`

الجامعة الإسلامية العالمية ماليزيا

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

يونيبرسيتي إسلام انتارابغسا مليسيا

Garden of Knowledge and Virtue

## MCTA 3203 SYSTEM INTEGRATION LAB

# WEEK 3:
# PARALLEL, SERIAL, AND USB INTERFACING

SECTION : 1
GROUP NUMBER : 7 (G)

LECTURER'S NAME : DR ZULKIFLI BIN ZAINAL ABIDIN

GROUP MEMBERS :

| NO : | | COURSE | MATRIC |
|---|---|---|---|
| 1. | MUHAMMAD NAQIB AIMAN BIN YUSNI | BMCT | 2117765 |
| 2. | MUHAMAD AZRI BIN ABDUL AZIZ | BMCT | 2113537 |
| 3. | MUHAMMAD HAFIDZUDDIN HANIF DANIAL BIN NORIZAL | BMCT | 2123651 |
| 4. | SHAREEN ARAWIE BIN HISHAM | BMCT | 2116943 |

# TABLE OF CONTENTS

# Abstract

## Part A:

The aim of this experiment is to create a serial communication link between an Arduino and Python so that data can be sent from the Arduino's potentiometer to a computer via a USB connection. This makes it easier to figure out how to set up serial connectivity and enables Python applications to use real-time data from the Arduino.

An Arduino board will be programmed to continuously transmit potentiometer data via serial in order to establish a serial communication. On the other hand, a Python script will create a serial connection and read the data sent by the Arduino using the **pyserial** library.

The main findings of this experiment show that a successful serial communication link between an Arduino and Python was established, allowing for the real-time transfer of potentiometer readings. The experiment proved that there is potential for dependable communication between Python and Arduino, enabling other uses of the data, such as charting.

To sum up, this experiment shows how to successfully create a serial communication link between an Arduino and Python to exchange data, in this case, potentiometer readings. This communication configuration works well for real-time data usage in Python programs when both ends are connected. The results of this study establish the basis for future applications and projects by providing a fundamental understanding of serial communication between microcontrollers and computers.

## Part B:

This lab report investigates the connection between a Python software for data input and an Arduino microcontroller. The objective of this project was to create a two-way communication channel between the Arduino and Python software, enabling the sharing of data. This interactivity was accomplished by using the serial communication protocol.

The setup involved connecting an Arduino board to a computer via USB and developing a Python script to communicate with it using the Pyserial library. Data transmission was initiated from both ends, enabling data to be sent from the Arduino to the Python program and vice versa.

Overall, this experiment confirmed that an Arduino microcontroller and Python could successfully establish connection, showing the importance of serial communication in crossing the gap between hardware and software applications. This useful implementation gives a flexible and effective means to enter and output data between the two platforms, with great potential for a variety of applications such as data logging, real-time monitoring, and Internet of Things projects.

## Introduction

The objective of this experiment is to create a serial communication link between an Arduino and Python so that data can be sent from the arduino potentiometer to a computer via USB connection.

An Arduino microcontroller is used to read data from sensors and to control devices. An Arduino would be applied to communicate with these electronics and transmit data to the computer because a computer cannot directly access these sensors and hardware. Additionally, Arduino boards are much less expensive than the standard computer, so replacing one due to a malfunctioning actuator or motor is not as expensive as replacing a computer.

Python is an universal programming language that's commonly used for data science applications. A Python script can read from and write to a serial port via the pyserial package. Thus, a Python script can be used to establish connection between a computer and an Arduino. Furthermore, complicated calculations can be performed with access to more computational power because of Python's versatility in manipulating data.

Part B:

This experiment aims to demonstrate hardware and software integration. In this instance, Python is the software and the Arduino is the hardware. It shows how two different platforms can cooperate to control a physical object through communication. Preparing engineers for understanding the fundamentals of software-based hardware control can be beneficial.

This experiment also demonstrates the concept of serial communication, that is sending and receiving data between two platforms between devices like an Arduino board and a

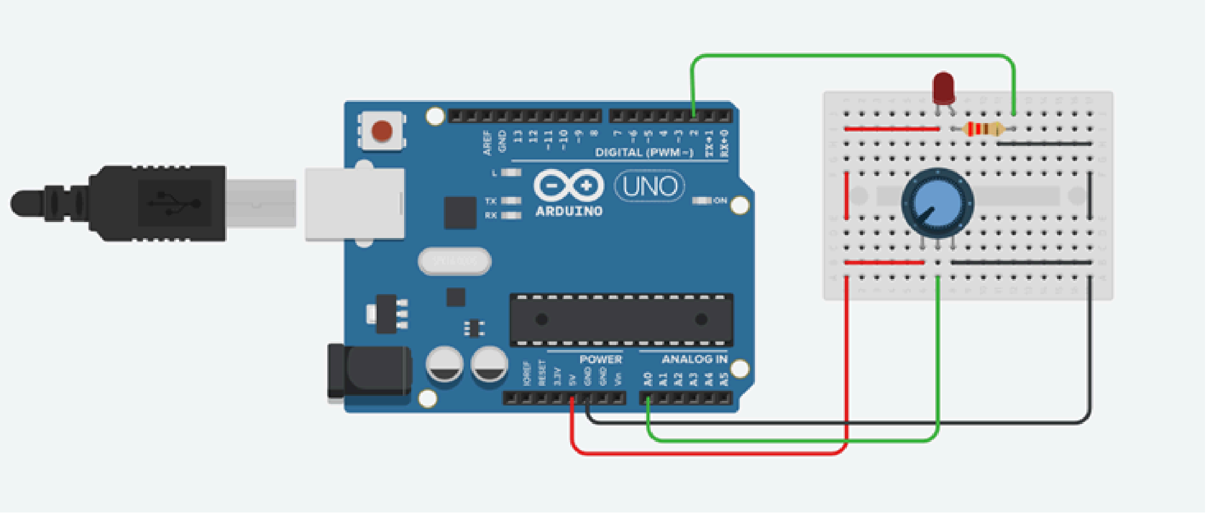
Python script implementing a serial connection. Understanding the functions of servo motors and the signals and orders needed to operate devices can also be beneficial.

## Material and Equipment

- Arduino Board
- Potentiometer
- Jumper Wires
- LED
- 220 resistor
- Breadboard

## Circuit & Experimental Setup

Experimental Setup for 3a:



1. Connect one leg of the potentiometer to 5V on the Arduino.

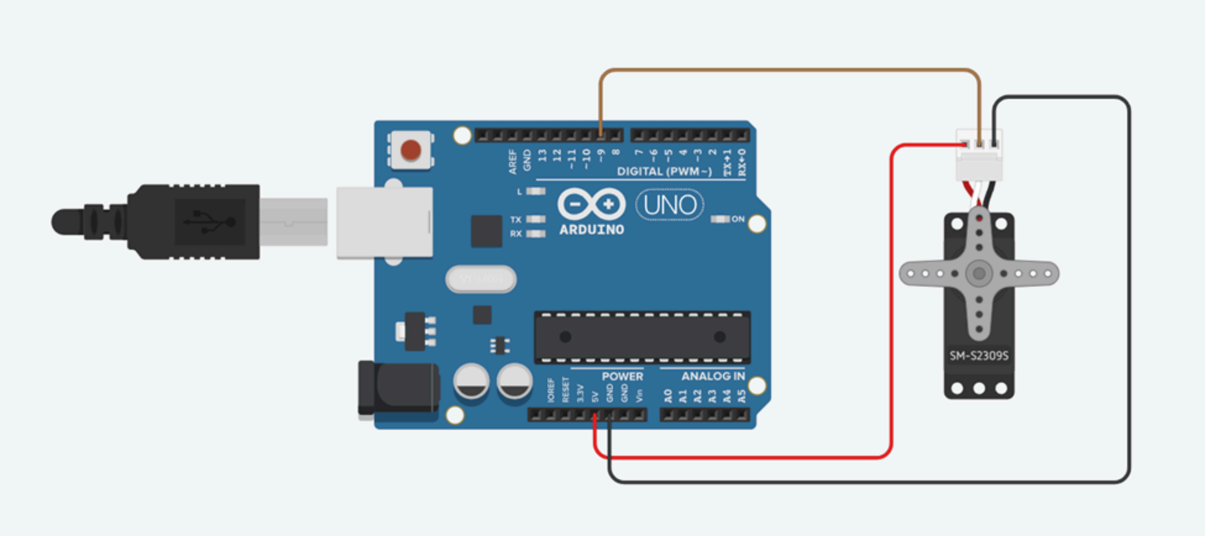
2. Connect the other leg of the potentiometer to GND on the Arduino.

3. Connect the middle leg (wiper) of the potentiometer to an analog input pin on the Arduino, such as A0.

4. Connect one leg of the LED to GND.

5. Connect the other leg to digital pin D7 using a resistor.

Experimental Setup for 3b:



1. Keep the setup from part 1 of the experiment.

2. Connect the Servo's Signal Wire to a digital pin (e.g., D9).
3. Connect the servo's power wire (usually red) to the 5V output on the Arduino board.
4. Connect the servo's ground wire (usually brown) to one of the ground (GND) pins on the Arduino.

## Coding

### Experiment 3a:

Arduino

```
int sensorPin = A0; // the potentiometer is connected to analog pin 0
int ledPin = 3; // the LED is connected to digital pin 13
int sensorValue; // an integer variable to store the potentiometer reading
void setup() { // this function runs once when the sketch starts up
  pinMode (ledPin, OUTPUT);
  // initialize serial communication :
  Serial.begin(9600);
}

void loop() { // this loop runs repeatedly after setup() finishes
  sensorValue = analogRead(sensorPin); // read the sensor
  Serial.println(sensorValue); // output reading to the serial line
  if (sensorValue < 500){
    digitalWrite(ledPin , LOW );} // turn the LED off
  else {
    digitalWrite(ledPin , HIGH );} // keep the LED on
  delay (100); // Pause in milliseconds before next reading
}
```

Python

```
Import serial
ser = serial.Serial('COM7', 9600)
Try:
While True:
```

```
Pot_value = ser.readline().decode().strip()
print("Potentiometer Value:", pot_value)
except KeyboardInterrupt:
 ser.close()
print("Serial connection closed.")
```

## Experiment 3b:

### Arduino

```
#include <Servo.h>
Servo servoMotor;
void setup() {
Serial.begin(9600);
servoMotor.attach(9);
}
void loop() {
if (Serial.available() > 0) {
int angle = Serial.parseInt();
servoMotor.write(angle); // Move servo to the specified angle
}
}
```

### Python

```
import serial
import time
import sys
ser = serial.Serial('COM7', 9600)
def move_servo(angle):
ser.write(str(angle).encode()) # Send angle data to Arduino
time.sleep(0.1) # Delay to allow time for Arduino to process
```

```
if __name__ == "__main__":
print("Enter an angle (0-180) or 'q' to quit.")
while True:
try:
user_input = input("Angle or 'q' to quit: ")
if user_input.lower() == 'q':
print("Exiting program.")
sys.exit()
angle = int(user_input)
if 0 <= angle <= 180:
move_servo(angle)
else:
print("Angle must be between 0 and 180.")
except ValueError:
print("Invalid input. Please enter an integer or 'q' to quit.")
```

## Results

Part A:

We can see when the potentiometer has been rotated the output can be read from our PC by using python.We can control the value of output by using the potentiometer by turning it up or down.Therefore if the output value reaches 500 and above then the light will turn on , while the light will turn off if the value is lower than 500.

Part B:

In this lab, we can see how to control the servo by using python.We have set it for the servo to rotate 0 to 180 degree. By writing the value of degrees in the python , we can observe the servo rotated according to the value of degrees that we have written in the python. Other than that if we press ‘q’ in the input the coding will be reset.

## Discussion

Experiment 3 part A demonstrates the use of a potentiometer to regulate an output value that controls whether a light goes on or off. This is a typical illustration of digital output control with analogue sensor input. It demonstrates the fundamental idea of controlling an electrical signal using a variable resistor (the potentiometer) so that the computer can read it and decide whether to turn on or off the light by using Python. This configuration can serve as the basis for more intricate systems that employ analogue inputs to initiate a range of reactions or activities.

Experiment 3 part B illustrates the use of servo motors controlled using python. Servos are frequently used in automation and robotics to regulate the position of mechanical parts. The servo motor may be programmed to rotate to different locations using Python by providing it with certain angle values. As demonstrated in the video, the The control system is made more interactive by the inclusion of an input command ('q' to reset the coding), which enables dynamic changes or resets.

## Conclusion

In conclusion, this experiment was a success in demonstrating the integration of both potentiometer and servo motors using arduino and python. These experiments demonstrate the practical application of Python in interfacing with hardware components for control purposes. Part A showcases how a potentiometer can be used to control digital output, while Part B illustrates the control of a servo motor. Both experiments highlight the versatility and interactive capabilities of Python in engineering and automation. Overall, they provide valuable hands-on experience and lay the foundation for more advanced projects in robotics, electronics, and system control.

## Student's Declaration

Declaration:
We certify that this project/assignment is entirely our own work, except where we have given fully documented references to the work of others, and that the material in this assignment has not previously been submitted for assessment in any formal course of study.

azri

______________
Name: Muhamad Azri bin Abdul Aziz

hafidzuddin

______________
Name: Muhammad Hafidzuddin Hanif Danial bin Norizal

arawie

______________
Name: Shareen Arawie bin Hisham

naqib

______________
Name: Muhammad Naqib Aiman bin Yusni

Date: 18/3/2024