Wireless communication design of internet of things based on FPGA and WiFi Module

F W Wibowo

1 Informatics Department, Universitas Amikom Yogyakarta, Yogyakarta 55283, Indonesia
Email: ferry.w@amikom.ac.id

Abstract. The implementation of the internet of things (IoT) has been many researched. Some applications have their own designs and special uses. Low-cost designs and multi-usage are always targeted on the implementation either using a microcontroller or field programmable gate arrays (FPGAs). The reconfigurable device of FPGA in some cases has advantages in its uses. This paper has designed the communication of the radio frequency (RF) for the IoT using wireless communication based on the FPGA and wireless fidelity (WiFi) of ESP8266 WiFi module. The research methodology used in this paper was a simulation using Xilinx ISE software for the device of the Xilinx Spartan-3E FPGA that has been configured using Very High Speed Integrated Circuit Hardware Description Language (VHDL). The result of the ESP8266 driver based on the FPGAs is sent to the JSON based website.

1. Introduction
The internet of things or usually it is called IoT because it connects all things is a system where the internet is connected to the analog world via ubiquitous sensors, and then it is processed, transformed, and transmitted into the digital world. The IoT has a role in the dynamic control of the industrial process and daily life that can be applied to the appliances [1]. This can also improve the resource utilization ratio so it takes the efficiency and effectiveness in its use, so the relationship between humans and nature can be better and forming an intellectual entity. It can also emerge communicating between the physical world and the information world. It has been assumed that the IoT will be implemented on the devices up to 50 billion units by 2020 [2], but this number can be lower than what was predicted. Data is established from the sensors and processed by the control or processor points so these machines or devices can communicate with each other. It can also manage the data and enforce proper access control while the data could be large since the transmitted data is in a real-time process and involving a wide array of sensors. In providing the real-time data analytics, it should be aware of the sensor inputs. For this purpose, it can apply the Kalman filter to detect the value error of the input sensors [3]. The scalable should be done in the use of the large-scale device networks and monitor various parameters.

The implementation of the field programmable gate array (FPGA) can be a solution for the challenges at the core of IoT implementation. It can promise the design approach that addresses on the power efficiency, incompatible interfaces, and migration to a new processor due to increased requirements [4]. The technology implementation paradigm for IoT using FPGA has many roles in reliable systems [5]. Even in the era of industry 4.0, the IoT absolutely could be applied in the production activities with its capabilities [6]. So it can’t be doubt that the FPGA implementation in the field of the IoT is still having place in the low-cost experiment and implementation since it may reconfigure on this device so that it can design with high interoperability with others device or signal. This paper addresses the wireless communication design using reconfigurable device which is FPGA.
and the WiFi module of ESP8266 as the data transmitting and receiving based on the radio frequency. The ESP8266 WiFi module is usually used in implementing IoT or embedded systems applications within the Arduino microcontroller. So this paper also will analyze the WiFi interface design of the IoT implementation.

2. Related Works

The applications of the IoT based on the microcontroller has been spread widely and many of the project has been used to get low-cost, effectively, and efficiently application. In a monitoring system that has been employed in the wireless sensor network (WSN) is also using a part of the IoT [7]. The elements of this monitoring consist of sensors, processors, and wireless modules. The sensor is functioned to catch the analog signal or data, thus this signal or data is sent to the processors which are processing in digital signal or data to yield information. The IoT is also using transmitting (TX) and receiving (RX) data wirelessly in its implementation. In its implementation for this TX/RX parts for the radio frequency (RF) module can be designed using field programmable gate arrays (FPGAs) [8].

The application of WSN could be used to collect information from the nodes that are designed like IoT but using a wireless connection. FPGA could be involved in the implementation of the monitoring system. According to the implementation of [9], it employs FPGA using sensors i.e. pH sensor, water level sensor, gas sensor, and temperature sensor. Each sensor module has had own analog-to-digital converter (ADC) module so it employed the FPGAs for the digital data processor only. While transmitting data for the values, it utilizes ZigBee or XBee module for transmitting wirelessly. ZigBee module has characterization use indoor and outdoor as shown in [10]. As implementing multi-ports of the sensor inputs, it could apply the reconfigurable first-in-first-out (rFIFO) for the design in FPGAs since this method can provide an optimum solution at different frequencies of the IoT architecture [11]. In addition, the FPGAs are not used for signal/data processing only, but it can also design such an algorithm on it [12]. In implementing the algorithm on this device, it should know the properties of the hardware design and of course the work principles or mechanisms of the algorithm. The higher accuracy could be reached by computation and processing raw data that has been resulted by the FPGA system which has been designed and used to analyze with lower execution time [13].

This paper is describing to program the ESP8266 WiFi module using reconfiguring of the Xilinx Spartan-3E FPGA. For reconfiguring purposes, this paper is using a hardware description language of very high-speed integrated circuit hardware description language (VHDL). The implementation of this WiFi module programmed using FPGA has also been applied by [14], but this paper boundary is only applying the design on the scope of the communication between both FPGA and ESP8266 WiFi module and sending the data from the read-only memory (ROM) that has been programmed in the FPGAs to the website based on JavaScript Object Notation (JSON). JSON is a data format that is used to interchange and store the data so it is a subset of the JavaScript. JSON uses an extension format of .json, but when it is defined in the other format like .html it is performed in quotes as a string JSON and could be included onto a variable. This format is easy to transfer between web servers and clients.

3. Research Methodology

The research methodology used in this paper is a simulation of the design using FPGA and ESP8266 WiFi module. The ESP8266 WiFi module is a low-cost platform and effectively used in the communication between devices and control via wireless or internet. It can be applied standalone or connected to the other device [15]. It is often used in the IoT applications and attached to the microcontroller as mentioned before. There are some types in the markets and easy to find either online or physically electronic shops (offline), namely ESP-01, ESP-02, ESP-03, ESP-14, etc. For the functions hosted on the module, it is similar but the difference is on the general-purpose input-output (GPIO). The ESP-01 is a module that allows accessing the controller device via the internet. This module is categorized as a system on chip (SoC) and not always require the controller device to drive the input/output (I/O). In the market, there are two kinds of this type namely blue and black printed circuit boards. The difference is on its memory capacity that affects the setting, the blue one is 512Kb whereas the other is 1Mb. This module can act as a computer mini with limited GPIO. It contains 4
The ESP8266 WiFi module in this paper is only employing TX and RX to make communication with the FPGAs. The FPGA used in this paper is Xilinx XC3S500E Spartan-3E FPGA. The configuration in the connection among FPGA, ESP8266 WiFi module, and the website based on JSON can be shown in Figure 2.

The reconfiguration for this ESP8266 driver on the FPGA consists of clock divider, controller, read-only memory (ROM), register, TX, and RX modules. The clock divider module is used to yield the appropriate clock as the baud rate used to communicate between the devices i.e. 9600 bps. The controller module is used to control the process of interfacing with driving the ROM. The ROM itself contains AT command data that used to communicate with the ESP8266 WiFi module. The driven
data from the ROM will be conveyed to the register thus the data will be transmitted via the transmitter module. Since this configuration in communicating serial data is employing the mode of universal asynchronous receiver/transmitter (UART), so the module is not necessary transmitting/receiving the clock in both communications to the ESP8266 WiFi module but both TX and RX modules must use the clock as seen in figure 2. The data that programmed onto the ROM as below,

1. AT command to reset the ESP8266 that uses “AT+RST\r\n”
2. AT command to join the access router for connecting to the internet that uses “AT+CWJAP="<SSID>","<password>"”. SSID stands for service set identifier which is the name of the WiFi network used, while a password is a password for connecting to this SSID.
3. AT command to connect to the website that uses “AT+CIPSTART="<type>","<website_name>",<port>”. The type is choosing one between two types which are transmission control protocol (TCP) or user datagram protocol (UDP). The TCP is a connection-oriented when it is set then the data can be sent bidirectional, while UDP is a connectionless internet protocol.
4. AT command to specify the number of bytes with using “AT+CIPSEND=<number of bytes>”
5. Send the data with JSON format that uses sequential data as “<website_name>/ferry/fpga/add?data1=<number1>&data2=<number2>”. It means that <website_name> is the name of the destination website that the data is sent, /ferry is the folder on the website, and /fpga is the sub-folder on the website. This folder is created to simple in a grouping of the data if the data wouldn’t be grouped in one folder so the folder might not be necessarily made. While for adding the data can use “add?data1=<number1>&data2=<number2>”. The variable of the number e.g. something like the value of sensor data is written in <number1> and <number2> depends on the sequential of data.

In programming the data of ROM, the data must be written using an American standard code for information interchange (ASCII). The representation of this code can be shown in figure 3.

| 00 nul | 01 soc | 02 stx | 03 etx | 04 eot | 05 enq | 06 ack | 07 bs  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 08 ht  | 09 n1  | 0a vt  | 0b cr  | 0c so  | 0d si  | 10 dle | 11 dc1 |
| 12 dc2 | 13 dc3 | 14 dc4 | 15 nak | 16 syn | 17 etb | 18 can | 19 em  |
| 1a sub | 1b esc | 1c fs  | 1d gs  | 1e re  | 1f us  | 20 sp  | 21 t   |
| 22 "   | 23 #   | 24 $   | 25 %   | 26 &   | 27 '   | 28 (   | 29 )   |
| 30 _   | 31 1   | 32 2   | 33 3   | 34 4   | 35 5   | 36 6   | 37 7   |
| 38 8   | 39 9   | 3a :   | 3b _   | 3c <   | 3d >   | 3e ?   | 3f a   |
| 40 N   | 41 A   | 42 B   | 43 C   | 44 D   | 45 E   | 46 F   | 47 G   |
| 48 H   | 49 I   | 4A J   | 4B K   | 4C L   | 4D M   | 4E N   | 4F O   |
| 50 P   | 51 Q   | 52 R   | 53 S   | 54 T   | 55 U   | 56 V   | 57 W   |
| 58 X   | 59 Y   | 5A Z   | 5B [   | 5C \   | 5D ]   | 5E ^   | 5F _   |
| 60 `   | 61 a   | 62 b   | 63 c   | 64 d   | 65 e   | 66 f   | 67 g   |
| 68 h   | 69 i   | 6A j   | 6B k   | 6C l   | 6D m   | 6E n   | 6F o   |
| 70 p   | 71 q   | 72 r   | 73 s   | 74 t   | 75 u   | 76 v   | 77 w   |
| 78 x   | 79 y   | 7A z   | 7B {   | 7C |   | 7D }   | 7E ~   | 7F del |

Figure 3. ASCII Code.
An example code in the ROM that is used to connect to WiFi access point of ferry with password 123456 can be written as Table 1.

| Command | ASCII Code |
|---------|------------|
| A       | x"41"     |
| T       | x"54"     |
| +       | x"2b"     |
| C       | x"43"     |
| W       | x"57"     |
| J       | x"41"     |
| P       | x"50"     |
| =       | x"3d"     |
| "       | x"22"     |
| f       | x"66"     |
| e       | x"65"     |
| r       | x"72"     |
| y       | x"79"     |
| n       | x"0a"     |
| \n      | x"0d"     |
| \t      | x"0d"     |

In the example code above, the UART can convey 8 bits or 1 byte once transmitting so it can be represented by one character of ASCII. The data storage in ROM could be divided into 16 characters and the blank character could be filled as 0xFF. For connecting to WiFi, it uses AT+CWJAP with SSID = ferry and password=123456. The symbol of \r \n, it is also known as carriage return (CR) and line feed (LF), is used as a new line character or similar with the enter key function. In the ASCII code, they are represented by 0x0d (13) and 0x0a (10) respectively whereas in the VHDL code, bit vector constants of hexadecimal are written as the notation of x"--". Each AT command that has been defined must be ended by \r \n to be executed in programming.

4. Result and Discussion

The VHDL code of the ESP8266 driver that is programmed into the Xilinx Spartan-3E FPGA has a configuration of the device as 3s500efg320-4. The utilization of the components that has been consumed for the design of the ESP8266 driver based on Xilinx Spartan-3E FPGA (3s500efg320-4) can be concluded in Table 2.

| Components                      | Number | Total  | Percentage |
|---------------------------------|--------|--------|------------|
| Slices                          | 231    | 4656   | 4%         |
| Slice Flip-Flops                | 238    | 9312   | 2%         |
| 4-input Look-Up Tables          | 367    | 9312   | 3%         |
| Input/Output                    | 5      |        |            |
| Bonded IOBs                     | 5      | 232    | 2%         |
| BRAMs                           | 1      | 20     | 5%         |
| GCLKs                           | 1      | 24     | 4%         |

The data that is sent to the website based on JSON has format as "<website_name>/ferry/fpga/add?data1=4&data2=7\r\n". In this format, the data1 is equal to 4 and data2 is 7 are illustrating the inputs if there are inputs on both data. This can be varied depends on the inputs used to sensor input value. To check the result of the data on the website, if this data has been successfully sent, can be checked using the format "<website domain>/ferry/fpga" as shown in figure 4. This figure shows that the transmitting data has been successfully done. With some modifications, it can be used for transmitting and receiving data.
Figure 4. Result of transmitting data on JSON based website.

5. Conclusion
The design of wireless communication using FPGA and the ESP8266 WiFi module has been configurated. The design used in this paper is using the communication mode of UART. The reconfiguration module for this ESP8266 driver on the FPGA consists of clock divider, controller, read-only memory (ROM), register, TX, and RX modules. For future works, the input values of data can result from the sensor input values. The input from the smartphone can also be used to control using a JSON based website to communicate with the ESP8266 WiFi module that is connected to the FPGA in controlling devices.

6. References
[1] Wibowo F W and Hidayat F 2017 A low-cost home automation system based-on internet of things Journal of Telecommunication, Electronic and Computer Engineering Vol 9 No 2-4 pp 155-159
[2] Evans D 2011 The internet of things: how the next evolution of the internet is changing everything White Paper of Cisco Cisco Internet Business Solutions Group (IBSG)
[3] Wibowo F W and Putra W S 2019 Sensor array fault detection technique using Kalman filter IEEE Proceedings of the 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) pp 124-128 DOI: 10.1109/ISRITI48646.2019.9034591
[4] EDN 2016 FPGAs solve challenges at the core of IoT implementation accessed at https://www.edn.com/fpgas-solve-challenges-at-the-core-of-iot-implementation/ on 4 August 2019
[5] Rupani A, Saini D, Sujediya G, and Whig P 2016 A review of technology paradigm for IOT on FPGA International Journal of Advanced Research in Computer and Communication Engineering Vol 5 Issue 9 pp 61-64
[6] Wibowo F W 2018 A dynamic intelligent control analysis on the wireless smart machine environments in the industry 4.0 IEEE Proc. of 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI 2018) pp 389-393
[7] Aditya T W and Wibowo F W 2016 Wireless sensor network and geographic information system based monitoring system Asian Journal of Information Technology vol 15 no 15 pp 2571–2577
[8] Wibowo F W 2015 Implementation of Viterbi algorithm based-on field programmable gate array for wireless sensor network Advanced Science Letters Vol 21 No 11 pp 3521-3525
[9] Balaji N, Vijayalakshmi S, Durgadevi K, Mohanraj K, and Mangayarkarasi 2019 FPGA implementation of smart water quality monitoring system International Journal of Innovative Technology and Exploring Engineering Vol 8 Issue 11 pp 3136-3139

[10] Perdana R C and Wibowo F W 2016 Quality of service for XBEE in implementation of wireless sensor network Journal of Engineering and Applied Sciences Vol 11 No 8 pp 692-697

[11] Gupta S, Sharma M, and Chawla R 2020 FPGA implementation of r-FIFO-based high-speed data acquisition IOT architecture model SN Applied Sciences Vol 2 pp 661

[12] Wibowo F W 2019 An analysis of FPGA hardware platform based artificial neural network Journal of Physics: Conference Series Vol 1201 No 1 pp 012009

[13] Satpathy S, Mohan P, Das S, and Debbarma S 2019 A new healthcare diagnosis system using an IoT-based fuzzy classifier with FPGA The Journal of Supercomputing

[14] Abdul A M, Krishna B M, Murthy K S N, and Khan H 2016 IOT based home automation using FPGA Journal of Engineering and Applied Sciences Vol 11 No 9 pp 1931-1937

[15] Ai-Thinker Inc 2017 ESP-01/07/12 series modules user’s manual V1.1

[16] Espressif Systems 2019 ESP8266: AT Instruction Set Version 3.0.2

Acknowledgments
Author thanks to publisher who has given chance to publish this paper.