Lab 2

Robot Arm

In this lab we will make the robot arm move through a predefined sequence of movements. Utilizing the work you already did in homework, you will make the arm move.

Task 1: Calibrate

The robot arm uses toy RC servos[1] to move. These servos are commanded by a pulse width modulated signal[2] (PWM) to set their position. Unfortunately, these toy servos are produced for their low price and not their performance. Therefore, you must determine the correct PWM signal to get your servo to move correctly . . . every servo is different.
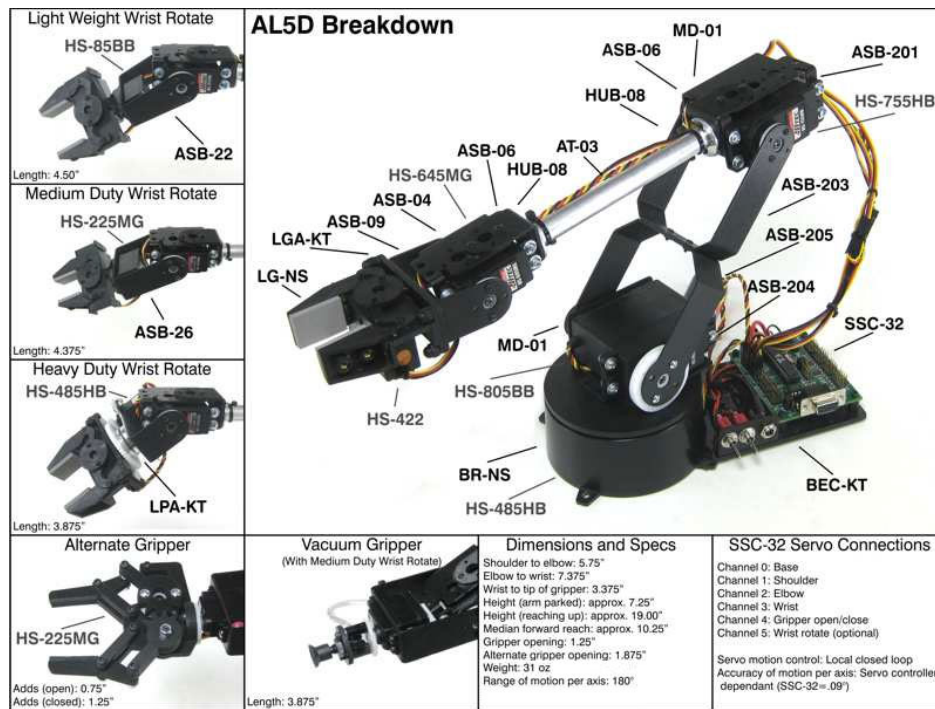

Figure 1: AL5D Robot Arm

The Lynx Motino AL5D has 5 servo motors that are able to turn between 0 and 180 degrees. The nominal PWM signals are shown below:

---

[1]https://en.wikipedia.org/wiki/Servo_(radio_control)
[2]https://en.wikipedia.org/wiki/Pulse-width_modulation

Angle	PWM	Gripper	PWM
0	800	Open	800
180	2300	Closed	2300

```python
#!/usr/bin/env python
from __future__ import print_function, division
import pyserial
import time

# open serial port
ser = pyserial.Serial('COM3', 115200)

def angle2pwm(angle):
    # your code here

def move_servo(servo, angle):
    # send a command to a single servo
    # example:
    #     servo 1  angle 0
    #     send '#1 P800 T2000\r'
    pwm = angle2pwm(angle)
    cmd = '#{} P{}\r'.format(int(servo), int(pwm))
    ser.write(cmd)

if __name__ == "__main__":
    # your code here
```

Task 2: Arm Movement

Once you have figured out the best pwm settings for your robot arm, now use your code to move the arm through a sequence of positions.

Step	Position	Orientation	Gripper
1	(10.75, 0, 5.75)	0	open
2	(9.5, 0, 4)	0	open
3	(9.5, 0, 0)	0	open
4	(9.5, 0, 0)	0	closed
5	(9.5, 0, 4)	0	closed
6	(7, -4, 0)	0	closed
7	(7, -4, 0)	0	open
8	(7, -4, 5)	0	open
9	(10.75, 0, 5.75)	0	open

```python
#!/usr/bin/env python
from __future__ import print_function, division
import pyserial
from math import atan2, acos, sqrt, pi, cos, sin
import time

# open serial port
ser = pyserial.Serial('COM3', 115200)
```

       Robots are cool!

```python
def inverse(x, y, z, orientation):
    # your code here

def angle2pwm(angle):
    # code

def move_arm(joint_angles):
    # your code here

if __name__ == "__main__":
    sequence = [
        [10.75, 0.0, 5.75, 0, 0], # x, y, z, orientation, gripper
        [...],
        ...
    ]

    for angles in sequence:
        move_arm(angles)
        time.sleep(5)
```

Robots are cool!