Development of Internet of Things (IOT) Based Electronic Reader for Medical Diagnostic System

M. N. Afnan Uda1,*, Asral Bahari Jambek1, U Hashim1,2, M. N. A. UDA3 and M. A. F. Bahrin1

1School of Microelectronic Engineering Group, Universiti Malaysia Perlis (UniMAP) Putra Campus, 02600 Pauh, Perlis Malaysia
2Institute of Nano Electronic Engineering (INEE), Universiti Malaysia Perlis (UniMAP), 01000 Kangar Perlis Malaysia
3School of Bioprocess Engineering Group, Universiti Malaysia Perlis (UniMAP) 02600 Arau Perlis

Email: nurafnan92@yahoo.com

Abstract. This paper is about an experiment for performing foodborne pathogens electronic reader using wireless sensing Internet of Thing (IoT). There are limited number of electronic readers for biosensors application with wireless internet connection. This research is to overcome the problem of commercial available electronic reader based on biosensor application method that only can be perform in offline or standalone device. This paper shows a complete system on how the data from electronic reader can be collected, easily understand by user and transfer data through the wireless internet connection via platform of IoT. There are three stages that is coding modification, android application development and transmit data to cloud storage. The NodeMCU microcontroller was used as a transfer medium for transfer data to internet. The Android Studio software was used for mobile application development. While, Arduino software was used to create a programming code to upload in NodeMCU microcontroller.

1. Introduction

Internet of Things (IoT) is an open and comprehensive network of smart objects capable of self-organizing, sharing information, data and resources, reacting and acting in the face of environmental situations and changes. The IoT in the IT arena is a novel paradigm shift. The phrase "Internet of Things" is coined from the two words, the first word is "Internet" and the second word is "Things". The Internet is a global computer network system using the standard Internet protocol suite to serve billions of users around the world. It is a global network system comprising millions of private, public, academic, business and government networks, ranging from local to global, connected by a wide range of electronic, wireless and optical networking technologies [1-2]. IoT is a technology that uses physical objects or ‘things’ embedded with communication capabilities, and facilitates intelligent data exchange among such devices, or between these devices and other internet-enabled systems. This is a new era of advanced technology, where each and everything is connected with each other and information can be accessed with existing of internet connection [3-5].

Currently, many of commercial electronic reader based biosensor application are standalone devices and do not have the capability to transmit information with internet connectivity [6-14]. Hence, the problem is the process of transmitting data from the electronic reader to the relevant party becomes slow...
Thus, no immediate action cannot be taken. In order to overcome this problem, the existing electronic reader-based biosensor need to be designed and improved the systems by embedded with the IoT system for improvement of connectivity capability of device. The use of wireless input and output terminal capable of accessing the internet to transmit and receive data from the electronic reader when sensors are attached to the electronic reader [17–19].

This research is proposed to design a smart electronic reader-based biosensor by embedding the IoT feature into the system. This project will focus on hardware and software according to electronic reader circuitry with ESP 8266 Wi-Fi module were used as microcontroller. Then, to design the algorithm to build a smart device application for user interface by using android apps maker such as “Android Studio”. The expected outcome of this project is to develop and introduce the electronic reader for biosensor application with IoT system. With this capability now the reading result of electronic reader can be access from the smart phone through mobile app. In addition, the result will be transmit to computer for display and record data. This system can develop a rapid news for taking an action in order to ensure the cleanliness and health awareness of pathogenic bacteria that caused food poisoning.

2. Material and methods
In the era of technology innovation, everything requires monitoring and controlling. Therefore, this research is about to offers an IoT framework for assisting to the electronic reader based on foodborne pathogens detection for food safety monitoring system in order to food seller serve the customers with quality and safe food. The system is focus on the rapid detection, the data can be transfer to the medical officer for the record and analysis the data for fast action. Hence, making the medical officer to manage the data in real time using personal computer or mobile phone. Figure 1.0 shows the complete system via Internet of Things (IoT).

![Figure 1. Complete system via Internet of Things (IoT).](image)

2.1. Coding Modification
The first step is to modify the current coding. It needs to enable Wi-Fi connection to push the data at cloud server that need internet connection. The NodeMCU ESP8266 can communicate in two different ways via Wi-Fi. It can be linked to an existing wireless hot spot or access point which same method the way people connected their phone or computer to the Internet. This is stated to as "station" mode. The NodeMCU can reach the internet in station mode and can also connect with other devices along with
other NodeMCU that have been connected to the same system. Figure 2.0 shows the example coding that the user needs to replace the SSID and password variables to wireless network ID and password.

```cpp
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
const char* ssid = ".......";
const char* password = ".......";
```

**Figure 2.** Coding Architecture Connect the NodeMCU ESP8266 Unit to an Existing Wi-Fi Access Point.

The existing coding of the electronic reader need to be modified based on the IoT architecture. The objective of this method is to achieve how to send JSON messages over MQTT using the ESP8266. Thus, the user will be using two libraries that handle the low-level details and expose us both the JSON encoding and the MQTT publishing functionalities in easy to use interfaces. These libraries are the PubSubClient, for the MQTT related functionality, and the ArduinoJson, for the JSON related functionality [19-21]. Figure 3.0 shows the coding modification to push the data transfer on cloud server.

```cpp
if (selectCurrent_number == 1) {
  String type = ";
  if(selectSensor_number == 0) type = "0";
  if(selectSensor_number == 1) type = "1";
  if(selectSensor_number == 2) type = "2";

  if (output_value > 40) {
    if(wifi_state == 1 && sended == false)
      String sendstring = "\"identifier\\"\"\"\"ESL READER 1\\"\"\",\"detected\\"\" true\\"\"\"\" type\\"\"\" +type\\"\"\"\\";
      char copy[100];
      sendstring.toCharArray(copy, 100);
      client.publish("foodDetector", copy);
      sended = true;
    }
    display.setCursor(0, 23);
    display.print("DINCTION: YES <<");
  }
}
```

**Figure 3.** After Modification of IoT System Architecture.

### 2.2 Mobile Apps development

Android Studio is the official IDE development environment for create android application. It's designed for android to speed up development and help a user build the highest quality apps for each android device it provides android developer specific tools, including rich code editing, debugging, testing, and profiling tools. Firstly, create an Android project by installing the latest version of Android Studio. Then, create a new Android project for your first app. This app displays the string the electronic reader reading result on the screen of the Android virtual or physical device as shown in Figure 4.0.

Usually, each screen in Android app is associated with one Java class known as a main activity. The single screen with "Hello World" displayed is created by one activity, called MainActivity.java. This was generated for user when created the new project. Each visible activity in an Android app has a layout that defines the user interface for the activity. Android Studio has a layout editor where it can create and define layouts.
Figure 4. Android studios create new project windows.

Figure 5. Developing layout of ESL READER application.

Figure 5 shows the developing layout of ESL READER application. The layouts are defined in XML. A part from that, the layout editor lets the user define and modify the layout either by coding XML or by using the interactive visual editor. Every element in a layout is a view. In this project, it needs an android device such as a android phone to run the apps. A data cable is need to connect the android phone with the computer via the USB port. After that, turn on USB debugging on the android phone to allow the android studio to communicate with android phone as shown in Figure 6.0. Then, install the app on android phone and run it.

Figure 6. Select deployment target panel.
2.3 MQTT Cloud Setup

MQTT is the machine-to-machine protocol of the future. It is ideal for the IoT world of connected devices. Moreover, MQTT protocol is a light weight and an extremely simple protocol. MQTT has a few advantages including providing the relevant data to any intelligent and decision-making resource that can use it [18]. In additions, it allows management and deployment to IoT solutions to be extremely scalable. To create the MQTT cloud server, login to the CloudMQTT website and create an account as shown in Figure 7.

![Main CloudMQTT Panel](image)

**Figure 7.** Main CloudMQTT Panel.

The instance is provided instantly after sign up and can view the instance details, such as connection information on the details page. When connecting via client libraries, the user is often forced to format a connection URL as shown in Figure 8.

![Information details panel](image)

**Figure 8.** Information details panel.

Figure 9 shows how the data will be publish from electronic reader and the data will be retrieve by subscriber which is the smart phone. Therefore, the electronic reader need to connect with the smart phone Wi-Fi for internet connection. After that, the electronic reader automatically linked with cloudMQTT. The MQTT will hold the data topic in data base. Then, the data topic will be display in the smart phone through mobile application. MQTT is based on clients and a server. Similarly, the server is responsible for handling the client’s requests of receiving or sending data between each other. MQTT server is called a broker and the clients are simply the connected devices. For example, if a device (a client) wants to send data to the broker, its call this operation a “publish”. While, if a device (a client) wants to receive data from the broker, its call this operation a “subscribe”.

![Data flow](image)
3. Result and Discussion

After turn on the Wi-Fi by pressing Wi-Fi button on the electronic reader, it takes around 5 seconds to connect the electronic reader to the Wi-Fi. If the electronic reader failed to connect to the Wi-Fi, the electronic reader will reconnect again unless the user pushes off the Wi-Fi button. Figure 10.0 shows the serial monitor that shows the electronic reader has successfully connected to the Wi-Fi hotspot of the smartphone device. Then, the electronic reader has the capability to send messages to the MQTT server.

After completed the inspection using the electronic reader to detect the presence of foodborne pathogen using AI IDE biosensor, the result gained will be sent to the cloud server according to the condition specified in the coding. There are three sensors used to detect food pathogen which is E. coli, Salmonella and Listeria. The cloud server monitor shows the type of sensors and the presence of the foodborne pathogen from the sample. Figure 11 shows the data that sent by electronic reader has been successfully received. Based on the JSON format that was sent to the cloud server it shows some conditions in the following table below.

| Type         | E. coli | Salmonella | Listeria |
|--------------|---------|------------|----------|
| Sensor Number| 0       | 1          | 2        |
| Condition Value (nA) | < 3.5 | > 3.5 | < 3.5 | > 3.5 | < 3.5 | > 3.5 |
| Reading Result | False | True | False | True | False | True |
Table 1 shows the specified condition format data to detect the food pathogens which are E. coli, Salmonella and Listeria. The value of output current condition has been set in the Arduino programming, based on the average value from the previous I-V characterization target ssDNA E. coli, Salmonella and Listeria biosensor reading [22-24]. All the detection reading is in nano ampere range.

**Figure 11.** Received data from electronic reader.

![Figure 11](image)

**Figure 12.** Mobile application (a) Home Page, (b) organize folder page and (c) record page.

![Figure 12](image)

Figure 12 shown the layout editor for ESL READER mobile application. The ESL READER mobile application system was designed and tested successfully. The mobile application system focusing for Android supportive mobile devices in Android Studio software using the Java programming language.
The ESL READER mobile application comprises user interface layout for observing the reading result from the electronic reader. Each layout serving various purpose of information. The main layout interface consists of electronic reader reading result such as the type of food pathogen, the presence state indicator (YES/NO), the count sample and the percentage alert.

4. Conclusion
The Internet of Things (IoT) system for electronic reader has been designed and established. It was able to enhance the communication between electronic reader and internet connectivity through mobile application for better efficiency. The IoT system were assisting to the electronic reader to transmit the reading result from the electronic reader to smart phone through mobile application for record, control and monitoring. It is providing the real time monitoring system. Last but not least, the data can be transfer faster to medical officer about breeding bacteria in food that caused illnesses for taking immediate action.

Acknowledgement
The author would like to thank all staff members of the Institute of Nanoelectronic Engineering in Universiti Malaysia Perlis (UniMAP) for their technical advice and contributions, directly and indirectly. Collaborative Research in Science and Technology Center (CREST) is acknowledged for providing grant for this research program.

References
[1] Y. Zhu and P. Liang, IOP Proceedings, 908, 78 (2017).
[2] G. Rateni, P. Dario, and F. Cavallo, “Smartphone-Based Food Diagnostic Technologies: A Review,” 2017.
[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, Futur. Gener. Comput. Syst., 29, 1645–1660, (2013).
[4] A. Whitmore, A. Agarwal, and L. Da Xu, 60, 261–274 (2015).
[5] B. Liu, “Research on Application of Internet of Things in Electronic commerce,” pp. 13–16, 2010.
[6] R. D. A. A. Rajapaksha, N. A. N. Yahaya, M. N. A. Uda, U. Hashim, AIP Conf. Proc., 2045, 1–7 (2018).
[7] R. D. A. A. Rajapaksha, M. N. A. Uda, U. Hashim, S. C. B. Gopinath, and C. A. N. Fernando, IEEE Int. Conf. Semicond. Electron. Proceedings, ICSE, 2018, 93–96 (2018).
[8] M. A. N. Uda et al., Adv. Mater. Res., 832, 410–414, (2014).
[9] M. A. N. Uda, R. D. A. A. Rajapaksha, M. N. A. Uda, U. Hashim, A. B. Jambek, AIP Conf. Proc., 2045, 2018.
[10] R. D. A. A. Rajapaksha, U. Hashim, N. Z. Natasha, M. N. A. Uda, V. Thivina, C. A. N. Fernando, Proc. 2017 IEEE Reg. Symp. Micro Nanoelectron. RSM 2017, 5, 191–194 (2017).
[11] S. A. B. Ariffin, T. Adam, U. Hashim, S. Faridah Sfaridah, I. Zamri, and M. N. A. Uda, Adv. Mater. Res., 832, 113–117 (2013).
[12] M. A. N. Uda et al., Elsevier Inc., 2019.
[13] M. N. A. Uda et al., AIP Conf. Proc., 1808, (2017).
[14] M. N. A. Uda et al., AIP Conf. Proc., 1808, 10–14, (2017).
[15] N. A. N. Yahaya, R. D. A. A. Rajapaksha, M. N. A. Uda, U. Hashim, AIP Conf. Proc., 1885, (2017).
[16] R. D. A. A. Rajapaksha, U. Hashim, M. N. A. Uda, and C. A. N. Fernando, Mater. Sci. Eng. B, 10(1), 61–64 (2017).
[17] R. K. Kodali and S. Soratkal, Automation System, 23, 35 (2015).
[18] R. D. A. A. Rajapaksha, N. A. N. Azman, M. N. A. Uda, U. Hashim, S. C. B. Gopinath, C. A. N. Fernando, AIP Conf. Proc., 2045, (2018).
[19] S. Grassini, S. Corbellini, M. Parvis, E. Angelini, and F. Zucchi, Meas. J. Int. Meas. Confed., 8
114, 508–514 (2018).

[21] I. Luiz, B. De Moura, L. Carlos, D. S. Monteiro, F. A. Soares, Int. J. Electr. Robot. Electron. Commun. Eng., 8, 310–314 (2014).

[22] O. A. Olumodeji and M. Gottardi, Integr. VLSI J., 58, 438–445, (2017).

[23] R. D. A. A. Rajapaksha, U. Hashim, M. N. Afnan Uda, C. A. N. Fernando, S. N. T. De Silva, Microsyst. Technol., 6, 70 (2017).

[24] R. D. A. A. Rajapaksha, U. Hashim, M. N. Afnan Uda, C. A. N. Fernando, S. N. T. De Silva, Microsyst. Technol., 6, 70 (2017).

[25] N. Z. Natasha, R. D. A. A. Rajapaksha, M. N. A. Uda, and U. Hashim, 020235, (2017).