

# Exploring IOT Application Using Raspberry Pi

Ayush Prabhu

Roll no: 24

Student, Dept. of Information Technology, Keraleeya Samajam (REGD.) Model College Maharashtra, India

**Abstract** – There are thousands of sensors in an industry with different usage, such as, pressure transmitters, flow meter, temperature transmitters, level transmitters, and so on. Wired networks are mainly used to transfer data to base station by connecting sensor. It brings advantage as it provides reliable and stable communication system for instruments and controls. However, the cost of cables necessary is very costly. Therefore, recently low cost wireless networks are strongly required by customers, for example, temporary instrument networks and/or some non-critical permanent sites which require low data rate and longer battery life. In client/server model, file server act as a parent's node which allow multiple child node to connect with it. It is responsible for central storage and data management so that other computers enable to access the file under the same network. This article explores the use of Raspberry Pi to function as a server in which several laptops are connected to it to copy, store and delete the file over network. IT requires authentication for user login before granting access to the file to ensure data integrity and security. File server is widely used in many areas, for example in education for uploading study note into the serve and student immediate downloading it into their computer. Moreover this work also explores the use of Raspberry Pi B+ model and XBee (ZigBee module) to demonstrate wireless communication data transmission, proving the validity of usage as a mobile low-power wireless network communication. The main goal of the research is to explore the use of Raspberry Pi for client-server communication using various wireless communication scenario such as Wi-Fi and ZigBee.

**Keywords** – client/server model, Raspberry Pi, XBee.

## 1. INTRODUCTION

Internet of Things (IoT) can define as interconnection between people, animal or object that ability to exchange data over network without involving human-to-human or humantocomputer interaction. IOT offer various kinds of connectivity from devices, systems, and services that work within machineto-machine communications (M2M) and cover with applications, domains and protocols. [1]. Nowadays, there have many implementations of IoT devices, for instances, heart monitoring implants, automobiles built-in sensors, farm animals equip with biochip transponders, field operation device used by fire-fighters in search and rescue etc. [2]. In current market the implementation of smart thermostat systems and washer/dryer for remote monitoring by using Wi-Fi [3].

According to McKinsey Global Institute, IoT has wider market and potential to create economic impact of \$2.7 trillion to \$ 6.2 trillion annually by 2025[4]. IoT is not futuristic or aspirational technology trend; instead it already exists in our devices, cloud infrastructure, data, and sensor as well as business intelligence tools. Microsoft Corporation also delivers unique and integrated approach for all enterprises to capitalize on IoT by collecting, storing and processing data. This method was extends from broad product portfolio such as PC, tablets and industry devices on edge of enterprise network to backend system and services develop tool and diverse partner ecosystem [5].

In 2007, there was a bridge collapsed in Minnesota [6], the cause of this tragic accident is steel plates were inadequate to handle to bridge's load. If we are implementing smart cement which equipped with sensor to monitor cracks, stresses and war pages, the tragedy sure can be escaped. Therefore, IoT should be further exploring as object can represent itself digitally or becomes more powerful than it. It known as "ambient intelligence" when there are many object act in unison [7].

### 1.1. Raspberry Pi

The Raspberry Pi is a low cost, small and portable size of computer board. It can be used to plug-in to computer monitor or television, keyboard, mouse, pen-drive etc. Raspberry Pi has built in software such as Scratch which enables users to program and design animation, game or interesting video. In addition, programmers can also develop script or program using Python language; it is main core language in Raspbian operating system [8]. Raspberry Pi B+ is evolution of Model B. Python language has been used in this work to write the script for client/server communication. Moreover, there are improvements such as adding more GPIO header PIN, more USB ports, lower power consumption etc. It is recommended to use model B+ for school learning because it offers more flexibility than model A especially for embedded projects and require low power as well as providing more USB ports compared to Model B [9].

Samba is open source software that provide seamless file and print service to SMB/CIFS clients [10]. This software is considered a good and user friendly development because it allows interoperability between Linux/Unix server and Window-based clients. Samba server consists of two key programs which is smbd and nmbd [11]. The four main services

provided by Samba included: file and print services, authentication and authorization, name resolution and service

simultaneous applications used by users. It enables dynamic run-time changes in application and switch between

announcement. In this project, samba is use for as a server in raspberry pi for remote file access with multiple computers.

### 1.2. ZigBee

ZigBee refers to high-level communication protocol which can be used to build a personal area networks (PAN). It is a small and low-power digital radio that based on IEEE 802.15.4 standard. Due to its low power consumption, the transmission distance is up to 10 – 100 meter line-of-sights. However, it increases the output power and less resistance in signal [12].

ZigBee is mostly used in low data a rate application that does not require high power consumption and secure transmission. ZigBee networks use of 128 bit symmetric encryption keys [13]. ZigBee has pre-defined rate of 250 kbit/s, which is best for intermittent data transmissions from a sensor or input device.

There is various kind of existing application such as wireless light switches, traffic management system, and consumer equipment etc. are use of short-range low-rate wireless data communication such as ZigBee. ZigBee is considered better than other technology (Bluetooth or Wi-Fi) because it is simpler and less expensive [14].

## 2. RELATED WORKS

Megat N.M. Mohamed Noor (2013) has conducted work on community based home security using wireless mesh network [15]. In this work, the author builds a prototype for home security system that is based on wireless mesh network. The system is considered cost effective and scalable. It is considered a community based application as it uses open mesh wireless router and raspberry PI device and the deployment is shared among house residential. The raspberry Pi B+ board is interfaced with PIR sensor to detect movement. Moreover, it uses Filezilla server to transfer captured image from Raspberry Pi to host over TCP network. The results show the benefit of implementing this system over home security as it produces a clear human face detection and recognition. The system can also be expanded by deploying additional open mesh wireless router, camera and sensor node.

A research on “An agent-based middleware for decentralized dynamic data-gathering in wireless sensor networks” has been done by Haghighi, M. & Cliff, D. (2013) [16]. This article describes the design and development of Sensomax which is written in Java and run on network ranging from limit resource as Sun Spot nodes to resource rich Raspberry Pi board. Sensomax facilitate fully distributed and decentralized bulk programming and update of sensor node to serve multiple

timedrive, data-driven and event-driven operational model. Sensomax are able to operate and provide response to application request without interrupting outgoing network operation.

Cagnetti, M. et. al. (2013) has conducted work on “A New Remote and Automated Control System for the Vineyard Hail Protection Based on ZigBee Sensors, Raspberry-Pi Electronic Card and WiMAX” [17]. The author proposed systems that perform action to protect vineyard and relay message to remote control. For the overall design, an “umbrella” solution is used in which ZigBee sensor used to send the data to central control unit (Raspberry Pi). However, WiMax are chosen because it overcomes both distance limitation of Wi-Fi and the absence of ADSL. They provide result to prove that the system is user friendly and effective to protect over hail and maintenance cost.

## 3. METHODOLOGY

In this project, Raspberry Pi is used as a server by connecting to several laptops as clients in the same network. The work was explored in three different scenarios, i. client-server communication using Wi-Fi, ii. Client-server communication using ZigBee and iii. Client-server communication within localhost for remote file sharing facility. Router is used to create a wireless network communication between the devices. To create remote file sharing environment, wireless adapter and router is used and 2 units of XBee S1 series ZigBee communication is used to explore an alternative wireless communication platform. In order to carry out the research, several hardware and software platforms were required as shown below.

### 3.1. Hardware List

Raspberry Pi B+: Small and portable board that can perform as a computer. It acts as a file server in this work.

SD card: Raspbian OS in SD card

Asus WL-167G V3: Wireless adapter for Raspberry Pi Wifi connection.

Wireless router: Provide wireless environment for file access between client and server.

Computer: Client model that gain remote access to raspberry Pi.

XBee: It used connect to Raspberry Pi to act as data transmitter and another XBee connect to Window to act as receiver.

### 3.2. Software List

- Raspbian OS: It is the official image for RaspberryPi, software development for this OS includes python language.
- Window 7: is used by client model to gain remote access to file server. It can store, copy and delete file in server.
- Samba: is an open source server mainly for file transfer.

### 3.3. Client-Server Communication using Wi-Fi

In this experiment, client-server communication is explored by transferring file wirelessly to the server using Wi-Fi communication. Raspberry Pi supports wireless communication by allowing connection to wireless adapter. In order to set up a server in Raspberry Pi, samba software is used by setting configuration such as permit user to read only, write only or read write access. Part of the configuration in samba is shown below:

```
[PiShare] comment=Raspberry Pi Share path=/home/pi/share
browseable=Yes writeable=Yes only guest=no
create mask=0777 public=no path: is the folder
location for sharing file. browseable = Yes: enable client to
search for the file. writeable = Yes: enable client to modify the
file. public = no: whenever client want access to the shared
folder must login with a valid user. create mask = 0777: allow
directories/folders created within the parent have read/write
permissions. only guest = no: refer for authorized user only.
```

Figure 1 shows the overall system architecture of client server files sharing in which laptop (client) can remote access to the file in raspberry pi (server). Figure 2 shows raspberry Pi connected with different components such as power source, HDMI to VGA cable and wireless adapter.

Figure 2 Raspberry Pi connection with different component.

### 3.4. Client-Server Communication using ZigBee

ZigBee is a short range communication technology that is very suitable for low data rate short distance communication forming a mesh network. Raspberry Pi has features to allow for wireless communication using ZigBee. The following experiment shows how Raspberry Pi can be configured for wireless

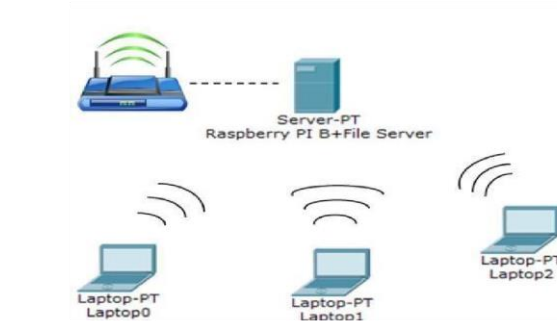


Figure 1 System Architecture of client/server files sharing



communication for short range file transmission. Together with ZigBee, XCTU is required to view ZigBee connections and range test. 3.4.1. Setting up XCTU in Windows

The XBee manufacturer, once called maxstream, now 'Digi', built software called X-CTU and it works on all versions of windows. This software can be downloaded for free from: <http://www.digi.com> [18]. Figure 3 shows the basic installation of XCTU.



Figure 3 Installation of XCTU

The installation process for this program is just like any other, so can download and get it installed in the laptop or desktop. Next, two XBee and a Raspberry Pi is used to setup the



communication environment. Figure 4 shows wireless sensors connected to XBee (Unit A) through Cytron board, whereas another XBee is connected to the laptop as shown in Figure 5. Figure 6 shows raspberry Pi connected with different components such as XBee and mouse.

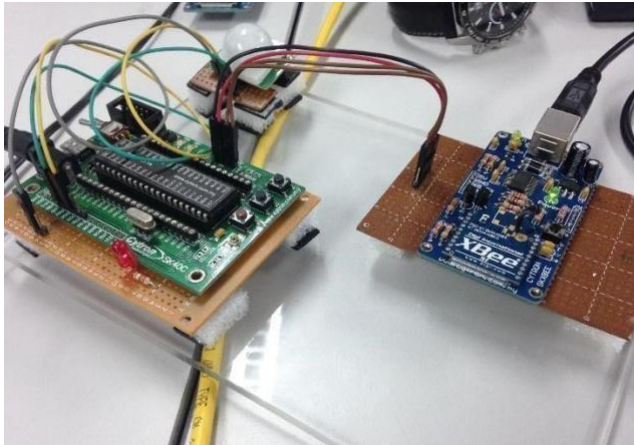


Figure 4 Wireless sensor connected through Cytron board to XBee (Unit A) for communication



Figure 5 XBee (Unit B) to be connected to Laptop

After designating the XBee to PC interface and building up the hardware, testing need to be done to make sure the PC can talk to the XBee module. Figure 7 show the testing communication with XCTU.



Figure 6: Raspberry Pi model B+ connected to

HDMI, Mouse, Keyboard, LAN and XBee (USB)

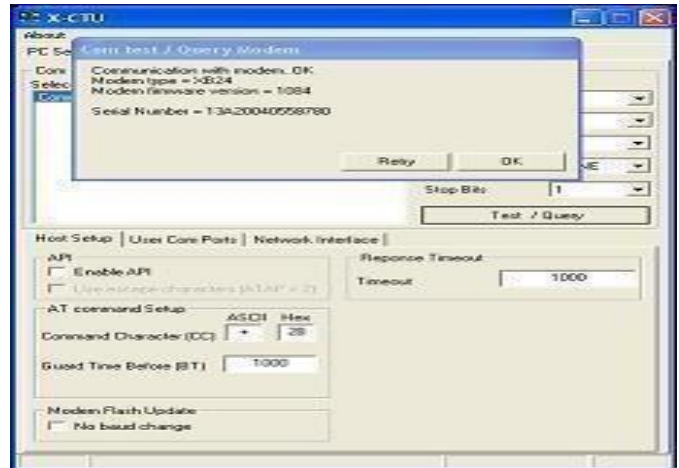


Figure 7: Testing XBee communication with XCTU 3.4.2.

### Setting up Minicom in Raspbian OS

X-CTU cannot be installed on Raspberry Pi since Wine does not support ARM processors. In order to communicate XBee with the Raspberry Pi, Minicom must be installed and configured correctly [19]. The tool Minicom can be downloaded through Aptitude. After installation, the following steps are required to configure the XBee.

- Upon plugging in the ftdi cable+adapter/xbee module, it is detected in /dev as ttyUSB0, so make sure to use ttyUSB0 below.

Using minicom, the steps for getting serial communication with the XBee are as follows.

- Enter this command: minicom --device /dev/ttyUSB0
- Can see the minicom screen. As prompted, hit Ctrl-A and Z to enter setup. (To quit at any point, hit Ctrl-A and Q.)
- Hit "o" and <enter> to 'configure minicom'. From the next screen, select 'serial port setup'.
- On the port setup screen, choose 'E' to set 9600 8N1
- Hit <enter> and return to port setup screen. Toggle F to set hardware flow control to off.
- Leave port setup and go to 'modem and dialling setup'. Hit A and remove all the characters from init string; hit B and remove all characters from reset string so those fields are empty.
- From configuration menu, select 'save setup as dfl'.
- Finally, return to the main minicom menu ('Command Summary').
- Hit 'A' to add linefeed then return and hit 'E' to add local echo. Their actual toggled status is not displayed. Since linefeed and echo are not saved in default

settings, these need to be changed each time minicom is used.

Now minicom main screen can be used as a serial terminal to send commands to the XBee device. Type '+++' and should get 'OK'. The communication only stays open for a few seconds. If more time elapses, enter '+++' again to re-open. If type 'atid' then hit enter, it should return your device's default PAN id. From there it is ready to use AT commands to configure the device on the serial terminal. Figure 8 shows testing connection and checking firmware and communication channel (unit A).

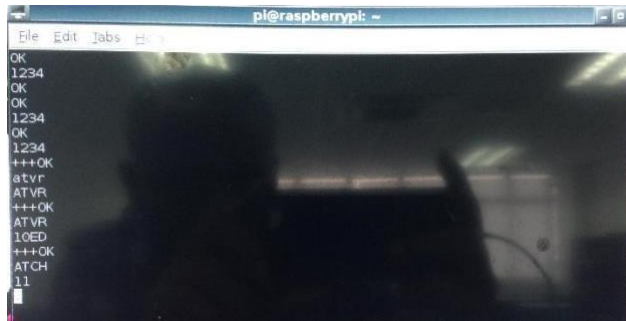


Figure 8: Testing connection and checking firmware and communication channel (Unit A)

### 3.5. Client-Server Communication within Localhost

In this scenario, python script is written for communication between client and server within localhost [20]. It uses raspberry pi as a medium in this communication. Client will send a request message to the server to get some information, then server will process it and reply the request message to the client. Figure 9 shows the architecture of client server communication within localhost.

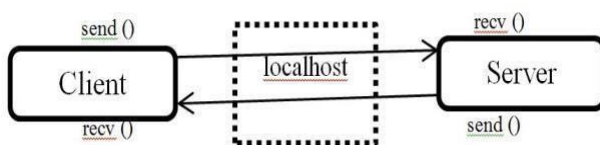


Figure 9 client server communication within localhost

- Code below shows part of server configuration:

```
PORT = 8018
TIMEOUT = 5
BUF_SIZE = 1024
# set up socket
sock = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
sock.setsockopt(socket.SOL_SOCKET,
socket.SO_REUSEADDR, 1) sock.bind((HOST, PORT))
sock.listen(5) print '-= WhatsUp Server'
```

```
== ' print '>> Listening on:', PORT print
'>> Author: Raspberry Pi' print
"
```

- Code below shows part of client configuration:

```
HOST = '127.0.0.1'
PORT = 8018
TIMEOUT = 5
BUF_SIZE = 1024 def __init__(self,
host=HOST, port=PORT):

self.sock
=
socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
self.sock.connect((host, port))
logging.info('Connecting to %s:%s' % (host, port))
while 1: try: buf = self.sock.recv(BUF_SIZE)
sys.stdout.write(buf) cmd = raw_input()
if cmd.strip() ==
'!q': sys.exit(1) self.sock.send(cmd)
except: self.sock.close()
```

- 
- 
- 
- 
- 

## RESULTS AND DISCUSIONS

### 4.1. File server communication result

After configuring raspberry server site, windows users can now connect to the RASPBERRYPI network. It is necessary for users to login before they can browse the file as it has been configured in the server site. This is important to improve data integrity and privacy of the file, so that unauthorized users cannot gain access. Figure 10 shows login authentication for RASPBERRYPI network. Figure 11 shows the file that is stored in the RASPBERRYPI shared folder such as images, documents and drivers.

Figure 11 File inside RASPBERRYPI shared folder

### 4.2. ZigBee communication result

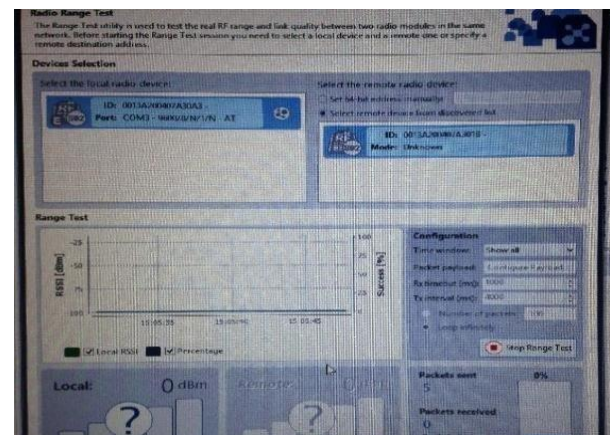




Figure 10 Login authentication for RASPBERRY PI network

Figure 12 Running XCTU showing locally connected XBee unit B on the left and wirelessly detected XBee (Unit A) on the right.

After completing all the setup and configuration of both XBee units on Raspberry Pi (Raspbian OS) and the laptop (Windows OS), the following test are conducted. Figure 12 shows XCTU program running, and showing locally connected XBee (Unit B) on the left panel and wireless detected XBee (Unit A) on the right panel.

After detecting wireless XBee, XBEE (Unit B) is used to send test packet to XBee (Unit A) as shown in Figure 13. Figure 14 shows test packets are generated and being converted to hexadecimal form in order to be transmitted to XBee(Unit A). The original text or plain text of test packets are shown in Figure 15.

Figure 15 Sample packet content in XCTU

To ensure reliability of data transmission and to avoid packet loss, XBee (Unit A) is set to receive packets before another XBee (Unit B) is transmitting message. Figure 16 shows the data flow control of XBee unit in which XBee (Unit B) tries to send message to XBee (Unit A).

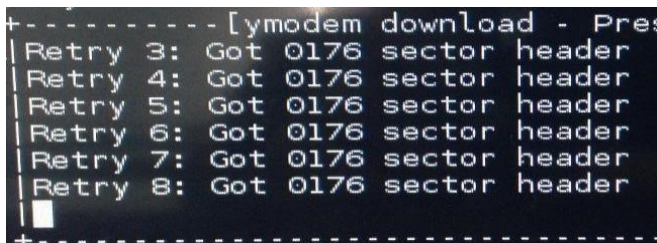


Figure 14 Test packet generated and converted to hexadecimal values in XCTU to be transmitted to Unit A

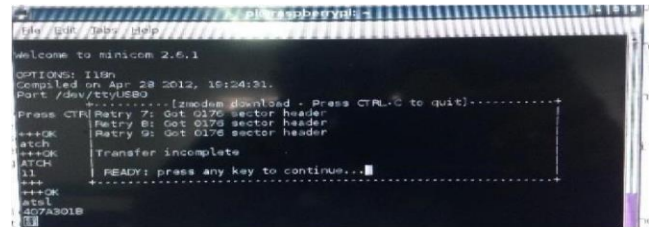


Figure 13 Unit A ready to receive test packets from Unit B

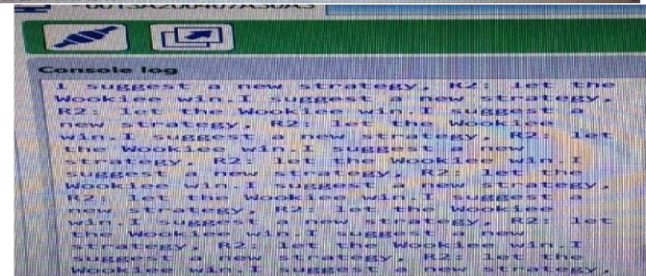
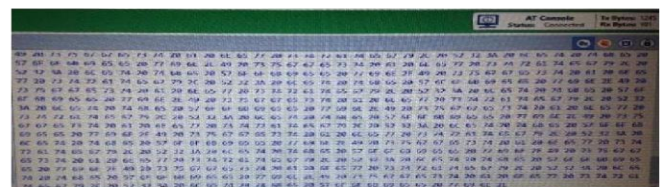


Figure 16 Data flow control of XBee unit.

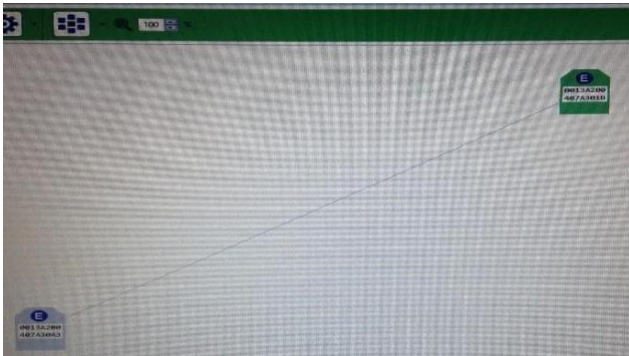


Figure 17 Network model of locally connected Unit B and wirelessly connected Unit A in XCTU



ID	Time	Length	Frame
558	15:16:08.074	16	Remote Command Response
559	15:16:13.909	5	AT Command
560	15:16:13.946	5	AT Command Response
561	15:16:13.946	4	AT Command
562	15:16:13.986	5	AT Command Response
563	15:16:13.987	4	AT Command
564	15:16:16.498	18	AT Command Response
565	15:16:16.499	15	Remote AT Command Request
566	15:16:16.849	5	AT Command Response
567	15:16:16.859	17	Remote Command Response
568	15:16:16.859	15	Remote AT Command Request
569	15:16:16.921	17	Remote Command Response
570	15:16:16.921	15	Remote AT Command Request
571	15:16:17.000	16	Remote Command Response
572	15:16:17.001	15	Remote AT Command Request
573	15:16:17.060	16	Remote Command Response

Figure 17 shows the network model locally connected with Unit B and connected wirelessly to Unit A in XCTU program. The overall communication frame log between XBee is being recorded in XCTU program. Users can view the record in frame log as shown in Figure 18. Figure 20 Creating custom payload packet for range test

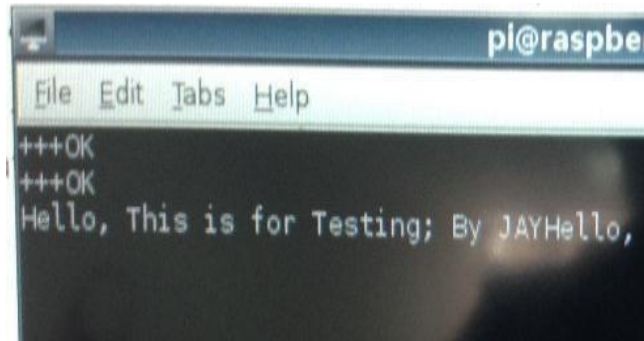


Figure 21 Unit A receives that packet from Unit B through ZigBee network

#### 4.3. Client server communication within localhost result

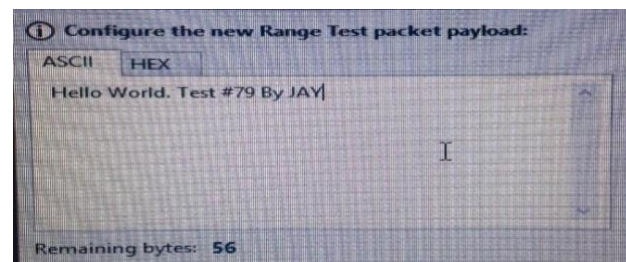




Figure 18 Frame log communication between Units A & B

Before completing the experiment, a range test was done for XBee. Range test is the amount of data packet can be sent in a message. Figure 19 shows the range test running and Figure 20 shows users create custom payload packet for range test whilst result is shown in Figure 21.

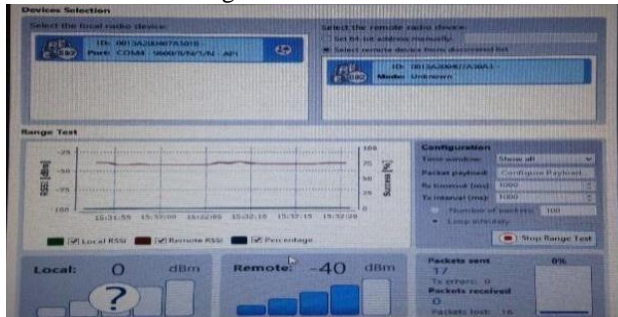


Figure 19 Doing a range test browser.

When client sends a request message to server localhost, the server processes the request and reply back to the client. Figure 22 shows client request is displayed in the localhost server through web browser.

## 6. CONCLUSION

Raspberry PI is useful for small application development because it can be used to integrate with many components such as speakers, LED lights, sensors, cameras and wireless communication units to develop smart applications. In this project, two ZigBee module is being communicated with two different devices (laptop and raspberry PI). With the successful communication implementation, it can be further developed by adding sensors to develop smart applications such as smart alarm system, population of customer walk-in and walk-out, temperature alert system and so on. This project also demonstrates how files are being shared between raspberry pi and windows users. Raspberry PI acts as server Figure 22 Localhost server display client request in web by using samba. Samba was chosen because it is free software and enables admin to configure for file access to protect the privacy or personal information of a company. It is very important to protect the file in order to maintain data consistency and

accuracy. File sharing is widely used ranging from small/medium sized company to big company. One of the reason is that, admin can easily manage the file and client can access the file efficiently. For future development, whenever user login failed more than three times it will generate sound and send a message to alert admin because there could be an intruder into the system. In conclusion, Raspberry Pi can be the future for smart applications and client-server communications. This project is a stepping stone to introduce the various features and possibilities available in Raspberry Pi and opens up an avenue for researchers who wish to embark into this new embedded invention. Raspberry Pi can also be used as an educational tool as it comes with Scratch as an animation tool for young learners.

## REFERENCES

- [1] J. Höller, V. Tsiatsis, C. Mulligan, S. Karmouskos, S. Avesand, D. Boyle: From Machine-to-Machine.
- [2] Margaret Rouse, (June 2014), Internet of Things (IOT), [ONLINE] Available: <http://whatis.techtarget.com/definition/Internet-of-Things>
- [3] Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., ... & Whitehouse, K. (2010, November). The smart thermostat: using occupancy sensors to save energy in homes. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (pp. 211-224). ACM.
- [4] Manyika J. et. al. (2013), disruptive technologies: Advance that will transform life, business and the global economy retrieved from McKinsey & Company [ONLINE] Available: [http://www.mckinsey.com/insights/business\\_technology/disruptive\\_technologies](http://www.mckinsey.com/insights/business_technology/disruptive_technologies)
- [5] Internet of Things (2014), retrieved from Microsoft [ONLINE] Available: <http://www.microsoft.com/windowsemerged/en-us/internet-of-things.aspx>
- [6] Libby Sander & Susan Saulny (2 August 2007), Bridge collapse in Minneapolis Kills at least 7, from New York Times [ONLINE] Available: [http://www.nytimes.com/2007/08/02/us/02bridge.html?\\_r=0](http://www.nytimes.com/2007/08/02/us/02bridge.html?_r=0)
- [7] Jack M. Germain (9 October 2014), The Importance of Openness to the Internet of Things [ONLINE] Available: <http://www.linuxinsider.com/story/81024.html?rss=1>
- [8] Anonymous, (n.d), Raspberry Pi [ONLINE] Available: <http://my.rs-online.com/web/generalDisplay.html?id=raspberrypi>