ROBOTICS FROM SCRATCH

Robotics Basics

- Today we will look at some of the basic problems of robotics and how we will be approaching and solving them in this class

Basic Robotics Problems

- Movement and steering
- Obstacle avoidance
- Direction finding and navigation
- Path following
- Combining everything

Problem: Movement and Steering

- We will need to make multiple motors work together to drive forward, backward, and turn smoothly and accurately.
- How do we control these motors?
- How do we turn the voltage on and off to drive the motors?
- How do we change their speed?

Problem: Obstacle Avoidance

- Our robot must avoid bumping into objects as it moves around, keeping itself and its environment safe.
- How does the robot know when it’s getting close to an obstacle?
- How can it measure distances without touching anything?
- How does it decide to stop or steer around something?

Problem: Direction Finding and Navigation

- The robot needs to know which way it’s facing to drive straight or turn to the correct angle.
- How can the robot sense its own orientation and heading?
- How does it keep track of direction while moving or turning?

Problem: Path Following / Line Detection

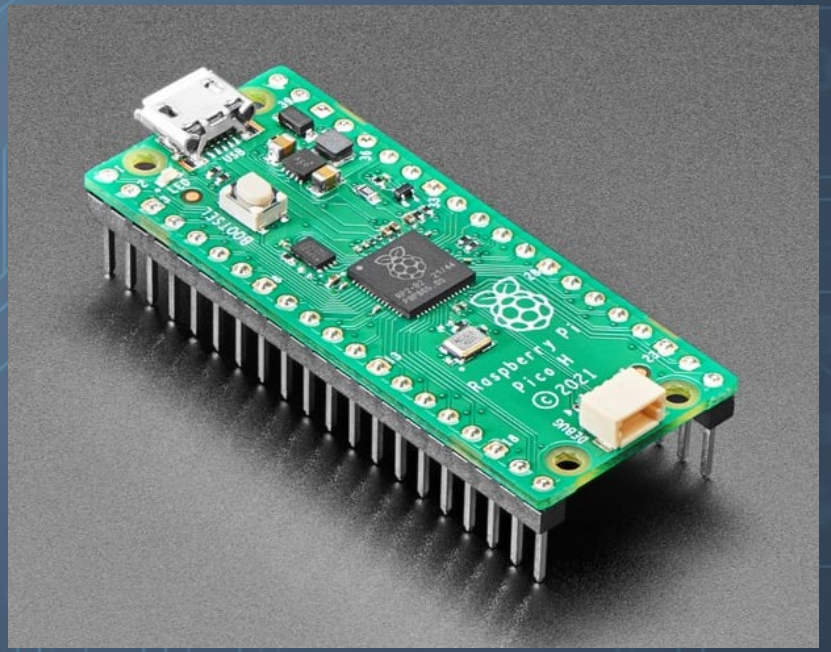
- The robot often has to stay on a line or a marked path to complete a course or challenge on its own.
- How does the robot detect and follow lines or edges on the ground?
- How does it know when to turn slightly or make sharp turns?

Problem: Combining Everything

- A successful robot uses all these abilities together to solve real-world tasks.
- How can we connect all the sensors and motors?
- How does the robot make decisions using information from different components?
- What acts as the robot’s “brain” to control everything at once?

Solution: Combining Everything

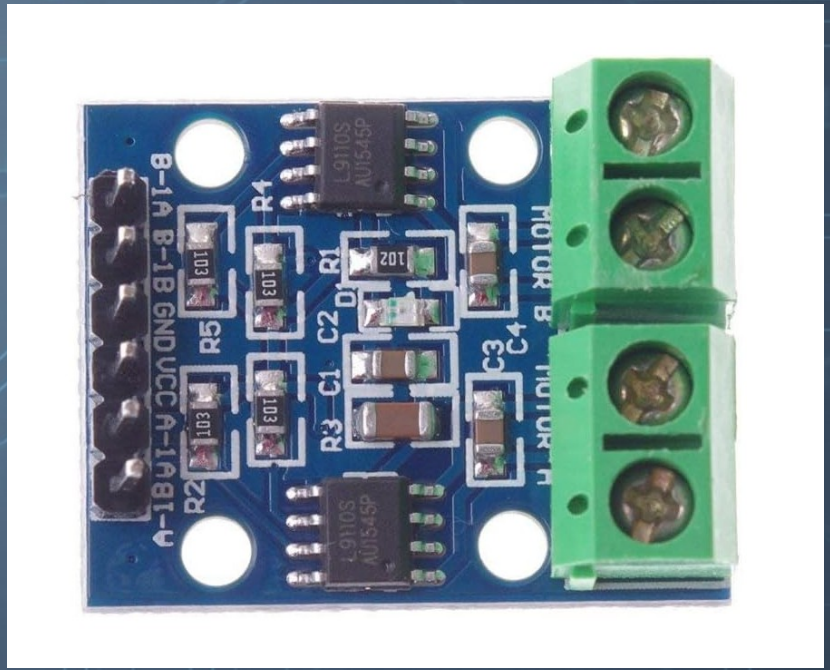
- The brain for these robots will be a microcontroller called a Raspberry Pi Pico
- A microcontroller is basically a tiny computer with hardware pins for controlling external devices.
- We will use a version of Python called MicroPython to program this.



Raspberry Pi Pico

Solution: Movement and Steering

- This motor driver takes input from the Pico.
- The pins on the Pico cannot provide enough power to drive a motor. It would let out the “magic smoke”.
- The motor driver takes input from the Pico and provides enough power and current to drive the motors.
- It handles direction changes, turning, speed, everything.



L9110S Motor Driver

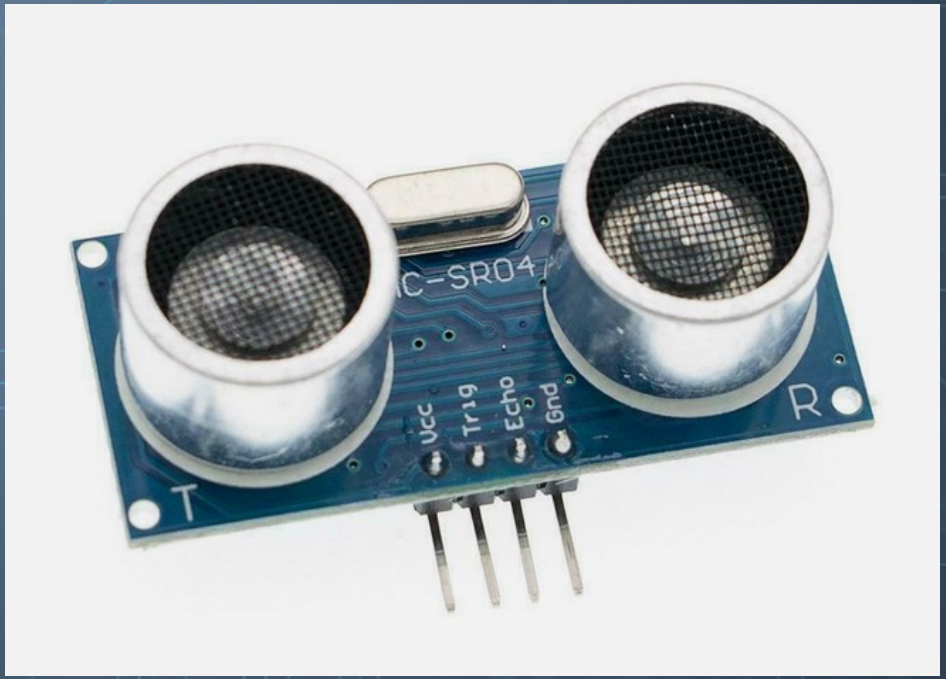
Solution: Movement and Steering

- These are the drive motors we will be using.
- They are 6VDC motors.

Drive Motors and Wheels

Solution: Obstacle Avoidance

- The HC-SR04 is an ultrasonic sensor.
- It works like RADAR, except using sound instead of radio waves.
- It sends out a pulse of sound, waits for the echo, then uses the time delay and the known speed of sound to calculate the distance.



HC-SR04 Ultrasonic Sensor

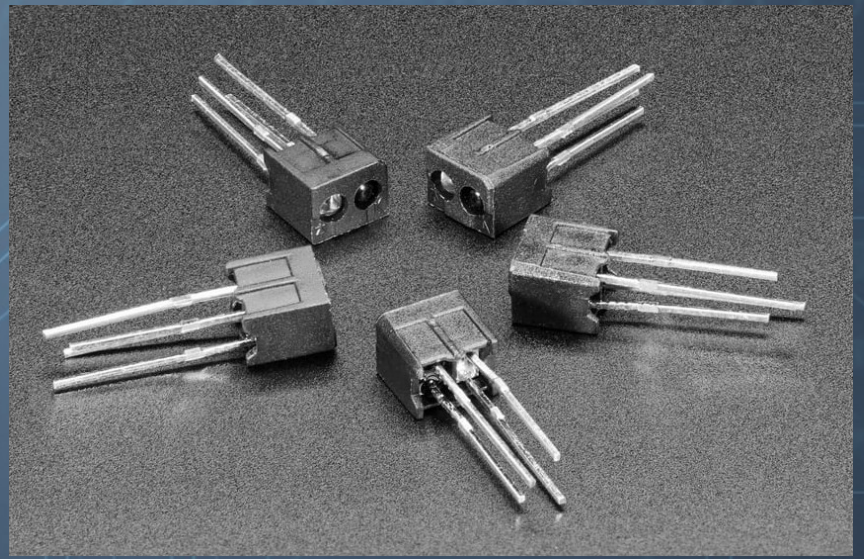
Solution: Direction Finding and Navigation

- This is an IMU or Inertial Measurement Unit.
- It can tell how fast something is accelerating or rotating.
- With a bit of math this can tell us how far the robot has turned or make sure that it is driving straight.

MPU6050 IMU

Solution: Path Following / Line Detection

- These sensors have two parts: an infrared LED and an infrared light detector.
- The lines are made of more reflective material so when the return level goes up, the robot knows it is over a line.



ITR20001 Infrared Sensor

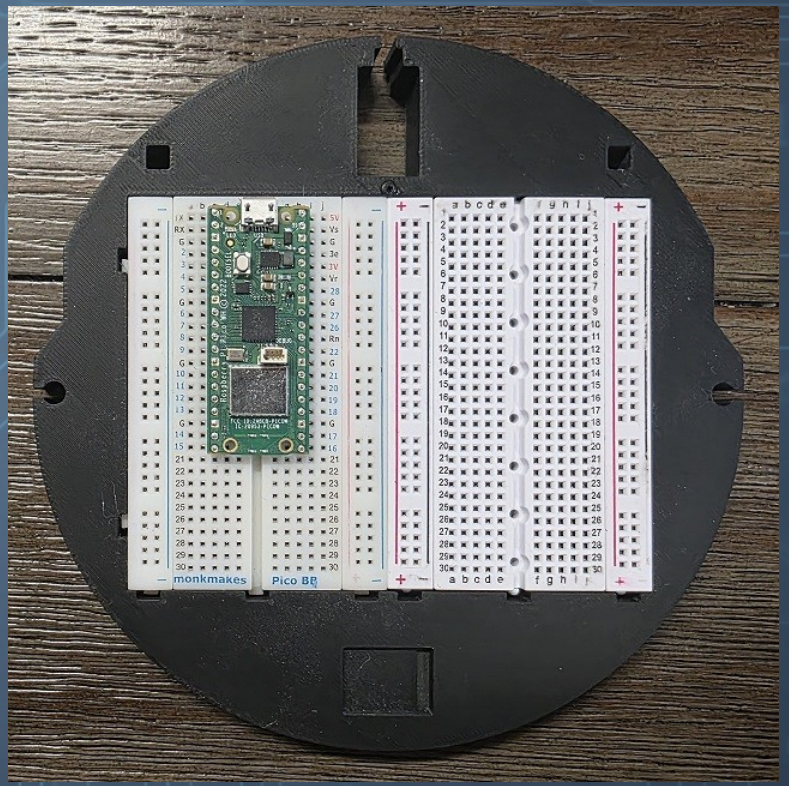
Solution: Servo Motors

- Most DC Motors simply turn when you apply voltage.
- Servo motors have a signal line which lets you turn them to a specific angle.
- Servos are used for moving sensors, manipulators, arms, and more.

9g Servo Motors

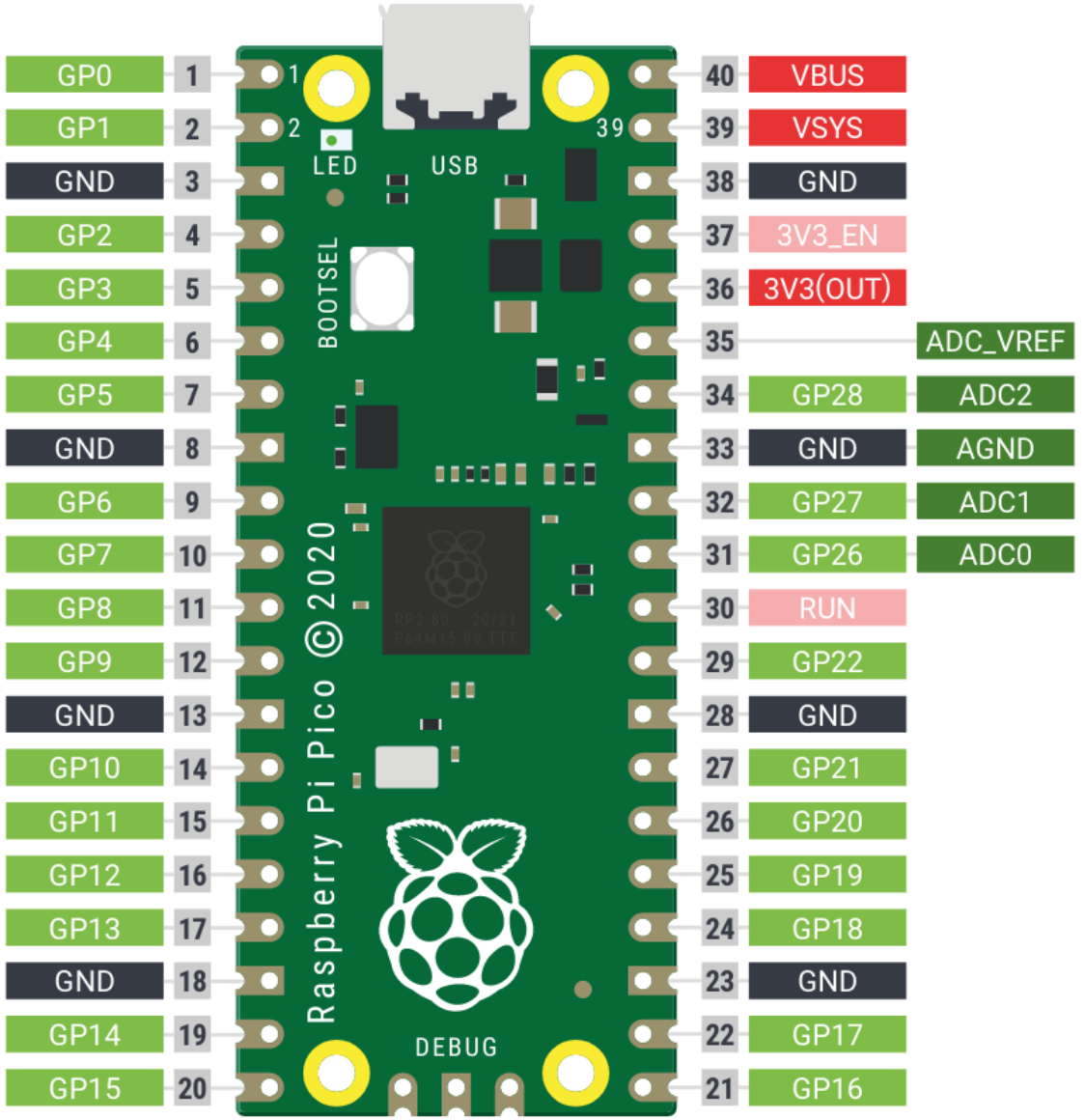
Combining Everything

- We will start with a blank base like this; only breadboards with a Pi Pico.
- As we progress we will add more hardware and sensors to build out a working robot.

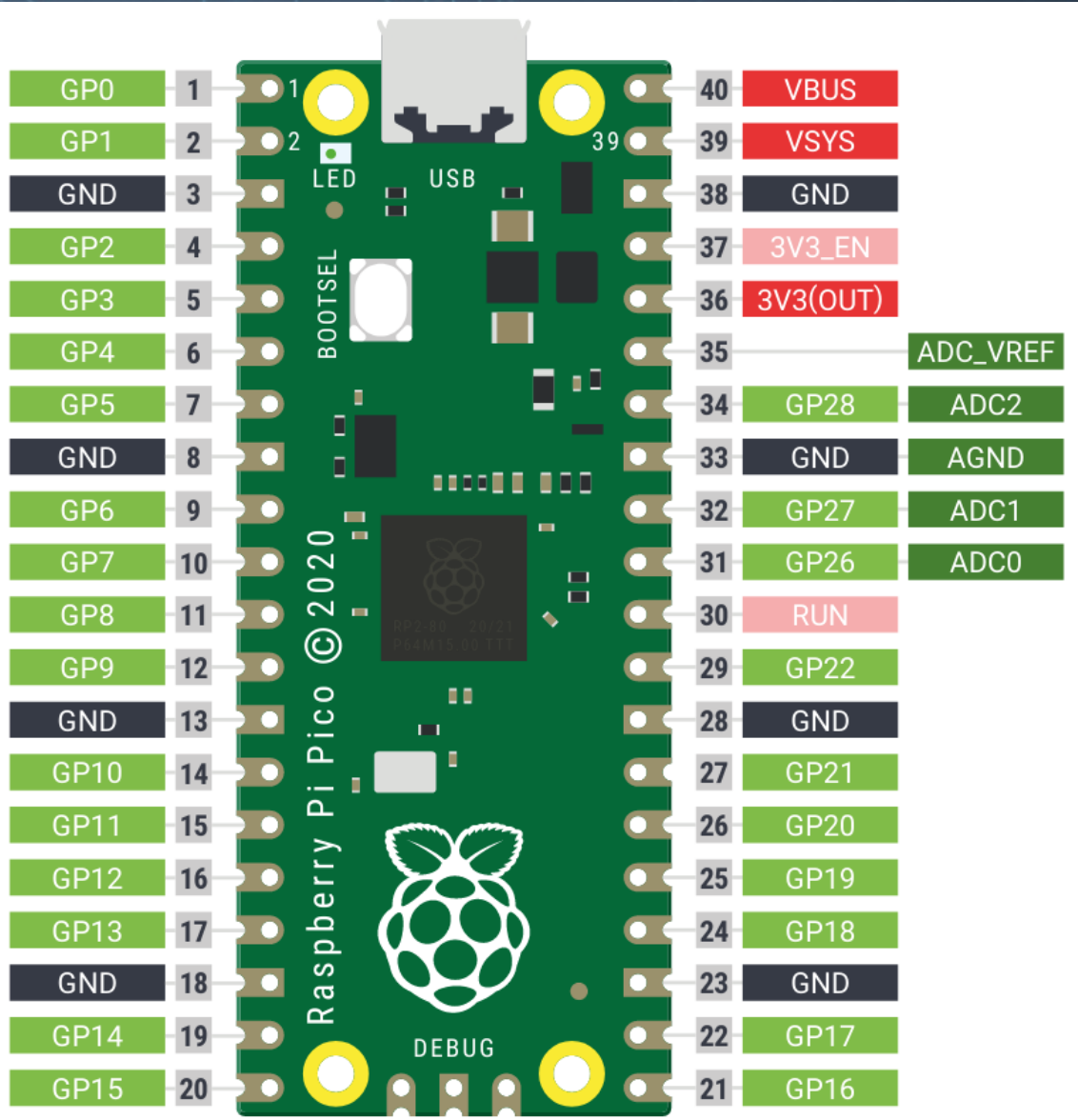


Blank Robot Base

- Power gets fed into VSYS (+) and GND (-). We will be supplying 5V in, and the Pico converts it to the 3.3V it needs.
- All GND pins are connected, it doesn't matter which one you use.
- The GP* pins are Input/Output (IO) pins. These are the pins that will be connected to devices to control them or read sensors.



- At the top is a micro USB connector. This is what you will connect to your laptop to program the Pico. It can also provide power to the Pico when the robot isn't driving around.



- The goal is to create and program robots which can autonomously navigate a course or complete a task, instead of being controlled remotely.
- You will need your laptops for the next class. We will be programming the Pi Picos and will start learning to interface with hardware.
- Please if at all possible find a Windows, Linux, or OSX laptop. Even if your Chromebook can be made to work, it will still be more limited than a full laptop.

Introduction to Digital Logic

- Remember the boolean data type in python?
- It was only had two possible states:
- True, also known as 1, HIGH, or ON
- False, also known as 0, LOW, or OFF
- The digital IO pins on the Pico work the same way.
- They are either ON or OFF
- This is often referred to as HIGH or LOW

Introduction to Digital Logic

- Remember the boolean data type in python?
- It was only had two possible states:
- True, also known as 1, HIGH, or ON
- False, also known as 0, LOW, or OFF
- The digital IO pins on the Pico work the same way.
- They are either ON or OFF
- This is often referred to as HIGH or LOW