IoT Based Water Level Control System

Steven Sachio[1], Agustinus Noertjahyana[1], Resmana Lim[2,*]
[1]Informatics Engineering Dept, Petra Christian University
[2]Electrical Engineering Dept, Petra Christian University
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
[*]Corresponding Author email: resmana@petra.ac.id

***Abstract*— Nowadays, almost all communication using the internet, but is not all, now object also can communicating each other, this concept is calls the Internet of Things (IoT). Things in IoT can be everything that we use every day. In this project, the "thing" is a water container, or in Indonesia, it called ”bak mandi". Why water container? Because, after we use water in the water container usually the level of the water container is reduced and we must fill it up for other people that use the water container. Generally, in the process of filling the water container, we forgot to close the valve or pump so water is overflowing and the water is wasted. That is why we propose to use the Internet of Things concept that can solve the problem. Our approach utilizes a controller of ESP8266 that can provide monitoring level of the water container. That controller will open and close the pump or valve automatically so that the water is not overflowing and wasted. We use an ultrasonic sensor to sense the level of the water. We utilize the Blynk IoT service incorporated with PHP web programming in providing water level monitoring and control. We have tested the system on a 64 cm water container. The system has an error of 2 cm in controlling the water level.**

***Keywords:* Arduino ESP8266, Internet of Things, IoT, Microcontroller, Water level control**

I. INTRODUCTION

The internet has become a requirement for the world community. With the internet, people can communicate with each other very easily and quickly. With internet people from different continent easily can interact with people from another continent. The internet is not just connecting people but things can also communicate with other objects. That concept is called an Internet of Things.

The concept Internet of Things (IoT) is the concept where object have the ability to transfer data over the internet without requiring human to human or human to computer interaction [1]. A Things in IoT can be an object that we use every day such as heart implant monitoring, a sensor that can remind driver when tire pressure is low, a trash can that will inform a user when the box is full. Internet of things is very closely associated with the machine to machine communication.

Internet of Things is usually supported by the device/controller that used to carry sensors and actuators such as Intel Edison, Intel Galileo, Raspberry Pi, Arduino based processor and so on. By using those boards, data from sensors can be monitored and controlled remotely via the Internet.

In the future Internet of Things is used to get all kind of data from the sensor, so the data can be used to analyze the market, provide data for the company for reference when creating a product, and many more.

The water container is one of the items we use daily to hold water before being used for bathing, washing clothes and so on especially in Indonesia. Whenever it is used, the water in the container must be decreases and we have to fill it up with the water so that it can be reused by other people who will use it later. In the process of filling the water usually, we have to wait for the water in the water container to be full. That's time to wait we often forgot to close pump or valve when filling the water container because the time to wait for the water to be full is so long and often we lazy to wait and leave the pump or valve. So When we forgot to close the pump or valve and the water container is becoming full, this is the point when water is overflowing and being wasted.

II. MATERIALS AND METHODS

A. *Internet of Things with ESP8266*

The Internet of Things has two keywords, internet and things. Internet, which stands for interconnection-networking, is a network of the computer that connect to each other using TCP/IP (Transmission Control Protocol/ Internet Protocol).

Things in the Internet of Things are the object that we use daily where the information of that things retriever from sensors that read environment around in real time without human intervention. Such as room temperature, humidity, and pressure.

So Internet of Things means objects that can generate data through sensors and the data sent to the server or computer via the internet connection. The Internet of Things is also very closely related to the communication Machine to Machine (M2M) that can communicate without any human intervention in it [1]. Internet of Things is also the first step in making Smart World where mixing of data from objects and smart city [2]. The Internet of Things has 3 crucial component:

1. A Hardware that consist of sensors, actuators, microcontroller and communication device.
2. A Middleware that consists of data storage a data analytic capability.
3. A presentation that consist of easy understand visualization data from middleware and provide useful information.

ESP8266 Microcontroller is a microcontroller designed by Espressif System, ESP8266 designed to have Wi-Fi directly integrated with [3], so ESP8266 doesn't require Wi-Fi module. If ESP8266 compared with Arduino, ESP8266 has some weakness it come from the number of analogue pins that

ESP8266 has Arduino has five analogue pins while ESP8266 only has 1 [4]

Ultrasonic is the sound vibration at frequencies above 20khz. In ultrasonic sensor, there are two modules in one circuit package, the receiver module and transmitter module. The transmitter is used to generate and transmit ultrasonic waves toward the observed objects. When an ultrasonic wave touches or hits an object, the ultrasonic wave bounces and is captured by the receiver module. The similar project of performed water level measurement by using an ultrasonic sensor, however, the level was just transmitted via wireless transceiver locally [5].

In this project, we have extended the water level measurement and control by using the Internet. That possible multi-device connected by using Blynk apps [6] and Blynk Server [7]. By using Blynk, the device can be controlled using mobile device and web control.

III. SYSTEM DESIGN

In system design, there are consists of two parts, The first is software system and the second is a hardware system. The software system combines between Water Container Monitoring Information System and Blynk IoT server

A. Hardware System

In fig 1 there is a system diagram of the design of IoT Based Water Level Control System. The system consists of a microcontroller, pump, adaptor, sensor ultrasonic, and a relay module to control the pump or valve on the hardware side. At the image microcontroller that we use is Wemos D1_R2 where ESP8266 based, this microcontroller generates a trigger to make relay switch open and close. On the top of the water container, we place an ultrasonic sensor to read the level of the water container, the level that read from ultrasonic transducer is to decide when the pump is working and not working. On the ultrasonic sensor there a two cable for trigger the ultrasonic sensor and receive the echo of ultrasonic sound.

The microcontroller communicating via Wi-Fi to the internet to connect with Blynk server and Web server (Water Container Monitoring Information System)

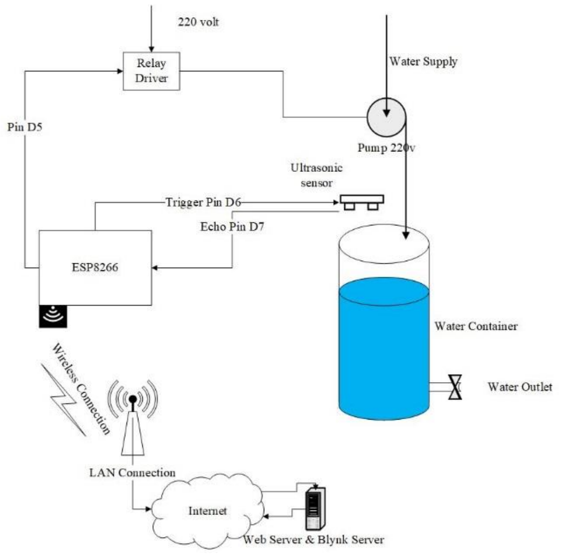

Figure 1 Hardware System

B. Software System

Software system in this project using Water Container Monitoring Information System incorporated with a Blynk IoT server. The combination is necessary because if we only use Blynk system, the Blynk system is only limited to be accessed on smartphone application so it would be less flexible for Concept of the Internet of Things.

The communication schema between Water Container Monitoring Information System and Blynk server is provided at fig 2

On fig 2, the microcontroller is communicating on Blynk server and a PHP web-based Water Container Monitoring Information System. Wemos D1 R2 send data level every 4 seconds to Water Container Monitoring Information System using protocol GET. Data that send via GET protocol is using the token of microcontroller, the water level of water container and valve or pump status. From microcontroller, Water Container Monitoring Information System check is there any token on the database? If token exists, so the data from microcontroller will be saved in the database, otherwise if the token from microcontroller is not exist so the data will not be collected in the database.

In Water Container Monitoring Information System, data of water level and valve/pump status is saved with a timestamp, so every data has its timestamp.

Wemos D1_R2 also send data to Blynk server but data that send to Blynk server only save into file not into the database so that why we need a Water Container Monitoring Information System. Beside microcontroller send to data to Blynk server microcontroller also receive configuration data from Blynk server. The configuration data is water container height, Max upper-level limit, lower level limit, mode and on/off the pump on configuration mode divided into two modes, which is auto and semi-auto. In auto mode, pump/valve automatically fill water container if the level is below the lower level limit and stop when the level of water is above or same as the upper-level limit. In semi-auto mode, if on/off in position on, the microcontroller works like auto mode, but if on/off in position off, the pump/valve will stay off until on/off in place on.

A PHP Web-based Water Container Monitoring Information System is also communicating with Blynk server for change configuration using API (Application Programming Interface) so the user can change configuration data on Blynk server using Water Container Monitoring Information System also from Blynk Application on the smartphone.

On fig 3 there is a picture that shows a flowchart of microcontroller coding. On the flowchart, there is a process that gets level from the sensor. On that process calls ping process, where sensor ultrasonic shoot ultrasonic pulse to the water surface and capture ultrasonic pulse reflection from the water surface. The different time between sensor ultrasonic shoot pulse and receive pulse must be converted into the distance (cm) using the formula of [8]:

$$length(cm) = milisecond\ (ms) \div 29\ (microseconds\ per\ centimeter) \div 2$$

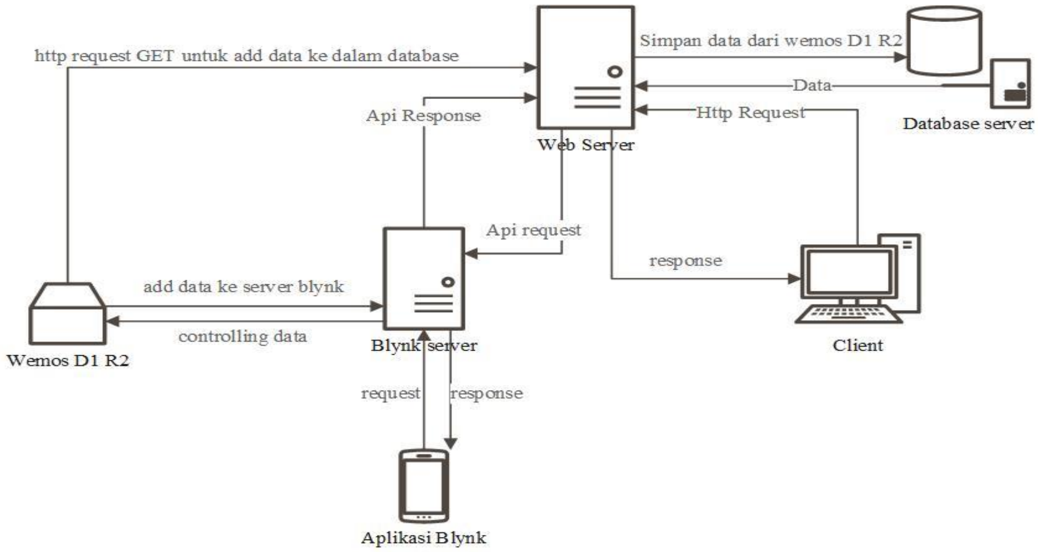

Figure 2 Communication schema

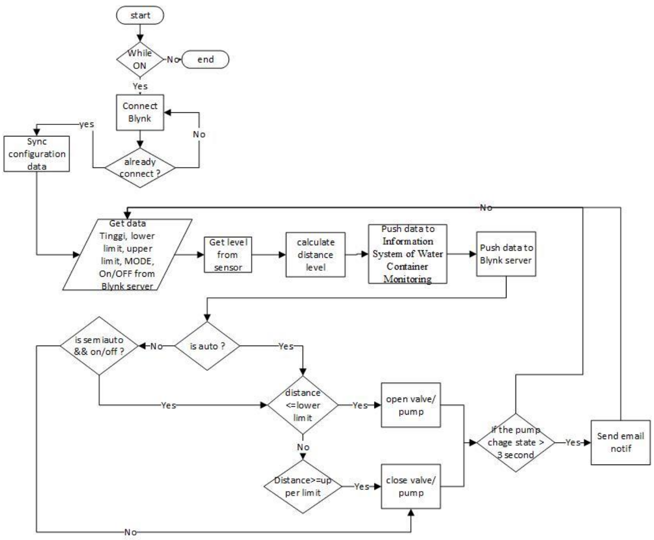

Figure 3 Flowchart

After getting water level from the sensor, the next step is subtracting level from the sensor with a height of water container because if the water container is near full, the level from the sensor is shrinking, so to eliminate this we must subtract level from the sensor with the height of water container.

Push data to Water Container Monitoring Information System is run every 4 seconds so the server is not overloading. Push data to Blynk server run every 1 second.

IV. Implementation and Testing

A. Hardware System

On fig 4 is an image show how hardware system connected each other. For the ultrasonic sensor, it has 5 pin vcc, trig, echo, out, gnd. Vcc pin connected with 5 v output on Wemos D1_R2 microcontroller, Trigger pin connected with pin D6 on Wemos D1_R2 microcontroller, echo pin connected with pin D7 on Wemos D1_R2, out pin is not used, gnd pin is connected with ground pin on Wemos D1_R2. For relay module trigger using pin D5 on Wemos D1_R2. Vcc and gnd on relay module connected same as ultrasonic sensor. Relay module common pin is connected with pump cable with normally close configuration. An Adaptor is used for providing power to Wemos D1_R2 microcontroller.

After all hardware is connected, wevassembly to the water container model as fig 5. An ultrasonic sensor is installed at the top of the water container. Microcontroller, adaptor and relay module is mounted on the little container, so it's not exposed with water. A pump is installed in the water container model.

Figure 4 Hardware System

Figure 5 Hardware System & water container

B. Software system

Results of the Water Container Monitoring Information System. Overview of the application, in general, can be seen in Figure 6

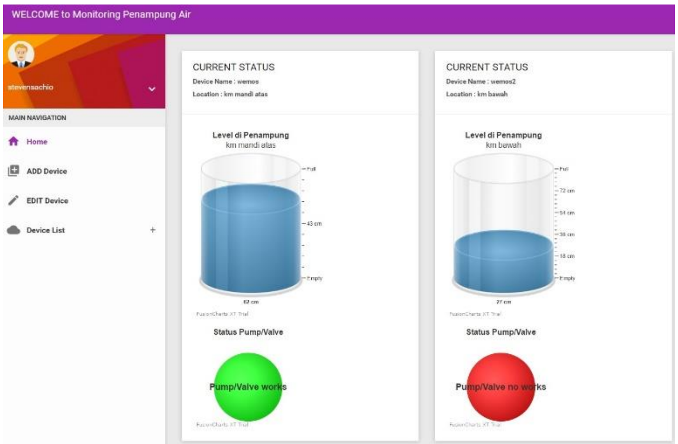

Figure 7 Home Page

The device list page is the page that displays the data stored in the database in the form of a chart and to change the configuration. This can be seen in Figures 7 and 8

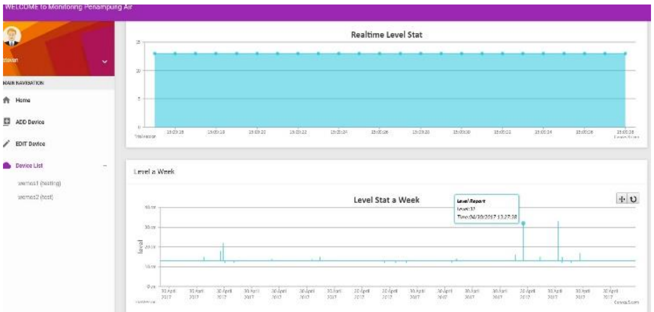

Figure 9 Device List Page (Chart)

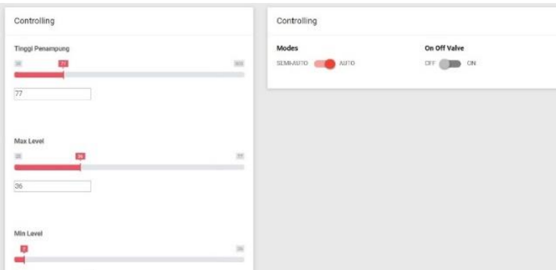

Figure 8 Device List Page (Controlling)

C. Testing of Ultrasonic sensor on a static water level

The Testing is a process to answer the hypothesis: could the ultrasonic sensor read level of the water container and how good the accuracy? From testing the ultrasonic sensor to read water level on static condition 0cm, 10 cm, 25 cm, and 45 cm(maximal) with several timers on ping process: without a timer, timer 1ms, timer 10ms, 60ms. Data is taken from 50 data in the database

Data result is:

Table 1
Comparison reading level at water container empty using the ultrasonic sensor

Comparison of the reading level at water container empty				
	Without Timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Average Error Result	35.36 cm	16.68 cm	0.02 cm	0 cm

Table 2
Comparison reading level at water container 10 cm using ultrasonic sensor

Comparison of the reading level at water container 10 cm				
	Without Timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Average Error Result	1.32 cm	0.58 cm	1 cm	1 cm

Table 3
Comparison reading level at water container 25 cm using ultrasonic sensor

Comparison of the reading level at water container 25 cm				
	Without Timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Average Error Result	0.14 cm	0.02 cm	0 cm	0 cm

Table 4
Comparison reading level at water container 45 cm using ultrasonic sensor

Comparison of the reading level at water container 45 cm				
	Without Timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Average Error Result	0.02 cm	0.04 cm	0 cm	0 cm

That's the result of the accuracy of the ultrasonic sensor on the static water level.

D. Testing of the ultrasonic sensor on working mode

The testing data on working mode can be shown on tab 5. In this testing configuration, the setting is the height of water container is set at 64 cm, upper limit 35 cm, lower limit 15 cm, the mode is auto.

Table 5
Water Container work experiment with high configuration 64cm, upper limit 35cm and lower limit 15cm

Trial no-	Pump on at (cm)	Pump off at (cm)
1	15	34
2	14	31
3	14	34
4	15	32
5	14	34
average	14.4	33
difference	0.6 cm	2 cm

On working mode, ultrasonic sensor makes an error at the pump on and pump off. At pump condition on an average error with actual level is 0.6 cm, but with the pump status OFF, the average error is increased to 2 cm. This increase of error due to whenever water container is filling the water surface is not flat any more.

E. Connection Microcontroller Testing Esp8266

This section is testing how connection microcontroller (ESP8266) to Wi-Fi router using security protocol WPA-PSK, WPA2-PSK, and WEP. The result is on table 6.

Table 6
Microcontroller Connection with Wi-Fi Security Protocol

Security Protocol	Encryption Protocol	Connection
WPA-PSK	AES	Ok
	TKIP	Ok
WPA2-PSK	AES	Ok
	TKIP	Ok
WEP	-	Ok

V. Conclusion

From the project experiment, we conclude as follows:

- From experiments between pings without using a timer and by using a timer. It can be concluded that the longer the delay given the more accurate the sensor readings to the water level.
- From experiments, the water container at working mode can be concluded that when the pump from running to stop, the error of reading level from the sensor because it is caused by the flow of water that causes the surface of the water is not flat so the reflection pulse not perfect any more.
- From the connectivity experiments with Wi-Fi it can be concluded that Microcontroller ESP8266 can be connected with Wi-Fi with security protocol WPA-PSK, WPA2-PSK, as well as WEP protocol.
- From experiments, ultrasonic sensor can be used as water monitoring with error threshold between 2 cm

References

[1] C. G. Garcia, D. M. Llorian, B. P. G-Bustelo, J. M. C. Lovelle and N. Garcia-Fernandez, "Midgar:Detection of people through computer vision in the Internet of Things scenarios to improve the security in SmartCities, Smart Towns, and Smart Homes," *Future Generation Computer System,* 2015.

[2] J. Gubbi, R. Buyya, M. Palaniswami and S. Marusic, "Internet of Things (IoT):A Vision, architectural elements, and Future Directions," *Future Generation Computer System,* pp. 1645-1660, 2013.

[3] Espressif System IOT Team, "ESP8266EX," Espressif System, 2015.

[4] M. Singh and S. Goel, "Review of Arduino Microcontroller," p. 2, 2013.

[5] S. Jatmiko, A. B. Mutiara and M. Indriati, "Prototype of Water Level Detection System With Wireless," pp. 52-59, 2012.

[6] BlynkTeam, "Overview of blynk app," 2015. [Online]. Available: http://www.blynk.cc/. [Accessed 21 October 2016].

[7] BlynkTeam, "Blynk-Server," 2015. [Online]. Available: https://github.com/blynkkk/blynk-server. [Accessed 21 October 2016].

[8] M. Margolis, Arduino Cookbook, United States of America: O'Reilly, 2012.