A Water Level Detection: IoT Platform Based on Wireless Sensor Network

1st Anton Prafanto
Faculty of Computer Science and Information Technology
Universitas Mulawarman
Samarinda, Indonesia
anton.prafanto@gmail.com

2nd Edy Budiman
Faculty of Computer Science and Information Technology
Universitas Mulawarman
Samarinda, Indonesia
edy.budiman@ymail.com

***Abstract*—Wireless Sensor Network (WSN) has been used in various fields and for various applications. This includes environmental monitoring, volcanic activity monitoring, supporting components in order to provide a smart city, and monitoring activity in the agricultural field. The detection of water level in the river is still not able to work automatically and real-time to determine the water level of the river. This causes the locals generally do not know when the river on the river level grows higher and higher. In this study designed a water level detection sensor-based wireless network that works automatically by reading the height of the water using ultrasonic sensor and then the elevation data is sent to a website so that the public can monitor the height of the river in real time. This system was designed using two main components HC-SR04 and NodeMCU. HC-SR04 periodically transmit data of water level in real time, then NodeMCU upload this to the monitoring platform. The test results show that the design can achieve the collection and stably upload of surrounding environment data, and meet the design requirement of remotely monitoring the living environment conditions. The invention has follow advantages: small power consumption, simple structure, low cost, convenient installation and urban communities of Samarinda no longer confused to monitor the water level in flood-prone points, with this study, now everything can be monitored in real time through all kinds of gadgets.**

Keywords—arduino nano, HC-SR04, nodeMCU, wireless sensor network

I. Introduction

As one of the countries that have a fairly high rainfall in the world. Indonesia is a tropical country where weather phenomena occur very dynamic or rapidly changing. This is because we have very much sun exposure (tropics), which acts as the major energy generator coupled with the weather and geographical conditions of our country is an archipelago. Both of these things are speeding up the interaction between the ocean and land to form a weather system that is changing so rapidly [1]. Keep in mind that not only internal factors above, Indonesia weather is also strongly influenced by many external factors, for example the phenomenon of El Nino, La Nina, Dipole Mode and MJO (Madden-Julian Oscillation) [2].

Kalimantan itself has a large amount of rainfall throughout the year. This is true even for the driest month. The climate here is classified as Af by the Köppen-Geiger system. Temperatures here average of 26.7 ° C. The annual rainfall average is 2992 mm. For example, one of the capital of the island of Borneo, namely Samarinda, factor very high rainfall can affect river water levels that can cause flooding. At the time it rains, people generally know the Samarinda city flood information only through social media and the public cannot monitor even knowing dots flooded areas in the city of Samarinda in real time.

In this paper, by using the proximity sensor HC-SR04 and Internet technologies are wireless sensor networks that use NodeMCU as the sender of the data of the level of real time into a monitoring platform that is placed in several locations prone to flooding, thus allowing urban Samarinda monitor in real time regions Samarinda city is prone to flooding.

II. Methodology

A. Wireless Sensor network

Wireless Sensor Network, also called Wireless Sensor and Actuator Network (WSAN), is a sensor network spatially distributed to monitor the physical and environmental condition, such as temperature, voice, pressure, etc. [3]. WSN was initially developed for the military interest which is used for battlefield surveillance, such as monitoring and controlling industry, machine health monitoring, etc. [4].

WSN is supported by a few nodes while each node connected to some sensors. Each WSN node has an internal antenna and connection to external antenna MCU, an electronic circuit for the interaction with the connected sensor and typically using battery or its own power supply. In addition, WSN has various topologies that range from simple network to multi-hop wireless network. Propagation technique for multi-hop networks can be directed or flooded [5]. Following here are a few examples of the implementation of WSN, Fig. 1.

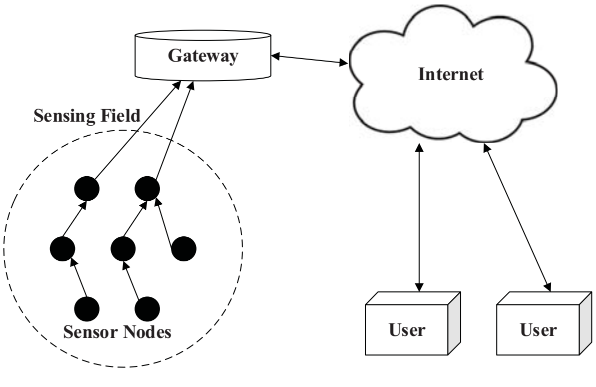


Fig. 1. Wireless Sensor Network System [6]

978-1-5386-8050-6/18/$31.00 ©2018 IEEE

B. NodeMCU main board

The NodeMCU is a development board and firmware is open source software that can help developers to create a product IoT. The NodeMCU developed to facilitate the use of the API for the input and output of a hardware. APIs can significantly reduce the complexity of configuring and manipulating the hardware. Design of The NodeMCU almost similar hardware like Arduino input and output, but the difference NodeMCU have Wi-Fi module ESP8266 very useful to be applied in the field of IoT and the price is very cheap [7].

ESP8266 is very well known by the developers of the field of IoT as Wi-Fi chips at its best, it is small and very cheap. Application ESP8266 only requires some additional circuitry that is integrated with 32-bit Tensilica microprocessor unit, a device interface digital and analog signals, antenna RF, amplifier power, low noise receive amplifier, filter and a module for managing power, it exists in a chip ESP8266 [8].

Fig. 2 shows the NodeMCU that use chips ESP8266 with several different functions on each pin. In this study, using 2 digital pins (D6 and D5), 1 pin VCC and GND pin. D6 pin is used to provide a trigger signal to the pin HC-SR04 and D5 are used to receive the status of 1 or 0 is sent by the HC-SR04.

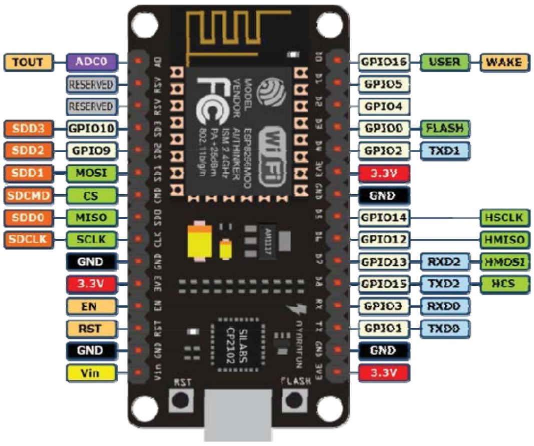


Fig. 2. NodeMCU. [https://www.marginallyclever.com/product/nodemcu-lua-wifi-v3-development-board-esp8266/]

C. HC-SR04 Sensor

Sensor HC-SR04 is a distance measuring sensor based on ultrasonic waves. The working principle of this sensor is similar to the ultrasonic radar. Ultrasonic waves are radiated then received back by the ultrasonic receiver [9]. The distance between transmit and receive time is a representation of the object distance. These sensors are suitable for electronics applications that require detection distance including to sensors on the robot, Fig. 3.

Fig. 3. Ultrasonic ranging module HC-SR04.

At the ultrasonic sensor, an ultrasonic wave generated by a piezoelectric device called a certain frequency. This will produce a piezoelectric ultrasonic wave (generally frequency 40 kHz) when an oscillator is mounted on the object. In general, these tools will be fired in the direction of ultrasonic waves or the target area. After a wave of target audiences, then the target will reflect the wave. Waves reflected from the target to be photographed by the sensor, then the sensor measures the distance between a transmitting and a reflected wave, Fig. 4.

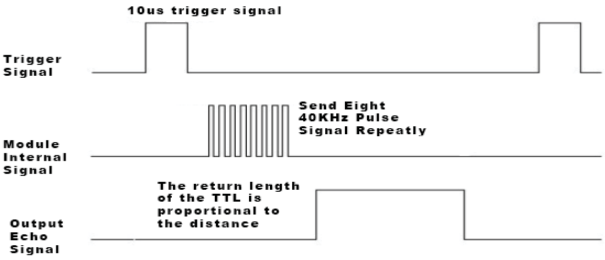

10us trigger signal
Trigger Signal
Send Eight 40KHz Pulse Signal Repeatly
Module Internal Signal
The return length of the TTL is proportional to the distance
Output Echo Signal

Fig. 4. Timing Diagram

In detail, the workings of the ultrasonic sensor are as follows: The signal emitted by the ultrasonic transmitter with a certain frequency and with a specific time duration.

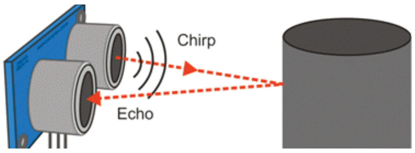


Fig. 5. HC-SR04 Sense Distance

The signal frequency above 20 kHz. To measure the object distance (proximity sensor), commonly used frequency is 40 kHz. The transmitted signal will propagate as sound waves travel at about 340 m / s. When light strikes an object, then the signal will be reflected by the object. Upon reflection wave to the receiving device, then the signal is processed to calculate the distance of the object. The object distance is calculated based on the formula, S = 340.t/2 (S is the distance between the ultrasonic sensor with an object (field reflection), and t is the time difference between the emission of waves by the transmitter and the time when a reflected wave received by the receiver) [10], [11].

III. Results and Discussion

A. Hardware structure Design

The system was built consisting of several sensor nodes placed at some point in the famous Samarinda city prone to flooding as seen in Fig. 6.

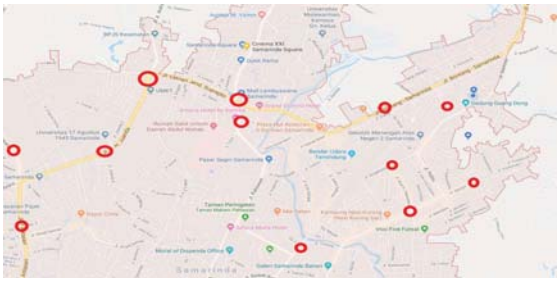


Fig. 6. Flood Prone Areas

B. Wireless Sensor Network System

Fig. 7 shows the overall structure of the node sensor. Sensor nodes can be seen consists of two main components, namely HC-SR04 a proximity sensor and NodeMCU as Wi-Fi module that sends data to the real-time monitoring platform.

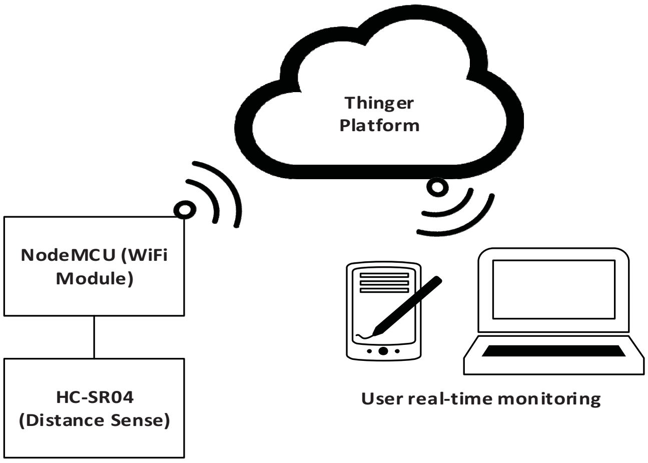


Fig. 7. Overall Structrue A Detection Water Level Based On Wireless Sensor Network

C. Designing Prototype

1) Hardware Module

The NodeMCU in programs using the C programming language that is compiled by an open-source Arduino software IDE. Use of the Arduino IDE software give the opportunity and convenience to users Arduino Kit to explore the advantages possessed by NodeMCU for the application of IoT in modern times. All of that is also easy with the many libraries that provided by the developer NodeMCU IoT for devices such as ESP8266WiFi.h, ThingerESP8266.h, and ArdinoJson.h. The following schematic electronics designed for this study, Fig. 8.

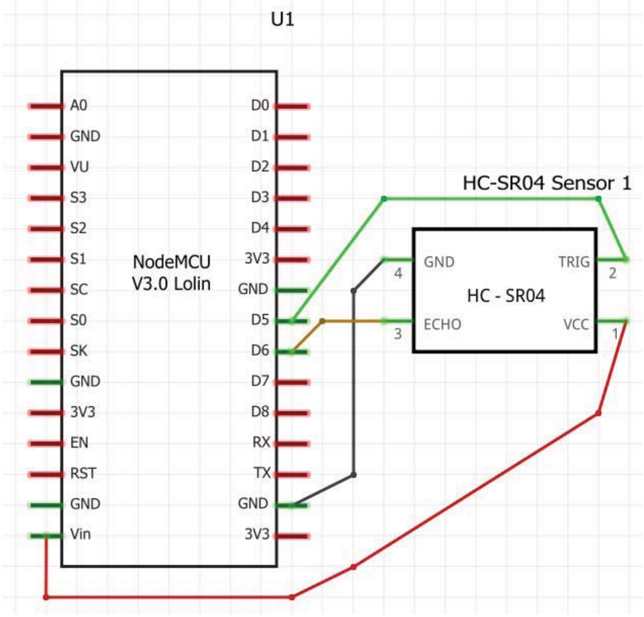


Fig. 8. Distanse Sense WiFi module schematic

The circuit in Fig. 9 is a circuit design on each sensor nodes deployed at some point prone to flooding in the city of Samarinda.

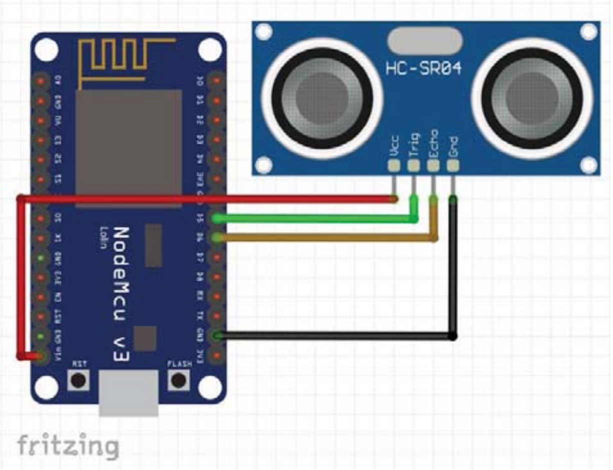


Fig. 9. Sensor Node Circuit Interface Diagram

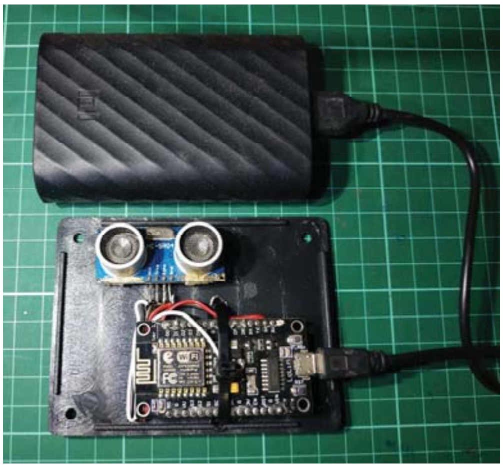
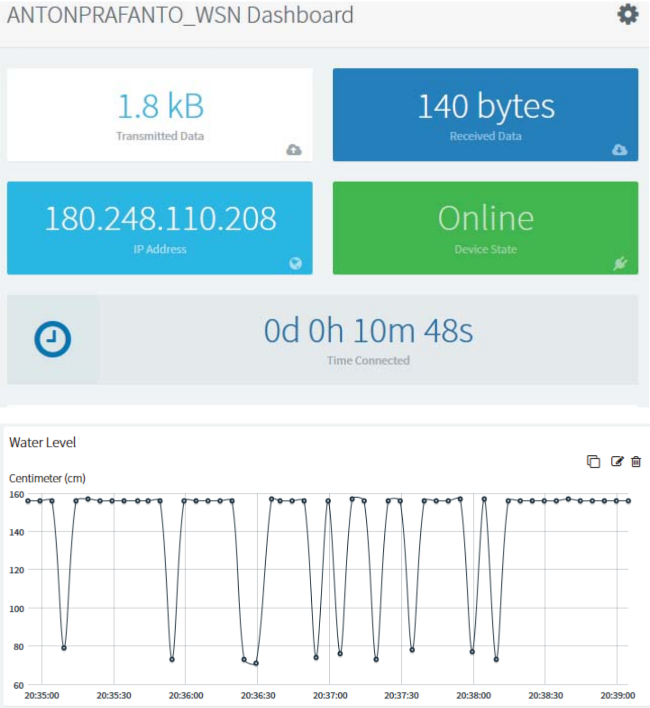


Fig. 10. Hardware Design

Fig. 10 shows when the NodeMCU sends HIGH signal to the proximity sensor through D5 pin connected to the TRIG pin HC-SR04, proximity sensor will transmit 40 kHz ultrasonic wave through a funnel transmitter. After that funnel HC-SR04 receiver receives the reflected wave and a LOW signal will be issued by ECHO pin, then sent to pin D6 at The NodeMCU. During transmission of ultrasonic waves has not been received by the mouthpiece receiver then sends a signal ECHO pin to pin D6 HIGH the NodeMCU.

D. Implementation

In this study, the NodeMCU programmed using the Arduino IDE Software libraries are known to have a very complete thinger.io platform, so the NodeMCU can easily connect with thinger.io IoT platform. Thinger.io is an IoT platform that can be integrated with many types of components is easy, for example, Arduino, Raspberry and the NodeMCU used in this study. At the time of the NodeMCU program, we need to include some identifier id The NodeMCU Wi-Fi to connect to the access point are available.



ANTONPRAFANTO_WSN Dashboard

1.8 kB
Transmitted Data

140 bytes
Received Data

180.248.110.208
IP Address

Online
Device State

0d 0h 10m 48s
Time Connected

Fig. 11. Dashboard of HC-SR04 Sensor in thingerio

Thinger.io provides multiple settings so that users can use the set of sensors and actuators, and data readings of the sensors is sent via the Internet and displayed on thinger.io platform.

In Fig. 11, the data is displayed in the form of a timing diagram that displays the water levels continuously during the appliance on and connected to the Internet network.

At the time of the experiment, the water level is fairly quiet but there is a little ripple so that the range of the measurement results water level changes are only in the range of 80 to 150 cm, plus another at the time of data collection, Samarinda not experience heavy rainfall. -

IV. Conclusion

In part III has described how connections between electronic circuits and programming is done on NodeMCU. ESP8266 Wi-Fi module on NodeMCU IoT technology lets developers connect the hardware to the server easily through the access point based on the ID which is embedded in the NodeMCU (credential). When connecting NodeMCU to thinger.io platform, make sure all the configuration and initialization data sent from the NodeMCU and will be accepted at thinger.io IoT platform is correct. Current water levels of the river at any point Samarinda city prone to flooding will be detected and all sensor data will be transmitted to and displayed thinger.io platform in order to facilitate urban Samarinda to monitor river water levels in real-time by utilizing the method of Wireless Sensor Network.

References

[1] D. A. Adyanti, A. H. Asyhar, and D. C. R. Novitasari, “Forecasts marine weather on Java sea using hybrid methods: TS-ANFIS,” in *4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), September 2017*, 2017.

[2] R. Zhang and C. Ting, “The study of the effects of El Nino on northeast monsoon using ERS scatterometer data,” in *Symposium Proceedings. (Cat. No.98CH36174). 1998 IEEE International Geoscience and Remote Sensing*, 1998.

[3] E. Kanagaraj, L. M. Kamarudin, A. Zakaria, R. Gunasagaran, and A. Y. M. Shakaff, “Cloud-based remote environmental monitoring system with distributed WSN weather stations,” in *2015 IEEE SENSORS, November 2015*.

[4] Y. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, “Mils-Cloud: A Sensor-Cloud-Based Architecture for the Integration of Military Tri-Services Operations and Decision Making,” *IEEE Syst. J.*, vol. 10, no. 2, JUne 2016, 2016.

[5] K. Sohraby, D. Minoli, and T. Znati, “Wireless Sensor Networks: Technology, Protocols, and Applications,” in *Wireless Sensor Networks: Technology, Protocols, and Applications*, John Wiley and Sons., 2006, pp. 1–307.

[6] V. Peiris, “Highly integrated wireless sensing for body area network applications,” 2013.

[7] S. S. and M. A., “Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system,” in *2017 Devices for Integrated Circuit (DevIC) March 2017*, 2017.

[8] L. K. B. Saputra and Y. Lukito, “Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266,” in *International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, 2017.

[9] C. Jeswin, B. Marimuthu, and K. Chithra, “Ultrasonic water level indicator and controller using AVR microcontroller,” in *International Conference on Information Communication and Embedded Systems (ICICES), February 2017*, 2017.

[10] M. A. Mutiara, A. A. G. Agung, and R. Handayani, “Sensor comparation for smart parking system,” in *2015 1st International Conference on Wireless and Telematics (ICWT), November 2015*, 2015.

[11] H. Jafari, X. Li, L. Qian, A. Aved, and T. Kroecker, “Evidence Theory Enabled Quickest Change Detection Using Big Time-Series Data from Internet of Things,” in *19th International Conference on Data Mining, Big Data, Database and Data System, June 15-16*, 2017.