Mechatronics System Integration (MCTA3203)

Week **3b**: Serial and USB interfacing with microcontroller and computer based system (1): Sensors and actuators.

Controlling a servo motor using Python involves interfacing with an Arduino board that's connected to the servo. Below is a step-by-step guideline to conduct an experiment *where you transmit angle data from a Python script to an Arduino, which then actuates a servo to move to the specified angle*:

Materials Needed:

- Arduino board (e.g., Arduino Uno)
- Servo motor
- Jumper wires
- Potentiometer (for manual angle input)
- USB cable for Arduino
- Computer with Arduino IDE and Python installed

Hardware Setup:

- Connect the Servo's Signal Wire: Usually, you connect the servo's signal wire to a PWM-capable pin on the Arduino (e.g., digital pin 9).
- Power the servo using the Arduino's 5V and GND pins. Servos typically require a supply voltage of +5V. You can connect the servo's power wire (usually red) to the 5V output on the Arduino board.
- Connect the Servo's Ground Wire: Connect the servo's ground wire (usually brown) to one of the ground (GND) pins on the Arduino.

  An example of the hardware setup is shown in **Fig. 1**

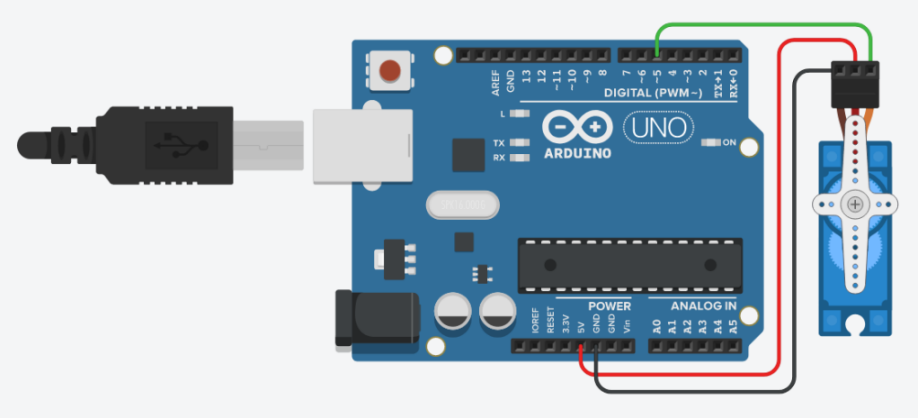

Fig. 1

Install Arduino Libraries:

- Open the Arduino IDE.
- Go to "Sketch" > "Include Library" > "Servo" to install the Servo library if it's not already installed.

Write the Arduino Code:

- Open a new sketch in the Arduino IDE.
- Write the Arduino code that reads angle data from the serial port and moves the servo accordingly.

```
#include <Servo.h>

Servo servo;
int angle = 90;

void setup() {
      servo.attach(9);        // Attach the servo to dig. pin 9
}

void loop() {
      servo.write(angle);     // Set to the desired angle
      delay(1000);            // Wait for 1 second
      angle = 180 - angle;    // Reverse the angle (e.g., 90 to
      90 degrees)
}
```

- Upload the code to your Arduino board.

Install Required Python Libraries:

- Open a terminal or command prompt.
- Install the *pyserial* library using pip, if you haven't it:

```
pip install pyserial
```

Write the Python Script:

- Create a new Python script using a code editor (e.g., Visual Studio Code, PyCharm, or a text editor).

```
import serial
import time

"""
      Define the serial port and baud rate
      (adjust the port as per your Arduino)
"""
ser = serial.Serial('COM3', 9600)

try:
    while True:
        angle = input("Enter servo angle (0-180 degrees): ")
        if angle.lower() == 'q':
```

```
            break
        angle = int(angle)

        if 0 <= angle <= 180:
            # Send the servo's angle to the Arduino
            ser.write(str(angle).encode())
        else:
            print("Angle must be between 0 and 180 degrees.")

except KeyboardInterrupt:
    pass          # Handle keyboard interrupt

finally:
    ser.close()   # Close the serial connection

print("Serial connection closed.")
```

Run the Python Script:

- Save the above Python script with a *.py* extension.
- Run the Python script, which will prompt you to enter the servo angle. You can enter an angle between 0 and 180 degrees to control the servo.

Experiment with Angle Input:

- Input an angle between 0 and 180 degrees and press Enter. The Python script will send the angle to the Arduino over the serial port.
- The Arduino will move the servo to the specified angle, and the script will display the angle set by the servo.
- You can exit the Python script by entering 'q' and observing that the serial connection is closed.
- You can input multiple angles to see the servo move accordingly.

Exit the Python Script:

- When you're done with the experiment, enter 'q' to exit the Python script.

Questions:

Enhance your Arduino and Python code to incorporate a potentiometer for real-time adjustments of the servo motor's angle. Ensure that, in the updated Arduino code, you have the ability to halt its execution by pressing a designated key on your computer's keyboard. Following the modification, restart the Python script to receive and display servo position data from the Arduino over the serial connection. While experimenting with the potentiometer, observe the corresponding changes in the servo motor's position.