GOPIGO3
SENSORS
ADDITIONAL ACTUATION

• GoPiGo3s offer many ways to alter the state of the robot
  • Wheels
  • LEDs
  • Servo motor

• Simply need to call the right method.
  • Together let's look up different methods for moving the robot.
USING THE SERVO

- The servo turns to a particular angle in the range $0^\circ, 180^\circ$
  - Note it cannot quite reach that far, more like $(10^\circ, 170^\circ)$
- Remember to initialize with `init_servo()`
- Set the position of the servo with `rotate_servo(x)`, where $x \in (10^\circ, 170^\circ)$
- Return the servo to its middle position with `reset_servo()`

**Servo.py**

```python
1. from easygopigo3
   import EasyGoPiGo3
2. robot = EasyGoPiGo3()
3. servo = robot.init_servo()
4.
5. servo.rotate_servo(45)
6.
7. servo.reset_servo()
```
SENSORS

• Sensors gather input data from the physical world
  • Images
  • Distances
  • Temperature
  • Light values
  • Etc.
GOPIGO3 SENSOR SETUP

• Light/Color Sensor
• Line Sensor
• Distance Sensor
• Inertial Measurement Unit (IMU)
USING SENSORS/ACTUATORS

• In the beginning, we will use the "EasyGoPiGo" versions of things. Eventually, around halfway through the semester, we will start using the richer, advanced version "GoPiGo"

• Same it true of sensors

• When using, we need to specify the port that each sensor is connected to. This will always be the same for our purposes.
DISTANCE SENSOR

- The distance sensor determines the distance to whatever is immediately in front of it.
- The sensor has a range of 2.3 meters and reports the distance in millimeters.

**Distance.py**
```
1. from easygopigo3 import EasyGoPiGo3
2. robot = EasyGoPiGo3()
3. distance_sensor = robot.init_distance_sensor()
4. 
5. distance = distance_sensor.read_mm()
```
LIGHT AND COLOR SENSOR

• The light and color sensor simply allows reading of a light intensity value
• The light value is broken down into RGBA components
• This sensor also has an LED that can be turned on and off

LightColor.py
1. from di_sensors.light_color_sensor import LightColorSensor
2. light_color_sensor = LightColorSensor(bus="GPG3_AD1")
3. light_color_sensor.set_led(True)
4. r, g, b, a = light_color_sensor.get_raw_colors()
5. light_color_sensor.set_led(False)
LINE FOLLOWER SENSOR

• The line sensor detects whether the robot sits on top of a black line or not
  • Essentially reads the color of the floor
• Six different values are read by the sensor

LineFollower.py
1. from easygopigo3 import EasyGoPiGo3
2. robot = EasyGoPiGo3()
3. Line_follower= robot.init_line_follower()
4. 
5. ll, l, lc, rc, r, rr = line_follower.read()
**INERTIAL MEASUREMENT UNIT**

- Inertial Measurement Units (IMUs) measure orientation, velocities, and forces about the state of the sensor
  - Also can measure temperature
- We will always have to calibrate the sensor each time we use it (more on this later)

*IMU.py*

1. from di_sensors.inertial_measurement_unit
   import InertialMeasurementUnit
2. imu = InertialMeasurementUnit(bus="GPG3_AD2")
3. 
4. a, b, g = imu.read_euler()
5. t = imu.read_temperature()
EXERCISE

• Write a program that determines and outputs the distance in each "cardinal" direction
  • Let "north" be where the robot originally faces.

• After, work on program 1