Published online on the journal's website:https://jes-tm.org/index.php/jestm/index

Journal of Engineering Science and Technology Management

| ISSN (Online) 2828 -7886 |

Article

Prototype of Water Rise Detection Tool in Kampar River Fish Cages Based on Internet of Things (IoT)

Muhammad Bayu Setiawan[1]✉, Emon Azriadi[2], Joko Musridho[3]

[1,3]Informatics Engineering Study Program, Pahlawan Tuanku Tambusai University, Riau

[2] Industrial Engineering Study Program, Pahlawan Tuanku Tambusai University, Riau

DOI: 10.31004/jestm.v6i1.349

E-mail:sbayu4605@gmail.com

ARTICLE INFORMATION

Volume 6 Issue 1
Received: 25 December 2026
Accepted: 11 February 2026
Publish Online: 10 March 2026
On line: at https:JESTM.org/

Keywords

Fish Cage
Water Level Detection
Waterproof Ultrasonic Sensor
ESP32
Internet of Things
Vessel
R&D

ABSTRACT

Fish cages, classified as vessels under the Indonesian Shipping Law Number 17 of 2008, must consider safety aspects due to risks such as sinking and drifting. Increasing river discharge causes damage to the cage structures, potentially worsening fish health and causing economic losses for fish farmers. This study aims to develop a water level detection device for fish cages in the Kampar River using Internet of Things (IoT) technology based on waterproof ultrasonic sensors and the ESP32 microcontroller. The device can automatically monitor water levels and provide real-time alerts to cage owners via wireless Wi-Fi or Bluetooth connections. The research uses a Research and Development (R&D) approach. This innovation is expected to help fish cage owners anticipate risks from changes in water levels and improve the management of fish farming operations

1. Introduction

Based on Law Number 17 of 2008 concerning shipping(Indonesia, 2008)The definition of a ship is an object that is propelled by wind power, mechanical power, or other energy, towed or towed, including dynamically supported vehicles, underwater vehicles, and floating devices and floating structures that do not move. In line with this definition, cages can be categorized as ships so that ship draft regulations apply. Every ship is required to implement safety aspects. This is because many cages have the potential to sink, drift, and suffer other damage, which can cause losses to the community. Cage farming is a fish farming system in a container in the form of a net that floats with the help of buoys and is placed in waters, such as rivers, lakes and reservoirs. Fish produced through cage farming systems can meet the requirements for export, namely uniform size, brighter and clearer fish colors, no muddy odor and clean meat.(Furqan & Maghfirah, 2022)

One of the main factors causing damage to fish cages is the increase in river water levels accompanied by strong currents. Rising water discharge can damage cage structures, cause fish to escape, and trigger changes in water quality that negatively affect fish health. In addition, debris and sediment carried by river currents can further deteriorate the water conditions around the cages. These problems often occur suddenly and are difficult to predict, highlighting the need for a monitoring system capable of detecting water level changes quickly and accurately.

Based on these issues, development of fish cage construction and safety systems is necessary, one of which is a water level monitoring device capable of providing early warnings. This water level detection device, based on a waterproof ultrasonic sensor and ESP32 integrated with the Internet of Things (IoT), allows fish cage owners to monitor water conditions in real time and take immediate action to minimize the risk of loss, especially during extreme weather conditions.

A water level detector is a device that automatically measures and monitors water levels. Waterproof ultrasonic sensors work by detecting water levels and sending data to a microcontroller for processing. When the water level reaches a certain threshold, the system will issue a warning via an LED or buzzer. This tool is commonly used for monitoring floods, water tanks, and controlling tides (Khair, 2020).

The Internet of Things (IoT) is a technology that enables various smart devices to connect to each other via the internet to exchange data and operate automatically in real time. With ESP32 wireless connectivity support based on Wi-Fi or Bluetooth, a water level detector can send data and notifications directly to the owner's device, such as a smartphone, so that preventative measures can be taken immediately (Selay et al., 2022).

This research employed a research and development (R&D) methodology, encompassing needs identification, design, testing, evaluation, and revision. Therefore, this study aims to develop an IoT-based water level detection device for fish cages in the Kampar River to help cage owners manage their aquaculture operations more effectively and minimize the risk of loss.

2. Literature Review

2.1 Fish cages

Fish cages are a widely used aquaculture method for raising fish in water bodies such as rivers, lakes, reservoirs, or the sea by suspending nets tied to floating frames. Cages allow for intensive and productive fish farming because they can be raised in large volumes of water with efficient management. Common fish species cultivated in cages include tilapia, catfish, gourami, and carp, each with specific environmental requirements and stocking densities for optimal growth. These cages are typically made from materials such as wood, bamboo, or iron, equipped with floats and anchorage systems to maintain stability in the waters.(Wardani & Karimah, 2022).

Analysis of fish cage farming indicates that it is economically viable, generating positive net income, although feed and labor costs significantly impact profitability. Furthermore, cages can also serve as fishing tourism attractions, supporting the local economy. Therefore, proper management and appropriate site selection are essential for optimal and sustainable fish cage farming.(Hidayati et al., 2020).

2.2 Internet of Things (IoT)

The Internet of Things (IoT) was first introduced by Kevin Ashton in 1999 as a technological concept that allows physical objects to transfer data over a network without requiring direct human interaction. IoT utilizes technologies such as Radio Frequency Identification (RFID) to increase efficiency and allows objects to transmit data autonomously over a network without the assistance of computers or humans (Turyadi, 2021).

2.3 Waterproof ultrasonic sensor

A waterproof ultrasonic sensor is a device that uses ultrasonic waves to detect the presence and distance of objects in watery environments or extreme weather. These sensors are equipped with special protection so they remain functional even when exposed to moisture, rain, or submersion in water, making them suitable for use in flood detection devices to measure water levels. The sensor works by emitting a signal to an object and receiving the reflected signal back (Admanugraha & Damsi, 2025).

Table 1. Sensor specifications

Specification	Information
Operating Voltage	5V DC
Workflow	30 mA (maximum)
Measurement Distance	20 cm – 600 cm
Accuracy	±1 cm
Detection Angle	Less than 15°
Ultrasonic Frequency	40 kHz
Output Type	TTL Level (Trig and Echo)
Response Time	Less than 100 ms
Sensor Type	Waterproof (waterproof, suitable for outdoor)

2.4 ESP32

The ESP32 is a high-performance microcontroller developed by Espressif Systems, equipped with a dual processor, integrated WiFi and Bluetooth modules, and approximately 36 GPIO pins supporting ADC and DAC. The ESP32 is widely used as the main controller in IoT-based monitoring and control systems, such as security systems, temperature control, and remote control. Its main advantages are wireless connectivity and large memory capacity.

adequate, so it is very suitable for Internet of Things (IoT) applications (Lesmana & Silalahi, 2020).

2.4 Research and Development (R&D) Methods

The research and development (R&D) method is an increasingly popular research method among academics today, especially in designing and testing product effectiveness. The main goal of this method is to create a product through the process of identifying potential problems and designing and developing optimal solutions. By using this research and development method, researchers can take scientific steps to collect the necessary data, making it easier for them to produce, develop, and validate the resulting product.(Waruwu, 2024).

2.5 Monitoring system

A monitoring system is a system used to continuously observe and measure changes in specific parameters to support decision-making. In environmental applications, it typically consists of sensors for data collection, a processing unit for data analysis, and an interface for data display. In water level monitoring, such a system plays an important role in detecting changes that may pose risks to safety and infrastructure, as continuous measurement enables early identification of abnormal conditions and supports preventive actions, making it a fundamental component of IoT-based water level detection for fish cage management.

2.6 Early Warning System (EWS

An Early Warning System (EWS) is a system designed to detect potential hazards early and provide timely warnings to reduce risks and losses. Generally, an EWS consists of risk identification, monitoring and detection, warning delivery, and response actions. The main function of an EWS is to convert monitoring data into clear and actionable alerts for users.

In water level monitoring, EWS commonly uses a threshold-based mechanism, where warnings are triggered when water levels exceed predetermined safety limits. The use of IoT technology enables warnings to be delivered in real time through digital platforms such as mobile applications, allowing fish cage owners to respond quickly to rising water levels and reduce potential damage.

3. Research Methodology

This study employed a Research and Development (R&D) methodology to develop and test a prototype Internet of Things (IoT)-based water level detection device for fish cages. This approach was chosen because the research focused not only

on data collection but also on designing, building, and testing a system that could be directly applied in the field.

3.1 Location and time of research

The research was conducted in fish cages in the Kampar River, covering Kuok, Bangkinang, and Air Tirs Districts. The study was conducted in stages from February to June 2025.

3.2 Research subjects

The research subjects consisted of fish cage owners and managers. The cage owners act as capital providers and business decision-makers, while the managers are responsible for daily operations and monitoring the cages' condition. The involvement of both parties is crucial for obtaining accurate data on rising water levels and the need for a developed monitoring system.

3.3 System development stages

The system development stages include:

Analysis and identification of needs

This stage was conducted to identify problems with rising water levels in fish cages and user needs for a monitoring system. Data was obtained through direct observation at the cage locations and interviews with fish cage owners and managers.

System planning

At this stage, a prototype water level monitoring system was designed, consisting of hardware and software. The hardware used a waterproof ultrasonic sensor and an ESP32 microcontroller, while the software was developed using the Arduino IDE and integrated with an IoT platform

.

Prototype Assembly

The assembly phase involves assembling and connecting all hardware components according to the system design. Next, the program is uploaded to the microcontroller to activate sensor reading, data processing, and data transmission to the IoT platform.

System Testing.

Testing was conducted to ensure the system performed as designed. Testing included the sensor's accuracy in detecting water levels, the ESP32's ability to process and transmit data, and real-time data display on the LCD and IoT platform.

Evaluation and Revision

Test results are analyzed to identify system weaknesses. Evaluation and improvements are then carried out to ensure the tool functions optimally and meets user needs.

3.4 Data collection technique

Data collection techniques included literature studies, field observations, and interviews with fish cage owners and managers. The data obtained served as the basis for system design and evaluation.

3.5 Data validity techniques

The validity of the data was tested using source triangulation and technical triangulation, namely by comparing data from observations, interviews and
literature studies to ensure the consistency and validity of the research data.

3.6 Data analysis

Data analysis was conducted descriptively, categorizing data based on rising water levels, user needs, and system testing results. The results were used to conclude the effectiveness of the IoT-based water level detection tool in supporting fish cage management.

4. Results and Discussion

4.1 Block diagram design

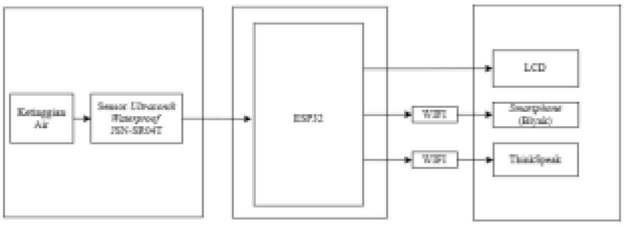

Figure 1. System block diagram

The system block diagram shows the integration of a waterproof ultrasonic sensor as a water level detector with the ESP32 as the main processing unit. The ESP32 processes the sensor readings and sends them to the Blynk IoT platform for water level visualization and alert notifications, and to ThingSpeak for monitoring and data storage in spreadsheet format. Additionally, an LCD is used as a local display medium to display water level values in real-time.

4.2 Hardware design

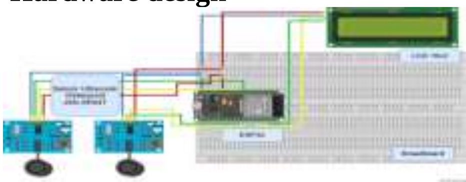

Figure 2. Hardware

The figure shows the system's hardware design. The JSN-SR04T waterproof ultrasonic sensor is used to measure the water level. The measurement data is then processed by an ESP32 microcontroller equipped with a Wi-Fi module for wireless data transmission. Additionally, a 16x2 I2C LCD is used as a display medium to display water level data in real time.

4.3 Software design

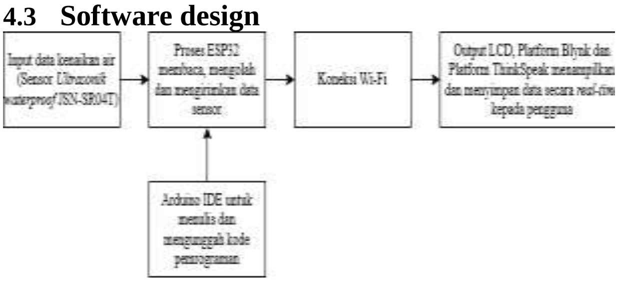

Figure 3. Software architecture

The system software consists of an embedded program on the ESP32 and the Blynk and ThingSpeak IoT platforms. Embedded software development was carried out using the Arduino IDE with the C++ programming language, which was chosen because it is open-source, supports the ESP32, and provides a comprehensive library. The ESP32 processes data from the ultrasonic sensor and sends it via a Wi-Fi connection to Blynk IoT for data visualization and alert notifications, and to ThingSpeak for storing and periodically monitoring water level data.

4.4 Hardware assembly

The circuit design is done by connecting the ESP32 microcontroller, sensors, and other supporting devices via a breadboard. The breadboard simplifies connection distribution due to the ESP32's limited pin count. The breadboard serves as a connection medium between components using jumper cables.

Figure 4.I2C 16x2 LCD Configuration

4.5 Software assembly

The software assembly of the water level monitoring system was carried out using Arduino IDE version 2.3.6 with the C++ programming language. The program was embedded in the ESP32 as a data processing unit for the JSN-SR04T ultrasonic sensor, which then displays data on a 16×2 I2C LCD and sends it to the Blynk IoT and ThingSpeak platforms via a Wi-Fi connection. System development requires the installation of an ESP32 board and supporting libraries, namely BlynkSimpleEsp32, LiquidCrystal_I2C, and NewPing, as well as the selection of the DOIT ESP32 DEVKIT V1 board so that the program compilation and upload process runs correctly.

Table 2.ESP32 program code on Arduino IDE

Program Code	Function
#define BLYNK_TEMPLATE_ID "TMPL6YBZEgKRJ" #define BLYNK_TEMPLATE_NAME "Monitoring of Water Level Conditions in Fish Cages" #define BLYNK_AUTH_TOKEN "uBAZldnUuhVvC8_WtYjXklIdI2W46mUC"	Defines the Blynk IoT platform id template Defines the Blynk IoT name template Defining Blynk IoT token authentication

Program Code	Function
#include <WiFi.h> #include <WiFiClient.h> #include <HTTPClient.h> #include <BlynkSimpleEsp32.h> #include <Wire.h> #include <LiquidCrystal_I2C.h> #include <NewPing.h>	Importing libraries: • WiFi.h : WiFi connection • Wifi Client.h : Wifi Client • BlynkSimpleEsp32.h : Blynk IoT • Wire.h : I2C • LiquidCrystal_I2C.h:LCD • NewPing.h :LCD I2c
Char ssid[] = "Masbay"; char pass[] = "55555555";	Save Wifi network name (SSID) and save Wifi password

const char* THINGSPEAK_API_KEY = "CKIYK3ZQYVGVBSF0";
const int THINGSPEAK_CHANNEL_ID = 2990854;
unsigned long lastThingSpeakTime = 0;
const unsigned long thingSpeakInterval = 60000;
	Manage data sending from ESP32 to ThingSpeak platform.
	
	Stores the authentication key so the device can send data to the appropriate channel, indicating the destination channel ID.
	
	The variable is used to record the last time the data was sent, determining the time interval between sends, namely every 60,000 milliseconds or 1 minute.

#define TRIG_PIN_1 5 #define ECHO_PIN_1 18 #define TRIG_PIN_2 25 #define ECHO_PIN_2 26 #define MAX_DISTANCE 200 NewPing sonar1(TRIG_PIN_1, ECHO_PIN_1, MAX_DISTANCE); NewPing sonar2(TRIG_PIN_2, ECHO_PIN_2, MAX_DISTANCE);	Setting up two ultrasonic sensors with libraries on ESP32. Defining Sensor 1 and Sensor 2 Set 200 cm as the maximum limit for distance measurement. Initialize each sensor to read the distance of an object using ultrasonic pulses.
LiquidCrystal_I2C lcd(0x27, 16, 2);	initialize I2C LCD

Program Code	Function

bool notifSent1 = false;
bool notifSent2 = false;
 	
void setup() {
Serial.begin(115200);
 	
Wire.begin();
lcd.begin(16, 2);
lcd.backlight();
lcd.clear();
lcd.setCursor(0, 0);
  lcd.print("Connecting...");
 	
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
 	
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("WiFi Connected");
}
 	
void loop() {
Blynk.run();
 	
int distance1 = sonar1.ping_cm();
int distance2 = sonar2.ping_cm();
 	
if (distance1 == 0) distance1 = -1;
if (distance2 == 0) distance2 = -1;
 	
Blynk.virtualWrite(V0, distance1);
Blynk.virtualWrite(V1, distance2);
 	
// Notification if water level rises
if (distance1 > 0 && distance1 < 50 && !notifSent1) {
    Blynk.logEvent("water_approaching_sensor",
"Warning: Water rising approaching Sensor 1!");
notifSent1 = true;
} else if (distance1 >= 50) {
notifSent1 = false;
}
 	
if (distance2 > 0 && distance2 < 50 && !notifSent2) {
    Blynk.logEvent("water_approaching_sensor",
"Warning: Water rising near Sensor 2!");
notifSent2 = true;
} else if (distance >= 50) {
notifSent2 = false;}
	Setting Notifications on Blynk IoT for monitoring results of sensor 1 and sensor 2

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sensor1:");
  lcd.print((distance1 != -1) ? String(distance1) + "cm"
: "Danger");
 	
lcd.setCursor(0, 1);
lcd.print("Sensor2:");
  lcd.print((distance2 != -1) ? String(distance2) + "cm"
: "Danger");
	Set the LCD display to display the monitoring results.

Program Code	Function

if (millis() - lastThingSpeakTime > thingSpeakInterval)
{
sendToThingSpeak(distance1, distance2);
lastThingSpeakTime = millis();
}
 	
delay(1000);
}
 	
void sendToThingSpeak(int data1, int data2) {
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
String url =
"http://api.thingspeak.com/update?api_key=" +
String(THINGSPEAK_API_KEY)
+ "&field1=" + String(data1)
+ "&field2=" + String(data2);
http.begin(url);
int httpCode = http.GET();
if (httpCode > 0) {
Serial.println("ThingSpeak Response: " +
String(httpCode));
} else {
Serial.println("Error Sending to ThingSpeak");
}
http.end();
}
}
	Sending monitoring data to the ThingSpeak platform.

Table 2 above explains the function of the program code developed using the Arduino IDE and uploaded to the ESP32 microcontroller.

Figure 5. View Blynk IoT Cloud Stored Events & Notifications

4.6 Testing

Testing was conducted in two stages, consisting of individual ultrasonic sensor testing and live system testing. The ultrasonic sensor was evaluated by comparing the actual distance between the sensor and the object surface with the measured sensor readings to assess accuracy, stability, and response to distance changes. The results showed that the JSN-SR04T waterproof ultrasonic sensor produced accurate and stable measurements within acceptable tolerance limits, indicating that the sensor is suitable for use as the main component

in a real-time IoT-based water level monitoring system.

4.6.1 System testing on the container

The initial testing was conducted using a water container to simulate the monitoring environment. This test evaluated the system's response to gradual increases in water level and its ability to classify water conditions into *Normal* and *Danger* based on a 50 cm threshold. The results served as preliminary validation of the system's performance before implementation in real and more complex environments, such as rivers.

Figure 6. System testing on the container

Test results on a water container showed that the system was able to respond consistently to changes in water level. Ultrasonic sensors detected the distance to the water surface with good accuracy, and the system sent notifications according to the detected conditions. When the water level reading fell below the 50 cm threshold, the system automatically classified the status as Danger and sent an alert. These results indicate that the system functioned well and was suitable for testing in real-world environments, such as fish cages and river flows.

4.6.2 Testing in real environment

The following are the thresholds for

determining the condition status in testing in a real environment, which can be seen in the following table.

Table 4. Testing Threshold Table in Real Environment

No	Condition Status	Water Level
1.	Danger	<50 cm
2.	Normal	>50 cm

Table 5. Testing in real environment

No	Sensor Distance 1 (cm)	Sensor Status 1	Place Description	Sensor distance 2 (cm)	Sensor Status 2	Place Description	Notification
1	22 cm	Danger	In the water surface	60 cm	Normal	Beyond the water surface	There is
2	24 cm	Danger	In the water surface	60 cm	Normal	Beyond the water surface	There is
3	26 cm	Danger	In the water surface	50 cm	Normal	Beyond the water surface	There is
4	25 cm	Danger	In the water surface	50 cm	Normal	Beyond the water surface	There is
5	30 cm	Danger	In the water surface	60 cm	Normal	Beyond the water surface	There is
6	28 cm	Danger	In the water surface	60 cm	Normal	Beyond the water surface	There is
7	31 cm	Danger	In the water surface	60 cm	Normal	Beyond the water surface	There is
8	33 cm	Danger	In the water surface	60 cm	Normal	Beyond the water surface	There is
9	38 cm	Danger	In the water surface	50 cm	Normal	Beyond the water surface	There is
10	32 cm	Danger	In the water surface	50 cm	Normal	Beyond the water surface	There is

Real-world testing demonstrated that the system can operate reliably under dynamic natural conditions. The ultrasonic sensor consistently produced fairly accurate water level readings, and the system successfully sent notifications when water levels reached danger levels. Despite fluctuations in readings due to environmental factors, such as splashes and surface waves, the system performed well overall. These results confirm the system's suitability for direct water level monitoring.

4.6.3 Testing onplatformBlynk IoT

Figure 7. Sensor notification testing 1and 2

Testing on the Blynk IoT platform was conducted to evaluate the system's ability to transmit and display sensor data in real time and ensure notification functions are functioning appropriately for detected water level conditions. The Blynk IoT platform serves as a mobile-based monitoring interface that users can use to monitor water level status.

4.6.4 Testing on the ThinkSpeak platform

Figure 9. Testing on ThinkSpeak

The figure presents water level monitoring graphs from two sensors connected to the ThingSpeak platform. Field 1 displays data from Sensor 1, while Field 2 shows data from Sensor 2, with a total of 75 data entries recorded over six days.The results indicate that the system can periodically transmit data to ThingSpeak and display it visually, facilitating effective monitoring of water level changes over time

5. Conclusion

Based on the test results of the sensor components and monitoring system, both in simulated and real-world conditions, it can be concluded that the system performs effectively in accordance with its design objectives. The ultrasonic sensor is capable of detecting water levels with a good level of accuracy, and the system successfully classifies water conditions into Normal and Danger based on predetermined thresholds.

Furthermore, the system's integration with IoT platforms such as Blynk and ThingSpeak performed well. Sensor readings processed by the ESP32 microcontroller were displayed in real time, and warning notifications were successfully sent when a dangerous situation was detected. These results demonstrate the system's suitability for use as a real-time water level monitoring tool.

References

Admanugraha, A., & Damsi, F. (2025). Design of a Flood Detection System Using the Jsn-Sr04t Ultrasonic Sensor Based on the Internet of Things in Tanah Abang. 18(I), 31–38. https://doi.org/10.5281/zenodo.15052567

Alfan, AN, Ramadhan, V., Komputer, S., Informasi, FT, Raya, US, Pendahuluan, I., & Module, F. (2022). Prototype of Gas Detector and Temperature Monitoring. 9(2).

Azhar, MF, & Nurpulaela, L. (2024). 10201-Article Text-40042-1-10-20240630. 8(4), 7248–7253.

Fauzan, M., & Nurdin, N. (2024). Implementation of Thingspeak as a Database in a Flood Detection Tool Using Esp32. Content: Computer and Network Technology, 4(1), 66–70. https://doi.org/10.31294/conten.v4i1.3596

Fitria, RI, Tulodo, RP, Isralestina, F., & Firmansyah, H. (2023). Prototype of Monitoring System for the Effect of High and Low Water Levels on Shrimp Cultivation Using the Internet of Things (IoT). Engineering: Jurnal Bidang Teknik, 14(1), 47–56.

Furqan, F., & Maghfirah, FR (2022). The Impact of Freshwater Fish Cage Cultivation on the Welfare of the Toweren Village Community, Lut Tawar District, Central Aceh Regency. Jurnal Al-Ijtimaiyyah, 8(2), 305.https://doi.org/10.22373/alijtimaiyyah.v8i2.15545

Irhamna, A., Pratama, A., Fhonna, RP, & Ula, M. (2024). Water Monitoring System in Catfish Cultivation Ponds Based on Internet of Things. Sisfo: Scientific Journal of Information Systems, 8(1), 65. https://doi.org/10.29103/sisfo.v8i1.18089

Kamal, K., Tyas, UM, Buckhari, AA, & Pattasang, P. (2023). Implementation of Arduino IDE Application in Digital Systems Course. Journal of Education and Technology (TEKNOS), 1(1), 1–10.

Khair, U. (2020). Automatic Water Level and Faucet Detection Device Using Arduino Uno-Based Water Level Sensor. Wahana Inovasi: UISU Journal of Research and Community Service, 9(1), 9–15. https://jurnal.uisu.ac.id/index.php/wahana/article/view/2632

Lesmana, T., & Silalahi, M. (2020). Design Of Iot-Based Home Security System Ari. Comasie, 3(3), 21–30.

Mindasari, S., As'ad, M., & Meilantika, D. (2022). Security System For Alms Boxes At Sabilul Khasanah Mosque Based On Arduino Uno. Mahakarya Informatics Engineering Journal (Jtim), 5(2), 7–13.

Nasrullah, M., Ramadan, Dn, & Hartaman, A. (2021). Water Level And Ph Control In Koi Fish Cultivation. E-Proceedings Of Applied Science, 7(6), 3198.