Project Title: Industrial Robotic Arm - 5 DOF

Team Members:

1. [Duaa Mohamed Ibrahim]
2. [Rowan Ibrahim Desouky]
3. [Asmaa Osama Eid]
4. [Rozan Mahmoud Ahmed]
5. [Zohir Raafat Hassan]
6. [Ziad Mohammed Fathy]
7. [Abdelrahman Said]
8. [Abdelrahman Ashraf Saad]

Supervisor:

Dr. [Mohammed Magdy]

Eng. [Soha Magdy]

GitHub Repository:

[Link to GitHub Repository: https://github.com/ziadmohamed0/RoboticProjects.git]

Hardware Connections and Configuration

Code 1: ESP8266 with MPU6050, PCA9685, and IR Sensor

MPU6050 (Accelerometer & Gyroscope Sensor):
VCC: 3.3V (from ESP8266)
GND: Common ground
SCL: D1 (GPIO5 on ESP8266)
SDA: D2 (GPIO4 on ESP8266)

PCA9685 (16-Channel Servo Driver):
VCC: 3.3V (from ESP8266)

GND: Common ground
SCL: D1 (GPIO5 on ESP8266)
SDA: D2 (GPIO4 on ESP8266)

Servo Motors (Channels):
Channel 0: Base motor
Channel 1: Shoulder motor
Channel 2: Elbow motor
Channel 3: Wrist motor
Channel 4: Gripper motor

IR Sensor:
VCC: 3.3V (from ESP8266)
GND: Common ground
Signal: D5 (GPIO14 on ESP8266)

Servo Motors Power:
Connect servo motor power lines to an external 5V source.
Ground the external power source with the ESP8266.

Functionality Overview:
The MPU6050 measures the arm's orientation in real-time.
Measurements are converted to servo motor positions via the PCA9685 for smooth arm movement.
The IR sensor provides end-effector (gripper) activation based on external input.
Real-time sensor data and movement parameters are displayed through the Serial Monitor.

Code 2: ESP-01 and ESP8266 for Wireless Communication

Connections:

MPU6050 to ESP-01:
VCC: 3.3V (from ESP-01)
GND: Common ground
SCL: GPIO2 (ESP-01)
SDA: GPIO0 (ESP-01)

ESP-01 Power Supply:
Ensure a stable 3.3V source.

ESP8266 to PCA9685 and IR Sensor:
PCA9685:
VCC: 3.3V (from ESP8266)
GND: Common ground

SCL: GPIO5 (D1 on ESP8266)
SDA: GPIO4 (D2 on ESP8266)

IR Sensor:
VCC: 3.3V (from ESP8266)
GND: Common ground
Signal: GPIO14 (D5 on ESP8266)

Explanation:
The ESP-01 reads data from the MPU6050 and sends it to the ESP8266 via Wi-Fi.
The ESP8266 receives and processes the data.
The PCA9685 controls the servos, and the IR sensor controls the end effector.

Code 3: ESP32 and ESP8266 with MPU6050 and PCA9685

ESP32 Hardware Connections:

MPU6050:
VCC: 3.3V
GND: Common ground
SCL: GPIO 22
SDA: GPIO 21

IR Sensor:
VCC: 3.3V
GND: Common ground
OUT: GPIO 34

ESP8266 Hardware Connections:

PCA9685:
VCC: 3.3V
GND: Common ground
SCL: GPIO5 (D1 on ESP8266)
SDA: GPIO4 (D2 on ESP8266)
V+: 5V (External power for servos)

Servo Motors:
Connect signal pins to PCA9685 channels (0–5).
Power and GND to external power supply.

Explanation:
ESP32:
Reads data from MPU6050 (accelerometer and gyroscope) and IR sensor.

Transmits data wirelessly to ESP8266 over Wi-Fi.

ESP8266:
Receives data from ESP32.
Parses the data to extract accelerometer, gyroscope, and IR values.
Maps accelerometer data to control base rotation and arm movement.
Uses IR sensor data to control the end effector's opening/closing.

Communication:
ESP32 sends data as a string (comma-separated values).
ESP8266 parses this string and maps the data to servo angles.

Code 4: STM32 MCU with MPU6050 and PCA9685

Hardware Description:

Microcontroller:
STM32 MCU configured with an I2C interface for communication.

Peripherals:
PCA9685 Servo Driver:
I2C address: 0x80.
Drives up to 16 servo motors (5 used in this configuration).

MPU6050 IMU Sensor:
I2C address: 0xD0.
Provides accelerometer and gyroscope data (Ax, Ay, Az, Gz).

Servos Controlled:
Channels:
Channel 0: Roll
Channel 1: Pitch
Channel 2: Yaw
Channel 3: Roll (duplicate for demonstration purposes)
Channel 4: Pitch (duplicate for demonstration purposes)

Functionality Overview:
The MPU6050 continuously measures orientation (roll, pitch, yaw).
These orientation values control the angle of the 5 servos through the PCA9685 driver.
Roll and pitch are applied to multiple servos for redundancy, while yaw controls a specific servo.