Design and Prototyping of Temperature Monitoring System for Hydraulic Cylinder in Heavy Equipment using ESP32 with data logging and WiFi Connectivity

Jimmy Linggarjati
Computer Engineering Department, Faculty of Engineering, Bina Nusantara University, Jakarta, Indonesia 11480
jimmyl@binus.edu

Abstract. Measurement and data-logging of temperature are important in industrial application. This paper discusses the implementation of monitoring temperature of cylinder hydraulic of a heavy equipment such as excavator in the field. For monitoring the temperatures, DS18b20 sensors are used to sense the hydraulic arms in the excavator. Due to sensing moving parts such as hydraulic arms, a WiFi technology is used to transmit temperature’s data to a browser. The ESP-NOW technology from Espressif ESP32 is chosen to minimize the electrical energy drained from the Lipo’s battery. Data-logging is done in a browser environment, by using javascript, websockets, css, and html technologies. The result shows that the usage of ESP32 in monitoring the temperatures, can be used in the real-time mode for the monitoring of excavator operations. And the 20000mAh battery-based system can last for approximately a month by using ESP-NOW technology from the Espressif.

Keywords – ESP32, ESP-NOW, DS18B20, data-logging, battery-based, javascript, websockets, html, css

1. Introduction
As we know, hydraulic equipment has superior advantage of electrical one, in which the power produces by hydraulic system is huge, such that it can lift heavy objects such as dirts at a large quantity. The excavator is an important equipment in the mining industries. The downside of hydraulic system is that it is expensive, due to the high manufacturing’s cost of cylindrical arm used in the excavator. This expensive hydraulic system can be ruined if the operator of the excavator operates it in a careless way [1]. The high pressure from the hydraulic oil inside the cylinder, has some limits that needs to be kept within certain boundaries. If the limits are exceeded within a long period of time, the lifetime of the hydraulic arm system is shorten significantly. Therefore, a temperature monitoring system is proposed to detect a spike of temperatures changes during the excavator operations. These spikes of temperature are caused by higher pressure built up inside the hydraulic’s arms.
The idea of using temperature sensor to detect earlier defect on the hydraulic ram of the excavator, had been explored previously in [2-3]. This paper will focus on using ESP32 IoT device to capture and send the temperature data in real-time during the excavator’s operation. Based on some earlier literatures [4-9]

Because of moving parts, a light and simple temperature sensor with a battery-power, is considered in the design of this system. The connectivity technology is also important to be able to send the temperature’s data in real-time, to be further analysis in a browser. Browser technology is chosen, due to its abundance in existence, such as in our daily usage of mobile phone. And due to an energy intensive usage of WiFi, this system is switched to using a point to point connectivity with a lower energy consumption, by using the ESP-NOW proprietary technology by Espressif [10-12]. By using ESP32 SoC, low energy, low cost, and WiFi connectivity can be achieved efficiently, compared to the first implementation of this temperature’s monitoring, which used cable for power and SD-card for storing the data. Fig. 1 shows design components of this proposed temperature monitoring system. The system characteristics are low-cost (due to cheap components) and low-energy consumption, promoting an electronic’s sustainability. In the following section, we will discuss the implementation details of this temperature monitoring system.

![Fig. 1. Design Components of Temperature Monitoring System](image1)

2. Methodology

The overall proposed system is illustrated in a block diagram, as shown in Fig. 2. There are two ESP32’s devices that are used, one is attached to the excavator’s arm, and the other one acts as an access point and as a station, in order to receive temperature’s data and at the same time connects to the client via a web-browser.

![Fig. 2. Proposed System using ESP-NOW](image2)
2.1 The Excavator
The excavator has three major parts of hydraulic arms, i.e. the boom, arm, and bucket actuators, as shown in Fig. 3. Therefore in total, three temperature’s sensors are needed for each hydraulic actuators. In a newer excavator machine, a hydraulic’s sensor is used to monitor the preassure inside the hydraulic’s cylinder. But, it is expensive, and by using temperature’s sensor, one can achieve the same importance’s information, whether the arm hydraulic is in a good state or not.

![Fig. 3. Parts of an Excavator](image)

2.2 ESP32
The ESP32 microcontroller is produced by Espressif. It is a dual core processor with WiFi connectivity in a single module, as shown in Fig. 4.

![Fig. 4. The ESP32](image)

The Arduino IDE will be used to program the ESP32. And in order to successfully used ESP32, in the Arduino IDE, a package of ESP32 board’s library is installed.

2.3 DS18B20
DS18B20 is one wire temperature’s sensor from Maxim/Dallas. The one-wire’s technology eases the connections for multiple sensors, by simply create parallel connection of this temperature sensor. The DS18B20 has three pins, i.e. Vcc, Gnd, and Data pins.

![Fig. 5. Sample of DS18B20 Temperature Sensor](image)
3 Design of Software Algorithm

There are two esp32 being used for temperature data transmission, as shown in Fig. 2. They communicate using ESP-NOW. The battery-power ESP32 is used to read the sensor’s value. While the other ESP32 acts as an Access Point (AP) and a Station Mode (STA) at the same time. Therefore, the second ESP32 hosts the web server’s program, so that clients can connect and see the temperature’s data in a browser (either from smart phone or from a desktop computer).

In the browser site, a Javascript programming language, along with html, css, and websockets technologies, are all hosted in ESP32. The websockets’ technology, in particular, enables the server and the client to communicate each other. This is very useful, in a situation where the client would like to change one or more parameters in the ESP32.

4 Implementation

With the methodologies set, the hardware’s decision is kept in low price, under 30 dollar, by using ESP32 SoC module. Another 30 dollar is invested in battery LiPo and its charging module. While the software has no additional cost, other than the time to spent on its coding in ESP32 system.

4.1. Hardware Parts

Fig. 6. shows the hardware parts in block diagram. It shows that a single LiPo battery 18650 is used in the ESP32, to power up the ESP32 which is responsible for reading the temperature’s data. While the other ESP32 that host the web-server-client code, is assumed to be powered by utility grid via a DC power-supply equipment.

As the hardware parts have been design as depicted in Fig. 6, Fig. 7. shows the implementation of hardware parts, using four modular parts, i.e. two ESP32 modules, one 18650 battery shield and the DS18b20 sensor’s module.
4.2. Software Parts

The overall architecture of software, utilises ESP-NOW, webSocket, and deep-sleep in the Arduino IDE (Integrated Development Environment). For the sender (ESP32 with temperature’s sensor), Fig. 8. depicts the sender flowchart diagram. While Fig. 8. depicts the receiver flowchart diagram.

In Fig. 8., a deep-sleep function is called and woke-up by timer peripheral within the ESP32. And also notice, a send’s callback is initialized within the initialization of ESP-NOW. When ESP-NOW sends data, a send’s callback will be called to report its status of delivery. And inside the send’s callback, a deep-sleep function is called, to further decrease current consumption from the battery.

In Fig. 9., the ESP32 is set as receiver and also as a server. Therefore, this ESP32 is set as Access-Point and also as a Station. The websockets technology is used to retrieve information from the client. While the ESP-NOW technology is used to receive temperature’s sensor from the hydraulic parts of the excavator.

---

Fig. 7. Real Hardware Implementation

---

Fig. 8. Sender Flowchart Diagram
5 Results
The system monitoring can display a real-time temperatures taken from the DS18B20 temperature’s sensors. Table 1. shows the temperature data-logging, that is seen from the web-browser.

Table 1. Temperature Data-logging

| ReadingId | Temperature1 (°C) | Temperature2 (°C) |
|-----------|-------------------|-------------------|
| 402       | 31.4375           | 31.8125           |
| 401       | 31.375            | 31.8125           |
| 400       | 31.4375           | 31.8125           |
| 399       | 31.375            | 31.75             |
| 398       | 31.4375           | 31.8125           |
| 397       | 31.4375           | 31.875            |
| 396       | 31.4375           | 31.75             |
| 395       | 31.4375           | 31.75             |
| 394       | 31.5              | 31.8125           |
| 393       | 31.4375           | 31.8125           |
| 392       | 31.4375           | 31.8125           |
| 391       | 31.4375           | 31.8125           |

The data are also displayed as a graphical line view, by using a ChartJS’s javascript. Fig. 10. shows the chart displaying the temperature’s value over time.
Current measurements are taken, to measure how many currents are needed by the ESP32-sender, resulting in how many mAh needed for a certain period of time.

Based on the current measurement, at deep-sleep, a current of less than 10 mA is taken. And at time of data temperature sending, a maximum of 100 mA is recorded. By using a stop-watch, an average of 10 data-logging (with 5 seconds interval) can be recorded for every 1 mAh of consumed’s energy. Therefore, given a power-bank with a capacity of 20000mAh, the system can last for 20000mAh * 50s/mAh = 1000000 [s]. If assumed that the system is used 8 hours/day, then this battery-based system can last for approximately 30 days or 1 month without recharging.

6 Conclusions
The temperature monitoring system is able to operate under battery-power system for a period of one month, using a 20000mAh battery bank. By utilising a deep-sleep feature in between transmission of data, and by using a more efficient communication protocol i.e. ESP32-NOW, the power consumption of this ESP32 system is improved.

An automated web-server, using websockets and javascript technologies, enables the collection of temperatures data automatically within a certain predetermined time-interval, in which the minimum interval is 5 seconds, or in another word, each sample data is taken with a frequency of 0.2Hz.

References
[1] Amrulloh A and Harjono 2015 Hydraulic oil overheating analysis excavator EX 2500-6. Gadjah Mada University.
[2] Ni L, Chen X S, Wang A L and Cao S K 2014 An intelligent fault monitoring system for hydraulic excavator. Advanced Materials Research, 989-94, 3114–6
[3] Li Y and Ru Z H 2012 Research to the infrared monitoring system of construction machinery hydraulic oil temperature. Applied Mechanics and Materials, 271-2, 621–4
[4] Jun Z, Sheng-jie J, Gui-mao S, Peng-long Y, Xiao-ming L, Jian-bo W and Wen-jin J 2010 Design of electronic control system of hydraulic excavator with CAN Bus and PID Method 2010 International Conference on Intelligent System Design and Engineering Application
[5] Budijono S 2021 Smart temperature monitoring system using ESP32 and DS18B20 In IOP Conference Series: Earth and Environmental Science (Vol. 794, No. 1, p. 012125). IOP Publishing
[6] Allafi I and Iqbal T 2017 Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring IEEE Electrical Power and Energy Conference
(EPEC)

[7] Al Abbas I 2019 ESP32 Based Data Logger International J. of Computer Science and Mobile Computing vol 8: 259-67.

[8] Yusri I K 2020 Solar panel remote monitoring and control system on miniature weather stations based on web server and ESP32 International J. of Recent Technology and Applied Science, 2(1), 1-24.

[9] Pasic R, Kuzmanov I and Atanasovski K 2020 Espressif ESP32 development board in wifi station communication mode TEMEL-ij, 4(1), 1-6.

[10] Pasic R, Kuzmanov I and Atanasovski K 2021 ESP-NOW communication protocol with ESP32 J. of Universal Excellence, 6(1), 53-60.

[11] Yukhimets D, Sych A and Sakhnenko A 2020 Designing a method for constructing distributed open ACS based on the ESP-NOW wireless protocol International Russian Automation Conference (RusAutoCon) (pp. 642-647). IEEE.

[12] Hoang T N, Van S T and Nguyen B D 2019 ESP-NOW based decentralized low cost voice communication systems for buildings International Symposium on Electrical and Electronics Engineering (ISEE) (pp. 108-112). IEEE.