

# **Controlling and Monitoring the Performance of a Robotic Arm using Internet of Things (IoT)**

Ola Khalaf Assi  
Iraq University College  
Iraq – Basrah  
Somar Building Al-Estiqal St.

Ali A. Abed  
University of Basrah  
Computer Engineering  
Iraq – Basrah

## **ABSTRACT**

Technology, if it is not well managed and used then it will be a huge waste. It is imperative to accept the fact that technology is taking over, and the internet of things (IoT) technologies are the future. This project aimed to design and develop a smart Robotic arm that is remotely controlled and monitored using IoT. The controller of the Robotic arm system is the Arduino UNO board, and the internet connectivity is provided to the system using a Wi-Fi model. Multiple sensors have been used to create a more comprehensive system. The IoT platform used to control and monitor the Robotic arm performance is the Blynk app, which is used to display the sensors output data and control the Robotic Arm movement. The accuracy and effectiveness of the IoT platform is tested and validated by comparing its results with accurate measurement devices.

## **Keywords**

Internet of things (IoT), Robotic arm, IoT platform, WBS network, Wi-Fi Model.

## **1. INTRODUCTION**

The rapid technology development of the industry 4.0 that can be seen in the Internet of Things (IoT) revolution, along with the increasing employment of robots in man's everyday life activities, made the integration of the robotics applications and the IoT technologies an inevitable fact that draws the new features of our forthcoming future. The IoT model encompass a network of heterogeneous interconnected objects that can be addressed unambiguously and it can create an interaction between them using standard protocols of communication [1]. These objects include e.g., sensors, actuators, computers, and cell phones, etc. The robotic arm was designed and fabricated according to the project requirement. The mechanical and electrical components were chosen to fulfil the objectives of this paper. The mind of this project is the Arduino UNO microcontroller, and the system is provided with the internet connectivity using ESP8266 Wi-Fi model. The sensors used in this project and the servomotor is connected to the Arduino board which is programmed to achieve the required functionality. And the gap between the hardware and the data network is filled by the Blynk IoT platform. The first attempt of interaction between the internet and the resource constrained aimed to connect smart thing into standardized Web service architectures (such as SOAP, WSDL, UDDI) [2] or apply the HTTP embedment servers into the devices. However, practically, it is very complex and heavy for the simplest devices [3]. For this project, an efficient secure way of communication is used to connect the smart Robotic Arm to the internet. A way that can solve the complexity of the IoT interfacing protocol.

## **2. LITERATURE REVIEW**

Earlier, some researcher did control their robot using IoT, they

used different protocols, most of them were very complicated and required a prolonged procedures to be able to connect the Robotic arm system to the internet, for example one of the researcher suggestion was to use SOAP protocol and PC server with ASP.NET framework, the protocol also required the usage of PC client that have been develop with ASP.NET framework. There were delays of 30 second in the response due to the overhead in the Web service, taking into account the web services operate as request-response base, which will negatively affect the simulator computing efficiency [4]. Meanwhile another one suggested using an industrial standard PC architecture running LINUX, and a network interface which supports a wireless TCP/IP Ethernet link. The robot is the NOMAD-200 that has a server workstation that runs on UNIX, and an industrial standard PC that runs on LINUX [5]. Reference [6] used relay infrastructure to enable the Robotic arm for connection via the internet. Relay infrastructure enable secure access to embedded systems behind a firewall or NAT. However it faced the problems of delay and complexity. The work presented in this paper use one integrated platform to interface with the internet, control, and monitor the performance of a Robotic Arm using only an App in a smart phone.

## **2.1 Internet of Things**

The Internet of Things (IoT) expression is the most recent trend in the IT world, where the physical and digital fields are connected by Internet of Things as a whole system [7]. IoT is the network of physical hardware of devices, vehicles, and other things supported with electronics, sensors, actuators, software, and internet network connectivity which permit the interaction between the embedded systems and enable the exchange of data between these hardware.[8][9][10] The embedded computing system distinguish each thing by features that differentiate it from other things, however that does not affect their ability to interact within the existed internet infrastructure. By 2020, expert's estimation suggests that the IoT will witness the inclusion of 30 billion objects [11].

## **3. METHODOLOGY**

The basic function of a Robotic arm is to simulate a human hand to perform certain type of tasks maybe regarded as risky or monotonic. A Robotic arm is mechanical, usually programmable system. In this section the mechanical and electrical design and material will be discussed.

### **3.1 Mechanical Design**

In this project a one degree of freedom Robotic arm is equipped with a servomotor which is a short for "servomechanism". This automatic device uses a position feedback mechanism to ensure that desired output is achieved. The servo is controlled by a micro controller (Arduino Uno R3). The robot has a square base with a dimensions of 15.2mm side length and a height of 3.2mm. The base consist of two

aluminum plates with 0.6 mm height each, and the two plates are connected together by a tubing with 2mm height as shown in Figure 1. The robot degree-of-freedom is powered by the servomotor. The basic structure of the Robotic arm is made of aluminum due to its ease of formation, cheap price. Also it is strong enough to bear the weight of motor and movement. The Robotic arm joint is constructed using servo brackets that are made of aluminum.

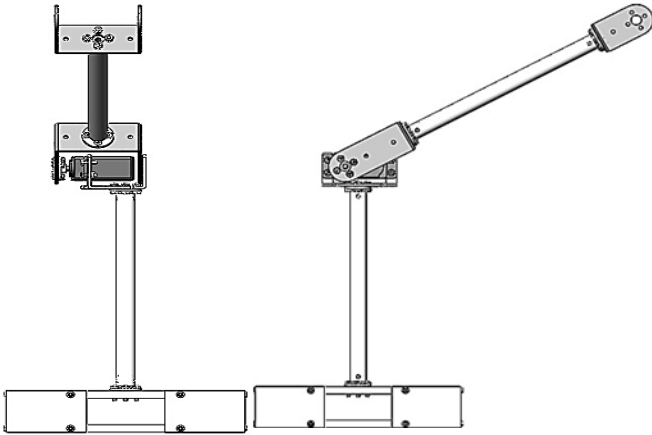


Fig 1: Robotic Arm Design

### 3.2 System Design

A Robotic arm is not an easy system to be built. To design it, several parameters need to be taken under consideration, and the basic most important one is the Degree of Freedom (DoF). Each joint on the Robotic arm is a degree of freedom.

### 3.3 Calculation of force

In order to design a robotic arm, the exerted force at each joint should be calculated to select the required motor. The selected motor should be able of handling not only the weight of the Robotic arm but also the weight the Robotic arm would carry [7]

The connection between the joint and the base is cold link1 and the other one with the holder is called link2 as shown in Figure 2(a). After labeling the FBD and the involved parameter which are the weight (W1) and length (L1) of the linking tubing and the weight end holder (W2) see Figure 2(b).

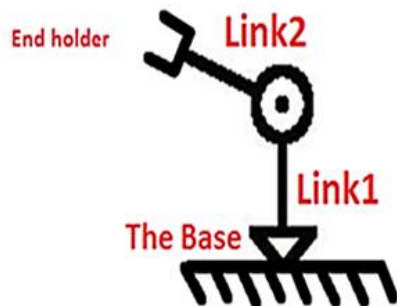


Figure 2: Robotic arm design - Free body diagram

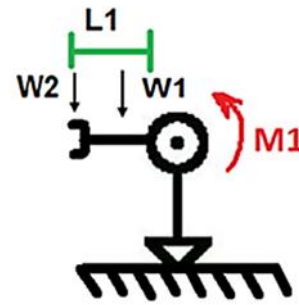


Figure 3: Calculation of the moment

The Moment (M1) is calculated multiplying the downward force with the length of the linking tubing as shown in Figure 2. The centre of mass of the linking tubing is assumed to be length / 2.

$$M1 = (L1/2) \times W1 + L1 \times W2 \quad (1)$$

Where:

M1= The Moment at the shoulder joint

W1 = the weight of the linking tubing Link1 = 20 g

W2 = the weight of the end holder = 19 g

L1 = the length of the linking tubing Link1 = 6 " = 15.24 cm

Torque about the joint:

$$M1 = ((15.24/2) \times 20) + 15.24 \times 16$$

$$M1 = 396.24 \text{ g.cm}^2$$

The servo motor have been selected based on the above calculations. The TowerPro MG995, This servomotor has a torque of 9.4kg/cm at a voltage of (4.8v), and 11kg/cm at (6v).

### 3.4 Electrical and electronic design and materials

The electrical system component depend mainly on the requirements of the other systems components such as the actuators voltage, amount of current draw, sensors used, size and weight of the system parts.

The electrical system provides power and collects data with sensors and outputs the power that drives the other systems. It gives the microcontroller the ability to exert control on the rest of the robotic arm, so it likes the nervous system of the robot. The block diagram as shown in Figure 3 presents the main structure of the robotic arm.

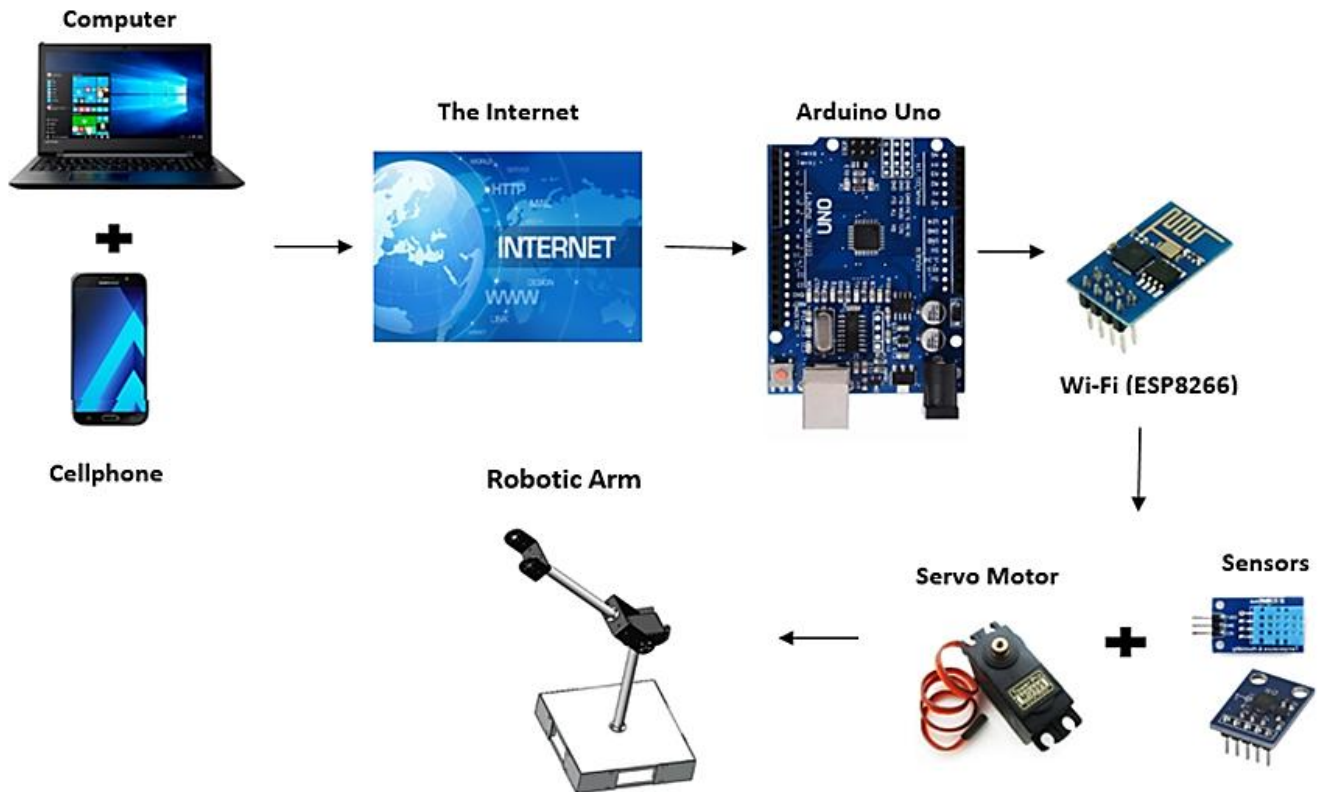


Figure 3: The main structure of the IoT controlled Robotic arm

Basically, this robotic arm has 3 outputs which is the arm movement generated the servomotor, temperature and humidity sensor (DHT11) output, and the accelerometer (ADXL355) reading. Also the Arduino Software (Integrated development Environment) IDE It is an open source electronic platform that allows the developer to write programs and upload them to the board used. The development and programming window allows the user to write new coding or modify an existing one; the language use is a set of C and C++ functions.

### 3.5 The Blynk IoT platform

A Blynk platform is used to fill up the gap between the devices sensors and actuators and the data network. The Blynk is one of the drag and drop platforms that was designed for the Internet of things. It allows the user to prototype and remotely control the hardware, display the output data of the sensors, and also store and visualize data. The Blynk IoT platform consist of three component which are:

1. **Blynk App** – provide the user with the ability to create easy interface for any project using the various widgets available in the App.
2. **Blynk Server** - this server allows the communications between the hardware and the smart phone.
3. **Blynk Libraries** - for all the popular hardware platforms enable communication with the server and process all the incoming and out coming commands.

### 3.6 The IoT protocol of controlling and monitoring

This system has two main parts which are the hardware and the software. In this project, the Arduino Uno is the controller of

the entire system. The Arduino Uno is connected to the internet via a Wi-Fi model (ESP8266). The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack, in simple words the Robotic arm system can remotely be controlled via Wi-Fi with a Smartphone using an App called the Blynk. The below steps will illustrate the set up and connection of the Blynk IoT platform.

- A. **Downloading the Blynk App:** The Blynk App is downloaded on the Smartphone from the play store for Android, and App Store for IOS.
- A. **Create a new Blynk account:** After downloading the Blynk App, an account is created. To do so, an E-mail address and a password are required. After signing up and successfully being logged in, a new project feature is activated.
- B. **Create a new IoT project:** To do that, the New Project plus button is pressed. Pressing the new project button activates the selection of the project hardware, and also gives certain options such as naming the project and so on.
- C. **Hardware selection:** In this step the hardware to be connected to the platform are selected, and by hardware it means the microcontroller used in the project, and for this project an Arduino board is being used. After selecting the required hardware the connection type is selected as well, whether it is a USB, Bluetooth, or Wi-Fi, after that the create button is pressed to continue.
- D. **Auth Token:** Is a unique identifier which is needed to connect the hardware to the smartphone. Every new project being created has its own Auth Token; it sent automatically to the Blynk account email address.
- E. **Add widgets:** First the project platform is empty, tapping on the widgets box displays the widgets list. The required widgets are selected from the list and added to the

platform. The slider widget is used to control the servomotor angular position. And the display widgets are used to display the sensors output data.

F. **Run the project:** when the settings are done, the PLAY button is pressed, switching the mode from EDIT to PLAY, where the developer can interact with the hardware.

#### 4. RESULTS AND DISCUSSION

In order to build a smart thing, it should be provided with the ability to sense and communicate. Boosting the abilities of a robotic arm by adding the IoT technologies to the system resulted in IoT controlled robotic arm. Figure 4 shows the completed IoT controlled Robotic Arm and the Blynk IoT platform for controlling and monitoring.

The dynamic part of the Robotic arm has been connected to the servomotor shaft using U shape aluminum holder to generate motion. The Robotic arm is connected to the Arduino board. The communication between the hardware and the software is made by programming the microcontroller. Uploading the coding to the Arduino is like sending the commands to the arm to move or to sense temperature, humidity, and acceleration. Furthermore, the communication ability is not limited to that only, a Wi-Fi model (ESP8266) have been added to the system, in order to connect it to the internet.

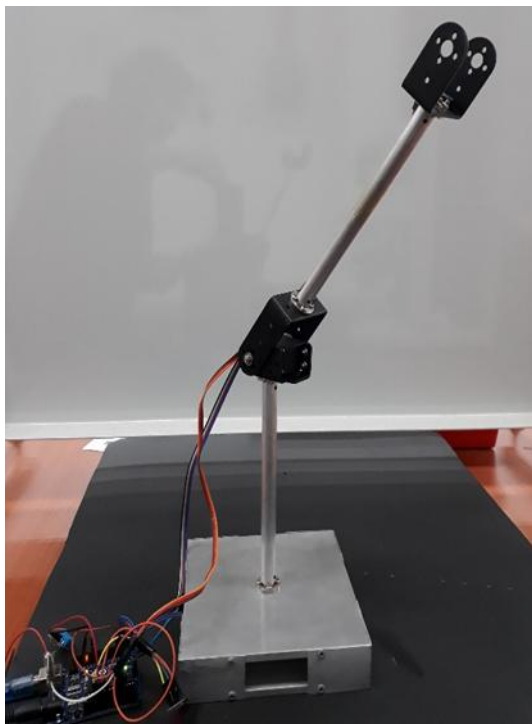


Figure 4: IoT Controlled Robotic Arm

To communicate with the robotic arm a smart phone has been used with an IoT platform that provides the means to control and monitor the performance. The Blynk App which can be found in the App store for the IOS device or the Play store for the android devices. Figure 5 shows the design and look of the Blynk IoT platform where the developer can add and use the widgets to operate hardware's such as motors and LEDs or display the data of sensors on screen.

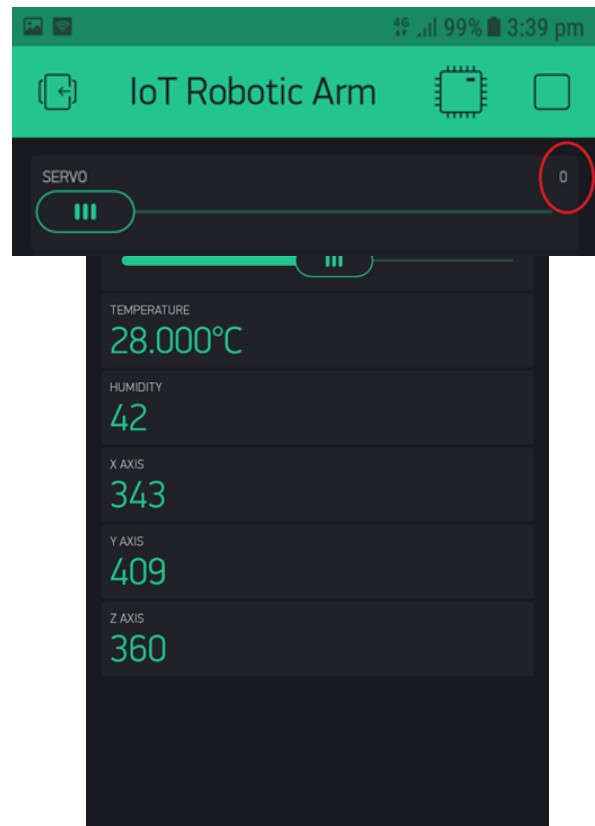
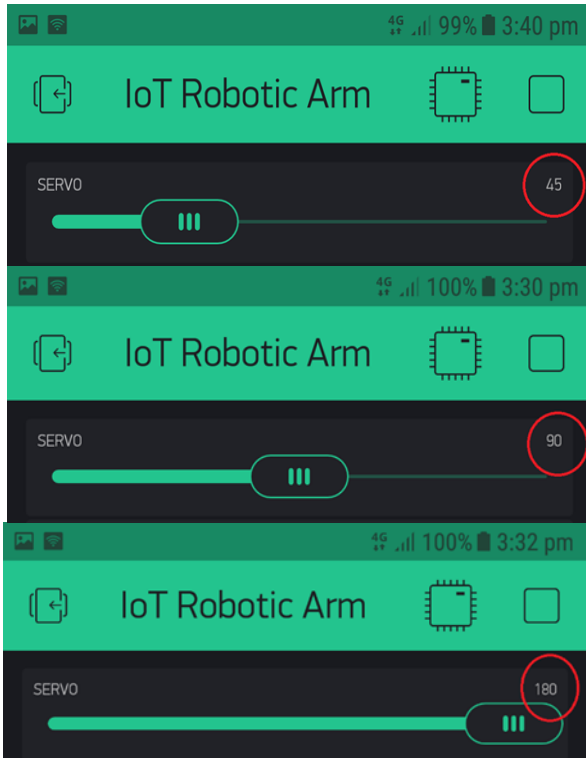


Figure 5: The IoT phone App platform for monitoring and controlling the robotic arm

#### 4.1 The Result of Connecting the Robotic Arm to the Internet

On the Smartphone, and precisely at the IoT platform, a slider widget has been attached to the virtual pin V3 for the servo. To virtual pin V6, a labeled value widget has been attached for temperature reading display. For humidity readings display, a value display widget is attached to the virtual pin V5. For the accelerometer X, Y, and Z outputs, a value display widgets are added to virtual pins V7, V8 and V9 respectively for displaying the three axes reading on the Blynk IoT platform.

The slider widget is used to control the arm actuator, it is set with a range of ( $0^{\circ}$  -  $180^{\circ}$ ) that reflect the angular degree of motion of the servo motor. The slider on the Blynk platform send a value to the Arduino UNO board via Wi-Fi, the value will be written on the virtual pin V3, and then the servo will rotate that much value of degrees. The display will show the rotation angle of the servo, which is from  $0^{\circ}$  to  $180^{\circ}$  as shown in Figure 5. There are two ways of displaying the sensor data to the Blynk App, the first one is the "Pull" and in this way, the data will be requested by the Blynk App only when it is open. The second is the "Push" technique and in this way, the hardware will keep sending the data to the Blynk cloud, regardless of the App is open or not. The user should decide the best way of sending the data according to the project requirements. In this project, a push technique is used. Sending the sensor's output data to the Blynk cloud in intervals through the virtual pins.



**Figure 6: Blynk App – IoT platform – Servomotor angular position**

**Table 1: Servo motor angular position**

No.	Slider Position M	Servomotor Angular Position
1	M1	0
2	M2	45
3	M3	90
4	M4	180

Positioning the slider will rotate the servo motor as tabulated in Table 1, which leads to moving the Robotic arm up or down, it depends on the required angle. Rightward sliding of the slider in the IoT platform leads to downward movement of the Robotic arm, and hence, leftward sliding results in upward movement.

The IoT platform allows the user to control the hardware remotely and from anywhere in the world as long as the Robotic arm is connected to the internet and the Smartphone is also connected to the internet.

## 4.2 Accelerometer Sensor (ADXL335) Output Data Display Using IoT

The accelerometer sensor was used to measure the acceleration or vibration of the Robotic arm during the motion. However, in the standstill position the acceleration sensed by the ADXL335 is due to the gravity pulling it downward. The accelerometer reading collected by positioning the Robotic arm in different angles from the ground as displayed in the Blynk IoT platform. The acceleration measured is a dynamic acceleration resulted from motion. The ADXL335 measures acceleration along X, Y, and Z axes see Figure 6 and gives analog voltage output proportional to the acceleration along these three axes. Changing the angular position of the servo motor resulted in the tabulated data shown in Table 2.

**Table 2: ADXL335 sensor X, Y, and Z axes readings Angle vs acceleration**

Angle	X	Y	Z
0	392	448	432
10	340	398	385
20	342	398	383
30	342	394	388
40	341	398	383
50	341	399	381
60	341	400	379
70	342	396	385
80	344	396	306
90	344	410	342

The accelerometer was tested in two positions, the first one while the arm in standstill position. The second test was made during the sweeping test of the servo motor. After performing the sweeping test on the Robotic arm, the accelerometer outputs of the X, Y, and Z axis values as tabulated in Table 2

The Robotic arm is programmed to perform any task that requires a routine movement with a delay to be set through the Arduino coding. The delay changes the speed of rotation; it is adjusted to suit the functionality of the servomotor. A direct connection is made between the servomotor and the Arduino board, and in order to determine the position of the shaft, the servomotor uses feedback. As a result the Robotic arm movement is controlled accordingly.



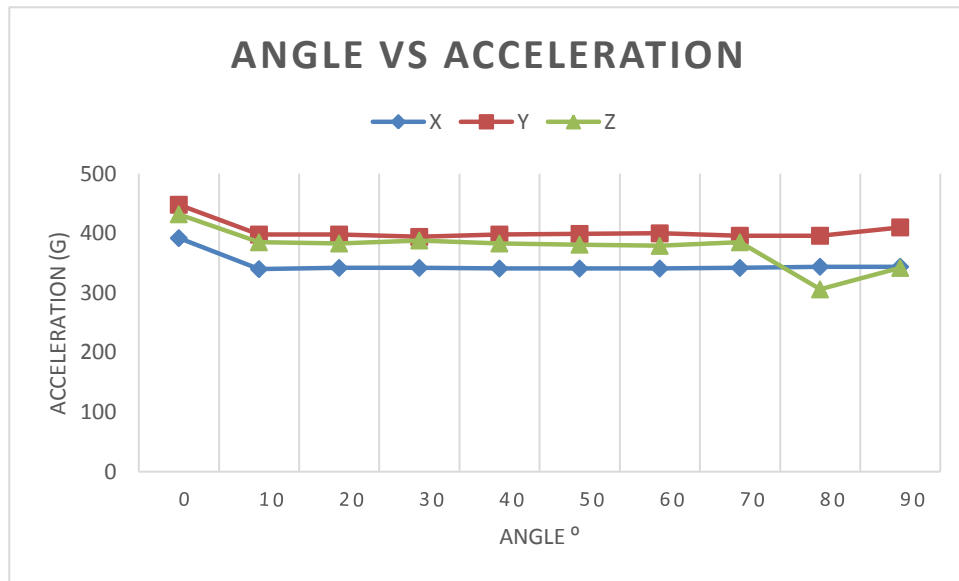


Figure 7: Angle Vs acceleration

From Figure 7 there is an initial rising edge showing and it is due the startup delay in the accelerometer. The acceleration data is mostly constrained between (300g – 450g) for the z-axis. The most acceleration (motion) activity is clear in the z-axis, due to the accelerometer position and the vertical movement of the Robotic arm.

### 4.3 The DHT11 Temperature Readings Validation

The accuracy of the IoT platform is tested experimentally through the calibration of the temperature readings measured by the DHT11 sensor, and displayed by the Blynk IoT platform. Using an infrared (IR) thermometer with wireless PC interface. Using an external heating source the temperature was raised and the readings of both the IR thermometer and the IoT platform were identical except for a slight difference of (0.5°C) between the two readings as shown in Figure 8.



Figure 8: Temperature readings (IoT platform Vs IR thermometer)

## 5. CONCLUSION

First, building a smart robotic arm is accomplished by using a microcontroller (Arduino board) to communicate and sensors to sense. The first part of the project was accomplished successfully, the Arduino board made that applicable. By uploading the right coding to the Arduino, the smart Robotic arm was able communicate and the sensors output data were displayed through the serial monitor of the Arduino IDE. Merging the two domains the robotic and the IoT in one project to achieve the second part, resulted in creating one integrated system that combines the abilities of the robotic applications and the availability and pervasiveness of the internet network. The Blynk IoT platform used in this project as so many other IoT platforms provided an easy secured communication with the hardware of the Robotic arm system via Wi-Fi. The developed IoT controlled Robotic arm has an expandable system, more sensors and actuators can be added to the system.

## 6. RECOMMENDATION

The robotic arm developed in this project has many features; however it can be improved according to the user requirement. More degrees of freedom can be added to the robotic arm. A camera can be added for security applications, and mobility feature can be added to test the capability of both the robotic arm and the IoT platform.

## 7. REFERENCES

- [1] Information Society and Media Directorate general of the European Commission and the European Technology Platform on Smart Systems Integration. [Accessed on 27 Oct 2017]. Available: [http://www.smart-systems-integration.org/public/documents/publications/Internet-ofThings\\_in\\_2020\\_ECEPoSS\\_Workshop\\_Report\\_2008\\_v3.pdf](http://www.smart-systems-integration.org/public/documents/publications/Internet-ofThings_in_2020_ECEPoSS_Workshop_Report_2008_v3.pdf).
- [2] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, Interacting with the so a-based internet of things: Discovery, query, selection, and on demand provisioning of web services, Services Computing, IEEE Transactions on 3 (3) (2010) 223–235.
- [3] D. Guinard, V. Trifa, F. Mattern, E. Wilde, from the internet of things to the web of things: Resource-oriented

- architecture and best practices, in: *Architecting the Internet of Things*, Springer, 2011, pp. 97–129.
- [4] Cuno Pfister, *Getting Started with the Internet of Things*, May 2011: First Edition
- [5] Tse Min Chen and Ren C. Luo, "Multisensor Based Autonomous Mobile Robot through Internet Control", Department of Electrical Engineering, National Chung Cheng University.
- [6] Internet Controlled Robotic Arm, International Symposium on Robotics and Intelligent Sensors, 2012 (IRIS 2012).
- [7] Ali A. Abed, "Internet of Things: Architecture and Design", IEEE Xplore, 2016.
- [8] Brown, Eric (13 September 2016). "Who Needs the Internet of Things?". Linux.com. Retrieved [27 November 2017].
- [9] Brown, Eric (20 September 2016). "21 Open Source Projects for IoT". Linux.com. [Retrieved 27 November 2017].
- [10] "Internet of Things Global Standards Initiative". ITU. [Retrieved 27 November 2017].
- [11] Nordrum, Amy (18 August 2017). "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated". IEEE.
- [12] Robotic arm Instructables, available <http://www.instructables.com=id=diy-rob>.