# Easy to Use IoT based Temperature and Humidity Data Logger

Stephanus Surijadarma Tandjung[1*], Eko Sutjipto Sujanto[2]

[1*] Teknik Informatika, Universitas Katolik Darma Cendika, Jl. Dr. Ir. H. Soekarno No.201, Klampis Ngasem, Kec. Sukolilo, Surabaya, Jawa Timur 60117

[2] PT Metro Instrument Abadi, Kompleks Pertokoan RMI Blok L-8, Jl. Ngagel Jaya Selatan, Surabaya, Jawa Timur.

stephanusst@ukdc.ac.id

**Abstract.** The instantaneous temperature and humidity reading cannot tell the overall condition of the environment in concern, but a one day data logger may give us a slight glimpse of an overall understanding of an environment. While the current high quality and lowest cost industry product may give us a good resolution, it still has various disadvantages such as the size of the goods and also complexity to program, the extra cost to purchase modules and to buy the customized application program, as well as the various IOT services. In this case, a temperature and humidity data logger is being developed with a complete set of IOT support and also with an android app to make the user feel easy to use. The comparison with industrial solutions is also shown in this article together with a syllabus of a specialized platform development course is also being developed to empower communities to build similar solutions for both industrial or commercial solutions, especially for toilet usages.

**Keywords:** Data Logger, ESP8266, MySQL, Google Charts, LOGO! Siemens, Node-red.

## 1 Introduction

### 1.1 Data Logger with IOT capabilities

In a common public toilet, sometimes, it may have a foul odor. Cleaner team may have been assigned with a task to clean it regularly or fully in charge of one or more toilets, however such work should be in a companion with a paper work pasted in the back of the door. To turn such paper into a digital work, an old well known in industry such as a data logger are accommodated here. Moreover, a temperature and an humidity sensor are included.

Recent years have been filled with a variety of Internet of Things (IOT) devices. As a result, an old version of classic data logger, now, should have IOT capabilities. Here, this paper discusses various hardware and software to build those devices are shown here. The device's industrial version may have a different specification compared to the commercial one. In this article, such choices are elaborated including the various products that have been developed.

Firstly, we have developed a data logger using LOGO! Siemens to record temperature and humidity. This Block Module required Analog Module to receive signal from the sensor devices. Lately, we are developing a logger using ESP32. The temperature and humidity sensor are selected to be logged. These devices are industrial versus commercial modules.

While the hardware has been installed, additional software and hardware should be added to further utilize the data being logged. The software includes custom C++ programming, awk, gnu-plot, Node-red, MQTT, web server, or Google Chart.

## 1.2 Industrial Development Platform

Siemens LOGO! Block Modules is a logic device if the system requires an RTD sensor, a LOGO! Expansion Modules should be added. The development platform for this device is LOGO! Soft Comfort LOGO! and optionally, the LOGO! Web Editor (LWE) to create a custom web server. Moreover, node-red on top of Raspberry-Pi industrial version can be added to display graphics.

The data in text mode saved in the SD Card are downloaded using LOGO! Soft Comfort and cut based on date. The software to cut is shell type that utilizes awk, and an c++ programming to analyze the data. GNUPLOT is also utilized to display the chart.

Siemens LOGO! network utilizes TCP to communicate between modules and PC as shown in Fig. 1. The orange display has a display while the gray display has no display. S7 Communication between LOGO! devices can be displayed as gray lines, while the unconnected lines are shown if the IP address is connected to an undrawn device. Moreover, S7 communication is a Transportation Service Access Point (TSAP) , while the yellow line is a modbus connection.

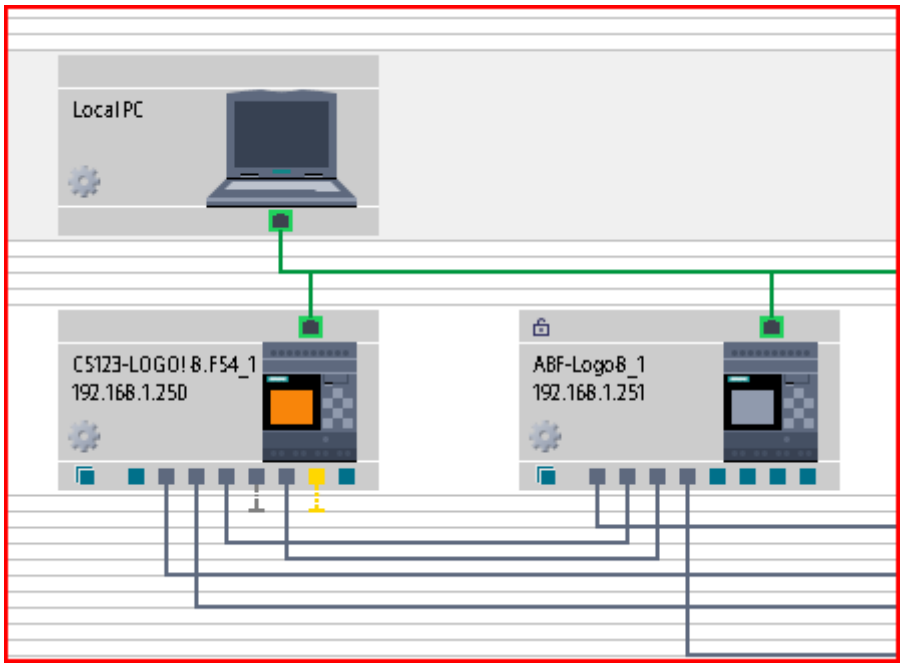


**Fig. 1.** LOGO! Block Modules Network consist of ethernet and S7/Modbus communication.

Fig. 2 is an expansion module to read temperature using PT100 temperature sensor. Fig. 3 shows TSAP connection between two LOGO! devices.The Analog Input and Output of the client with IP 192.168.1.250 update the Variable Memory of the server IP 192.168.1.252.


**Fig. 2.** Expansion Module AM2RTD to read PT100 temperature sensor

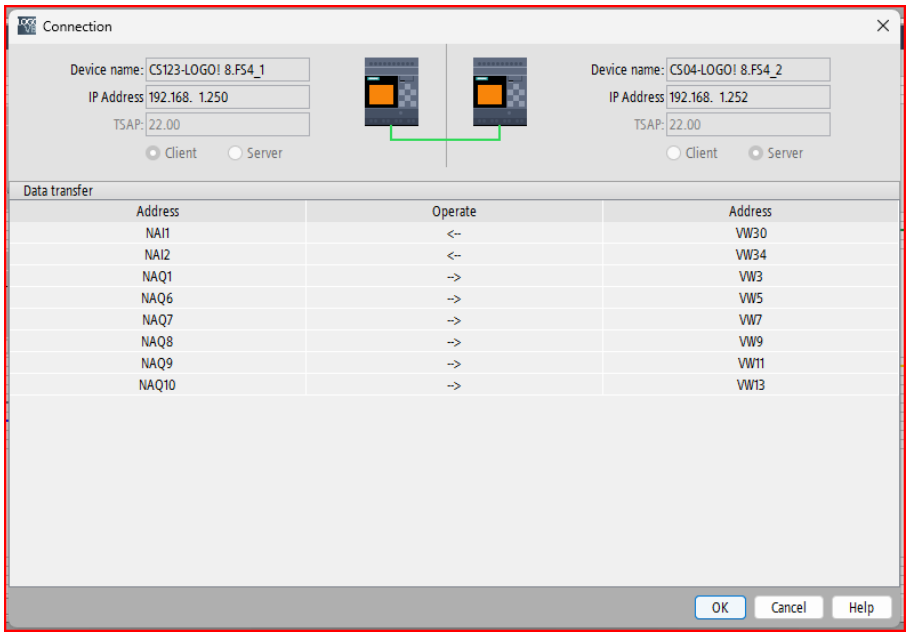


**Fig. 3.** TSAP connection between client and server. NAI means Network Analog Input while NAQ means Network Analog Output. VW is Variable Memory

In **Fig. 4**. Modbus Connection between Moxa TCP/RS485 Modbus converter and LOGO! device. In this figure, Holding Register updating Variable Memory for every 1 minute 100 milliseconds.

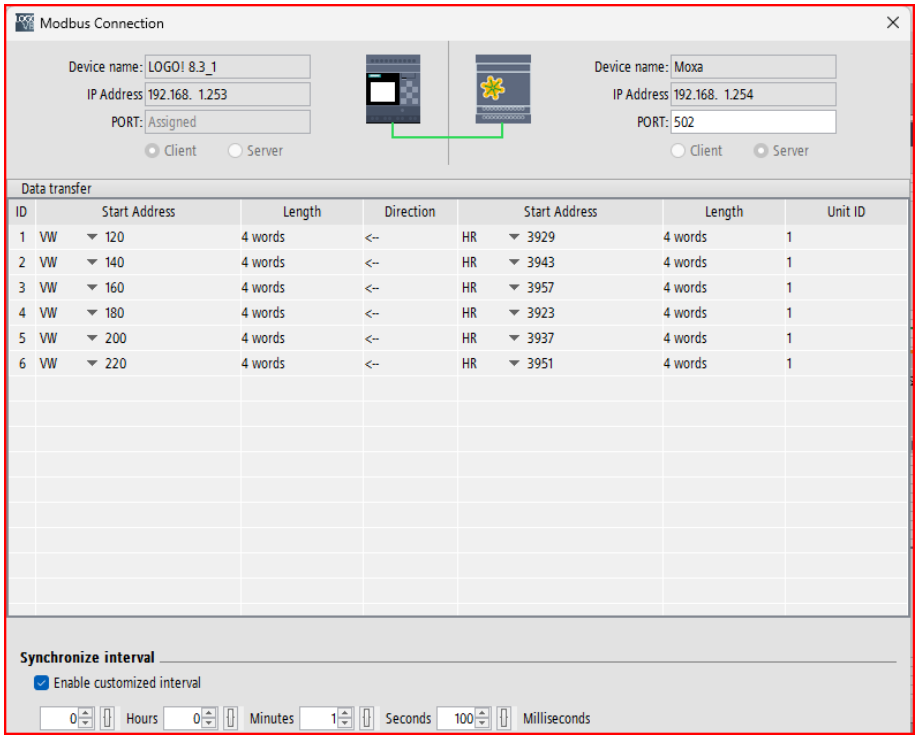


**Fig. 4.** Modbus Connection between LOGO! device and Moxa RS485 Modbus TCP Converter

Fig. 5. shows the mapped modbus address and address type of LOGO! devices. The block module has Digital Input (I), Digital Output (Q), Digital Memory Flag (M) and Variable bit memory (V). It also has Analog Input (AI), Variable Word Memory (VM), Analog Output (AQ) and Analog M Flag (AM). Those address types are mapped into modbus addresses.

**Modbus address space**

| Address Type | Range | Mapped Modbus Address | Direction | Unit |
|---|---|---|---|---|
| I | 1 - 24 | Discrete Input (DI) 1 - 24 | R | bit |
| Q | 1 - 20 | Coil 8193 - 8212 | R/W | bit |
| M | 1 - 64 | Coil 8257 - 8320 | R/W | bit |
| V | 0.0 - 850.7 | Coil 1 - 6808 | R/W | bit |
| AI | 1 - 8 | Input Register (IR) 1 - 8 | R | word |
| VW | 0 - 850 | Holding Register (HR) 1 - 425 | R/W | word |
| AQ | 1 - 8 | Holding Register (HR) 513 - 520 | R/W | word |
| AM | 1 - 64 | Holding Register (HR) 529 - 592 | R/W | word |

**Fig. 5.** Modbus address space

AWK is a scripting language used for cutting data based on date, while GNUPLOT is a graphic utility.
The system may require a converter if the device uses RS485. The converter mixed Modbus TCP and RTU/ASCII networks.

## 1.3 Commercial Development Platform

Arduino IDE is a platform development that simplifies hardware programming. This development system utilize C++ programming language.

A suitable board that has a WiFi, dual core and fast speed is ESP32. This is a powerful embedded system. To read temperature and humidity, the DHT 11 sensor terminal is connected to ESP32 as shown in Fig. 6.

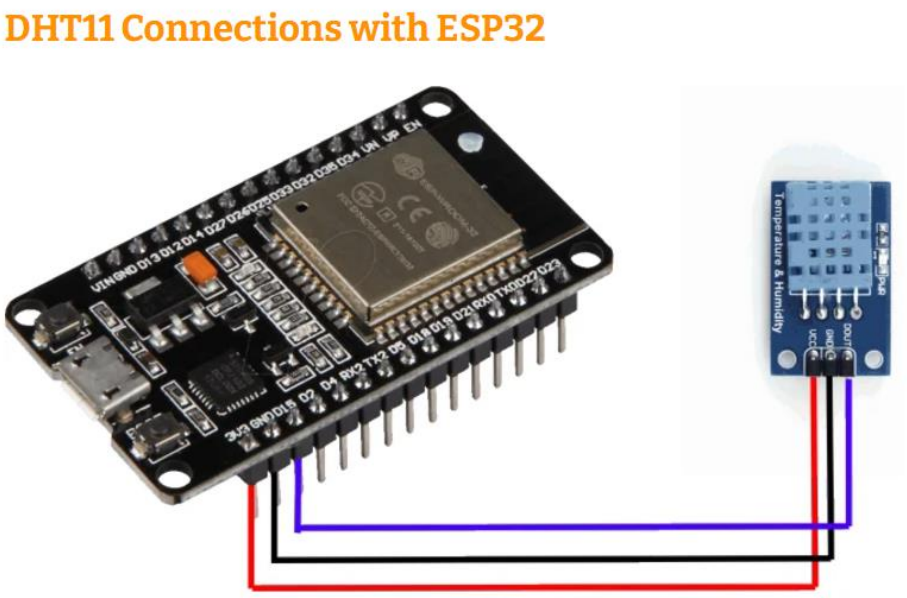


**Fig 6.** Embedded System ESP32 connected to temperature and humidity sensor, DHT11.

The values are uploaded into MySQL database server, and finally the chart is shown using Google Chart.

# 2 Methodology

## 2.1 An Industrial Solution

SIEMENS LOGO! Block Modules are promoted as an intelligent logic module.This module has 8 inputs and 4 outputs. The input can have digital inputs and analog inputs according to the specifications.The output can be a relay or PNP or NPN output.

As a data logger, we require analog inputs to read voltage input or expansion modules to read RTD temperature sensors.We may require digital inputs, so that if any push buttons or digital input devices are added, the block module can read its states.

The block module has an SD Card to keep the data, it also has an ethernet port. Each block module can also be programmed to send or receive data by using Modbus protocol.

The data logger reads the sensor every two minutes, and writes the I/O state on the SD Card. The block module has its web server to display the data, but it can be set as a modbus slave or master, while a raspberry Pi with its node-red can read and display the data for 24 hours.

Node-red modbus communication can be set to receive data when the data changed. This condition needs to be set to reduce the work of the node-red dashboard.

Fig. 7. shows the display of the Block Module, Fig. 8 shows the custom display created by LWE. To display graphics we may use Node-Red as shown in Fig. 9.

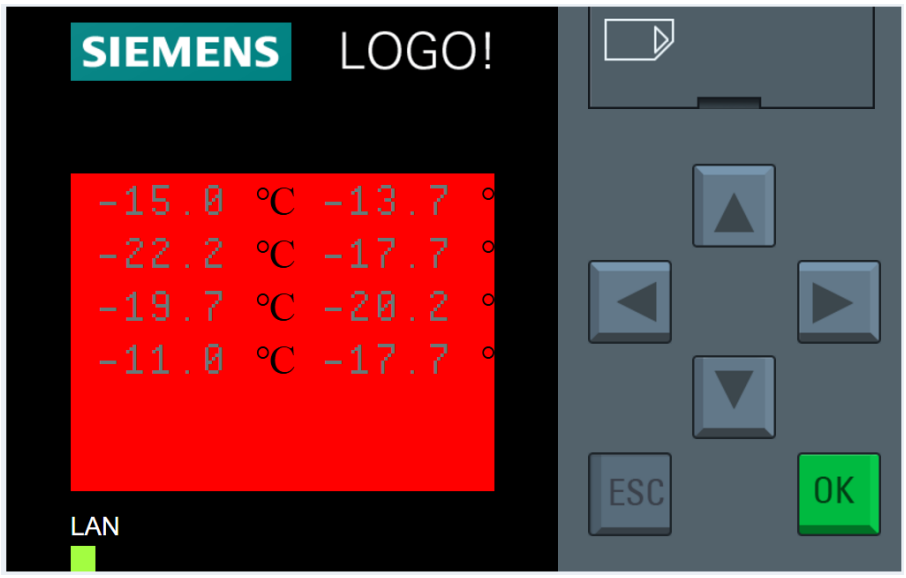


**Fig. 7.** LOGO! Display showing temperature of rooms.

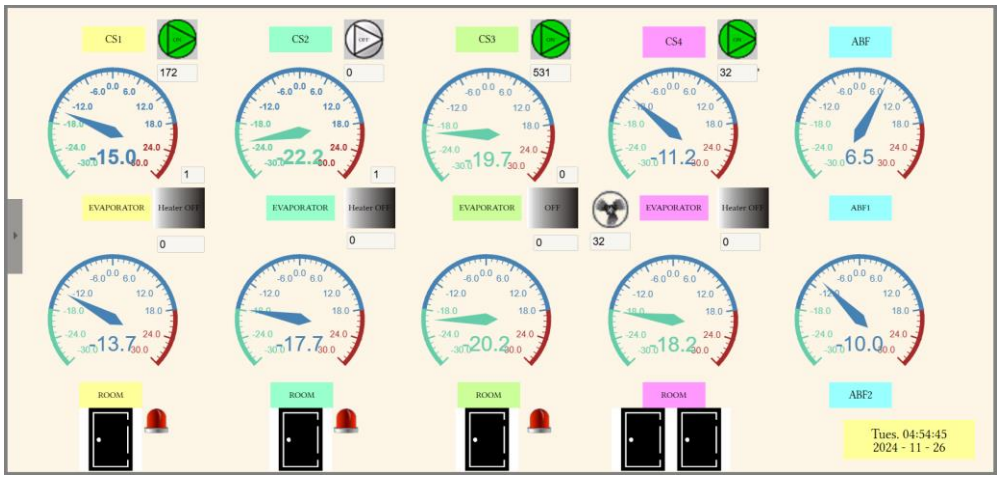


Fig. 8. The Custom Display.


Fig. 9. Node-Red Dashboard.

Awk is a scrip language, for example:

```
datestr=$(date +"%Y-%m-%d")
str=$(echo "$1" | awk '{print substr ($0,1,length($0)-4)}')
awk -v tanggal="$datestr" -f filtering-cs123-500-tanggal.awk "$1" > "$str-filtered-$datestr.csv";
```

to cut the text file based on one day.

Fig. 10. shows the temperature chart for a certain day and Fig. 11 shows the electrical consumption.

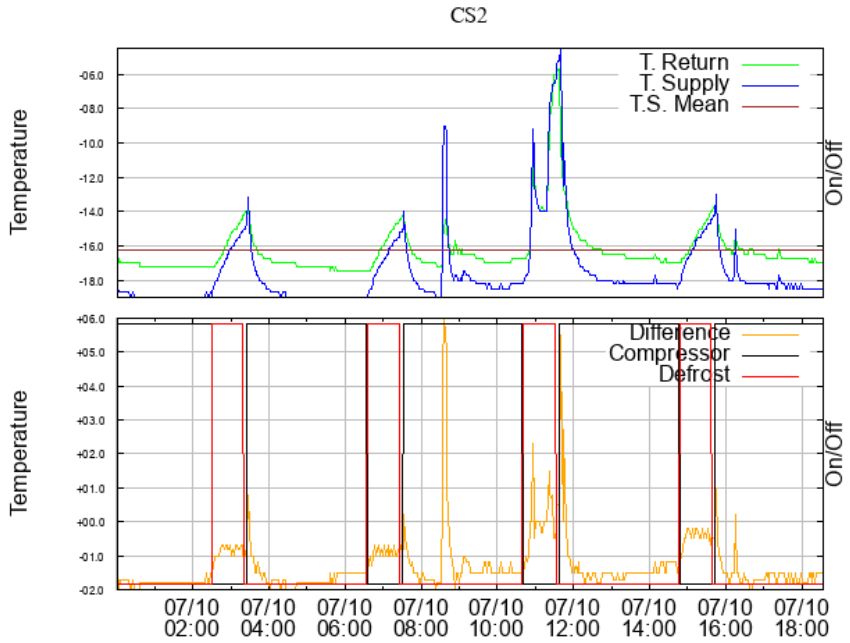


Fig. 10. Chart created by GNUPlot.

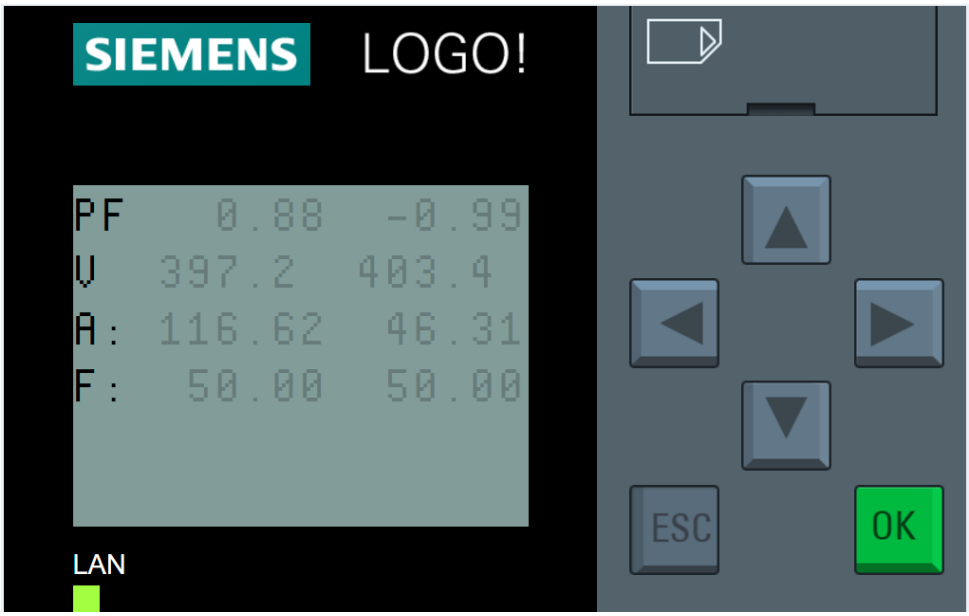


Fig 11. Schneider Multimeter Data shown in Block Module's Display.

## 2.2 A Commercial Solution

ESP32 is a series of low-cost, low-power system-on-chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. It can be connected to humidity and temperature chips. The value is sent to the database server, at the same time the data is displayed as a graph using Google Chart.

Fig. 12. shows web server display of temperature and humidity data, and the picture of serial communication of schneider power meter data. The temperature and humidity has the option to display every two days or every one day while the power meter report is under development.

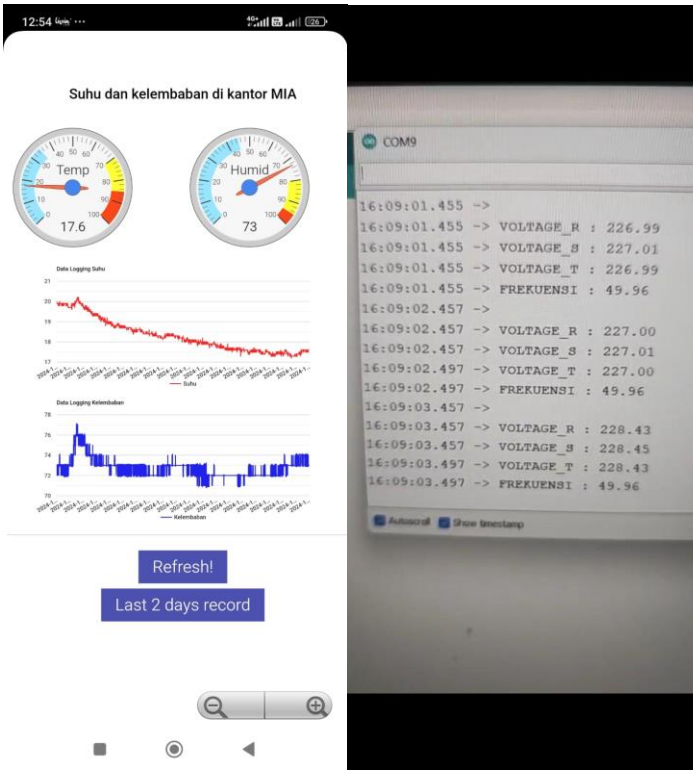


Fig 12. Web Server Display and PC based serial communication display.

## 3 Results and Discussion

The development of industrial solutions takes years, the aim is to show the performance of a machine to control the room temperature. The industrial solution has text data to be downloaded every day. The size of an SD Card can keep months of data. Having a Raspberry Pi and Node-Red installed, this data logger is able to display charts.

The development of a commercial solution takes a few months, it also involves a mysql database server. It also has several limitations if it uses ESP8266 such as serial communication, in this case ESP32 is the solution that gives two serial communication. The serial communication is to connect the sensor and the microcontroller.

The industrial solution has been used also in a part of Embedded System Course, and also a part of a commercial solution. If we consider the price, the commercial solution is a fraction of the industrial solution, but if the display, the SDCard is added, the price of the commercial solution may be a little bit lower than the industrial solution.

Both development platforms require the same knowledge to develop, both have certain difficulties. The most concerning matter for industrial solution systems is to display graphic data in realtime using LOGO custom web server, in the above solution, a raspberry-pi with node-red is added into the system. While the commercial solution system uses a paid server outside the premise and a certain knowledge to build charts using Google Chart.

## 4 Conclusion

In line with the topic of the seminar this article emphasises the importance of historical data of the condition under consideration. The appearance of the report consists of an instance temperature, and data consists of months data log.

Shown in this article, the development platform of both industrial and commercial versions. The Siemens LOGO! Device requires the logic developed in block, while the ESP32 Arduino requires logic written in C++

programming language, moreover extended knowledge of drag and drop programming such as in Node-red under Java programming is now required.  Furthermore, a classic mysql database is also employed in the commercial version.

## 5 Reference


Stephanus S. Tandjung, Yosefina F. Riti, Lasman P. Purba, *Rancangan Data Logger dan Analisis untuk Ruang Dingin*, Prosiding Seminar Nasional Riset dan Teknologi Terapan (Ritektra) X, 2021. Available at: https://journal.unpar.ac.id/index.php/ritektra/article/view/4998

Ibrahim A. Abbas, *ESP32 Based Data Logger*, International Journal of Computer Science and Mobile Computing, , Vol.8 Issue.5, May- 2019, pg. 259-267. Available at https://ijcsmc.com/docs/papers/May2019/V8I5201943.pdf

*LOGO! – the compact controller with a cloud interface.* Available at: https://www.siemens.com/global/en/products/automation/systems/industrial/plc/logo.html.

Download and installation instructions for *LOGO! Soft Comfort Upgrade V8.4* https://support.industry.siemens.com/cs/document/109826553/download-and-installation-instructions-for-logo!-soft-comfort-upgrade-v8-4?dti=0&lc=en-US

Effective AWK Programming: A User's Guide for GNU Awk. Available at https://www.gnu.org/software/gawk/manual/gawk.html

*Arduino*, Available at https://www.arduino.cc/

*Espressif ESP32*. Available at https://www.espressif.com/en/products/socs/esp32