

Python Lab 5

BAT-212: BAT Logic and Programming



This material is based upon work supported by the National Science Foundation Advanced Technical Education grant program, A New Technician Training Program for Advanced Building Technologies, DUE-2000190.

The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Python Lab 5

OBJECTIVES

Upon completion of this activity the student will be able to:

- Identify elements of the class structure.
- Perform basic interactions with the REPL function of the Mu platform.
- Describe and modify servo motor control.

PARTS AND EQUIPMENT

- Circuit Playground Express board.
- Computer
- Mu IDE
- microServo robot (3 servos total, SG90)
- alligator clip leads
- solid core hookup wire.

REFERENCES

The following are immediately relevant to the lab, but some may be at a higher level than the student wants read. These were used in developing this lab:

- Using REPL: <https://learn.adafruit.com/welcome-to-circuitpython/the-repl#interacting-with-the-repl-2977486-14>
- Circuit Playground Express Pinout: https://learn.adafruit.com/adafruit-circuit-playground-express/pinouts?gclid=EAIAIQobChMIxKvm8_Xg-wIVk0lyCh0zPQ0WEAAYASAAEgLR_D_BwE
- Example servo code: <https://davebsoft.com/programming-for-kids/classes/cpx/>
- Adafruit example servo: <https://learn.adafruit.com/adafruit-circuit-playground-express/circuitpython-servo>
- More about Adafruit example servo: https://docs.circuitpython.org/projects/motor/en/latest/modules/adafruit_motor/servo.html#Servo
- Another example servo: <https://www.youtube.com/watch?v=ynlGiPZk5VM>
- Tutorial on classes: https://www.w3schools.com/python/python_classes.asp

The following are general references:

- <https://greenteapress.com/thinkpython/html/thinkpython002.html#toc5>
- <https://www.geeksforgeeks.org/python-programming-language/?ref=lbp>
- <https://docs.python.org/3/tutorial/>

Going deep references:

- What is '@'? Decorators: <https://builtin.com/software-engineering-perspectives/python-symbol>

BACKGROUND

PROCEDURES

Part 1:

- Go to https://www.w3schools.com/python/python_classes.asp. Read and do each “Try it for yourself”.
- Reference “Using REPL” listed above. Connect CPX board and open Mu editor. Open serial monitor. Enter CTRL^D in the serial monitor. Type `help("modules")` into the REPL next to the prompt, and press enter. Verify that `pulseio` and `pwmio` are listed. Type `import pulseio`. Type `dir(pulseio)`. Now import `pwmio` and get the directory result. Compare the 2 outputs. We will be using `pwmio`.

Part 2:

- Copy `Lab5Servo.py` into Mu editor. This program has different modes of operation; for this lab we will only use mode 1. Look at the class definition for `Servo`. Look at the PINS array defining which analog pins are used.
- Connect one of the servo motors to the CPX: yellow to analog signal (figure out from the code which pin this is; look at the PINS array), red to Vout, brown to GND. Open serial monitor to verify values sent to servo. Verify servo operation.
- Adjust the range of the for loop so that the servo is traveling the desired range of motion for that part of the robot. Make note of this range.
- The PINS array identifies two more pins. Test the code for the other two motors and pins. This can be done simply by changing the pin sent for the `Servo` class. Test and make note of the range for each motor.
- Create a servo object for each motor. This means that you have a unique name for each object. For example, `servo1 = Servo(1)`. Copy the for loop for each object and set the ranges you have determined. Verify operation.

Part 3:

This program is a good example of a time when using functions can make writing the program easier. Each motor controls a different motion on the robot: open/close claw; move arm up/down; rotate arm right/left. The original code stepped the motor in one direction, then when the `while True:` loop repeated, reset to the first position. We want to write code that will step each motor in each direction, so that there are no swift resets.

- Write 6 functions, one for each direction of each motion of the robot. Give the functions meaningful names, such as *openClaw()*: Remember how for loops work – to reverse the direction of the motion, the greater value goes first in the range, the lesser second, and the increment is negative. Also remember that the second value is not actually executed, so small adjustments in the values are needed to accurately reverse the motion. Test operation of each function individually, and then the functions for each motor pairwise.
- Use the functions to program the robot to pick up a ball from one leg and place it on the other, then pick the ball up again and return it. **Demonstrate** operation to the instructor. **Comment and submit** this code.