

Mechatronics System Integration (MCTA3203)

Week 4: Serial interfacing with microcontroller: Sensors and actuators (ver. 3.0).

Designing a Motion-Activated RFID System

Learning Objectives

By completing this lab, students will be able to:

- Interface an MPU6050 sensor and RFID module with an Arduino.
- Process real-time accelerometer data using Python.
- Detect simple motion patterns (e.g., circular motion).
- Integrate motion sensing and RFID authentication to control a servo-based access system.
- Communicate between Python and Arduino over serial.

Part 1: Motion Detection with MPU6050

Goal

Track hand motion in real-time using the MPU6050 and display the path dynamically in Python.

Hardware Required

- Arduino board
- MPU6050 sensor
- Jumper wires/breadboard
- USB cable
- Computer with Arduino IDE and Python installed

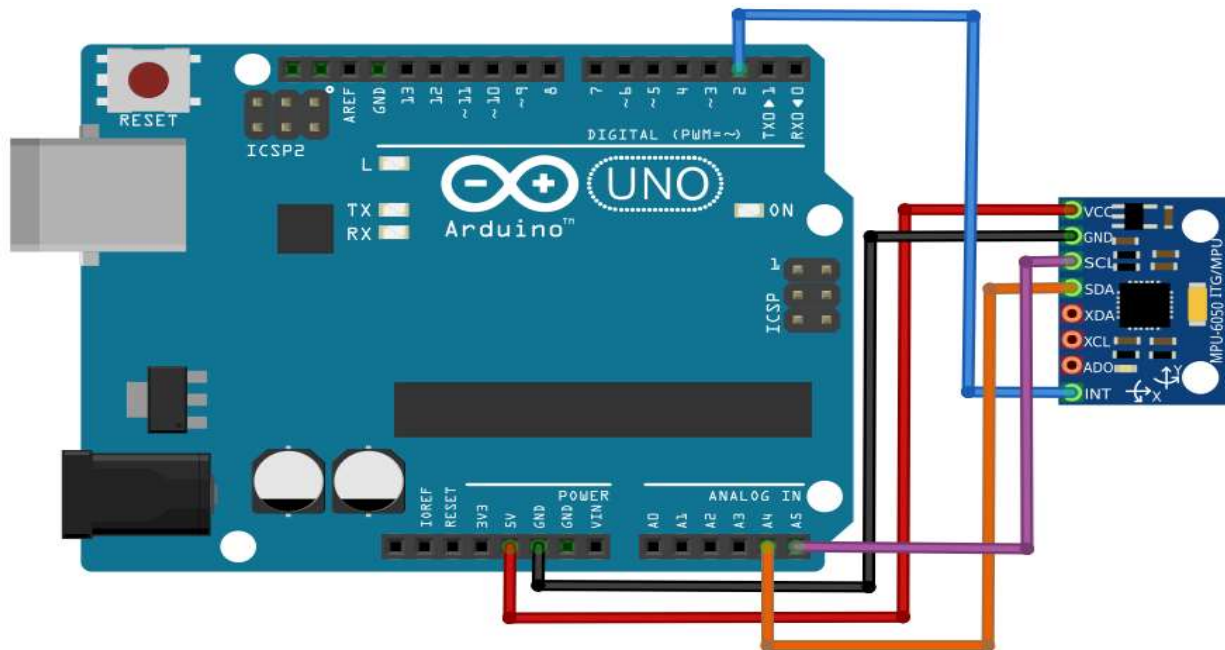


Fig. 1: Arduino-MPU6050 Connections

Connections

MPU6050 Pin	Arduino Uno
VCC	5V
GND	GND
SDA	A4
SCL	A5

Arduino Code to Read MPU6050 Data:

(This sends accelerometer and gyroscope data over serial at 9600 baud.)

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed!");
    while (1);
  }
}

void loop() {
  int ax, ay, az, gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  Serial.print(ax); Serial.print(",");
  Serial.print(ay); Serial.print(","); Serial.println(az);
  delay(100);
}
```

Task 1: Real-Time Motion Tracking Using MPU6050 Sensor

Your task:

- Capture real-time accelerometer X-Y data.
- Plot the motion dynamically using matplotlib.
- Detect circular motion gesture.

Starter Code (to be improved):

```
import serial
import matplotlib.pyplot as plt

ser = serial.Serial('COM4', 9600, timeout=1) # Change to match your system

x_vals, y_vals = [], []

plt.ion()
fig, ax = plt.subplots()
ax.set_xlim(-20000, 20000)
ax.set_ylim(-20000, 20000)
```

```

line, = ax.plot([], [], 'ro')

while True:
    try:
        data = ser.readline().decode().strip()
        if data:
            values = data.split(',')
            if len(values) == 6:
                ax_val = int(values[0])
                ay_val = int(values[1])
                x_vals.append(ax_val)
                y_vals.append(ay_val)
                line.set_data(x_vals, y_vals)
                plt.draw()
                plt.pause(0.05)
    except KeyboardInterrupt:
        break

```

Deliverables:

- A Python script that captures and visualizes the motion path in real time.
- Annotated code explaining improvements (what were made to the starter code?).
- Screenshots or screen recordings of your circular motion being plotted.

Part 2: RFID + Servo Access System

Goal

Control access to a system using RFID identification, validated by correct motion pattern (e.g., circular hand gesture), triggering a servo motor and LED feedback.

Hardware Required:

- Arduino board
- RFID reader (MFRC522) + RFID tags/cards
- Servo motor
- Red & Green LEDs + resistors
- MPU6050 sensor (from Part 1)
- Jumper wires and breadboard

Connections Summary

Device	Arduino Pin
RFID SS	D10
RFID RST	D8
Servo signal	D9
Green LED	D4
Red LED	D3
GND and 5V	Shared for all

System Behavior

Condition	Action
Valid RFID + correct motion	Servo unlocks, green LED on
Invalid RFID OR incorrect motion	Access denied, red LED on

Arduino Code for Servo Control:

```
#include <Servo.h>

Servo servo;
int servoPosition = 90;

void setup() {
  servo.attach(9);
  servo.write(servoPosition);
  pinMode(4, OUTPUT); // Green LED
  pinMode(3, OUTPUT); // Red LED
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    char command = Serial.read();
    if (command == 'A') {
      servo.write(180); // Unlock
      digitalWrite(4, HIGH);
      digitalWrite(3, LOW);
    } else if (command == 'D') {
      servo.write(90); // Lock
      digitalWrite(4, LOW);
      digitalWrite(3, HIGH);
    }
  }
}
```

Arduino Code for Reading RFID Tag:

Note: This project uses the MFRC522.h library to control the RFID reader. This library doesn't come installed in Arduino IDE by default, so you need to install it. Go to **Sketch > Include library > Manage libraries** and search for MFRC522 (see also [1]).

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 8
MFRC522 rfid(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
}
```

```

}

void loop() {
  if (!rfid.PICC_IsNewCardPresent()) return;
  if (!rfid.PICC_ReadCardSerial()) return;

  Serial.print("UID:");
  for (byte i = 0; i < rfid.uid.size; i++) {
    Serial.print(rfid.uid.uidByte[i], HEX);
  }
  Serial.println();
  rfid.PICC_HaltA();
}

```

Example of Python Script for Access Control (to be improved):

```

import serial
import time

SERIAL_PORT = 'COM4'
BAUD_RATE = 9600
authorized_cards = ["04AABBCCDD", "1234567890"]

def detect_circular_motion():
    # To do: implement real motion detection
    # You should reuse or enhance your motin tracking logic from Part 1
    print("Perform circular motion now...")
    time.sleep(3)
    # Return True if pattern is circular
    return True

def main():
    ser = serial.Serial(SERIAL_PORT, BAUD_RATE)
    try:
        while True:
            line = ser.readline().decode().strip()
            if line.startswith("UID:"):
                uid = line[4:].strip().upper()
                if uid in authorized_cards:
                    print("RFID authorized. Perform motion...")
                    if detect_circular_motion():
                        print("Motion verified. Access granted.")
                        ser.write(b'A')
                    else:
                        print("Motion invalid. Access denied.")
                        ser.write(b'D')
                else:
                    print("UID unauthorized. Access denied.")
                    ser.write(b'D')
            time.sleep(0.1)
    except KeyboardInterrupt:
        print("Program terminated.")
    finally:
        ser.close()

if __name__ == "__main__":
    main()

```

Task 2: Integrated Smart Access System

You must build a working system that:

1. Scans RFID tag
2. Checks if UID is authorized
3. Prompts user to perform a circular motion
4. Analyzes MPU6050 motion data (if motion detected: send 'A'; otherwise: send 'D')
5. Controls servo + LEDs based on result

Submission Requirements

1. Final Arduino sketch (.ino file)
2. Final Python scripts (.py files)
3. Brief report (PDF) including:
 - Description of your motion detection approach
 - List of authorized UIDs
 - Integration explanation (how parts work together)
 - Screenshot of working system

Troubleshooting Tips

Issue	Solution
Serial not working	Check port in Device Manager and ensure no other apps are using it
MPU6050 not detected	Use I2C scanner sketch to find address
Servo jittering	Use external 5V power for servo
Motion not detected accurately	Implement smoothing (moving average) or use gyro data

References

- [1] <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>
What is RFID? How It Works? Interface RC522 RFID Module with Arduino
- [2] <https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/>
Security Access using MFRC522 RFID Reader with Arduino
- [3] <https://randomnerdtutorials.com/arduino-time-attendance-system-with-rfid/>
Arduino Time Attendance System with RFID
- [4] <https://www.instructables.com/Arduino-MFRC522-RFID-READER/>
Arduino + MFRC522 RFID READER