

## Q1 :

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(9, 10); // CE, CSN

const byte address[6] = "00001";

void setup() {
    radio.begin();
    radio.openWritingPipe(address);
    radio.setPALevel(RF24_PA_MIN);
    radio.stopListening();
}

void loop() {
    int potValue = analogRead(A0); // Read potentiometer value
    int angle = map(potValue, 0, 1023, 0, 180); // Map to servo angle (0-180)

    radio.write(&angle, sizeof(angle)); // Send angle to receiver
    delay(15); // Small delay for stability
}

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>

RF24 radio(9, 10); // CE, CSN
Servo myServo;

const byte address[6] = "00001";

void setup() {
    myServo.attach(9); // Attach servo to pin 9
    radio.begin();
    radio.openReadingPipe(0, address);
    radio.setPALevel(RF24_PA_MIN);
    radio.startListening();
}

void loop() {
    if (radio.available()) {
        int angle;
        radio.read(&angle, sizeof(angle)); // Read the angle from transmitter
        myServo.write(angle); // Move servo to the received angle
    }
}
```

## Q2 :

```
#include <Wire.h>
#include <MPU6050.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

MPU6050 mpu;
RF24 radio(9, 10); // CE, CSN
```

```

const byte address[6] = "00001";

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

  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}

void loop() {
  int16_t ax, ay, az;
  int16_t gx, gy, gz;

  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // Read MPU6050 data

  // Map accelerometer data to control commands
  int forwardBackward = map(ay, -17000, 17000, 0, 255); // Forward/Backward
  int leftRight = map(ax, -17000, 17000, 0, 255); // Left/Right

  // Send data to receiver
  int data[2] = {forwardBackward, leftRight};
  radio.write(&data, sizeof(data));

  delay(50); // Small delay for stability
}

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(9, 10); // CE, CSN

const byte address[6] = "00001";

// Motor pins
const int IN1 = 2;
const int IN2 = 3;
const int IN3 = 4;
const int IN4 = 5;
const int ENA = 6; // PWM pin for speed control
const int ENB = 9; // PWM pin for speed control

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);

  radio.begin();

```

```

    radio.openReadingPipe(0, address);
    radio.setPALevel(RF24_PA_MIN);
    radio.startListening();
}

void loop() {
    if (radio.available()) {
        int data[2];
        radio.read(&data, sizeof(data)); // Read data from transmitter

        int forwardBackward = data[0]; // Forward/Backward control
        int leftRight = data[1]; // Left/Right control

        // Control motors based on MPU6050 data
        if (forwardBackward > 128) {
            // Move forward
            digitalWrite(IN1, HIGH);
            digitalWrite(IN2, LOW);
            digitalWrite(IN3, HIGH);
            digitalWrite(IN4, LOW);
            analogWrite(ENA, forwardBackward);
            analogWrite(ENB, forwardBackward);
        } else if (forwardBackward < 128) {
            // Move backward
            digitalWrite(IN1, LOW);
            digitalWrite(IN2, HIGH);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, HIGH);
            analogWrite(ENA, 255 - forwardBackward);
            analogWrite(ENB, 255 - forwardBackward);
        }

        if (leftRight > 128) {
            // Turn right
            digitalWrite(IN1, HIGH);
            digitalWrite(IN2, LOW);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, HIGH);
        } else if (leftRight < 128) {
            // Turn left
            digitalWrite(IN1, LOW);
            digitalWrite(IN2, HIGH);
            digitalWrite(IN3, HIGH);
            digitalWrite(IN4, LOW);
        }
    }
}

```