Fault detection and remote monitoring of robotic manipulator using Internet of Things (IoT)

*Faiq Ahmad Khan [a] *, Muhammad Hamza Khan [b], Fizza Khan [c], Mubashir Bashir [c] and Khalil Ullah [d]*

[a] *Department of Electrical Engineering, National University of Computer and Emerging Sciences (FAST) Islamabad*
i220182@nu.edu.pk

[b] *Centre for Wireless Communications (CWC), University of Oulu, Finland*
[c] *Department of Electrical Engineering, National University of Computer & Emerging Sciences, Peshawar, Pakistan*

[d] *Department of Software Engineering, University of Malakand, Pakistan*

**Corresponding Author*

Abstract

Remote monitoring of industrial machines is important in order to have high productivity continuous process control. New emerging technologies such as Internet of Things (IoT), are facilitating the process of automation and remote controlling in multiple ways. In this research work, a robotic manipulator based on PUMA robot structure and an intelligent remote maintenance system is developed aimed to ensure the continual accurate operation of a robotic manipulator. Entire manipulator is designed from different types of motors like DC motors, stepper motors and servo motors along with their motor drivers. Various types of sensors like Tachometer, DHT11 Temperature and Humidity sensor, Capacitance meter and EMF Detector are also designed and interfaced with the manipulator for identification of different faults. Android App and Wi-Fi module are used for the remote-controlled automation purposes. The authors were succeeded in detecting various faults in the robotic manipulator and notifying the operator in real-time.

Keywords: Robotic manipulator, Sensors, Internet of Things

1. Introduction

The sense of sound is a remote sense that carries information about remote events. Unlike sight, hearing is not limited by field of vision. Mechanical systems commonly produce sounds while operating. The operating sounds carry information about the state of the system. Thus, sound signals can potentially be utilized in fault detection. Robotic manipulators consist of serial and parallel connection of revolute and/or prismatic joints between base and end-effector frame [1]. They further consist of two main parts i.e. electrical and control part, where main function of electrical parts is to supply voltage, drive actuators and transmit the data among subparts of the manipulator, and the control parts are associated with adjustment of timing between subparts of the robot to make sure best possible performance [2].

In this project, a robotic manipulator based on PUMA robotic structure and an intelligent remote maintenance system is developed, aimed to ensure the continual accurate operation of a industrial robotic manipulator [3]. Dc motors are used to perform various rotatory movements. Similarly, various sensors are designed and installed to detect and identify faults in the system. The block diagram is shown in Figure 1.

Internet of Things (IOT) technology is also introduced in the project, which is based on a network of web associated objects that can accumulate and swap information using embedded sensors [4]. REST protocol is used in the project for communication between the microcontroller and a computer and Rest API which is an application programming interface for a webpage is a piece of code that enables two software programs to communicate among each other [5]. ARDUINO Wi-Fi Shield (CC3000) is interfaced with ARDUINO Board, which enables ARDUINO to connect with the internet. A web server on the ARDUINO board is created to accept REST commands from outside client, from an interface running on a computer or a cell phone.

The external client is a Web Application or an API accessed through a web browser. Library that handles the REST calls is called as aREST. Then, two applications for Relay Control are created: One is Web Application and the second one is BLYNK Android Application. For building the Web Application, a HTML file and a Java Script File are created. BLYNK Application is installed in an android phone to establish a connection between Raspberry pi and BLYNK android application. Once detected that the motor is damaged, an email is received by the responsible person for maintenance and monitoring of the Robotic Manipulator. A python script is used and Transmission Control Protocol (TCP) for sending the email.

The control portion of robotic arm is accomplished using the C-C++ language whereas the recording and sound recognition/processing of motors are done using Python.

The paper is organized as follows. In section 2, proposed methodology of the designing robotic manipulator and interfacing different modules and sensors are discussed. In section 3, the experimental results are presented and summarized. In the last section, the conclusion and future work are provided.

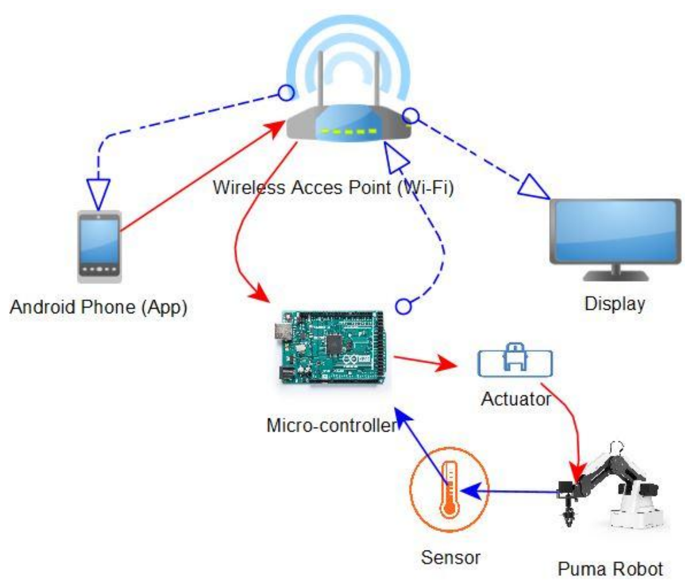

Figure 1: Block Diagram

2. Methodology

In this study, a real-time fault detection system is designed to ensure the smooth industrial processes. To do so, a robotic manipulator is designed and various sensors are connected to it. IoT technology is used to send the real-time sensory data to the WEB for user to monitor and notify the operator in case any fault is detected.

Puma Robotic Manipulator

Entire Robotic manipulator is built from plastic, steel and different types of motors. Motors that are used including servo motors, stepper motors and DC motors. ARDUINO Uno is used to control motors along motor control modules. Different sensors (speed, EMF, temperature & capacitance) are also added for identification of different faults. Then robotic manipulator is connected to ARDUINO. By creating a function, robotic arm moves in a particular direction back and forth, and meanwhile, sound generated by the motors is recorded using microphone. The control portion of robotic arm is accomplished using the C-C++ language whereas the recording and sound recognition/processing of motors are done using Python.

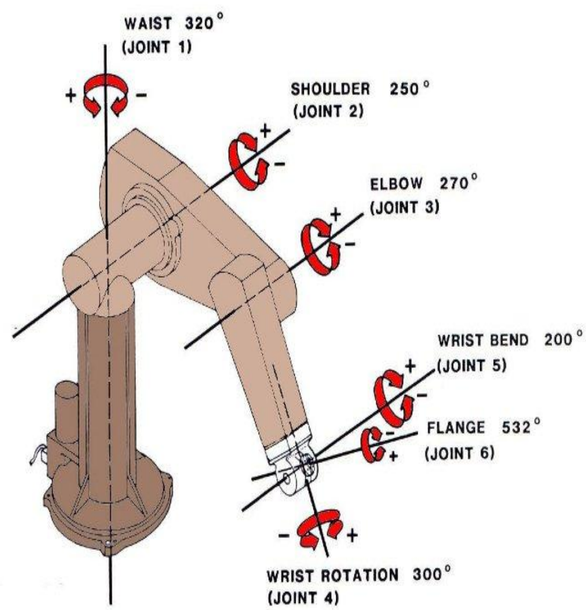

Figure 2: Puma Robotic structure

2.1 Sensors

2.1.1 Tachometer

Tachometer is an instrument which estimates the working speed of a motor commonly in revolutions per minute which displays on LCD that is connected with microcontroller. The circuit shown in Figure 2 was implemented to record RPM of motors.

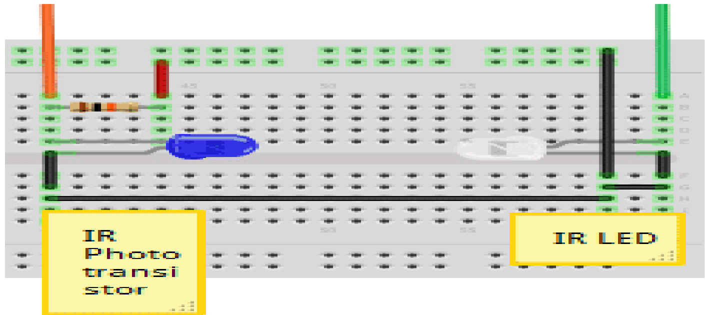

Figure 3: Tachometer

2.1.2 DHT11

DHT11 computerized temperature and humidity. It is a composite sensor contains an adjusted computerized signal output of the temperature and humidity[6]. Sensor data displays on LCD. It was used in order to prevent any severe damage in case of over-heating.

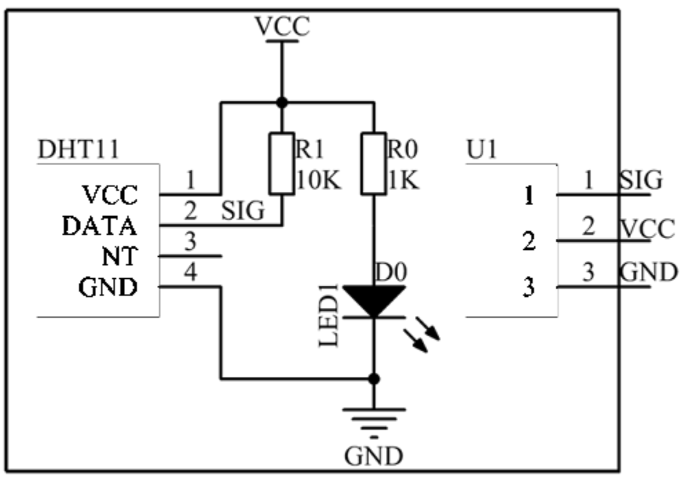

Figure 4: DHT11 Sensor

2.1.3 Capacitance Meter

A capacitance meter is a bit of electronic test gear used to check capacitance, generally of discrete capacitors [7]. Sensor data shows on LCD. The circuit shown in Figure 4 was implemented with the help of Arduino UNO.

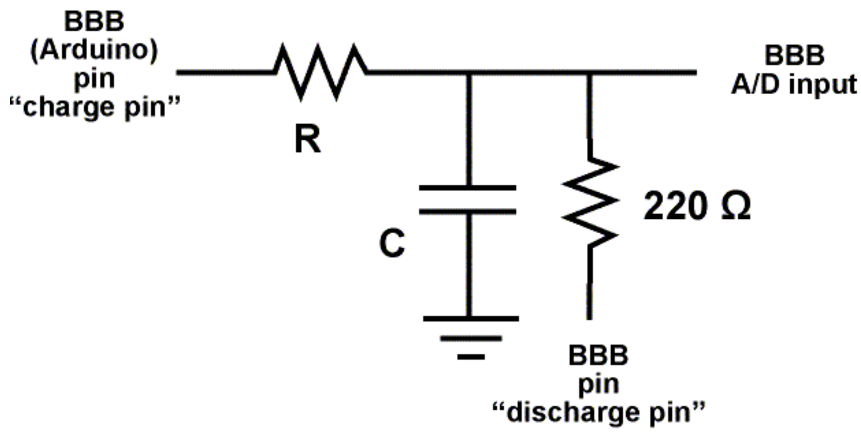

Figure 5: Capacitance Meter

2.1.4 EMF Detector

An EMF meter is a logical instrument for estimating electromagnetic fields[8]. As current is directly proportional to EMF so we can estimate the current as well through EMF Detector and sensor reading displays on LCD.

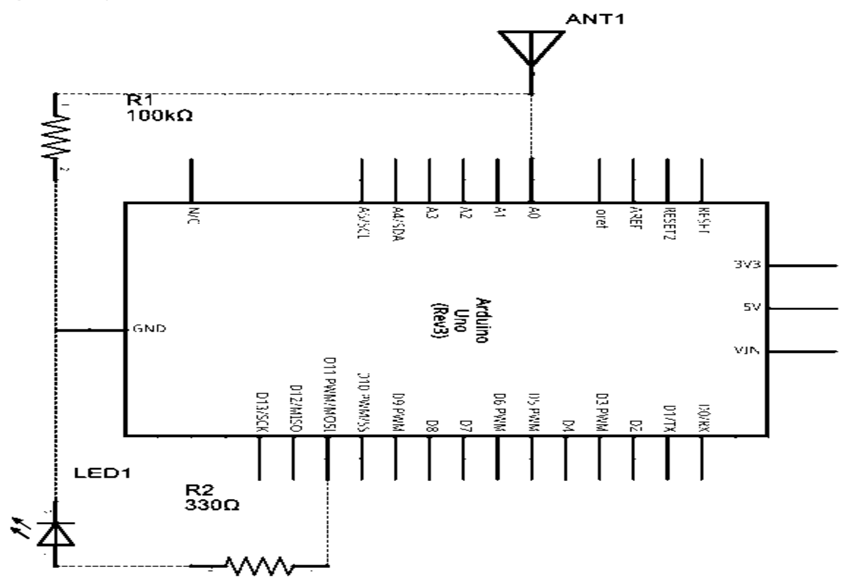

Figure 6: EMF Detector

2.1.5 Internet of Things

Internet of Things is a concept based on a web of internet-associated things that can gather and swap data using fixed sensors. An IoT gadget is any independently internet-associated appliance that can be supervised and/or managed from a distant position. We use REST protocol in our project for communication between microcontroller and computer. REST stands for Representational State Transfer. It is a communication design that was created back in 2000. REST defines many limitations that the design has to follow, the most important features are as following:

- A client/server communication: a client has to send a request to a server; the server has to answer the request.
- A stateless communication: every appeal by the client from the server should be a complete one so the server does not misinterpret anything, the client will than not have to rely on any data kept on the server. Making the state of communication independent of each other (client & server).
- A uniform interface to help the client easily point out resources on the server.

2.1.6 REST API

An application programming interface for a webpage is a piece of code that enables two software programs to communicate among each other. Using the API code you can request a server for some operation or information from the client side. The API we are using is based on REST protocol, the protocol has been explained above. The REST API is also known as RESTful API/RESTful web service it uses Hyper Text Transfer Protocol to collect, place, paste or remove data from a server. RESTful API is vastly used in web development services [9].

2.1.7 Enabling ARDUINO to Connect with the Internet

2.1.7.1 Approach

The ARDUINO is connected to CC3000 which enables ARDUINO to connect with the internet [10]. An internet server is made on the ARDUINO that accepts REST calls from an external user that uses a laptop/pc or a mobile phone to communicate with the ARDUINO. The external client is a Web Application or an API accessed through a web browser. Library that handles the REST calls is called as aREST.

Firstly, to access ARDUINO with a REST call a variable has to be declared in the sketch with its own alias name.

To use REST protocol, we write a code in C++. We use the REST library so we can use RESTful API to call for functions. First of all, we declare a variable for REST protocol usage in our program sketch as shown below:

```
// Expose variables to REST API
rest.variable("temperature",&temperature);
rest.variable("humidity",&humidity);
```

To access the value of temperature from client side you need to use the REST API variable you had declared in your code to call the server for collecting temperature information. Here we have declared the variable as temperature; arduino.local is the IP-Address of the server.

This http.RESTful API is a stateless concept that client is consuming the API on request.

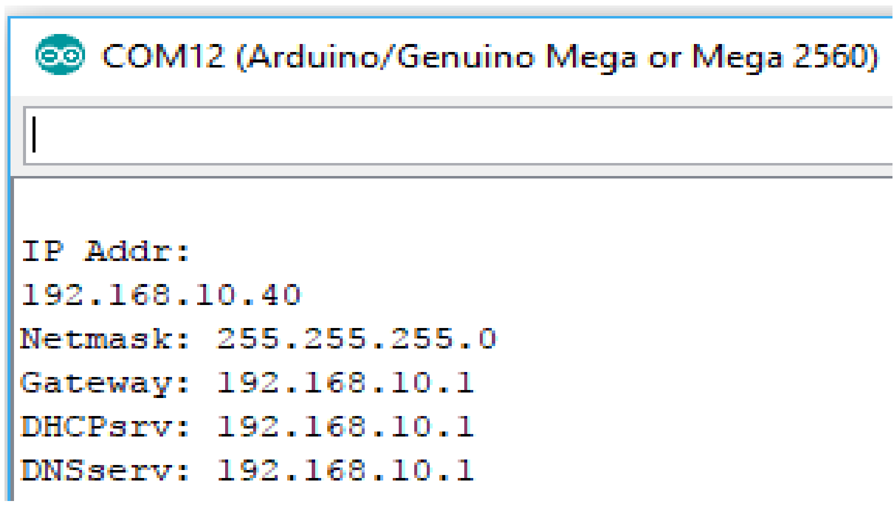

Figure 7:Server IP Address & other information displayed on Serial Monitor of ARDUINO

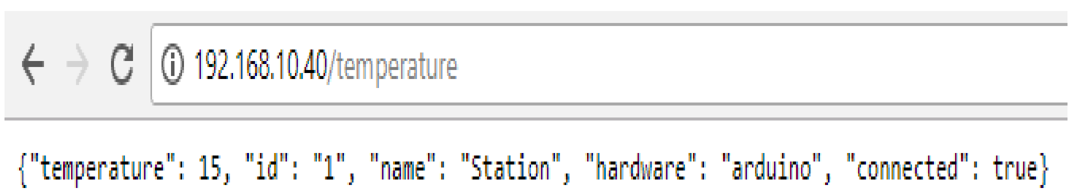

Figure 8:Client requesting temperature data using RESTful API in a web browser

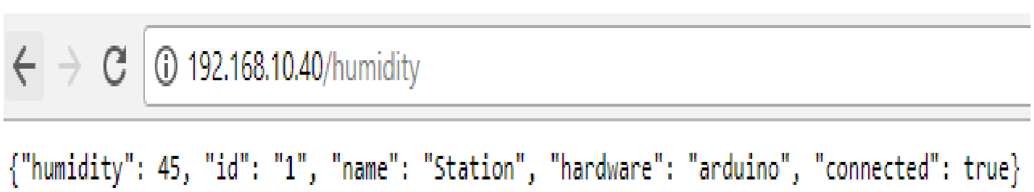

Figure 9: Client requesting humidity information using RESTful API in a web browser

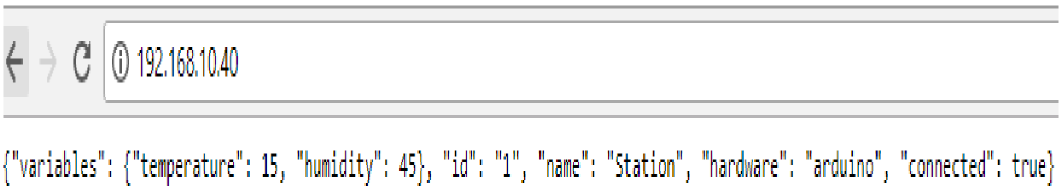

Figure 10: Arduino.local IP-Address Displaying all variable declared data at once

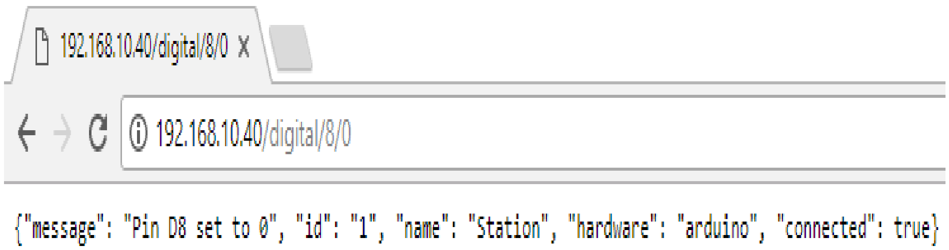

Figure 11: Client controlling digital pin 8 of ARDUINO using RESTful API in a web browser (setting pin to 0)

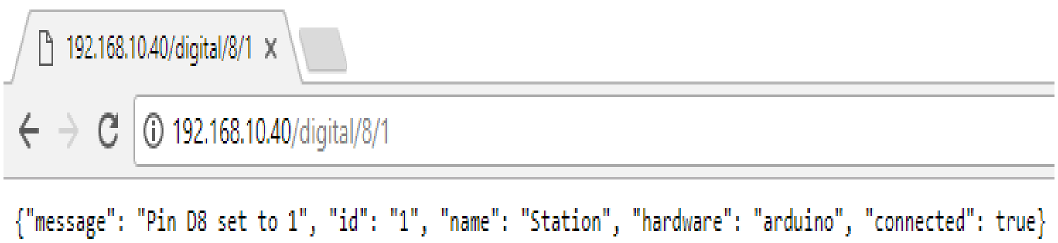

Figure 12: Client controlling digital pin 8 of ARDUINO using RESTful API in a web browser (setting pin to 1)

2.1.7.2 Web Application for Relay Control using ARDUINO Wi-Fi Module

2.1.7.2.1 Wi-Fi Module

Wi-Fi is the name of a well-known remote systems administration innovation that utilizations radio waves to give remote rapid Internet and system associations. It is utilized to transmit information, enable PCs to the web and enable PCs to associate with remote assets, for example, hard drives, printers, and other outside gadgets [10].

Wi-Fi has now turned into an exceptionally regular innovation and is accessible broadly and economically on a scope of gadgets. Cases of this include iPhone, other telephones, Cameras, Printers, Laptops, Game Consoles, iPod touch. Wi-Fi can make get to publicly accessible at Wi-Fi hotspots. In this project, we have connected Wi-Fi module with ARDUINO Uno for observing the temperature/humidity of motors and then controlling the relay via web app [9].

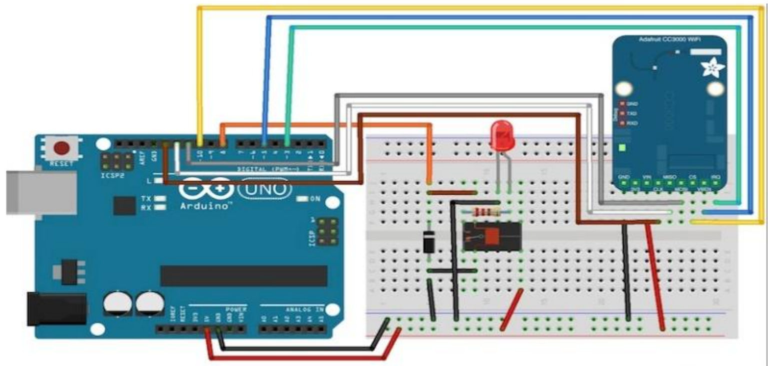

Figure 13: Connection diagram of ARDUINO UNO and Wi-Fi module

There are two portions for building the relay web application: a Hyper Text Mark-up Language file containing the interface, and a client-side JavaScript file that handles the clicks on the interface.

2.1.7.2.2 HTML File Creation

The html file contains all the important libraries for the interface. So, we import all libraries to the html file. After importing libraries, we declare two variables that are used for on and off buttons of the web application.

2.1.7.2.3 Java Script File Creation

Once the html file is created we save it. Now we create a java script file to handle on and off switching of the application buttons declared in the html file. We also have to establish a connection between our ARDUINO and the web application for that we declare a variable name that we connect to our ARDUINO's multicast Domain Name System.

The adafruit cc3000 Wi-Fi module enables your ARDUINO to connect with your wireless router, making it capable of communication over the internet. It uses Serial Peripheral Interface bus (SPI) for communication over short distances. The CC3000 shield an Interrupt pin IRQ that has asynchronous connection. It uses 802.11b/g and various types of wireless local area network security protocols for secured Wi-Fi connection. That shield also supports transfer control protocol and User Datagram Protocol in both modes, having up to four connections simultaneously

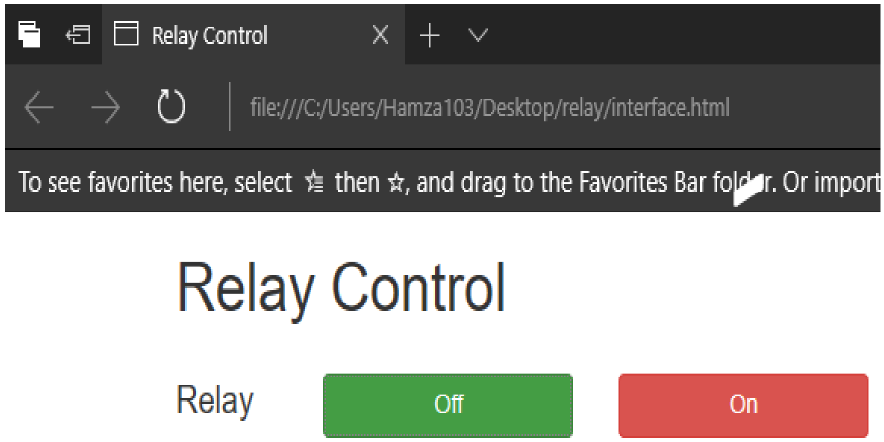

Figure 14: Graphic User Interface

2.1.7.3 ARDUINO Wi-Fi Shield (CC3000)

Figure 15: ARDUINO Wi-Fi Shield (CC3000)

Connecting CC3000 to the internet: To use CC 3000 Wi-Fi module with your home wireless router you need to enter your router name as displayed on your laptop and password. You also need to define your router security protocol for a successful connection.

```
#define WLAN_SSID       "PTCL_BBB"
#define WLAN_PASS       "61b8f405"
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

2.1.8 Raspberry Pi (Relay Control Using an Android Application and sending notification via email)

In the previous section we used ARDUINO mega and an ARDUINO Wi-Fi shield to control a relay. In this section now instead of using an arduino and a web application we will use a raspberry pi and an android application called as BLYNK to control the relay.

2.1.8.1 BLYNK Android Application

BLYNK is an Application supporting iOS and Android apps to control microcontrollers like ARDUINO & Raspberry Pi. BLYNK provides them access to the internet [10]. BLYNK is very useful in IoT applications.

It's a digital panel where you can develop a graphic user interface for your project by simply adding and removing icons. The icons are widgets. Once you add them you can specify the pin they are connected to on your microcontroller.

BLYNK uses RESTful API it allows users to easily collect and exchange values to/from Pins in BLYNK application and hardware.

2.1.8.2 Raspberry pi Email Notification (IOT)

Python Script: When the system identifies the motor as damaged, it needs to notify the responsible person that he/she could take measures for maintenance. In order to do this the system sends an email to that person. We have used a python script for sending the email [12]. The email is sent by raspberry pi once motor is identified as damaged by the classifier.

2.1.8.2.1 SMTP

Simple mail transfer protocol is a protocol used to send and receive emails over the internet [12]. SMTP uses TCP (Transmission Control Protocol) port 25 for communication between servers for mailing. Port 587 is used for client emails submission. SMTP connections are secured and they cannot be interfered with.

To send an email using raspberry pi you need to have the following information in your raspberry pi script:

- Your Account email address and password in a file named CONFIG that should be placed in the same directory as the email python script. This email address is the one raspberry pi would use to send an email. When the email is received by the responsible authority it would be through this email.
- Mention the subject in the python script so the person is aware of what the purpose of email is.
- Write body of message mentioning the reason of the email e.g., motor A in the robotic manipulator is damaged it needs maintenance.
- In the python script you will have to mention the email address of person to whom you have to send email.

3. Results and Discussion

In this study, an IoT based automated industrial robotic system is developed to detect fault using sensory data and enable the operator to be notified and take action wirelessly accordingly in real-time. The final hardware of the robotic manipulator is shown in Figure. Multiple sensors are interfaced with the microcontroller to detect different faults using the received data as illustrated in Table 1.

Moreover, Wi-Fi module is also interfaced to enable the microcontroller to communicate with the operator wirelessly. The operator is made capable of taking various actions with the help of Android application.

The threshold for temperature is set to 24±2 ^{o}C, and the operator is to be notified if the internal temperature violates the criteria, so that overheating or freezing can be avoided. Similarly, most of the capacitors used in the project are of 47µF, hence the threshold to identify the mal-functionality is set to 47±1 µF. Any value lower than the specified threshold results in sending an email to the operator to take necessary action using the Blynk App. EMF and tachometer are also used to detect if the system is receiving proper current and if the motors are rotating with the required speed, respectively. The thresholds of 5±0.3V and 320±10 rpm is set for the identification of any abnormality in current received by the system and the rotation of the DC motors respectively.

It is observed that, with the automated robotics system, time consumption and labor cost can be reduced immensely. It is also noticed that fully automated system can increase productivity effectively.

Table 1: Sensors Data

Sensor	Data	Threshold	Decision	Notification (Email)	Action
Temperature	35.5 ^oC	>24±2 ^oC	Too warm	Send	Turn on the internal fan remotely to cool down the machine
	24 ^oC	=24±2 ^oC	Normal	Do Not Send	Do nothing
	15.7 ^oC	<24±2 ^oC	Too cold	Send	Turn Off the fan remotely to maintain the moderate temperature
Capacitance meter	23.56 µF	<47±1 µF	Too low	Send	Alert the Operator
	46.3 µF	=>47±1 µF	Normal	Do Not Send	Do nothing
EMF	6.7 V	>5±0.3V	Higher Voltage detected	Send	Turn Off the main switch via IoT and alert the Operator
	4.8V	=5±0.3V	Normal	Do Not Send	Do Nothing
	3.2V	<5±0.3V	Lower Voltage detected	Send	Turn Off the main switch via IoT and alert the Operator
Tachometer	375 rpm	>320±10 rpm	Too fast	Send	Alert the operator
	320 rpm	=320±10 rpm	Normal	Do Not Send	Do Nothing
	197 rpm	<320±10 rpm	Too slow	Send	Alert the operator

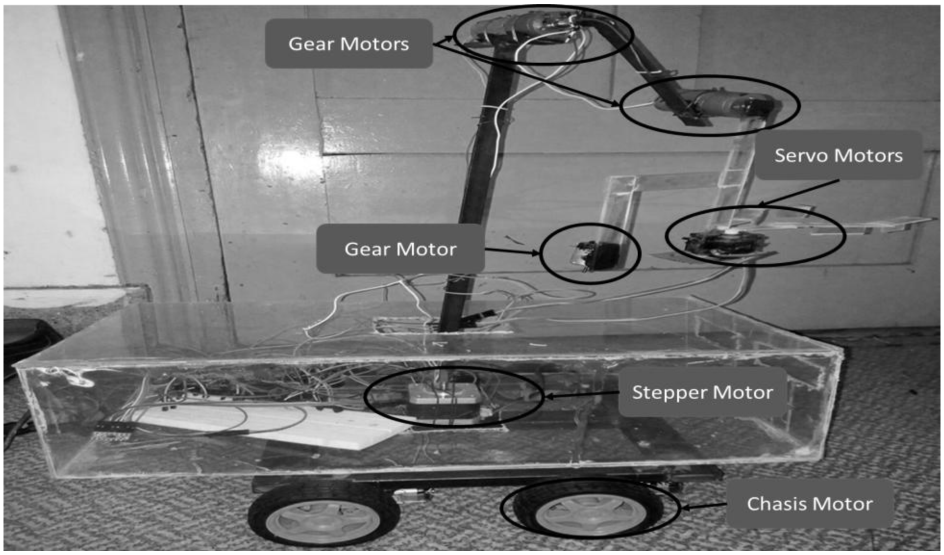

*Figure 16: Robotic Manipulator*s

4. Conclusion & Future Work

In this research work, state-of-the-art technique used to analyze sensory data received from different sensors and detect various faults in real-time. The same method can be used to differentiate other electrical equipment besides motors. Four different sensors (tachometer, capacitance, EMF, temperature) were designed for monitoring the robotic manipulator to reduce chances of error occurrence in machine. Each sensor has its own circuitry and assist in remote monitoring.

Faults in motors do not necessarily stop the motors from working but they do slow down or decrease the output leading to loss in industrial capital and wastage of time, the same is case for other systems. For modeling and experimentation of motor sounds a robotic manipulator based on a PUMA robot was built to demonstrate how our project could be implemented in an industry where robots are used as labor.

The use of IOT assist in monitoring and controlling the robot without any person being present at site or even near the site. The authors were able to successfully implement the system and detect various faults in real-time. It was observed, that IoT-assisted systems not only reduce the labor cost but also protects the systems from complete damage. In the future, the author would like to incorporate machine learning techniques to find correlation among different sensory data and enable machines to act upon accordingly.

Acknowledgements

The authors would like to thank Mr. Muhammad Kamil, Mr. Syed Mohsin Shah and Mr Waseem Ullah for their guidance and cooperation while working at the research lab in FAST-NUCES, Peshawar.

References

[1] T. R. Kurfess, Robotics and automation handbook: CRC, 2005.

[2] Piltan, Farzin, SH Tayebi Haghighi, N. Sulaiman, Iman Nazari, and Sobhan Siamak. "Artificial control of PUMA robot manipulator: A-review of fuzzy inference engine and

application to classical controller." International Journal of Robotics and Automation 2, no. 5 (2011): 401-425.

[3] Vivas, A., and V. Mosquera. "Predictive functional control of a PUMA robot." Conference Proceedings. 2005.

[4] Grau, Antoni, Marina Indri, Lucia Lo Bello, and Thilo Sauter. "Industrial robotics in factory automation: From the early stage to the Internet of Things." In IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, pp. 6159-6164. IEEE, 2017.

[5] Masse, Mark. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. " O'Reilly Media, Inc.", 2011.

[6] Tianlong, Ni. "Application of Single Bus Sensor DHT11 in Temperature Humidity Measure and Control System [J]." Microcontrollers & Embedded Systems 6 (2010): 026.

[7] Metze, S., S. Ruhl, G. Greller, C. Grimm, and J. Scholz. "Monitoring online biomass with a capacitance sensor during scale-up of industrially relevant CHO cell culture fed-batch processes in single-use bioreactors." Bioprocess and biosystems engineering 43, no. 2 (2020): 193-205.

[8] Shimatani, Masaaki, Shimpei Ogawa, Daisuke Fujisawa, and Satoshi Okuda. "Electromagnetic wave detector, electromagnetic wave detector array, and electromagnetic wave detection method." U.S. Patent Application 16/477,641, filed August 20, 2020.

[9] Sadowski, Sebastian, and Petros Spachos. "Wireless technologies for smart agricultural monitoring using internet of things devices with energy harvesting capabilities." Computers and Electronics in Agriculture 172 (2020): 105338.

[10] Rafyan, Afif, Ritzkal Ritzkal, and Yuggo Afrianto. "Prototype Control of Lighting Equipment Using Web-Based Arduino Microcontroller." Jurnal Mantik 4, no. 3 (2020): 1604-1610.

[11] Bohara, Bharat, Sunil Maharjan, and Bibek Raj Shrestha. "IoT based smart home using Blynk framework." arXiv preprint arXiv:2007.13714 (2020).

[12] Indukuri, Chaitanya Lakshmi, and Kottilingam Kottursamy. "Advanced Accident Avoiding, Tracking and SOS Alert System Using GPS Module and Raspberry Pi." In Artificial Intelligence Techniques for Advanced Computing Applications, pp. 167-178. Springer, Singapore, 2020.

[13] Krzyzanowski, Paul. "Sample SMTP Interaction." (2020).