Proposal Submission for COMP423 Project

Dear professor Ali Haidar, Eng Hafez Khatib, Eng Hasan Fayad

I hope this message finds you well. My team and I have agreed on our project idea for the Computer Architecture project.

In this project we aim to solve the issue of doctors not being in an emergency room 24/7 which may lead to severe damage to the victim addresses, given that If the doctor is not available in the hospital, the patient will have to wait for the doctor to arrive at the hospital.

Our project will aim to prepare a Wi-Fi controlled hand that the doctor can control and operate on the patient from home.

We already discussed it with prof. Ali Haidar and Eng Hafez Khatib, and got the green light.

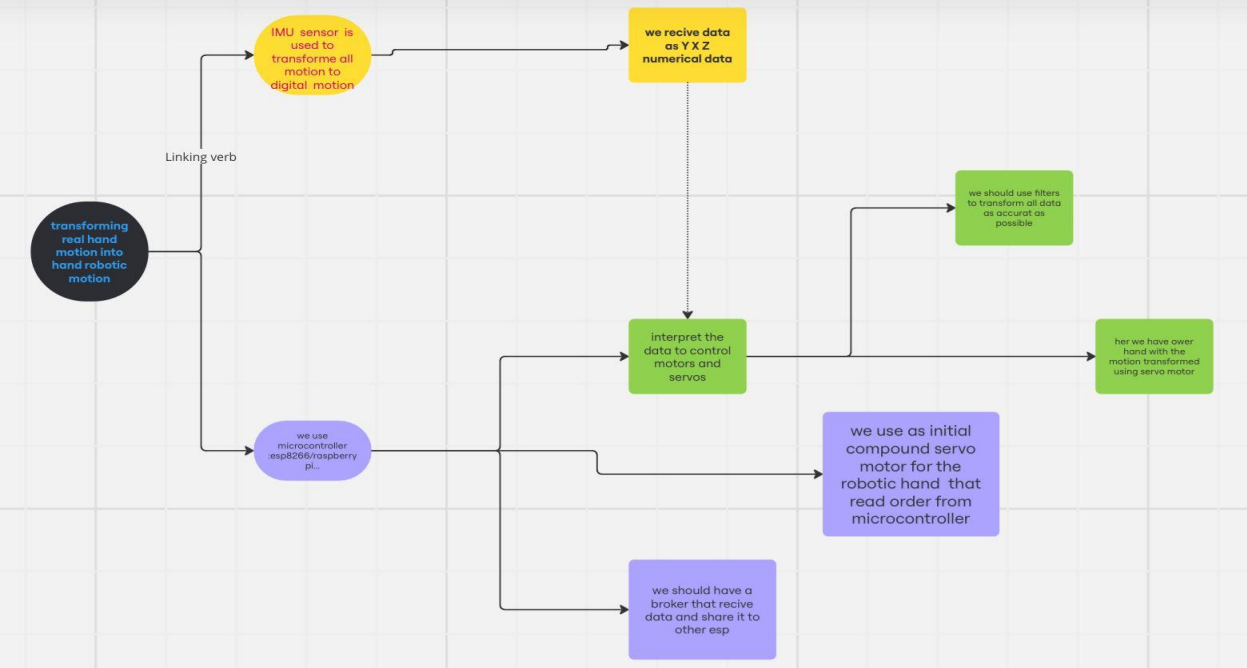
We realize this is a very ambient project, but hopefully with our hard work and Your expertise will help us overcome the obstacles and achieve a project close to perfection.

Our team members:

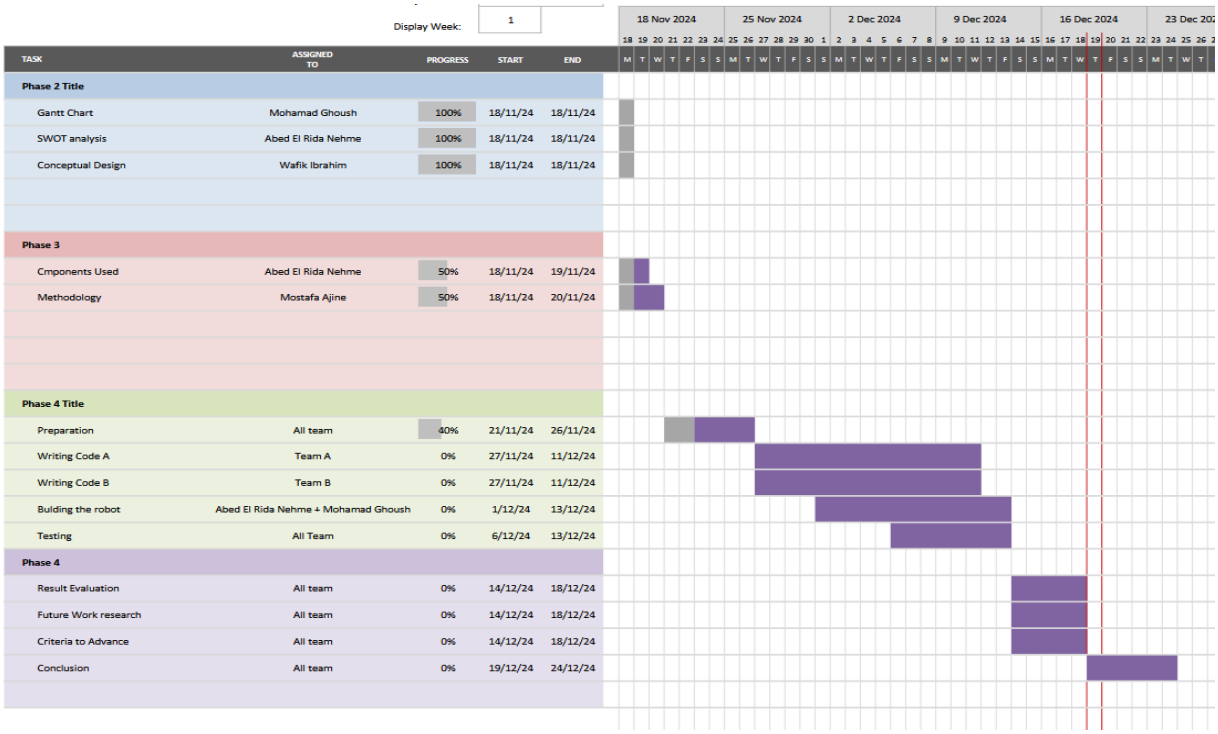
- Mohamad Ghoush 202300735
- Abed al Rida Nehme 202303149
- Mostafa Ajineh 202304458
- Wafik Ibrahim 202301238

Phase 2:

Conceptual Design:



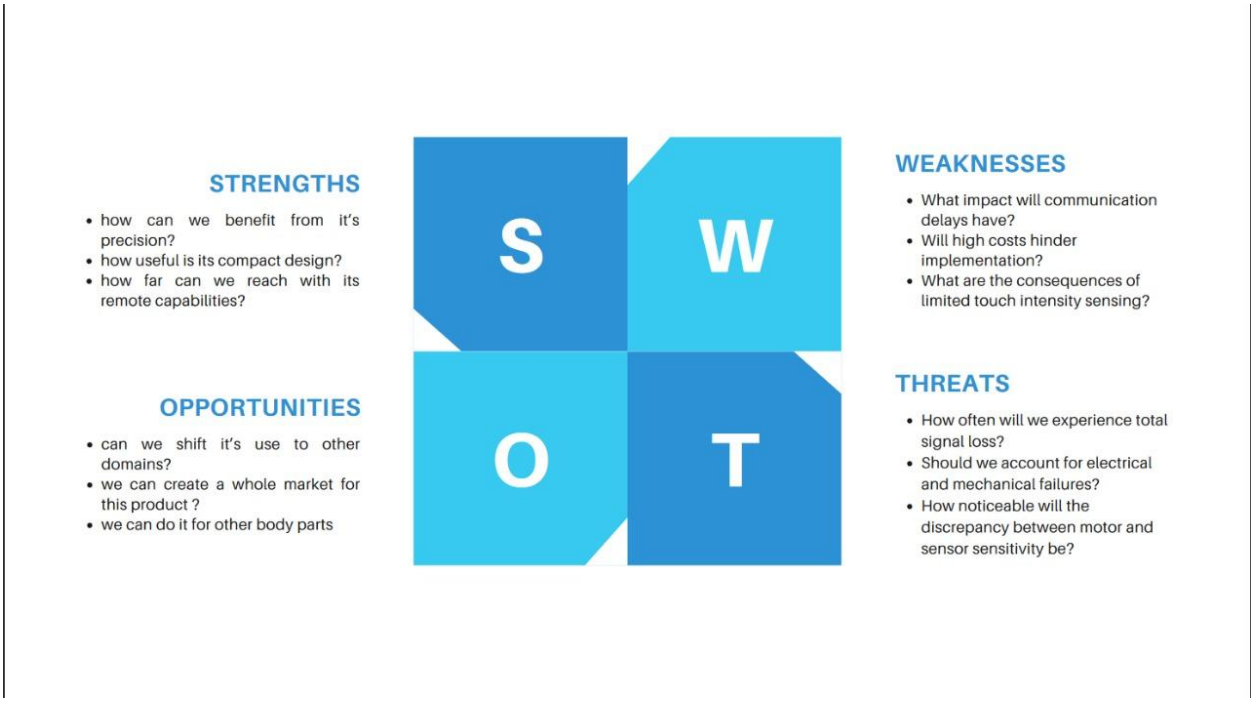
Gantt Chart:



Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 2 Title				
Gantt Chart	Mohamad Ghoush	100%	18/11/24	18/11/24
SWOT analysis	Abed El Rida Nehme	100%	18/11/24	18/11/24
Conceptual Design	Wafik Ibrahim	100%	18/11/24	18/11/24
Phase 3				
Cmponents Used	Abed El Rida Nehme	50%	18/11/24	19/11/24
Methodology	Mostafa Ajine	50%	18/11/24	20/11/24
Phase 4 Title				
Preparation	All team	40%	21/11/24	26/11/24
Writing Code A	Team A	0%	27/11/24	11/12/24
Writing Code B	Team B	0%	27/11/24	11/12/24
Bulding the robot	Abed El Rida Nehme + Mohamad Ghoush	0%	1/12/24	13/12/24
Testing	All Team	0%	6/12/24	13/12/24
Phase 4				
Result Evaluation	All team	0%	14/12/24	18/12/24
Future Work research	All team	0%	14/12/24	18/12/24
Criteria to Advance	All team	0%	14/12/24	18/12/24
Conclusion	All team	0%	19/12/24	24/12/24

Swott Analysis:



Methodology

1-Introduction

The primary purpose of this project is to provide reliable and accurate control over the robotic arm placed in the operation room to the doctor that controls this whole system remotely. The system will integrate multiple data inputs form the operation room and provide real time feedback for the doctor. The ai module will do some processing then offer some computed insights that help with the decision making and increase the overall accuracy and safety of the operation.

The problem we are trying to solve is that the doctor might be sometimes away from the hospital or where the operation is taking place. This might cause some critical and unwanted time waste due to the Doctor physical unavailability room, whether caused by travel or any other emergency.

In this system MQTT will be the bridge of connection between the multiple parts of the project, as it will be the utility we rely on to send the data form and to the doctor, considering that the Dr will send the IMU sensor data and will receive video feedback with Ai suggestions and vital signs added to it.

Like every computer system we have , this one also have key features like:
-gesture control: the robotic arms will be controlled according to the readings caused by the movement of the IMU sensor.
-patient monitoring: we spend a lot of efforts on the monitoring of the patient by using the Ai module and also displaying the video with the vital signs embedded in it.

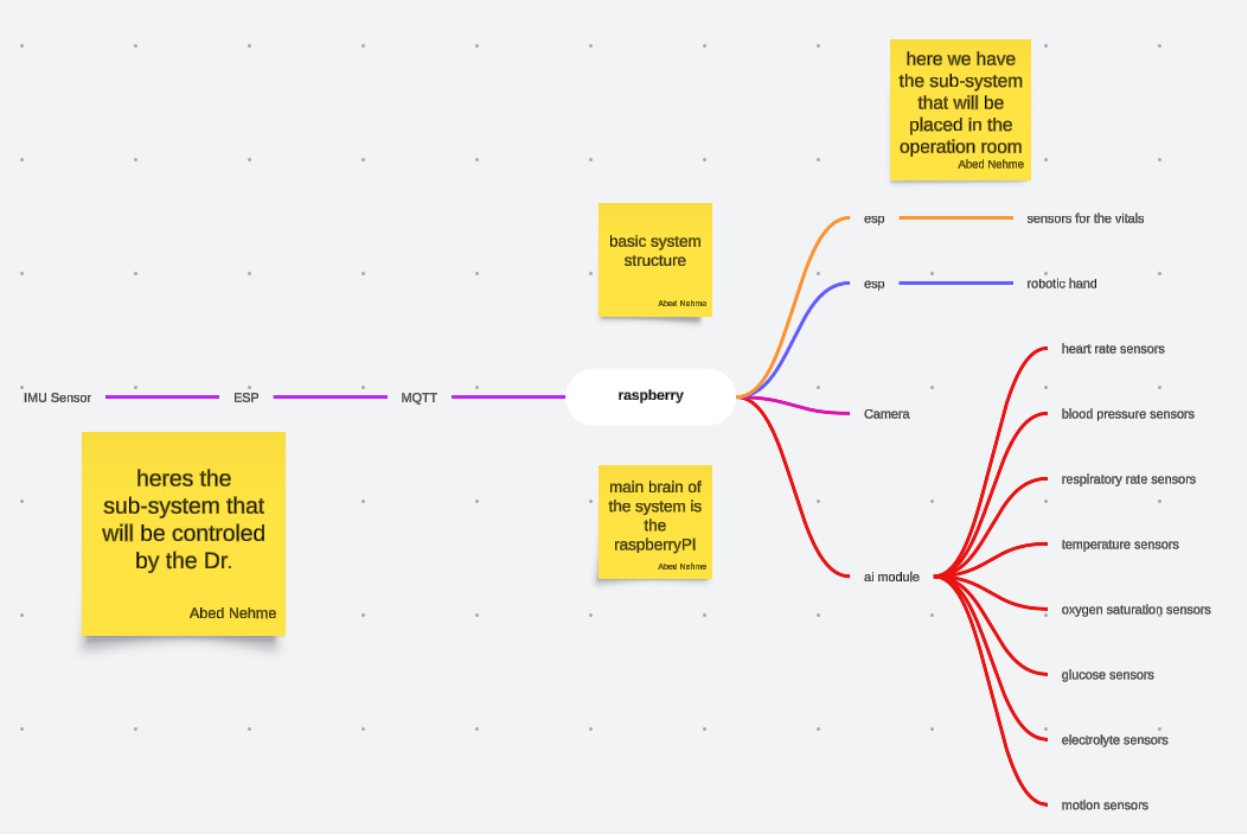
In this project we have the flow of data as following:
- sensors → Data sent via ESP8266 → Received by Raspberry Pi.
- Raspberry Pi → Processes data → Sends control signals to servo Motors.
- Raspberry Pi → Outputs to Display and integrate object detection results.

We used this combination of connections, sensors, methods, and we analyzed the obstacles and benefits of each decision we made to reach the most optimal productivity of the system in a way that helps with saving lives.

2-System architecture

An ESP device or devices (depending on the need) will be connected to a set of IMU Sensors (10DOF GY91 MPU-925 BMP280). The data collected will then be sent to the Raspberry Pi device through MQTT. The Raspberry Pi will go on and analyze this data and move the physical hand connected to the Raspberry Pi directly using a set of servo motors accordingly. Additionally, a camera will be connected to the raspberry pi device that will include object-

detection features according to the use of the whole device (surgeries, industrial, etc.). And finally, a screen will be installed beside the raspberry Pi to monitor its live movement.



3-Key-compnents

the controlling side

- ESP8266:

A microprocessor that can connect to Wi-Fi which we will use to send and receive data through wireless connections. This micro-processor has a clock-speed of 80 MHz, 160KB of RAM, and 1 MB Flash. These specifications are more than enough since we will be using it only to read data from the sensor and give PWM to the motors.

-IMU sensor:

The IMU sensor in a 10 degree of freedom sensor that detects and changes through the translational axis of the rotational axis. We use this to record any hand gesture the remote doctor makes. servo motors: Used to control robotic arm movement. We have decided to use this motor since although it is more expensive than the servo motor, it will provide more accurate and precise movement which is needed in this kind of operation.

-Gloves:

A simple glove with IMU sensors attached to it. We will put a sensor on each joint to track the movement of every movement. This method will allow us to change any broken sensor with ease.

Raspberry Pi 5

Will be the main brain of the operation that will receive images and vitals of the patient and through a designed knowledge graph fed by a well-organized medical and bio-medical databases to clear some work and analyze for the doctor. And the Raspberry pi will receive through MQTT connection data from the IMU sensors placed in the glove the remote doctor will be wearing to analyze these data and imitate the exact movement the doctor makes.

The controlled side

-servo motors:

Used to control the robotic arm movement. We decided to use this motor since although it is more expensive than the stepper motor, it will provide more accurate and precise movement which is needed in this kind of operation. In addition, it will need less connections than the stepper motor what will make the communication part easier.

-Camera:

We will use the camera of the raspberry pi and then we will install it directly to the raspberry and then we will use it to transport the video data to the ESP on the Doctor's side using the MQTT protocol, this whole process will for sure increase the accuracy of the work of the Doctor.

-Heart Rate (Pulse) Sensors

MAX30102:

It continuously tracks the heart rate of the patient while maintaining its output on the display, it Is widely used in ECG sensors for real time cardiac monitoring.

-Blood Pressure Sensors

Omron HEM-7320:
it measures the blood pressure continuously and it is also automatic blood pressure cuffs or invasive arterial lines.

-Respiratory Rate Sensors

Piezo Respiratory Belt:

used to monitor breathing rate through chest movement, it can also be connected to the ventilators for precise measurements.

-Temperature Sensors

MLX90614:

Body temperature is gathered as data using the skin or core temperature to increase the accuracy of the readings.

-Oxygen Saturation (SpO_2) Sensors

MAX30100:

It is clipped to the finger, and it reads the concentration of oxygen in the blood to ensure adequate oxygenation during surgery.

-Glucose Sensors

Freestyle Libre:

For non-diabetic patients, we can use it indirectly by measuring it once or in a discrete way before the operation.
For diabetic patients, we will measure the blood level continuously to monitor their sensitive reaction to multiple factors.

-Electrolyte Sensors

Medtronic Sodium Sensor:

It is used to monitor gases in the blood during long or critical surgeries.

-Motion Sensors

MPU-6050:
can be used to track equipment and surgeon movements during robotic operation.

4-Data transmission and processing

MQTT

Overview of data flow

-The input data of the MQTT protocol will be the IMU data collection from the controller side and it will be a kinetic reading to measure if there's a movement at the hand considering acceleration.

-the output data will be the live video feedback from the operation room, merged with the vital sensors' readings and the integration of the Ai module.

Why MQTT is Ideal for This System

-it is lightweight and doesn't require much processing which makes it a good option considering the small processor we have(raspberry).

- it is optimized for IOT.

-low latency with taking into consideration the local broker we can assign.

-It stands out from all the alternatives like "HTTP" and "WebSocket" in this specific use.

Role of MQTT

-the MQTT in this project will serve the role like the veins to the organs, it will send oxygen (Data) to the different parts of the system (organs).

Ensuring Reliable Communication

To make sure that the Data is received or sent correctly we can apply:

-error handling: we can double-check the received data to resolve any missing data or delayed messages.

-Encryption: we can later do an encryption protocol to the MQTT sent and received data to increase the security of the system.

Challenges and Mitigation Strategies

-latency issues.

-network consumption.

-system integration.

Raspberry PI

Gesture replication

- We are going to receive data from the IMU sensors through the MQTT Protocol and then we will replicate the movements of the hands and fingers and project it to the servo motors by mapping the data collected to fit the range of movement of each motor.
- While having multiple options to work with the data we chose the “numpy” and the “scipy” for motion data processing.
- For motor control and interfacing we chose the “RPI.GPIO” alongside “pigpio”.

Patient Data processing

- It will gather then vital sensors information and merge them will the Ai module that will be fed with all the data from medical books and research and it will analyze this data and predict illness or even warn the doctor before operating on the patient.
- We will also include an alert system that will for sure read the details of the data and notice any organ failure or any type of unwanted and abnormal bleeding during the operation.
- For better usage of these methods, we will use the “pandas” library in python for data handling and analysis. While we will use the “matplotlib” for the dynamic display of the vitals.

Object detection

- We are going to connect the camera to send us live video feedback, and to detect obstacles in the operation room.

- We haven't yet chosen an object detection algorithm since we want to use the machine learning approach.

Knowledge Graph Integration

- The purpose of using it is to provide a context-aware insight of everything that happens in the operation room.
- We haven't also reached any results considering this model since we don't have any experience with Ai models.

5-Implementation Plan

Prototype Construction:

For the prototype we will have a simple leather glove with holders for the IMU sensors (13 per hand) and a slot for the ESP8266 which we will have 1 per hand. All the sensors will be connected to ESP8266 using the I2C protocol. And in addition to this we will have a 3.3V external power supply to power these connections.
Then these ESP8266 will be connected to the raspberry pi through Wi-Fi using the MQTT protocol where the broker used is the mosquito broker hosted on the raspberry pi.
The raspberry pi will be connected to an ESP32 through Bluetooth communications to control the motors which will be connected to a PCA9685 to allow the connection of all the motors to the ESP32 at once.
We will need 14 total servo motors distributed as follows: 1 motor for the elbow (pitch only), 3 for the wrist (pitch, roll, yaw), 10 motors for the fingers (2 per finger).

Software Development:

For the software part, we first must collect the data effectively from the sensors and store them in an effective manner. Then we must send the data through MQTT while ensuring approximately no data loss which will prove challenging since the data rate will be fast. The next step will be to analyze this data using a certain algorithm to give orders to control the motors.

Now after we have finished controlling the arm remotely, we will work on the assistant that will help the doctor. First, we must detect the human parts and identify them and their status using a camera connected to a stand next to the patient and then display the image with the assistant's information on a screen next to the doctor at home.

We should not forget to calibrate the sensors at the start, so we will integrate it into the code so every time we power it on it will calibrate the sensors and set the arm to the position 0.

6-Challenges and Mitigation

Precision and Accuracy: we are working in a medical operation so precision must be our up most priority, so the algorithm used to compute the PWM's of the motors from the sensor data must be fast a

Mitigation:
-Use an advanced algorithm for calibration and analyzing sensors and motors.
- Use high precision IMU sensors and servo motors.

Control Latency: As we are using Wi-Fi to send data, the system is bound to have some lag that cannot be removed but we will try to provide fast Wi-Fi and a fast algorithm so that we do not have to waist more time.

Problems with the Power Supply: Powering such systems will of course need a stable power supply especially for the motors as the change of power will affect the accuracy of said motors, same for the sensors.

Mitigation: As we are in the operation room, we won't need any batteries, and for the motors we will use the PCA9685 shield to provide an outside source of power for the motors

Sensor Drift Challenge: If used for a long consecutive time, IMU sensors may drift and will grow and error which will result in fault positioning of the arm.

Mitigation: Provide a button for the doctor to recalibrate the sensors when possible.

Safety and Reliability: Making sure the patient or medical personnel are not harmed during the robotic arm's operation.

Mitigation: Include safety measures such as emergency stop buttons.

Resource and Cost Restraints: This system won't be cheap as the sensors and motors are costly and we need multiples, and the raspberry pi alone is a huge expense.
Mitigation: Unfortunately, this challenge can't be avoided since any cut in the price of this project will be paid in accuracy which we can't be messed with.

The Challenge of Regulatory Compliance: It might be intimidating to navigate the complicated regulatory environment for medical equipment.

7-Ethical and Safety Considerations

Ensure patient data is encrypted and stored securely:

Importance: Ensuring that the patients' data remain confidential, especially the video broadcast which may lead to sensitive data exploitation.
Method: We need to implement encryption algorithms both in the sending and receiving of data, this will need intensive research and some help from our doctors and lab instructors. And make sure to test these algorithms to find any possible flaw.

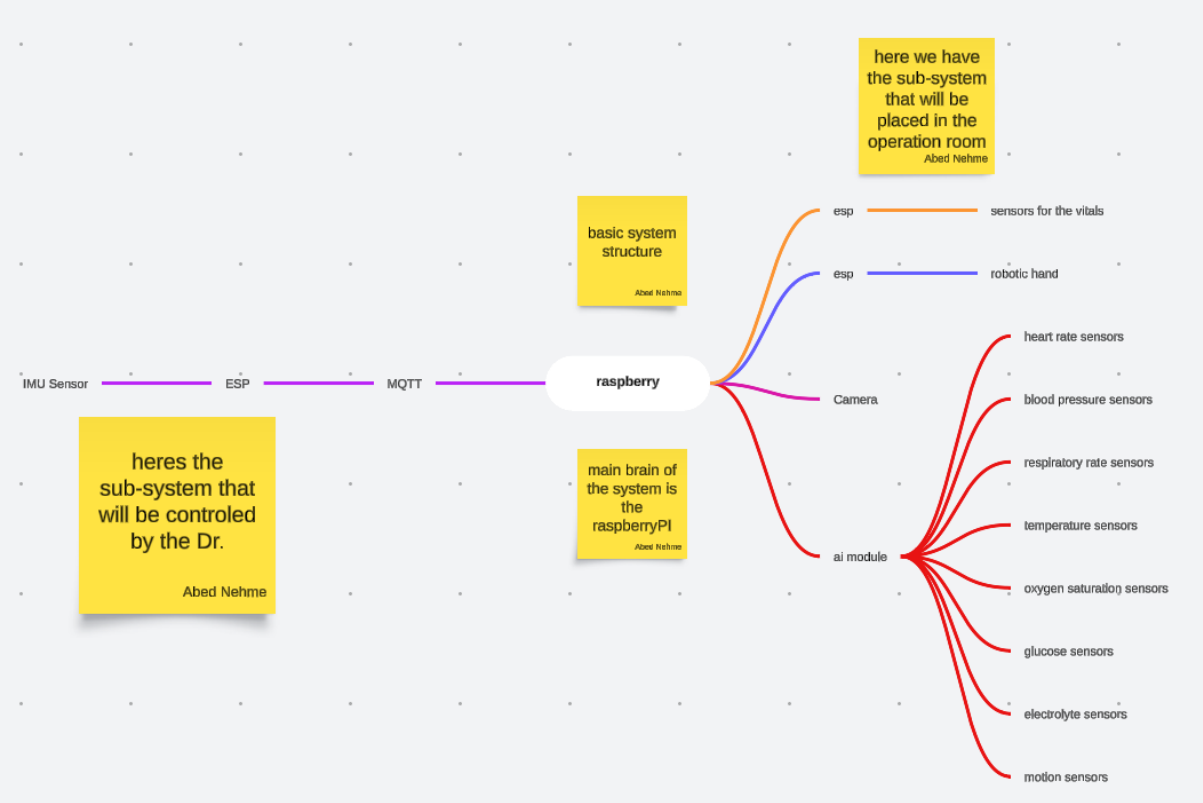
Conclusion:

So, in conclusion, we will be setting up an ESP8266 connected to IMU sensors via I2C to collect data send them for the raspberry pi using MQTT, analyze the reading and then give orders for the motors.
The AI assistant will show its recommendations and predictions on the screen next to the after analyzing the image taken by a camera connected to the raspberry pi.

The limitations that we will face will include the lag from the Wi-Fi connection, the budget, and the error in both the motors and the sensors.

But although we face some limitations and challenges, we are set to work on this project even if do not complete it in the set deadline, we will try our best to overcome the challenges and work with what we got to hopefully give the best rest possible.

Implementation:



Codes:

ESP8266 for collecting data:

```
#include <MPU9250_asukiaaa.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// ssid or the name of the Wi-Fi and its password

const char* ssid = "WING";

const char* password = "12345678910";

// MQTT server IP address of the Raspberry Pi

const char* mqtt_server = "192.168.43.173";

// Topic to publish to

const char* mqtt_topic = "sensor";
```

```
// Make the ESP8266 into a client
WiFiClient espClient;
PubSubClient client(espClient);
// The I2C pins of the ESP8266
#define SDA_PIN D2
#define SCL_PIN D1
float data[5];
// Connect to the Wi-Fi
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  // Wait till connected to Wi-Fi
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
}
// In case the MQTT connection is lost, try to reconnect
void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
```

```
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      delay(5000);
    }
  }
// Get the accX and accY to calculate the pitch angle
float get_angle() {
  float angle, aX, aY;
  if (mpu.accelUpdate() == 0) {
    aX = mpu.accelX();
    aY = mpu.accelY();
  }
  angle = atan2(aX, aY) * 180 / 3.14;
  return angle;
}
// Setup MPU9250 sensor, use only the accelerometer
void setup_mpu9250() {
  Wire.begin(SDA_PIN, SCL_PIN);
  mpu.setWire(&Wire);
  mpu.beginAccel();
}
void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
```

```
  setup_mpu9250();
}
void loop() {
  // If the MQTT client is disconnected, reconnect
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  // Get the angle and convert it to a String to send
  String payload = String(get_angle());
  // Publish data to MQTT broker
  if (client.publish(mqtt_topic, payload.c_str())) {
    Serial.print("Published: ");
    Serial.println(payload);
  } else {
    Serial.println("Failed to publish data.");
  }
  delay(10);
}
```

ESP32:

```
#include "BluetoothSerial.h"
#include "esp_bt_device.h"
BluetoothSerial SerialBT;
//function used to print the esp32 blutooth module MAC address
void printDeviceAddress() {
  const uint8_t* point = esp_bt_dev_get_address();
  for (int i = 0; i < 6; i++) {
```

```
    char str[3];
    sprintf(str, "%02X", (int)point[i]);
    Serial.print(str);
    if (i < 5) {
      Serial.print(":");
    }
  }
}
void setup() {
  Serial.begin(115200);
  Serial.println("\n---Start---");
  //set the name of the buetooth
  SerialBT.begin("ESP32");
  Serial.println("Device Name: ESP32");
  Serial.print("BT MAC: ");
  printDeviceAddress();
  Serial.println();
}
void loop() {
  //if bluetooth packets are available to recieve, recieve them and print them, later set the angle of the motor to the recieved value
if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
```

Rasberry pi code:

```
import paho.mqtt.client as mqtt
```

```
BROKER = "192.168.43.173"
PORT = 1883
TOPIC = "sensor"
OUTPUT_TOPIC = "sensor/output"  # Topic to send data to ESP
def send_to_esp(client, data):
    """Publish data to the OUTPUT_TOPIC."""
    client.publish(OUTPUT_TOPIC, data)
    print(f"Published to {OUTPUT_TOPIC}: {data}")
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to MQTT broker")
        client.subscribe(TOPIC)
    else:
        print(f"Failed to connect, return code {rc}")
def on_message(client, userdata, msg):
    try:
        payload = msg.payload.decode()
        print(f"Received angle: {payload}")
        send_to_esp(client, payload)
    except Exception as e:
        print(f"Error decoding message: {e}")
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect(BROKER, PORT, 60)
client.loop_forever()
```

Results and evaluation

Introduction to results

In this project we integrated multiple methods we learned from the learning outcome of this semester, first we used multiple microcontrollers each with its own characteristics, and we used multiple communication methods, and most importantly we integrated many computational libraries to manipulate the data input and output it to the network.

Presentation of results

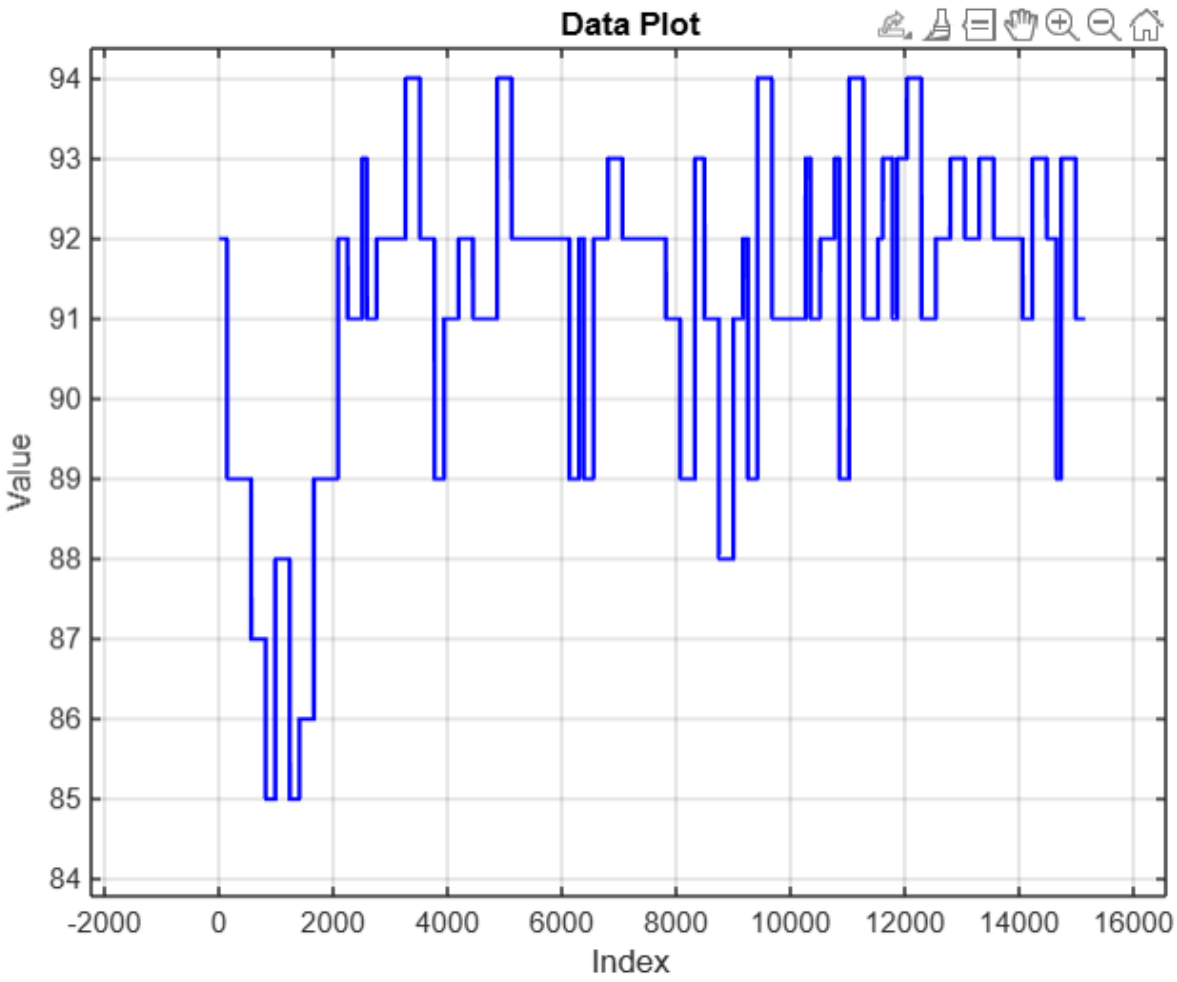
Key findings

The outcome of this project was evaluated based on different criteria:

- Data accuracy:
  the data accuracy represents the percentage of the usable data gathered from the IMU sensors we have.
- System performance:
  we faced some limitation to integrate the different methods on different platforms like “Wokwi”, “TinkerCad”, “ViltualBox”.
- Reliability:
  we found that the protocols are sometimes unreliable because of connection limitations so we had to search and choose other options rather than going with unified work.

quantifiable metrics

- While gathering the data we recorded the following readings form the IMU sensor:



- With comparing and monitoring data transportation we noticed that the system demonstrated an average response time of 200ms in the best cases with an accuracy of 95% for motion detection.

Comparison of objectives

- We talked previously about the objectives of this project being the following:

  Remote Control of Robotic Arm
  Integration of Data Inputs
  Real-Time Feedback
  Addressing Doctor Unavailability
  Gesture Control
  Patient Monitoring
  Data Processing and AI Insights
  Optimized System Communication
  Safety and Reliability
  Overcome Technical Challenges

- In this project we accomplished the remote control of the motor, and we met expectations when talking about the integration of data inputs as well as the gesture control alongside the safety and reliability. While on the other hand, our work wasn't enough when talking about communication and optimization of system communication.

Limitations or challenges encountered

- Connection limitations
- Networking complexity
- Lack of resources to fund the whole project
- Lots of errors when dealing with the IMU sensor
- Unavailability of a driver for the motors (considering the stepper motor choice) that can handle the big number of motors to control one or both hands
- Inaccuracy of the stepper motor when dealing with small angles.

Future work opportunities

Enhancements to Current System

- On the sensor level, we can better design the location of the sensors on the hand to reach the optimal case cost and performance wise.
- On data computation wise, we can move the Ai model to a computer to get enhanced results
- We can use a higher definition camera
- We can use a server dedicated to communication to ensure good connection
- We can search for better motors and robotic arm design to have better response and grip.

Expanding Scope

- This project has a very wide range of integration options, as we mentioned before, it can be used by a doctor, inside a chemistry lab, and even by NASA.
- We can merge two systems together. In the planning phase for this project we had two options, to integrate the motion detector by camera, or to use the sensors. And we decided to choose the IMU sensors since they are more accurate and more easier to lay hands on, but in the future we might integrate the two systems together to form a environment that detects the movement of the hand with sensor values and correct it with the camera feedback.
- We can use this technology anywhere and everywhere.

Research Directions

- After this we are going to focus on searching for a better working protocol, and better communication methods.

- Better motors
- Alternatives of the multiple vital sensors

Practical Applications

The robotic arm has a wide range of practical applications across multiple domains. In manufacturing, it can be used for assembly lines, material handling, and enhance efficiency and precision.

In biomedical domain, it could assist with delicate surgeries or rehabilitation exercises. The arm could also be integrated into logistics for sorting and packaging tasks, improving safety.

Additionally, its adaptability makes it suitable for research and development. With further refinement, the robotic arm could be a new solution for automation in many places.