Development of Internet of things-based Early Flood Detection using ESP32S Microcontroller and thinger.io Platform

Article *in* TEST ENGINEERING AND MANAGEMENT · April 2020

7 authors, including:

Imran Mohammad Sofi
University of Kuala Lumpur

Abdul Halim
Soil Resource Development Institute

Abdul Halim Abd Rahman
University of Kuala Lumpur

Mohd Azlan Abu
Universiti Teknologi Malaysia

March - April 2020
ISSN: 0193-4120 Page No. 8159 - 8165

Development of Internet of things-based Early Flood Detection using ESP32S Microcontroller and thinger.io Platform

[1]Imran Mohamad Sofi, [2]Mohammad Amirul Idham Zulkiply, [3]Abdul Halim Abd Rahman , [4]Zulhilmi [5]Mustafa, Mohd Azlan Abu, [6]Muhammad Noor Nordin

[1,2,3,4,5,6] Electronic Technology Section, Universiti Kuala Lumpur, Gombak, Selangor, Malaysia

[1]imranms@unikl.edu.my, [2]amirul.zulkiply@s.unikl.edu.my, [3]abdhalim@unikl.edu.my, [4]zulhilmim@unikl.edu.my , [5]mohdazlan@unikl.edu.my , [6]muhammadnoor@unikl.edu.my

Article Info
Volume 83
Page Number: 8159 - 8165
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 09 April 2020*

Abstract

Flooding is one of the highly destructive natural disasters in Malaysia and phenomenon happened almost every year particularly during the monsoon season. The lack of an early detection system implemented causes high lost in both properties and people's lives. The development of this early flood detection system via IOT is designed with the objectives to help in providing the platform to monitor the early signs of flooding by providing public and authorities with information of wind speed, water level, rate of rain within the period of one week of a particular location. This system able to give warning via alarm system and mobile application notification when the parameters detected increasing up to a certain dangerous threshold. The data collected from the sensors are processed and transmitted via Wi-Fi using ESP32S and are displayed on the thinger.io dashboard on the web and mobile application. The data are also stored up to five years for research or prediction about the flooding in the future. This system proves to be very crucial in providing critical information in monitoring flooding potential in an area where a high probability of flooding can occur.

Keywords; *IOT, ESP32S, thinger.io, flood, early-detection, open-source.*

I. INTRODUCTION

The most natural disaster in Malaysia is flooded since Malaysia isn't to encounter a portion of the catastrophic event that is going on around the globe, for example, the quakes, fountain of liquid magma, storms and others. Nearly every year during the monsoon season, flood become one of the major problems in Malaysia [1]. Flooding may result from the volume of water in a river or lake which overflows on lower areas. Flooding also happens when water in the dam is released when the dam is unable to accommodate a large amount of water and the result caused water flow exceeds the capacity of the river channel, particularly at bends or meanders. Because of this, Flooding in Malaysia has given an adverse effect to some part of the country's economy and the damage caused by floods is estimated to be worth millions and affecting certain states in Malaysia for many years [2].

Flood detection system must be created to minimize the damages caused by a flood. Because the flood warning system in Malaysia is not as effective as compared to other countries. Malaysia needs an advanced system flood warning so that the public, especially at rural area, would be better prepared against flooding and to reduced damage and destruction of property. This project will collect the water level data by a sensor every two minutes the provide notification and give an alarm when the parameters detected reaches a dangerous threshold level. This system will send the notification through

Published by: The Mattingley Publishing Co., Inc.

the open source thinger.io application and the buzzer will turn on when the sensor detects dangerous status. Therefore, the public and authorities would be better prepared to face floods and to reduce the level of damage and destruction of property and possibly preventing loss of lives. [3]

II. METHODOLOGY

This section discussed and explain the method used to design the early flood detection system. In this section, the block diagram and the part of the main coding will be discussed in detail with the information about project works and the components used.

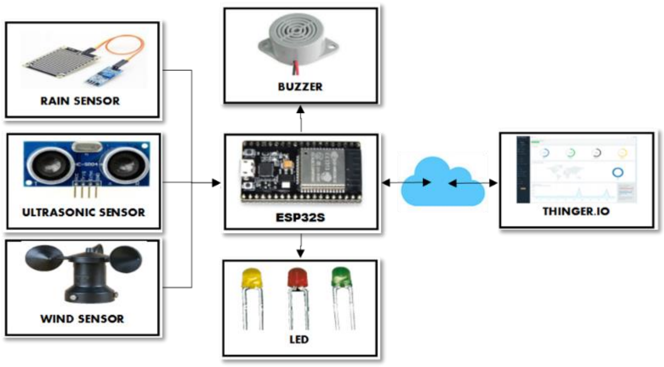

Figure 1. Block diagram of the Flood Detection System.

Figure 1 shows the block diagram of the project. In this project, The ultrasonic sensor, rain sensor, and water flow sensor used as the main input component to capture the important parameters for early flood detection. The rain sensor is used to capture the rate of raindrop occurring at a particular location by sensing wetness on the surface of the sensor during rain. The ultrasonic sensor is used to detect the water level by sending ultrasonic wave to the surface of the body of water and reading the time taken the ultrasonic waves to bounce back to the sensor which depending on the level of the body of water, the lower the water level is the more time taken for the ultrasonic waves need to travel from and back to the sensor. The wind sensor senses the speed of the wind blowing at a particular location the system is placed. The high speed wind blowing toward the sensor will turn the sensor's rotor and generating voltage according to the speed of the rotor rotation.

ESP32S is used as the main brain on this project to received inputs from all 3 main sensors and to process the input data and send the data to the open sourceIoT platform named thinger.io via Wi-Fi. The wind speed, water level and rate of rain can be viewed graphically on the thinger dashboard on the web and also on its mobile application, these data can be view and monitor from anywhere on a computer or a mobile phone with a stable internet connection. Thinger.io also stores data from all three sensors in its specified data bucket for every 2 minutes and these data can be stored up to 5 years and the data can be downloaded in excel files for processing and viewing.

3 color LEDs and a buzzer are attached to the ESP32S microcontroller as physical warning outputs for the system. The thinger.io platform is also able to provide notification of danger to users. The system has 3 stages of parameters condition which are Safe, Sandy and Warning. In the Safe condition, the green LED will be turned on and no buzzer or notification will be triggered. In the standby condition, yellow LED will be turned on and no alarm will be triggered but a notification will be sent reporting of the current situation and reading from the sensors at the particular moments to let the user be on standby mode. In the warning stage, the red LED will turn on blinking and the buzzer will be triggered together with the evacuate notification will be sent.

```
#include <WiFiClientSecure.h>
#include <ThingerESP32.h>

#define USERNAME "Amirul"
#define DEVICE_ID "ESP32"
#define DEVICE_CREDENTIAL "3fzTakAlbQw7"

#define SSID "ASUS_X00TD"
#define SSID_PASSWORD "amirul9504"
```

Figure 2. Coding for setting the Wi-Fi connection and device connection with thinger.io

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 8159 - 8165

Figure 2 shows the main coding needed to set the system to be connected to a specific network. Username and need to match the username registered on the thinger.io website. Device ID also needs to be the same ID where the ESP32S is set in the thinger.io website in order to enable the correct communication channel between the microcontroller and the thinger.io platform. Device credential is generated from the website in order to make sure that the particular device set on the website is paired correctly with the same unique device credential.

SSID is the service set identifier or the name of the wireless network needed to be connected to while the SSID password is the security phrases associated with the wireless network name needed to be connected to. All these setting is needed to be correctly configured in order to establish a wireless connection between the microcontroller and the Wi-Fi and communication between the microcontroller and the thinger.io platform.

```
thing.add_wifi(SSID, SSID_PASSWORD);
// digital pin control example (i.e. turning on/off
thing["led"] << digitalPin(2);
thing["RAIN"] >> [](pson& out){
float value = analogRead(35);
 out = 4095 - value;
};
thing["WIND"] >> [](pson& out){
float value1 = analogRead(33);
 out = value1 / 4095 *100;
};
thing["WATERFLOW"] >> [](pson& out){
float value2 = l_hour;
 out = value2;
};
thing["ULTRASONIC"] >> [](pson& out){
float value3 = distance;
 out = value3;
};
```

Figure 3. Main code required to established communication between sensors with the thinger.io platform.

Figure 3 shows the main coding used to configure the input and output parameters from the sensors, LEDs, and buzzer according to the thinger.io (thing) library instruction set to make sure on the thinger.io website the specified "name" able to carry the correct input or output data. The name set within the bracket after the instruction "thing" is the name associated with the value which is processed within the curly bracket. For example, the name "RAIN" is associated with analog input read from ESP32S pin number 33 which is connected to the rain sensors. Float variable is used since the value recorded from a rain sensor is not an integer. 4095 is the maximum value that the pin can read for analog input. "led" is associated with a digital output which will be sent out from ESP32 pin number 2. The same pattern applies for the "WIND", "WATERFLOW and "ULTRASONIC" name set in the coding.

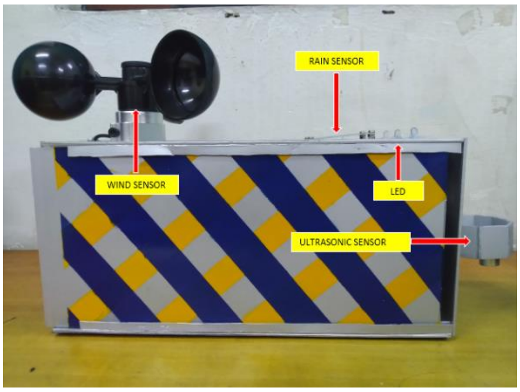

Figure 4. The final prototype of the system and position of all the sensors used.

Figure 4 shows the complete prototype constructed for testing. The ultrasonic sensor is placed at the right side of the system with the sensor facing down where during testing there will be a body of water underneath the system. 3 LEDs which consist of green, yellow and red color LEDs are placed on top of the system for better visibility for a user to see. The buzzer is placed on the side of the system to increase the sound propagation capabilities in order to enhance the effectiveness of a warning system. The rain sensor in position on top of the system to provide a large surface area for a raindrop to be collected on top of the sensor's surface during rain. The final sensor which is the wind sensor is placed on top of the system with the cup shape flags facing horizontally to improve the effectiveness in sensing the wind speed as the flags are aligned with the direction of the wind flow.

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 8159 - 8165

III. RESULTS AND DISCUSSION

Firstly, when the ESP32S receive the voltage supply. The ESP32S will start operated based on coding which has been set up on this module. This system also will be operated depends on the flowchart which was made before the project was created. Firstly, when the supply is given to the ESP32S. This module will stabilize with the connection to the Wi-Fi. After that, the thinger.io application will get online and ready for the internet of things when the microcontroller is linked to the internet.

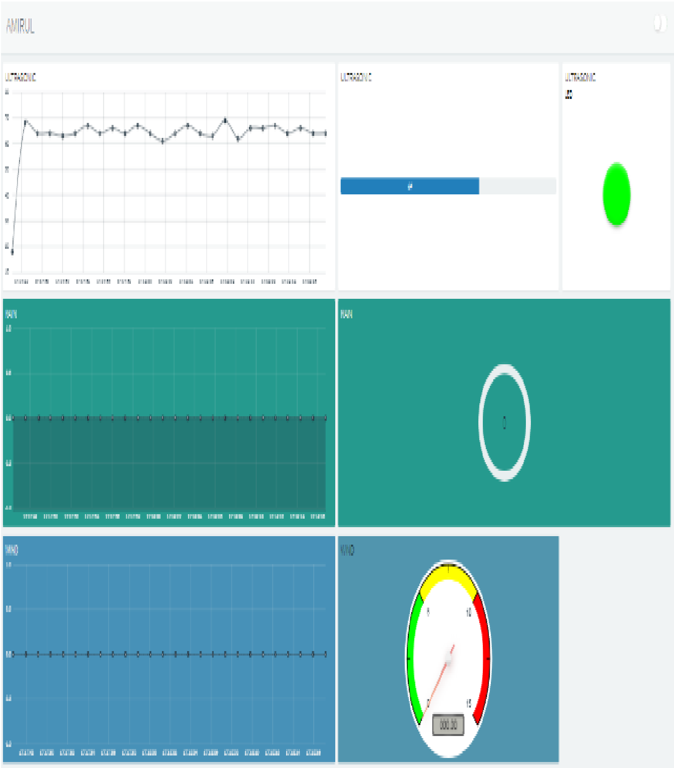

Figure 5. Thinger.io dashboard which can be monitored and controlled wirelessly.

Figure 5 shows the Thinger.io application as output for this project. Like the other digital dashboards, this application also can build a graphic interface for the project by simply dragging and dropping widgets and this application very simple to set everything up. At this part, this application will receive the data from every sensor used for this project and it will display in various forms that have been set on this application. In this project, it uses the graph as the main function to show the data for each sensor reading. Besides the thinger.io application, led and buzzer also used as an output on this project

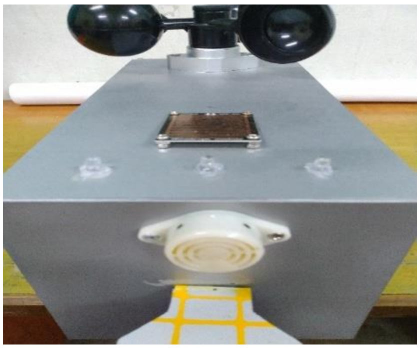

Figure 6. Position of the LEDs and buzzer.

Figure 6 shows that the led and buzzer that has been installed on this project. Three different color LEDs that are the green, yellow and red color used in this project. The LEDs

and the buzzer will trigger when the ultrasonic sensor detects the parameter of water level.

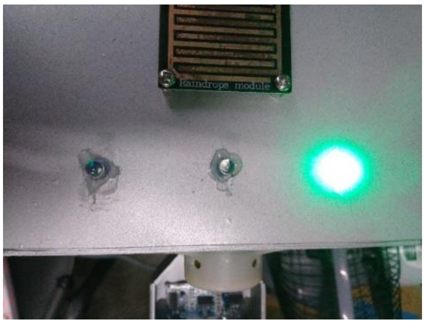

Figure 7. Safe condition, greed LED turned on.

Figure 7 shows that the LED green turn on when the ultrasonic sensor detects the normal water level, the LED green will turn on to indicate the water level is in a safe condition.

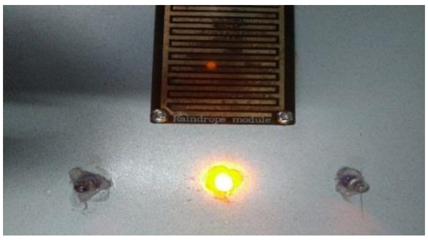

Figure 8. Standby condition, yellow LED turned on.

March - April 2020
ISSN: 0193-4120 Page No. 8159 - 8165

Figure 8 shows that the LED yellow turn on when the ultrasonic sensor detects the danger water level. At this stage, the residents must be careful and willing to be in any way possible.

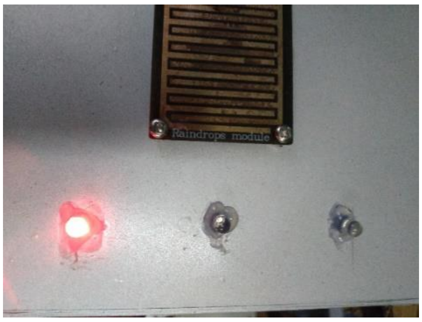

Figure 9. Warning condition, red LED turned on.

Figure 9 shows that the red LED and the buzzer will turn on when the ultrasonic sensor detects the very dangerous water level. At this stage, the residents must be moved to a safe and higher place. The buzzer and led works based on the coding that has been inserted on ESP32S.

Figure 10. Water level data collected within 30 minutes

Figure 10 shows the water level data collected on the thinger.io application within 30 minutes. The ultrasonic sensor will collect water level data every 2 minutes and display the data into Thinger.io application. Since this project is only a prototype, the sensor readings have been set up on three stages which is a normal stage, dangerous stage, and very dangerous stage water level. The normal water level readings stage is 0-25cm. The dangerous water level stages are 25cm-50cm and very dangerous water level stages is 50cm and up.

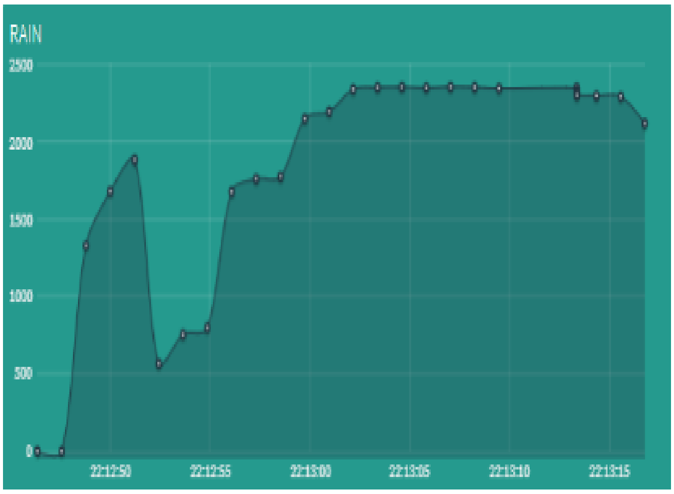

Figure 11. Rain rate data collected within 30 minutes

Figure 11 shows the rain rate data collected on the thinger.io application within 30 minutes. When the sensor does not detect the rain or water, the graph reading will be 0. The graph will be increased when the raindrop sensor detects the amount of water. The higher reading of the graph when the rain is getting heavy. When the rain stops, graph reading will be decreased.

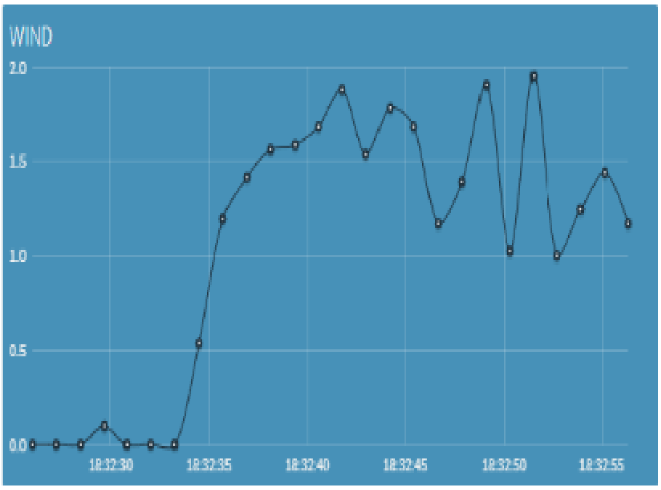

Figure 12. Wind speed data collected within 30 minutes

Figure 12 shows the wind speed data collected on the thinger.io application within 30 minutes. Wind sensors are used on this project as floods often occur

Published by: The Mattingley Publishing Co., Inc.

TEST Engineering & Management
March - April 2020
ISSN: 0193-4120 Page No. 8159 - 8165

during the monsoon season. The monsoon season usually causes a storm or wind. The graph will be increased when the wind sensor detects wind speed. The higher reading of the graph when the wind is getting heavy.

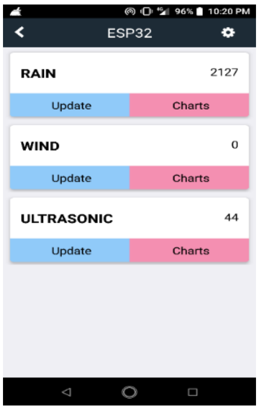

Figure 13. Thinger.io application on a mobile phone.

Figure 13 shows the sensor output from the thinger.io application on the thinger.io mobile phone application. The use of this application is intended for other users to know about sensor data readings. The users only need to scan the QR code to get into this system. This application also can display the reading sensor into the graph form and update the sensor data quickly.

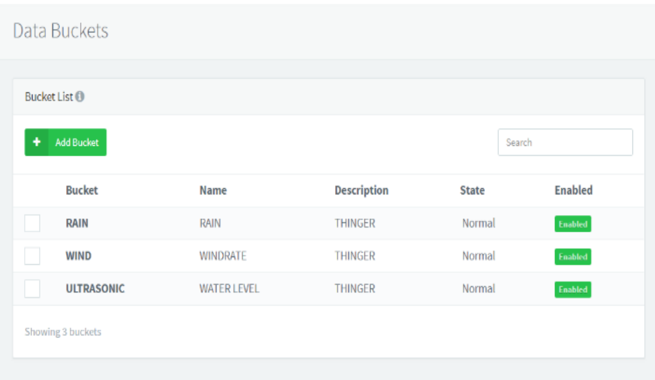

Figure 14. Data buckets for all sensors collected on the thinger.io platform

Figure 14 shows the data buckets from the thinger.io application. The use of data buckets is to store all the data taken by the sensor and it is very important for future research.

IV. CONCLUSIONS AND RECOMMENDATIONS

A. Conclusions

The flood detection system is very useful and it important for residents because this system is able to overcome the problem of floods that can cause many losses and lives. Nowadays, the problem of floods is still cannot be solved yet as the flood can happen at any time and therefore, the need for such a system to reduce the problem of this disaster. Besides, this project is also safe for use as it has no dangerous functions or components installed such as the machine. The four wheels' machine makes the machine easy to move in the oil plantation farm. This auto machine did not costly because it only uses the small components compared to the present projects and good to oil palm plantation workers that want to reduce their back pain, time and reduce cost.

The project has been successfully implemented according to the plan and has succeeded in following the objectives of this project. The concept that has been applied to this project is suitable for use and is based on the requirements of this project. This project can also add knowledge about each component such as ESP32S and Thinger.io application usage and give the opportunity to show the skills and creativity to create their own project. Additionally, this project was created based on previous reports and previous projects that enabled this project to be successfully built. Finally, early preparation and good planning are important to produce the perfect project.

B. Recommendations

Among the future recommendation that can be added in order to improve the efficiency and accuracy of the system is to use a higher quality water level, wind speed, and rain sensors. The buzzer may also be replaced with a better alarm system in order to improve the warning system and its coverage. The LEDs can also be replaced with a higher AC light bulb to improve visibility in stormy

Published by: The Mattingley Publishing Co., Inc.

weather. The data displayed on the dashboard can also be improve by adding threshold lines which can show the level of a safe, standby and dangerous condition on the graph itself.

V. ACKNOWLEDGMENT

I would like to express my sincere appreciation to Mohammad AmirulIdhamZulkiply, Ts. Abdul HalimZulhilmi Mustafa, Ir. Ts. Dr.MohdAzlan Abu and Dr. Muhammad Noor Nordin for their cooperation, contribution and support to make this project successful.

REFERENCES

[1]. N. W. Chan, "Impacts of Disasters and Disaster Risk Management in Malaysia: The Case of Floods," Resilience and Recovery in Asian Disasters, pp. 239-265, 2015.

[2]. Doraisamy, S. V., Akasah, Z. A., &Khamis, A. (2016). Factors Towards The Restoration Of Abandoned Residential Construction Projects In Malaysia: A Pilot Study. JurnalTeknologi, 78(5). doi:10.11113/jt.v78.8241

[3]. M. S. K. Shazwani Shafia, "Flood Disaster Management in Malaysia: A Review of Issues of Flood Disaster Relief during and Post-Disaster," in The European Proceedings of Social & Behavioural Sciences EpSBS, 2016..