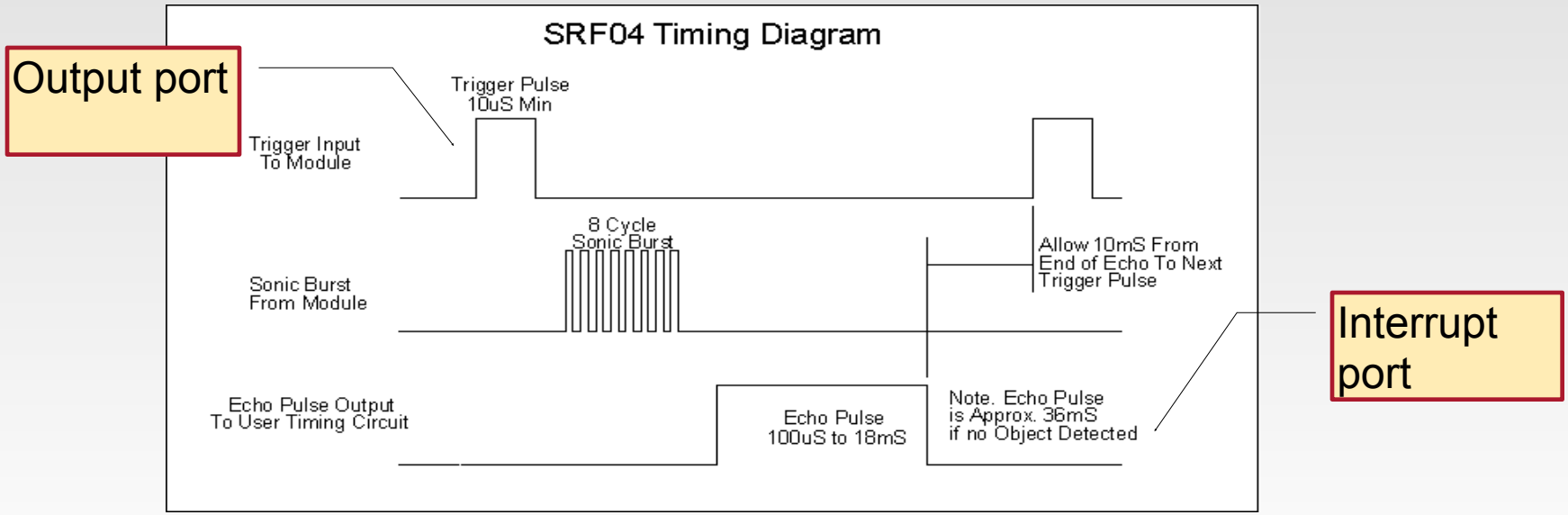
Other Echolocation Users



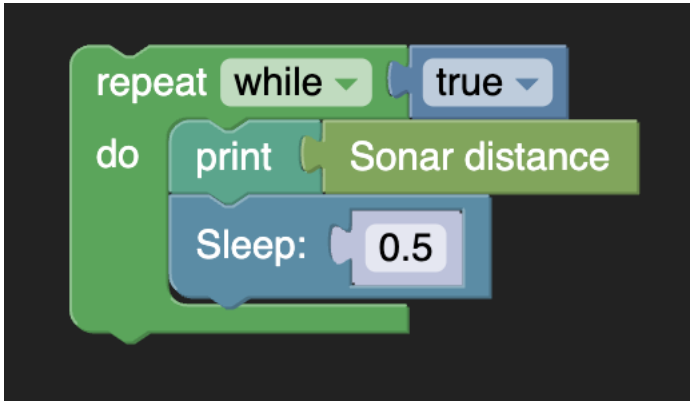
https://www.youtube.com/watch?v=laeE4icRYp4

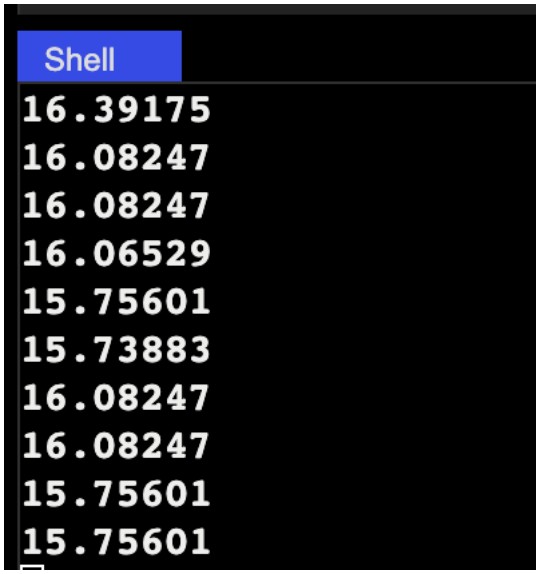
How to Compute the Distance

- Speed of sound (in air): about 343 m/s or 1125 ft/s
  - Distance to object is 1/2 the round trip time



Programming the rangefinder

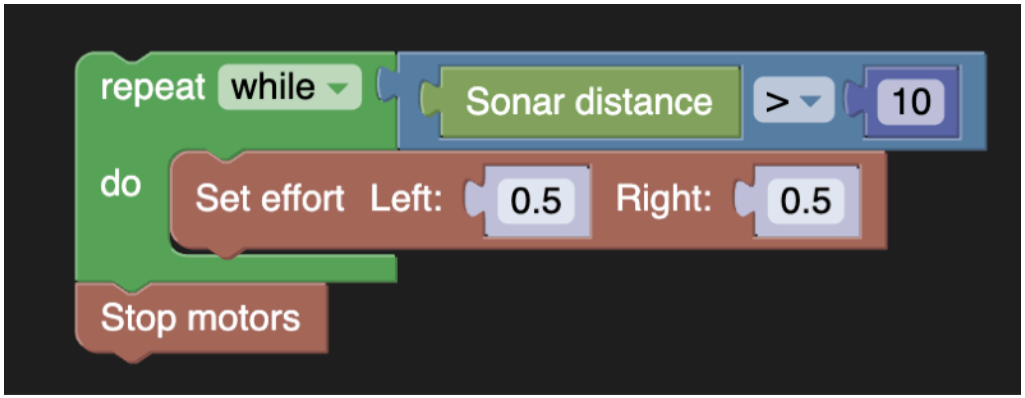


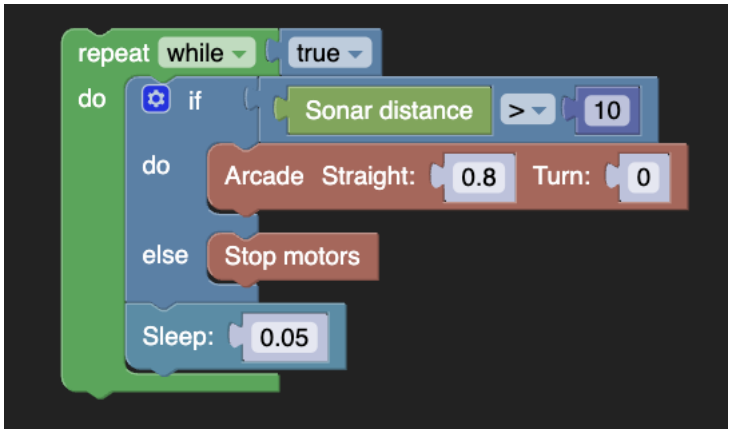


Get the distance every 0.5 seconds, then print it

Driving until near a wall

- Write a program to set the motor efforts to 0.5 while the rangefinder distance is greater than 10 cm.





Alternate solution using if/else blocks

Avoiding walls - Roomba style

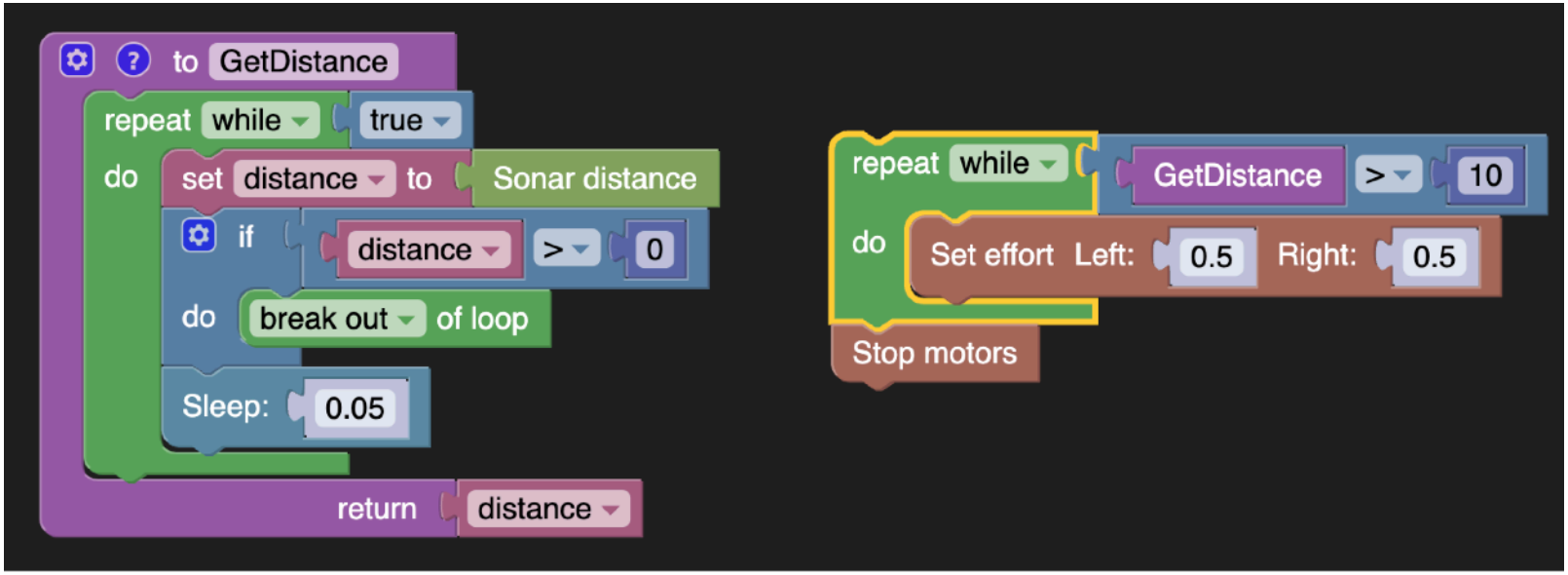
Write a program that drives until it gets to an obstacle, then turns around and continues driving

What can go wrong with this sort of program?

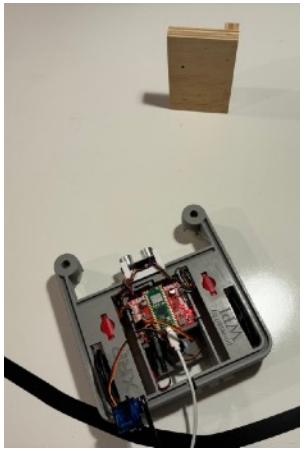
How can you fix it (think about random numbers)

Stop before the box

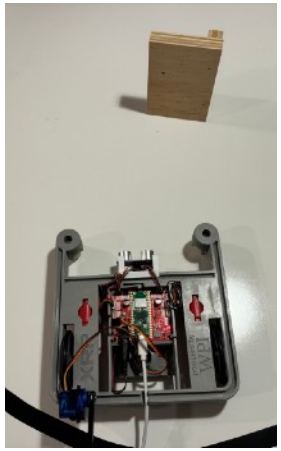
- This fixes a problem that I was seeing where the rangefinder occasionally returned a zero value



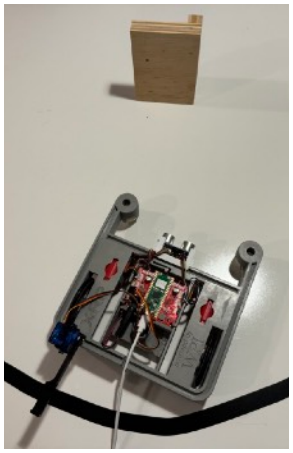
Find free-range bag



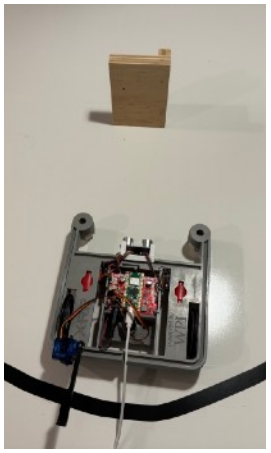
Robot is looking to the left of the target



Robot starts turning to the right and remembers the angle of the left edge

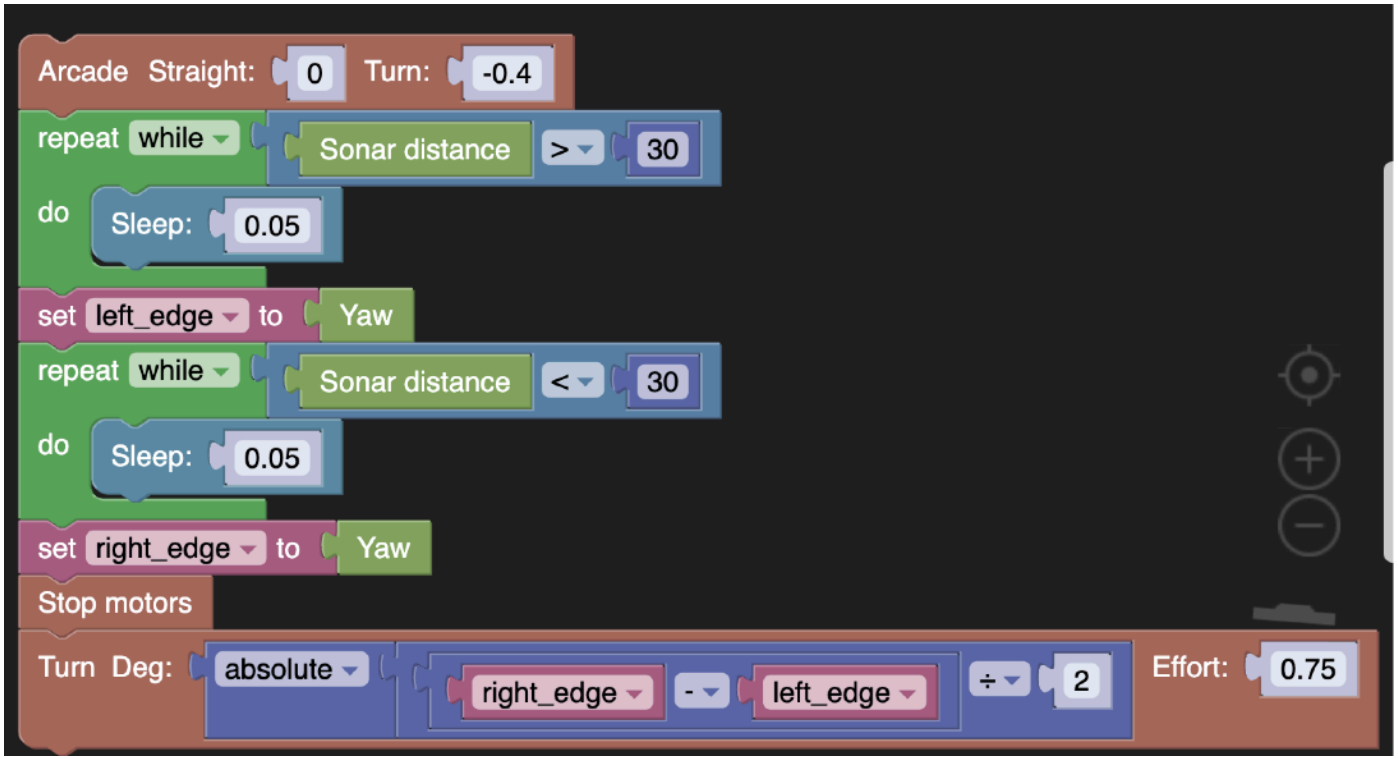


Robot continues turning and picks up the right edge



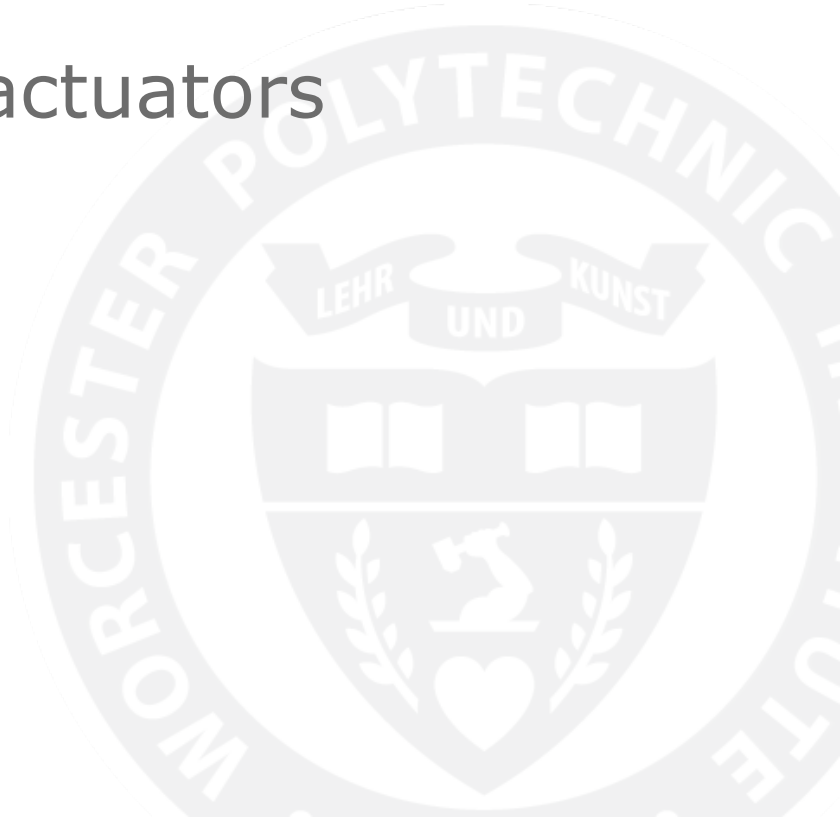
Robot turns back to the middle of the object

Find free-range bag



Robot control

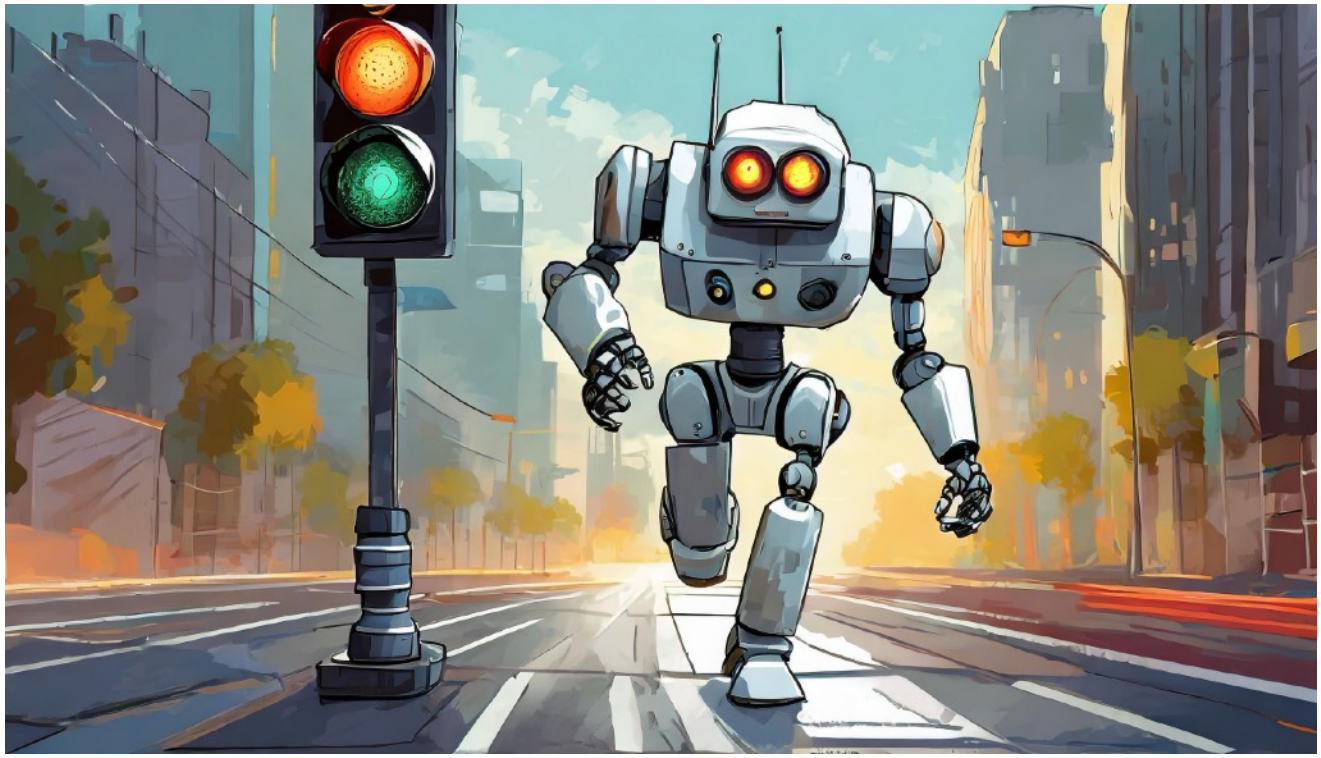
More precise control of actuators



Robot Control

- In this module you will:
  - Understand the basics of robot control and its application
- At the end of this module, students will be able to:
  - Implement simple controllers such as on-off controllers
  - Implement more complex controllers such as P controllers
  - Explain the difference between open-loop and closed-loop control

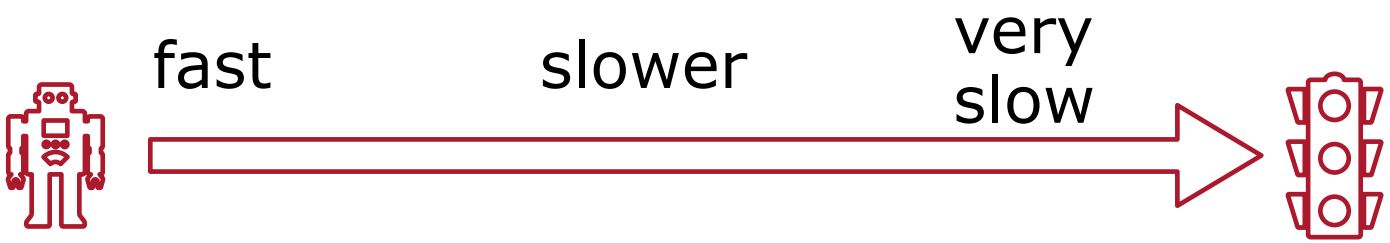
Driving fast, but stopping in time



If your robot is going fast, when should it slow down before an obstacle?

Driving fast, but not hitting the box

Think about cars… if you are coming up to a stop sign, when do you start to slow down?

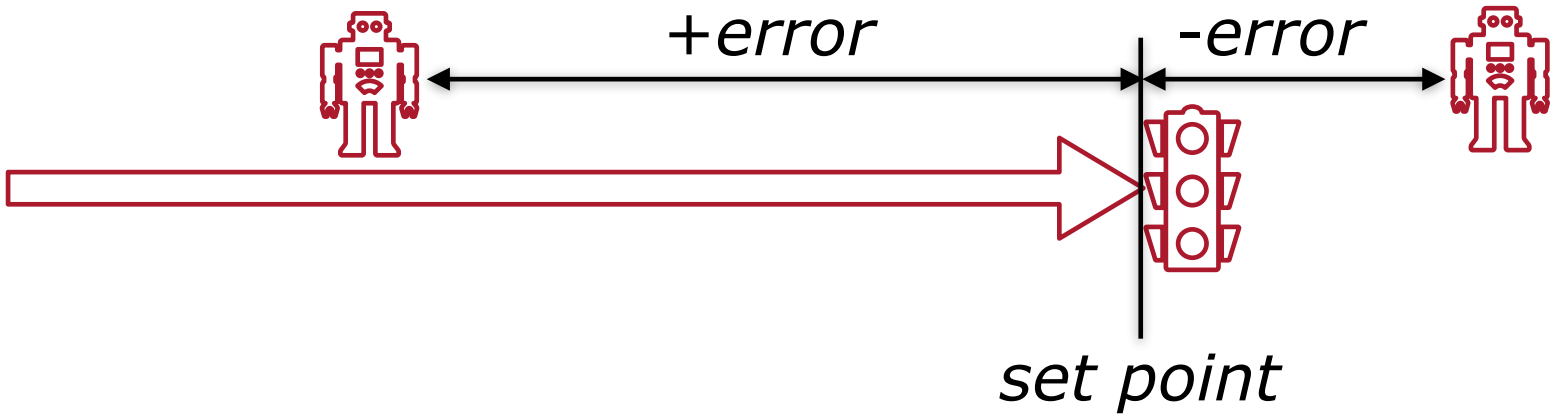


What can you say about your speed as a function of the distance to the stop sign?

As the distance decreases, the speed decreases

The speed is proportional to the distance

Proportional control

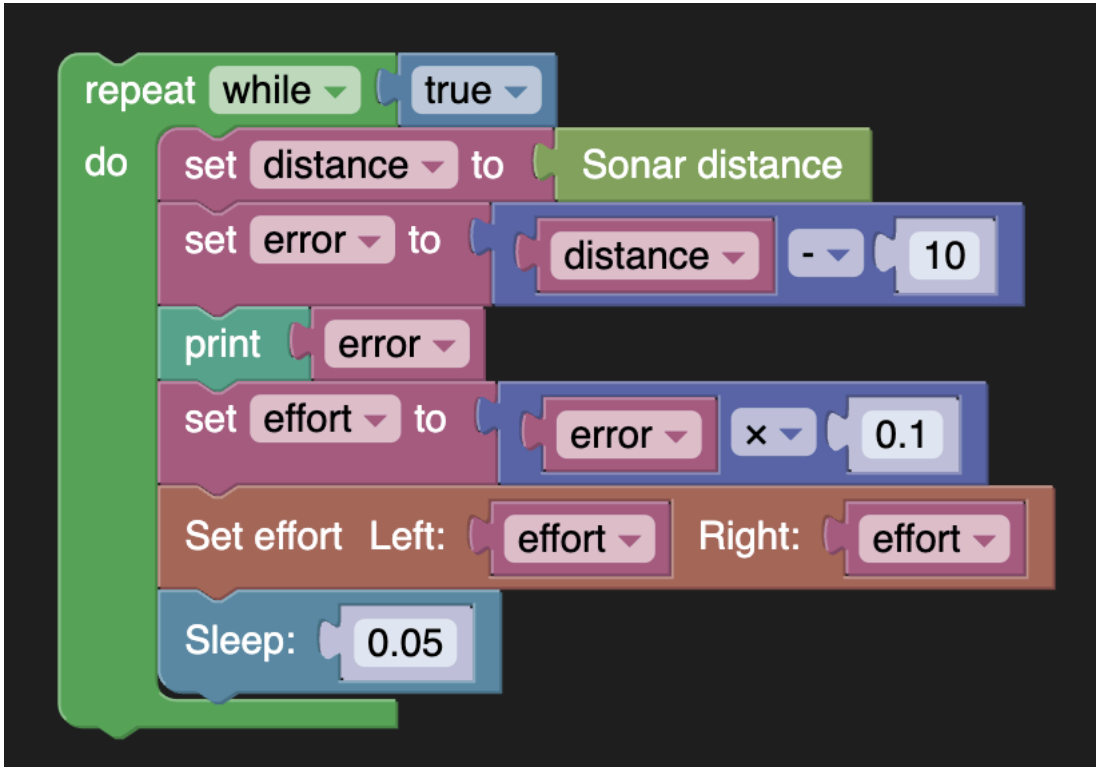


error = current position - set point

*effort = Kp * error*

Kp scales the error to a more "compatible" value

Proportional Control

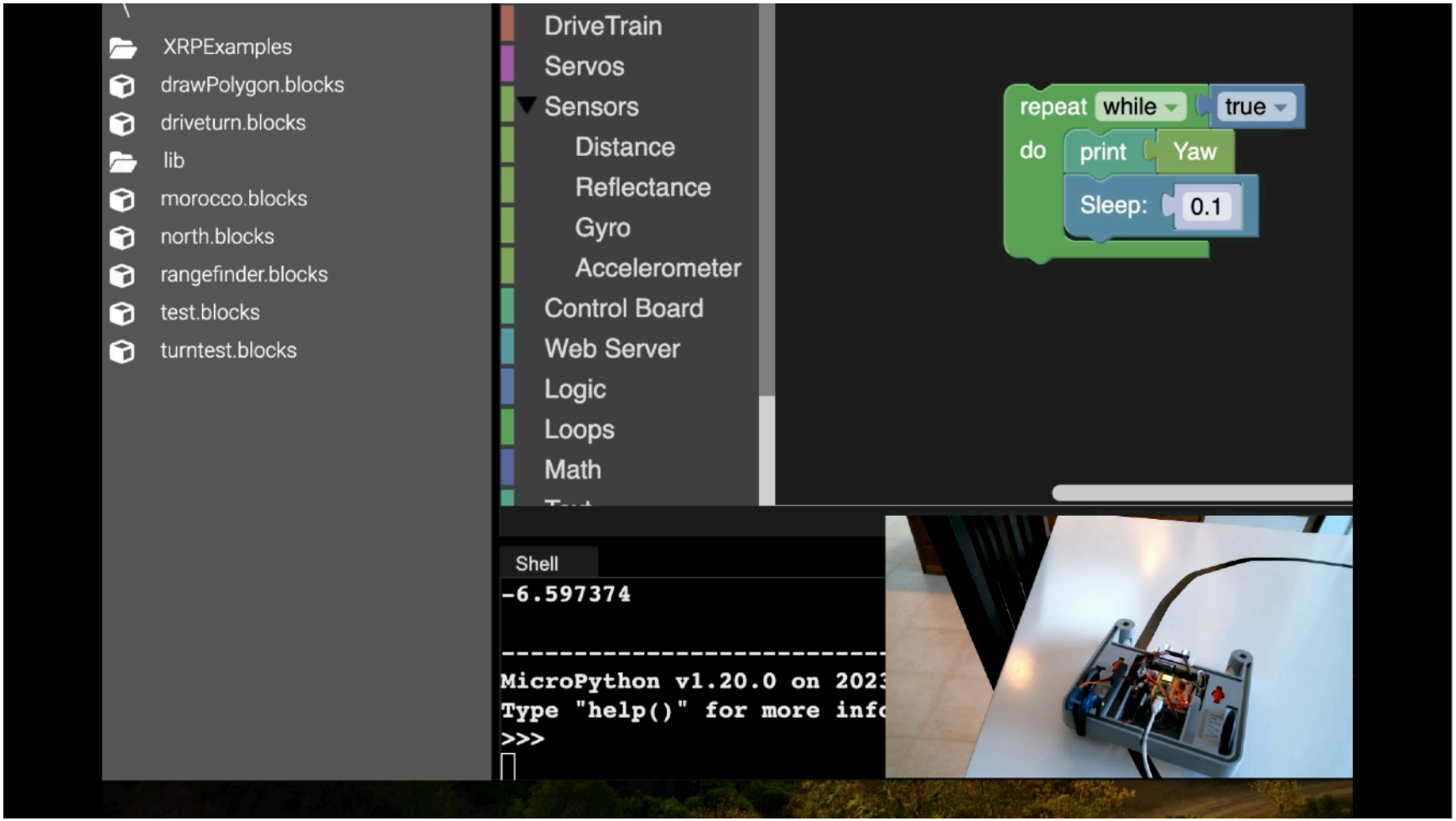


IMU (Inertial Measurement Unit)

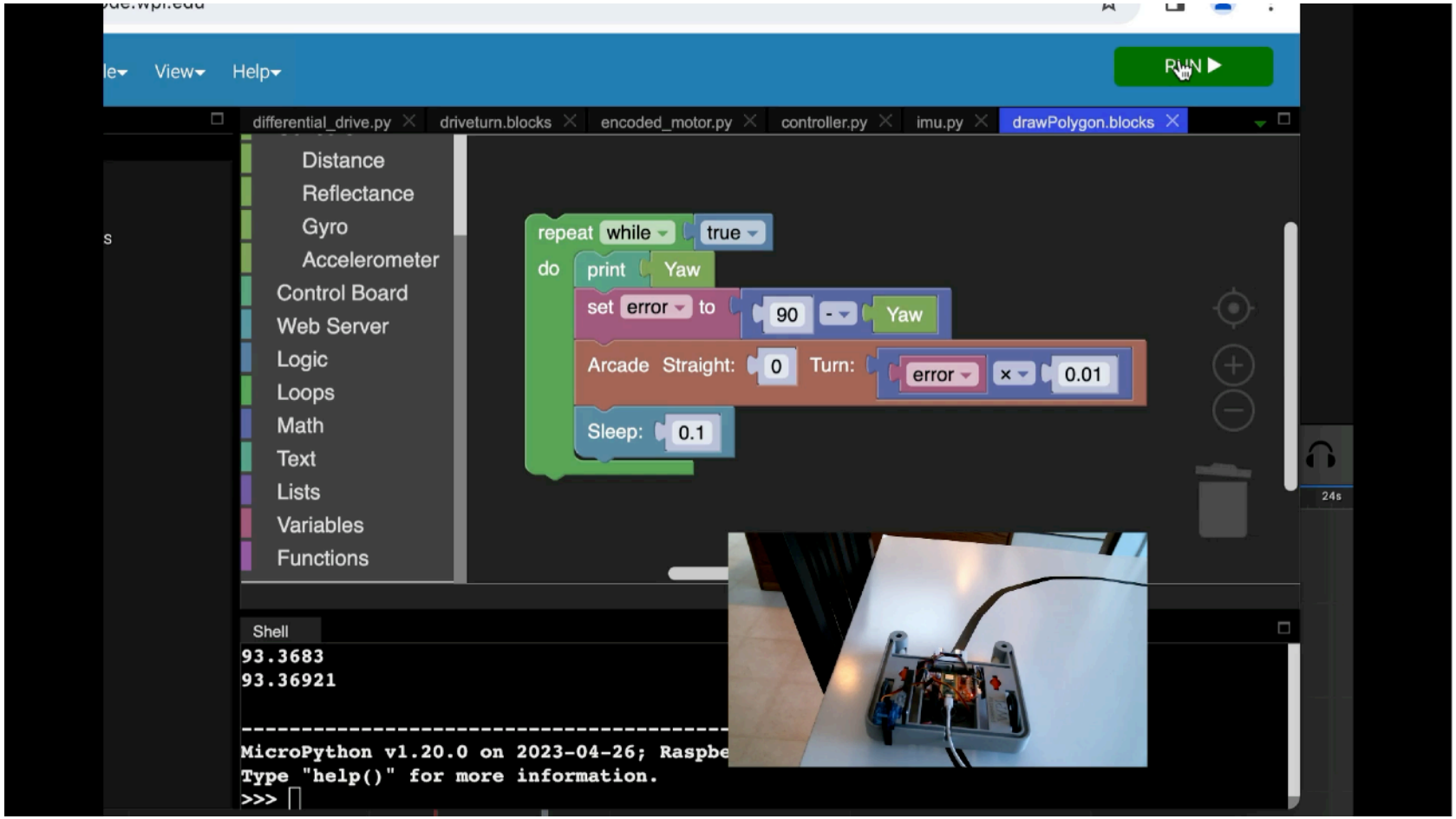
- The IMU has a 3-axis gyro and a 3-axis accelerometer
- The gyro measures rate of rotation in any access
- The accelerometer measure accelerations in any access

- The gyro (in particular, the yaw axis) can measure turn angle and can keep the robot driving straight

Gyro yaw sensor measuring turns



A program to turn 90 degrees



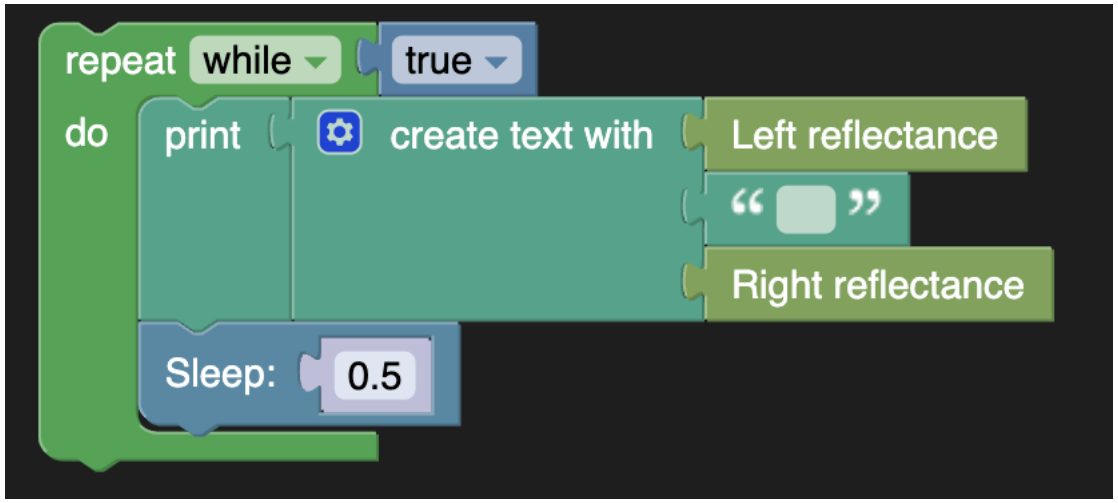
Sensing and following lines

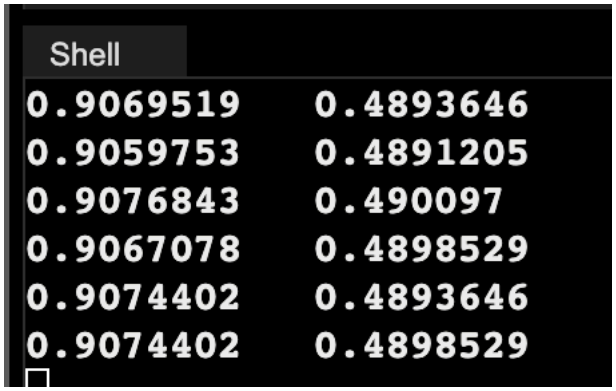
Driving a predetermined path along the ground

Sensing and Following Lines

- In this module you will:
  - Understand the purpose of following a line
  - Learn to use the reflectance sensor to detect the line
  - Write simple programs for detecting and following lines
  - Enhance the simple program with better sensor measurements and calculations to follow the line more smoothly.
- At the end of this module, students will be able to:
  - Discuss robot navigation concepts with their classmates
  - Program the robot to follow a line
  - Develop simple sensor logic control for robot decision-making

Understanding the reflectance sensors

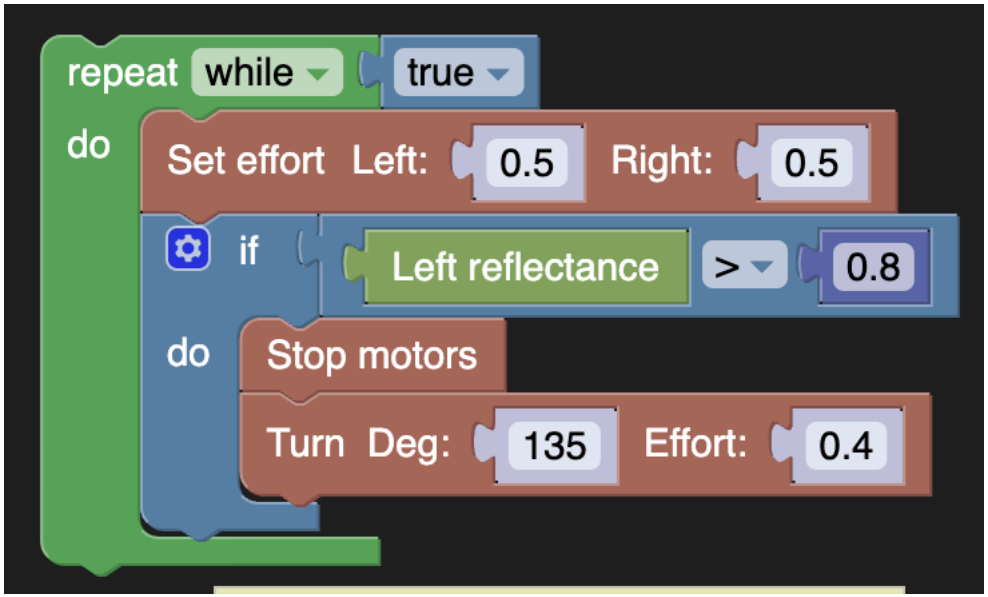




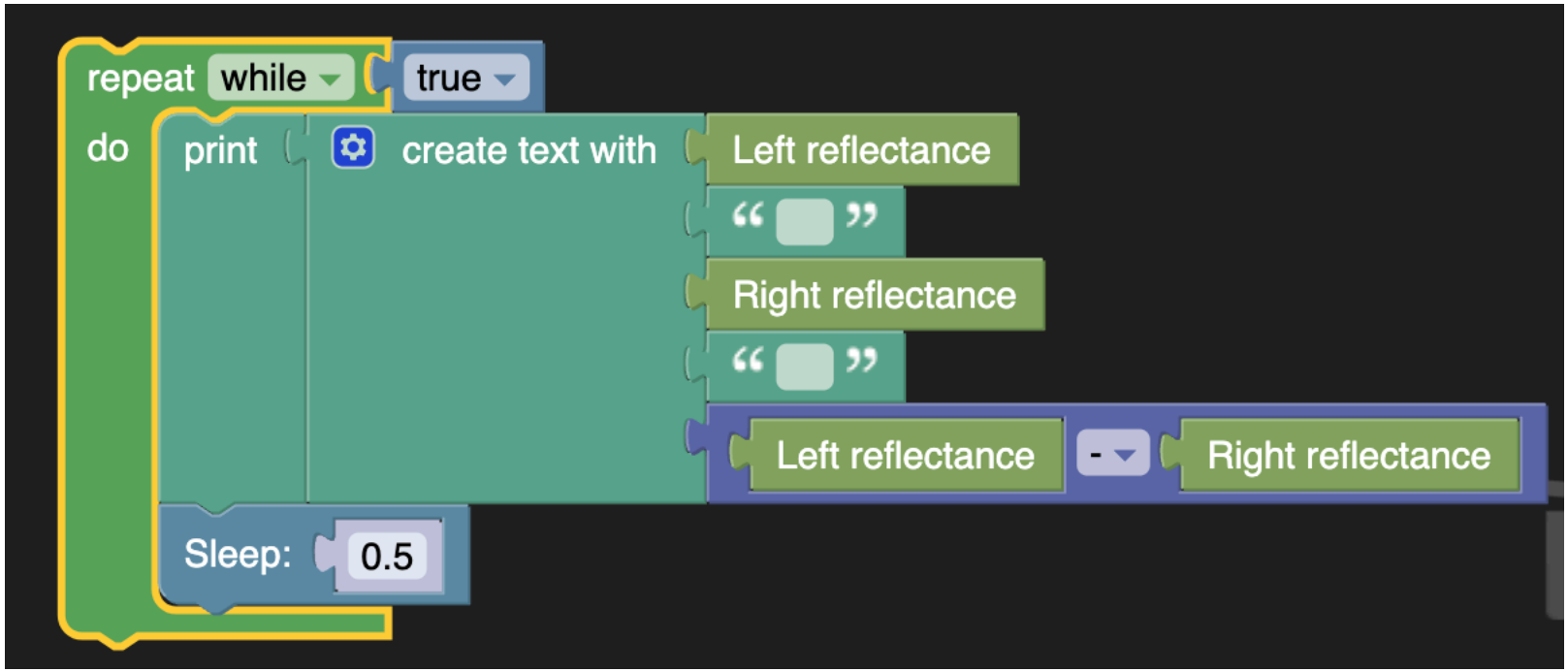
Black White

Look at the value of the left and right reflectance sensors as you move the robot across the line

Stay within the circle



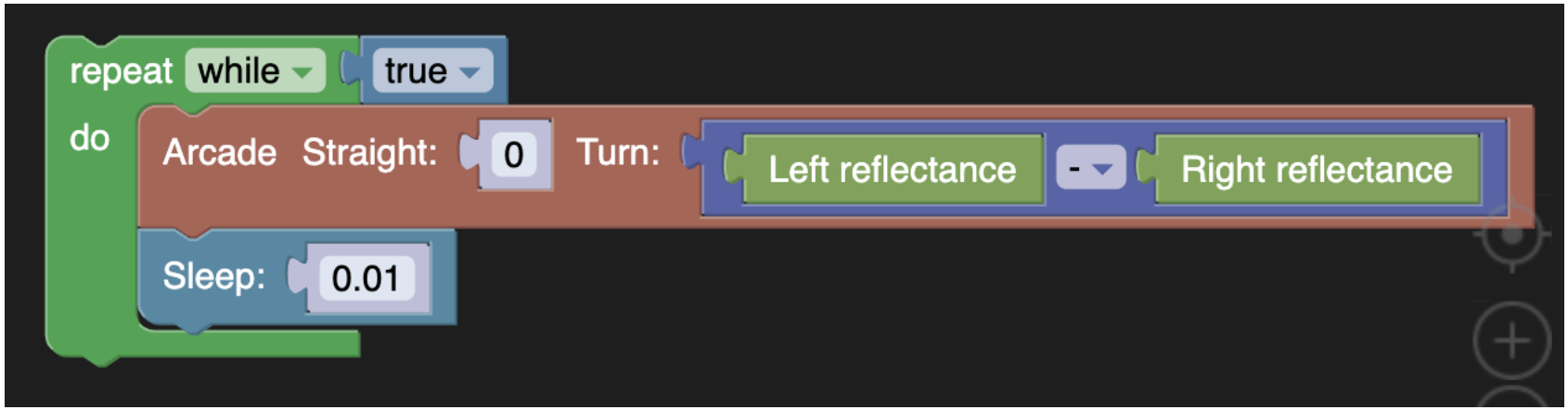
Look at the difference between left and right



How can you use that for line following?

Following lines with proportional control

Use the difference between the left and right reflectivity sensors



Fastest time around the circle

Drive around the circle turning around at cross and returning. Maybe multiple circuits.

There is a speed vs accuracy tradeoff that has to be made

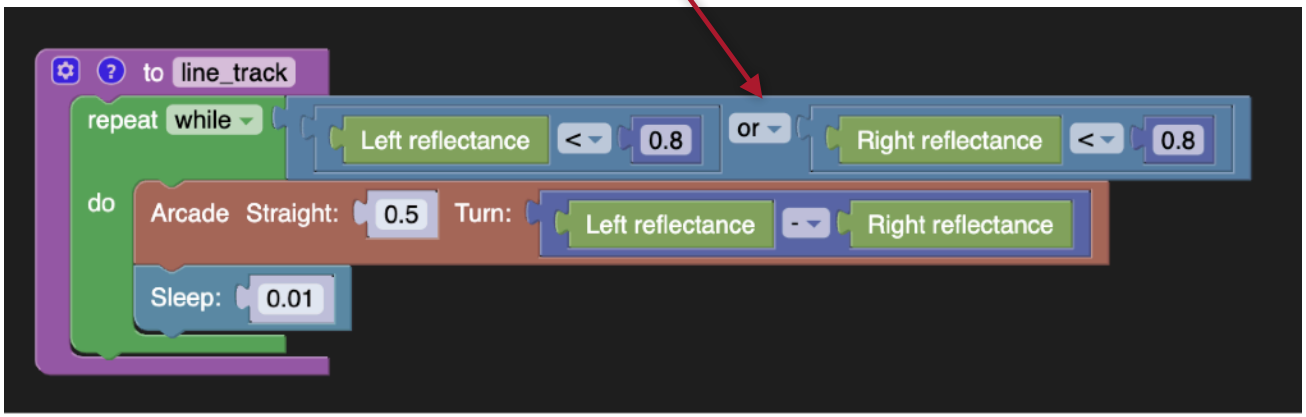
Sometimes, the robot will track in one direction, but not the other

Nuclear waste delivery challenge with rubric

Line track until the cross and turn around

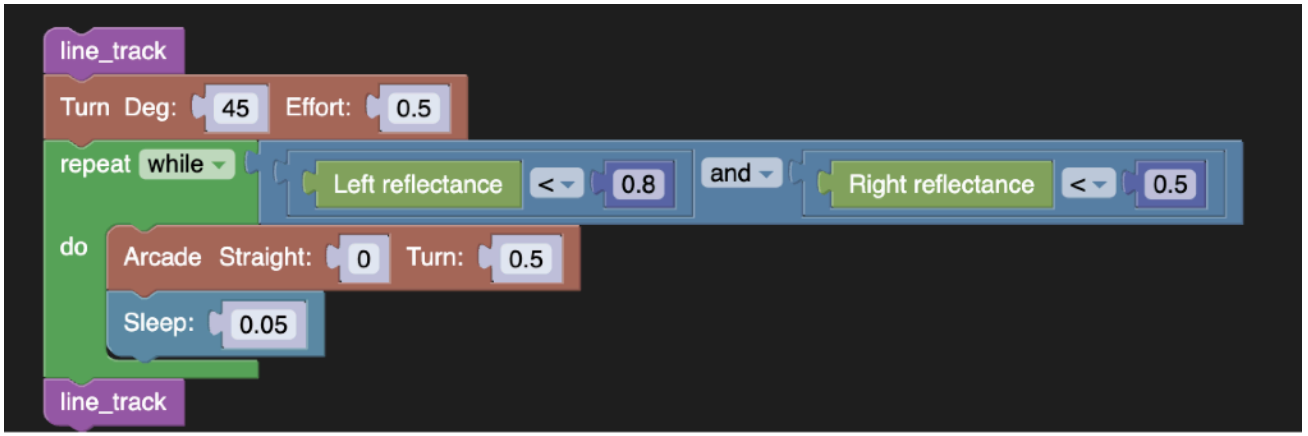
Create a line tracking function that stops at the cross

Recognize the cross



Line track until the cross then turn around

1. Line track until the cross
2. Turn away from the line
3. Continue turning until at the line going the other way
4. Line track some more



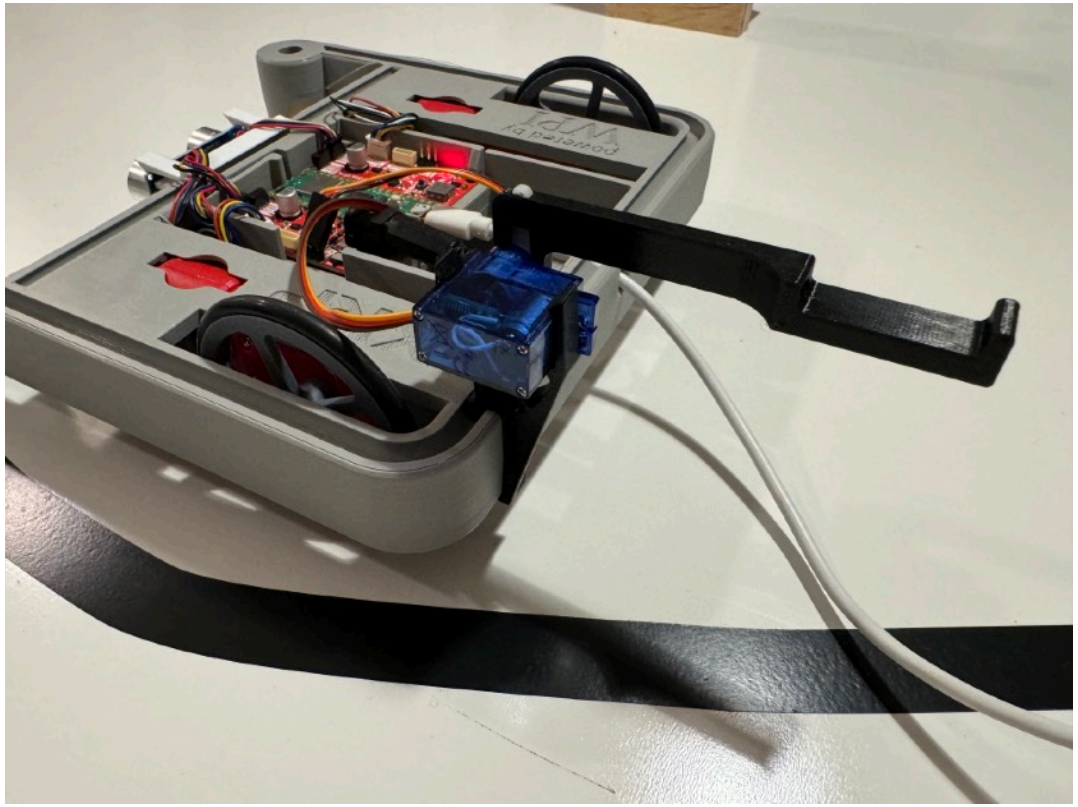
Manipulation

Interacting with objects

Manipulation

- In this module students will:
  - Understand the basics of manipulation and its application
  - Learn how to control the XRP robot arm
- At the end of this module, students will be able to:
  - Write a quick program for controlling the robot arm
  - Implement custom functions for controlling the robot arm
  - Manipulate their environment using the robot arm

Servo

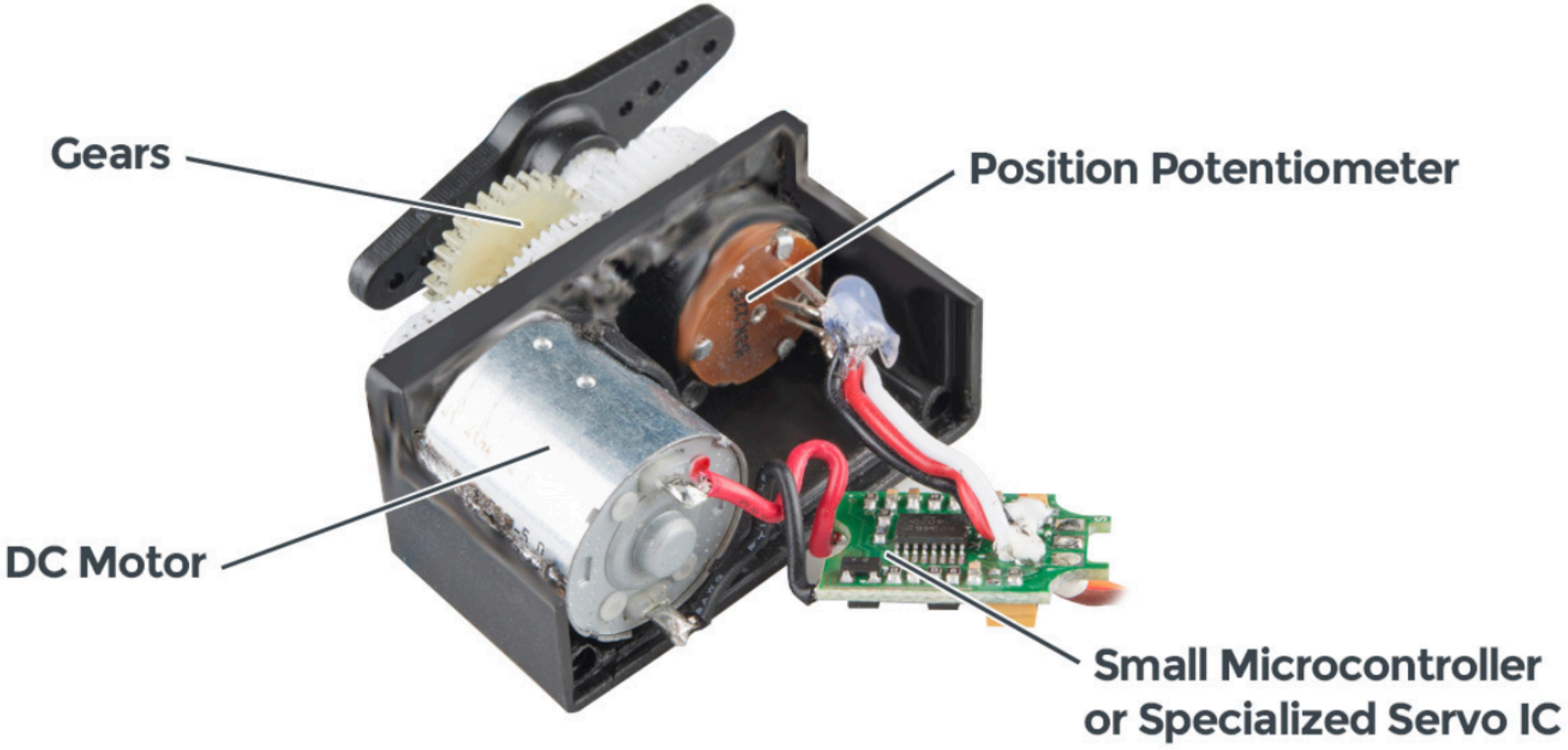


The servo is a motor with a built-in sensor that determines the position.

The arm can move to a specified position

All the control logic is built into the servo

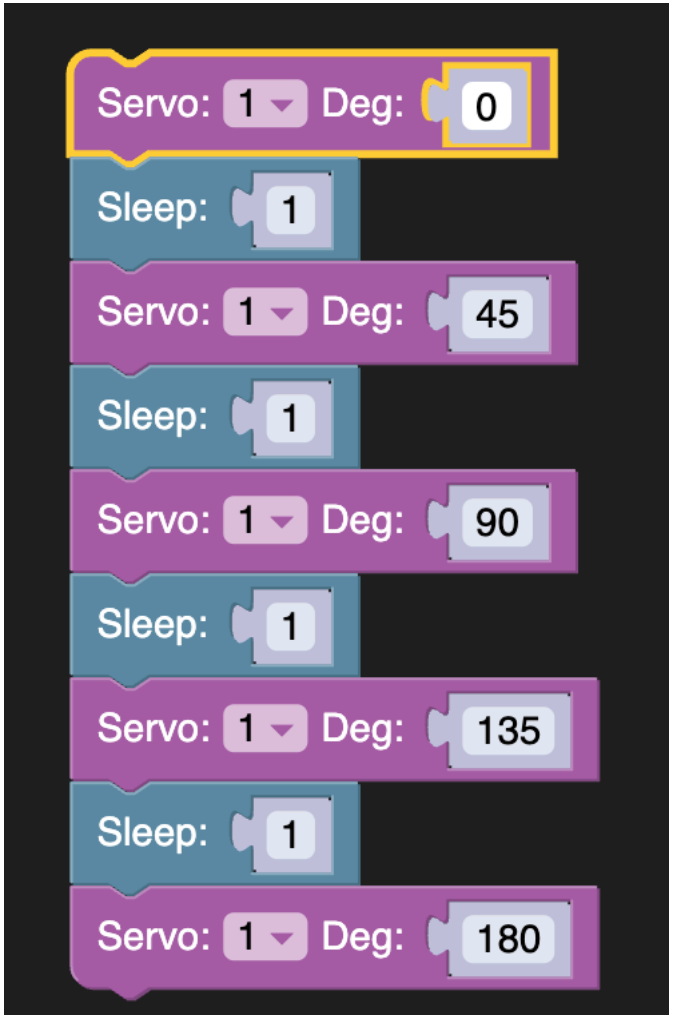
How servos work



Inside a standard hobby servo

https://www.sparkfun.com/servos

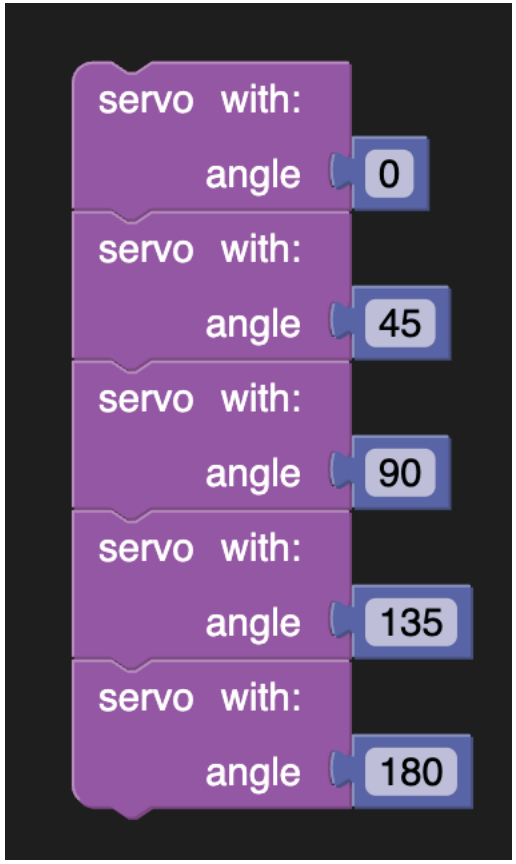
Servo program with five positions



The wait guarantees that the servo has time to reach the target angle

Servo program with five positions

Function that goes the the specified angle and waits

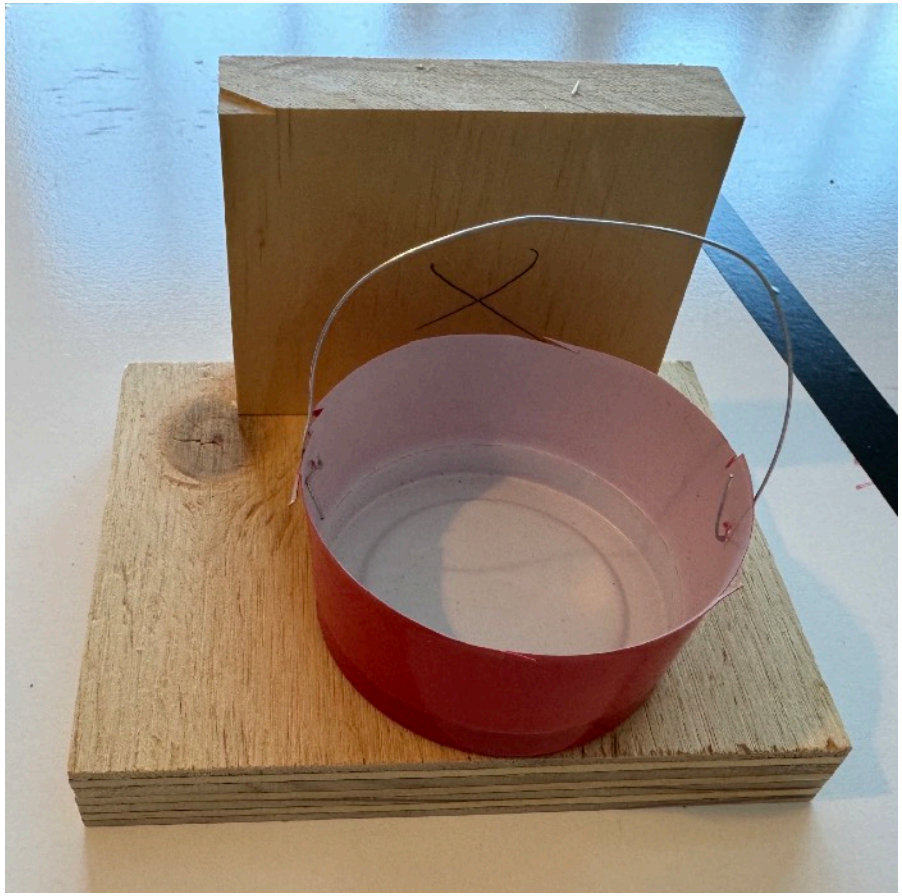


Same program, but using the function

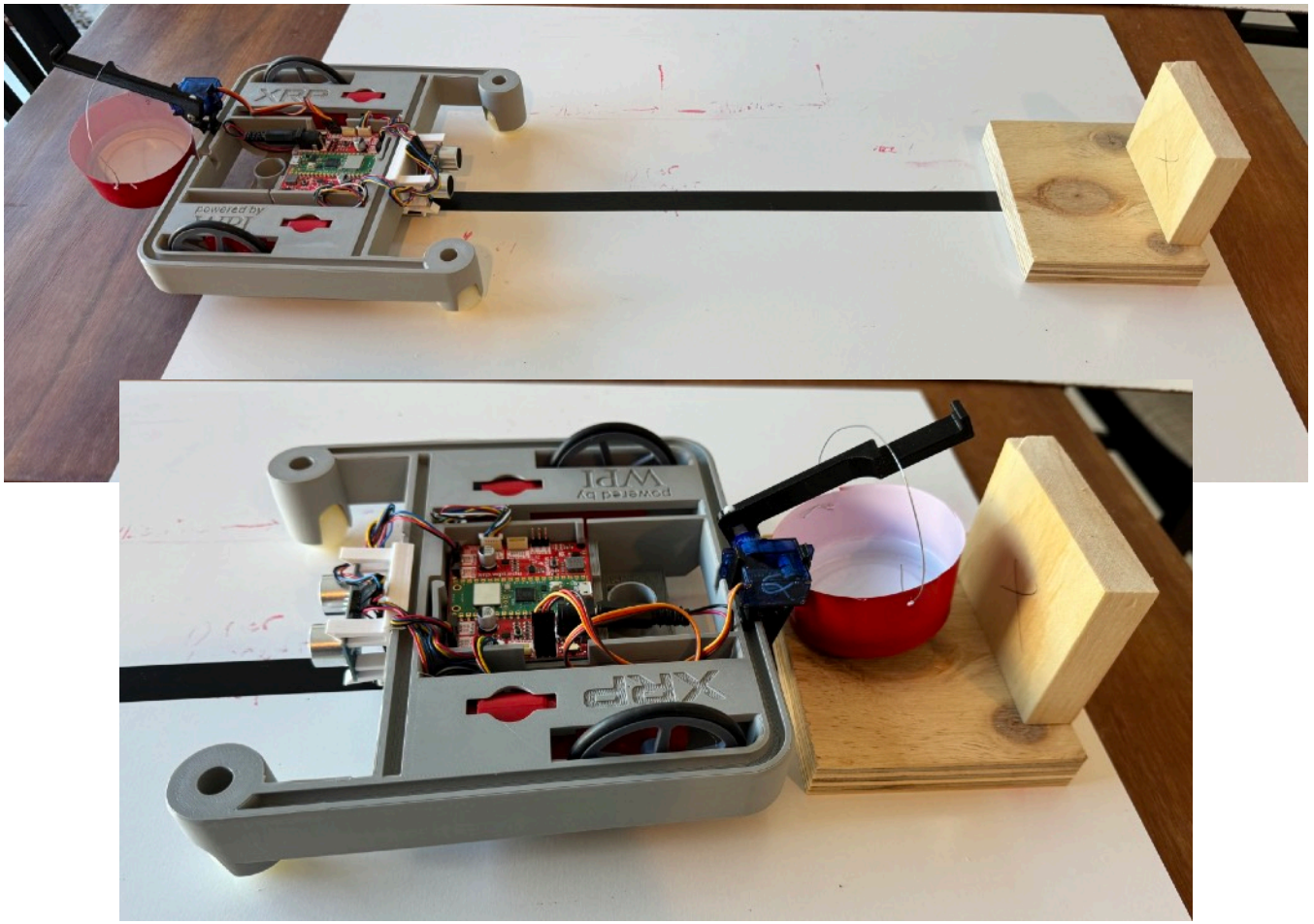
Delivery objects

The cup is cut down from a full size drinking cup.

The delivery platform and backstop make a target for the ultrasonic rangefinder



Lift an object and drive

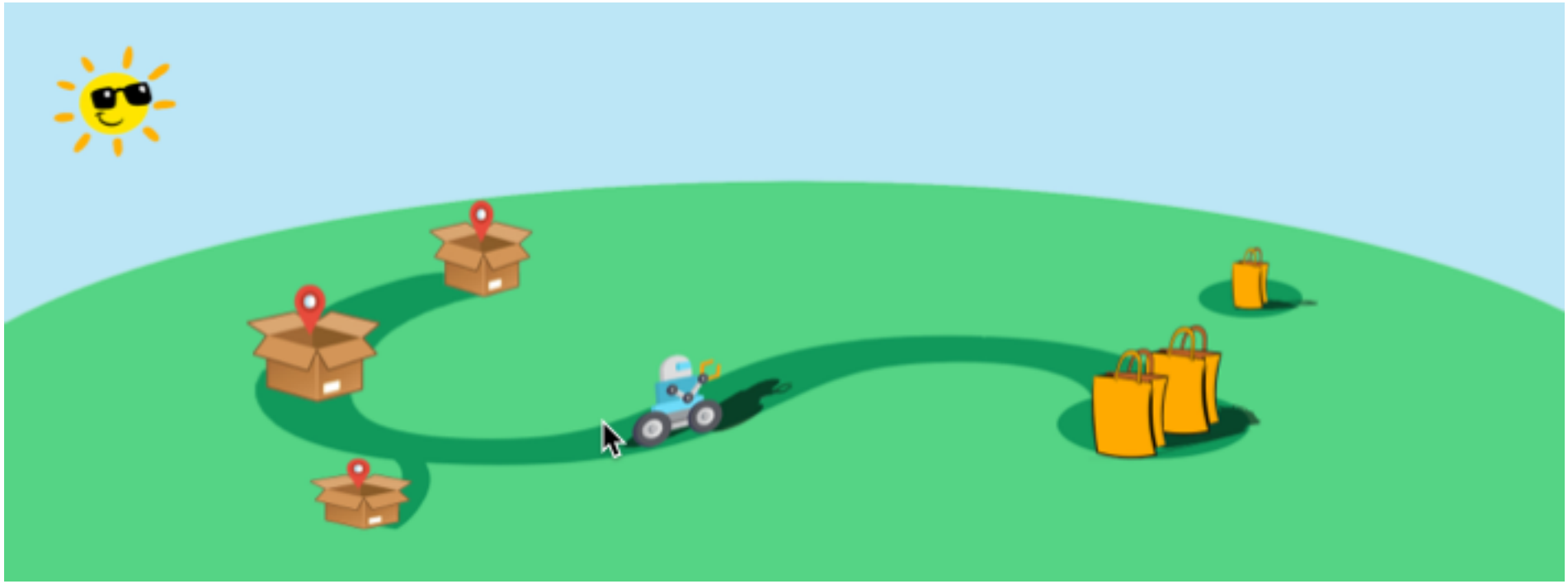


Delivery challenge

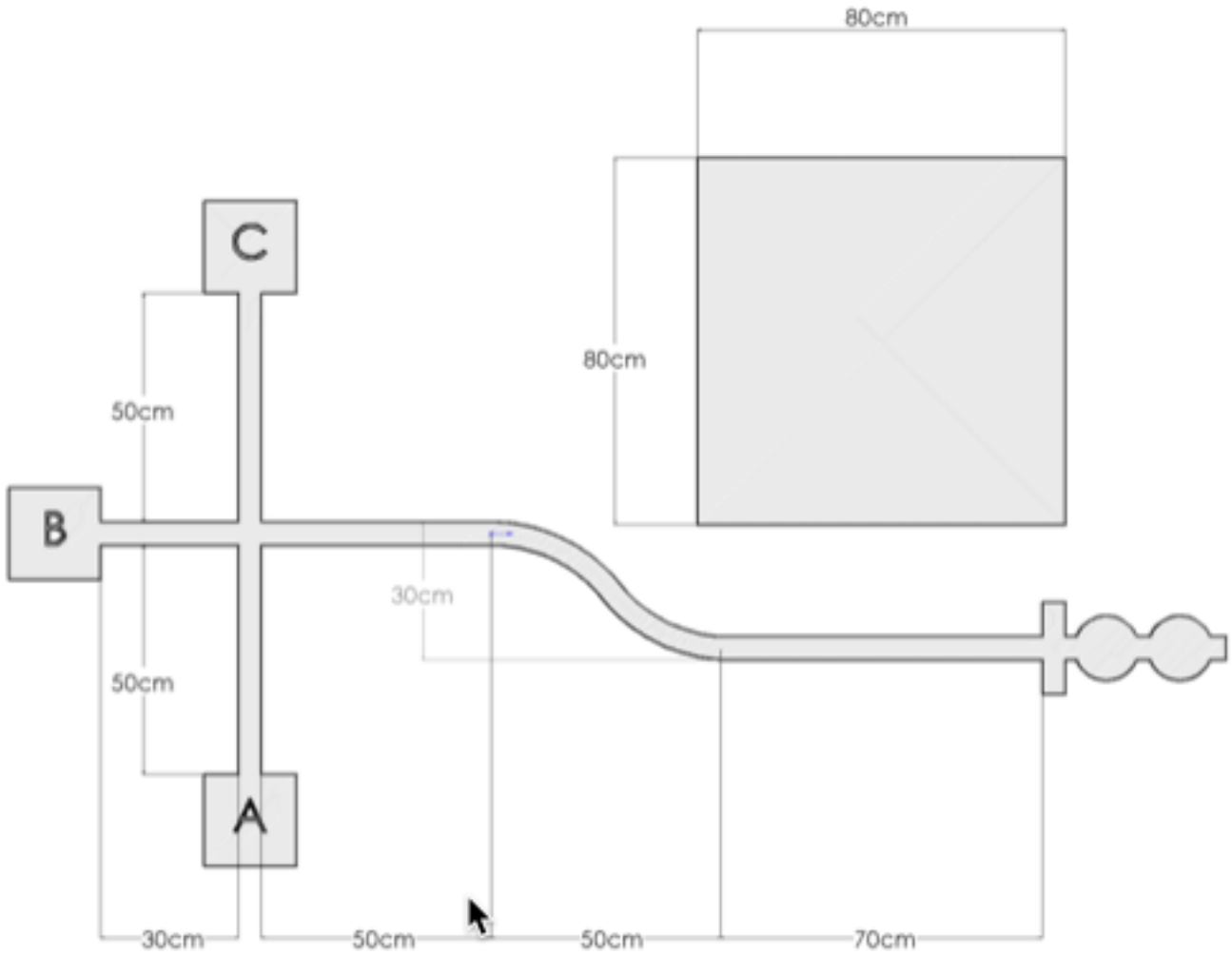
Putting everything together in one extended project

Final Project

- Discuss the Delivery Challenge final project
- Implement a scaled-down version of the challenge
  - Put together everything learned into a single program



Delivery Challenge



Options for challenge

- Points for number of cups delivered
- Extra points for covering all 3 destinations
- Extra points for free-range bag
- Points for time to complete challenge

- Can be a class competition or an individual skills competition

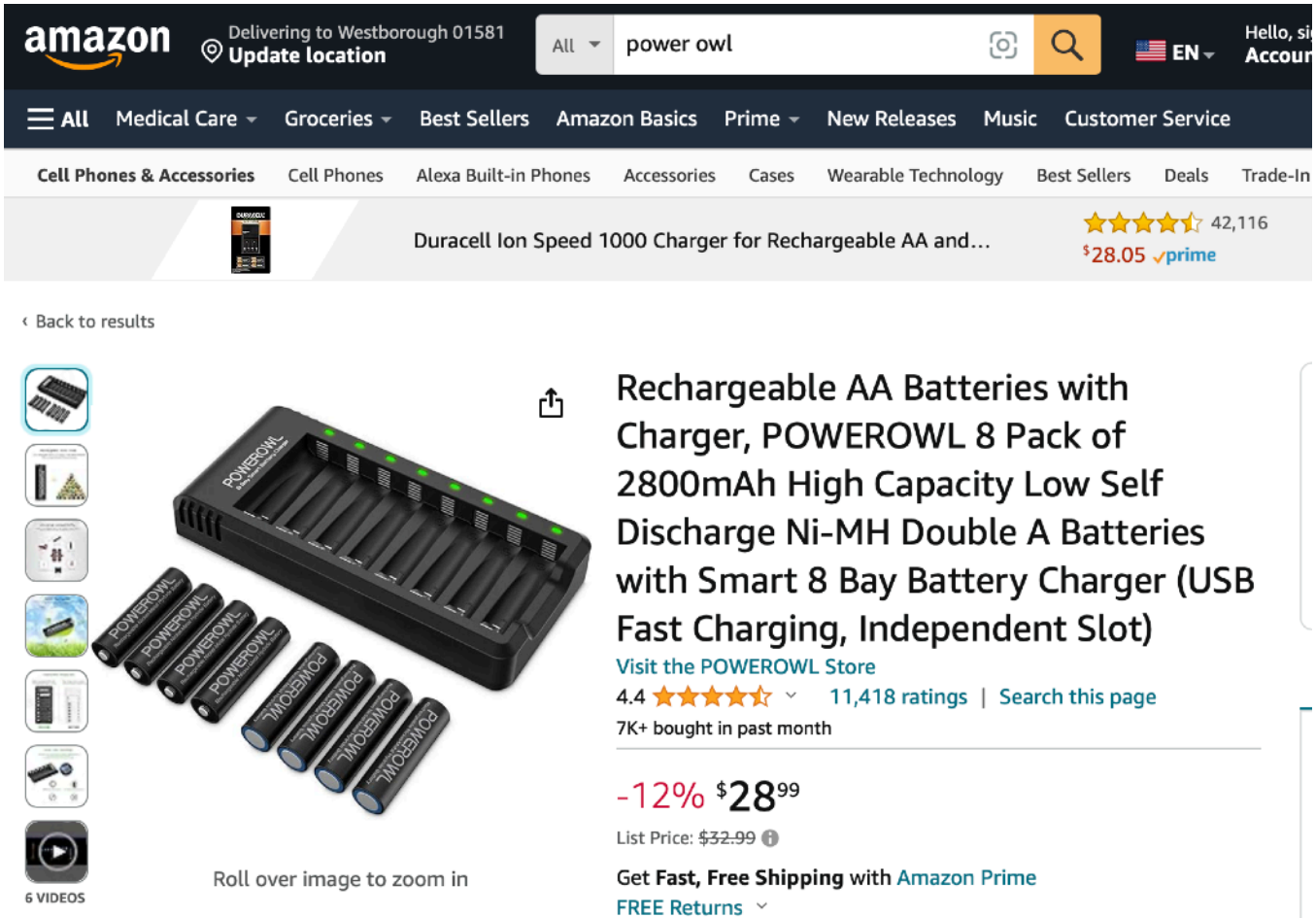
Resources for teachers

Tools to help you be more successful

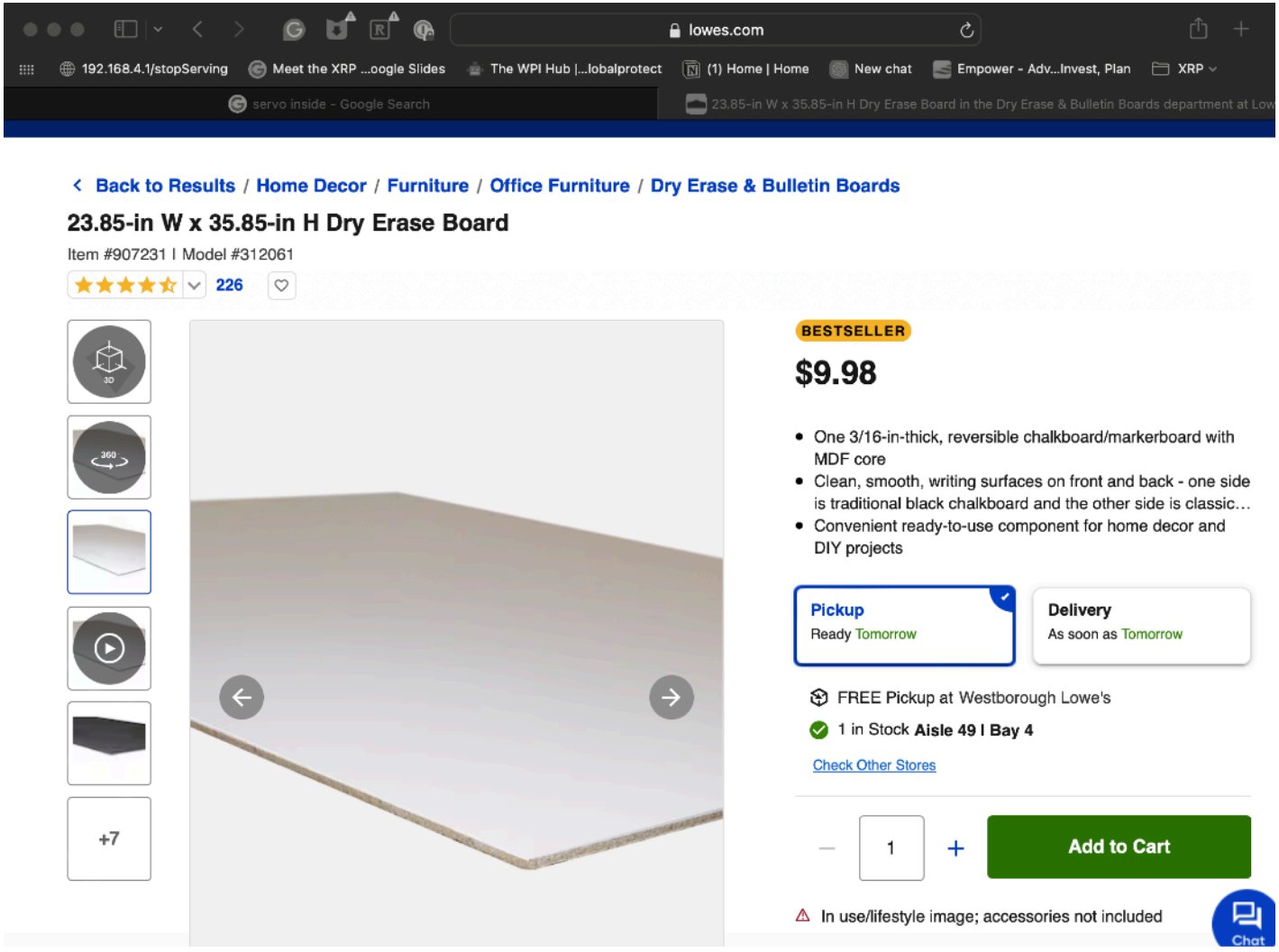
Resources for teachers

- Technical support
  - Discourse server
  - Contacts within WPI
- Materials required in addition to the robots
  - Whiteboards and markers
  - Batteries and chargers
  - Spare robot kits or spare parts

Suggested batteries



White board



Also available in 4x8

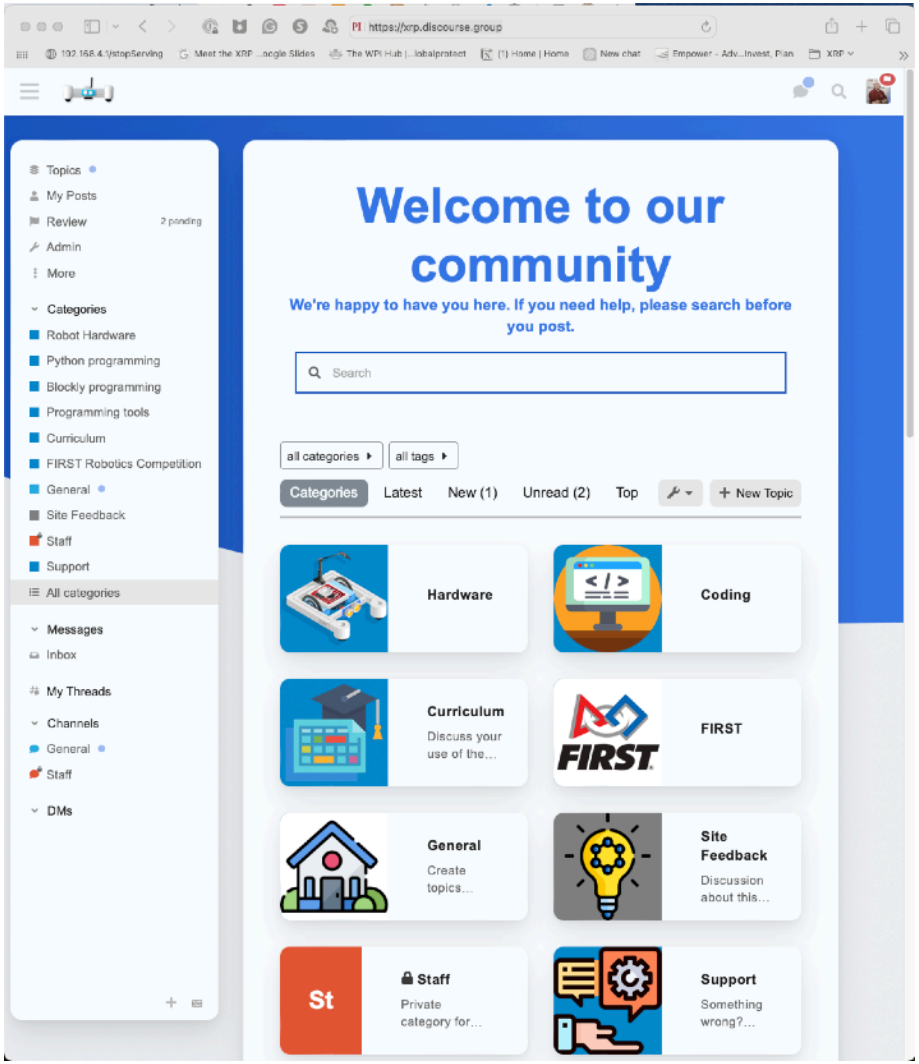
Teaching suggestions

- Try to intermix lecture with coding so students are always doing projects (project based learning)
- Problem: some students will have more experience than others, and will get ahead
- Have additional problems “in your pocket” ready to give those more experienced students so everyone in the class is working on the robot

Team sizes - my thoughts

- 1 student per robot - nobody to work through solutions
- 2 students per robot - students can work together and practice some teamwork
- 3 (or more) students per robot - inevitably two students will work on the robot and one will watch (or worse, play with a phone)

https://xrp.discourse.group/



Community general support and Q&A for XRP

Lots of topic areas and good resource for getting help

Links to related web sites

- https://experientialrobotics.org/ - all things XRP and a good place to see what is happening
- https://xrpusersguide.readthedocs.io - XRP User Guide that discusses how to make the XRP Robot work including assembly, motors, sensors, and other software things. Includes links to other sites
- https://www.canvas.net/browse/wpi/courses/introduction-to-the-xrp - The Introduction to Robotics curriculum that we discussed today
- https://www.printables.com/model/576581-xrp-robot-part-of-the-experiential-consortia/collections - XRP models for your own 3D printed parts

Next steps

- Integration plans for XRP curriculum in the classroom
  - Number of days and time per day?
  - Numbers of students and what grade?
- Assessment of class

Future Opportunities

- Beyond the curriculum, coming soon
  - Introduction to Programming, a Python programming course using the XRP as an engagement device
  - Design and 3D Printing, a course to teach mechanical design and actualization through 3D printing. Students will build accessories for the XRP Robot for more advanced games and challenges

FIRST Robotics

- We see joining a robotics team as the ultimate experience that a high school student can have
- Teaches teamwork, open-ended design, critical thinking, and the ultimate project-based learning experience
- We hope that using XRP will give teachers and students the experience and confidence to start of join a robotics team.

Q&A

- How would you use this in your classroom?
- How do you see other teachers using this?
- What are the barriers to getting it into the classroom?
- Who would you invite next?
- What do you like? What is unique? What isn't there?
- Are you planning on one to one or two to one?
- Thoughts on how we should change future days?