Software Industrial Internship Report

B.Eng. in Software Engineering, Academic Year 2017/2018

Internship period : 2018 June 04~2018 August 02 (*33 days/261 hours 42 minutes)*

58090002 Damian Satya Wibowo

International College

King Mongkut's Institute of Technology Ladkrabang

Table of Contents

1. Company outline

1.1 Company name and address

Company Name: PT Tigaresi Bangun Nusaperdana (TBN)

Office Address: (Indonesian) Jalan Sidomukti No. 56, Kecamatan Sukaluyu, Kota Bandung, Jawa Barat, Indonesia 40123

(English) 56, Sidomukti Rd., Sukaluyu Subdistrict, Bandung City, West Java Province, Indonesia 40123

Website: www.tigaresi.com, tigaresi.wordpress.com

The office is a house functioning as the headquarter, as a regular office and manufacturing site.

1.2 Main activities

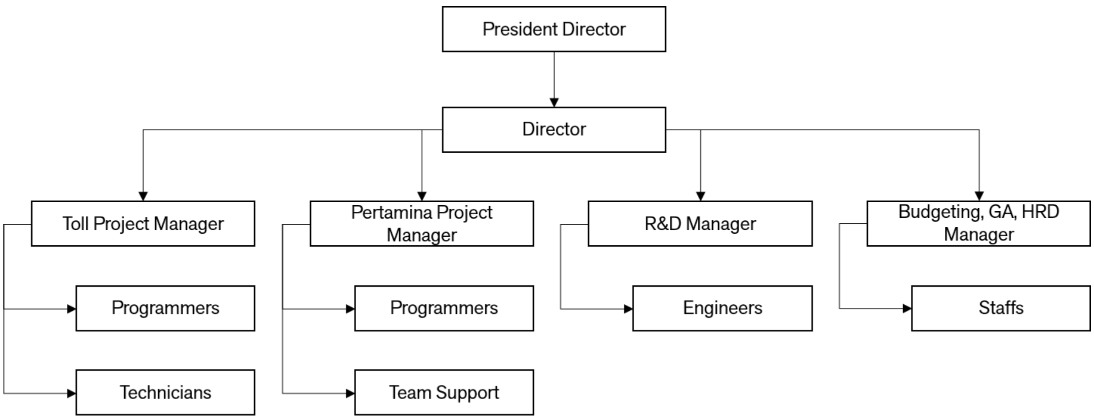
The company specializes in producing electronics, real-time systems, such as machineries, hardware and software, mainly in embedded system which involves the usage of microcontrollers and microprocessors.

TBN is currently handling five main projects:

1. Internet of Things (IoT), currently as a prototype to control air conditioners and view its status remotely;
2. Tollgate e-money payment and database system;
3. Tollgate vehicle class classification system, an embedded system which determines the vehicle type which passes the tollgate, to be passed to an automatic payment gate;
4. Gas pipe cathodic protection serving Perusahaan Gas Negara;
5. Gas/petrol depot automation, as a partner for Pertamina, Indonesia's biggest oil and gas company. Includes the system for automated driver fit-to-work check system, automatic gas/petrol distribution routing, et cetera. TBN also won the 2nd place in petrol/gas depot efficiency, worldwide.

PT Tigaresi Bangun Nusaperdana focuses on their motto: *"Teknologi Merah Putih" (Technology of the Red and White [Indonesian national symbol]),* and works on a full effort to provide appropriate and accurate solutions which are economical, with sustainable support. [1]

1.3 Organization structure



The company has around 40 employees.

1.4 Department and supervision

I worked in none of the mentioned departments, but contributed in the Internet of Things project, one of PT Tigaresi Bangun Nusaperdana's project.

Supervisor: Mr. Victor P. Limbong +6281321944400

2. Internship

2.1 Original description and objectives

Participate in the development team to implement or improve automation of Oil Truck Driver Fit To Work Check system. This activities may cover area such as medical instrument data acquisition, truck key management, and check in terminal panel.

2.2 Revised description and objectives

Phase 1 – 2018/06/22 to 2018/07/22 (22 work days)

The project assigned to me was changed on the second day of the internship by the supervisor. The new job description was: implementing IoT in an air conditioner to control the air conditioner remotely, publishing the air conditioner's status and surrounding environment data (temperature, current) online, also logging the sensor data inside a server. Below is the detailed list of my responsibilities:

- Planning and design

1. Configuring Raspberry Pi to cater all needs for the project;
2. Deciding which components to be used, hardware connection (schema) and embedded systems;
3. Designing hardware, software and server data flow;

- Hardware and sensor side

4. Connecting Infrared receiver and transmitter, temperature sensor and Analog-to-digital converter to Raspberry Pi development board, connecting a current sensor to Analog-to-digital converter;
5. Cloning an air conditioner remote using LIRC library and IR receiver, to map keys to be a virtual remote and testing each key clone's output to the infrared blaster;
6. Establishing communication between temperature sensor and Raspberry Pi, callibrating the binary temperature reading to Celcius, with some help from the web and outputting Celcius reading;
7. Establishing communication between Analog-to-digital converter and Raspberry Pi, callibrating the binary digital value to voltage, finding out the scaling between current level and voltage, and converting outputted voltage to current level then outputting it;

- Software, communication and server side

8. Building a controller~embedded system communication system using MQTT (including finding broker, initializing topic, writing the subscriber/publisher code, setting message events)
9. Setting up a web server (domain + hosting) which includes PHP and MySQL for passing and containing routine log data (deprecated in the second phase)
10. Building a HTML/JS status and control page which controls the embedded system via MQTT

Phase 2 – 2018/07/23 to 2018/08/02 (12 work days)

In the last two weeks, the project mission was upgraded to an extent that it supports more modularity and efficiency. The old code was migrated to a new framework, namely QtWebApp by Stefan Frings and with a standalone MQTT library. Some parts of this phase was supported directly by the president of the company.

3. Software Development

3.1 Background

The project's main topic is Internet of Things (IoT). IoT itself is defined as *"a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction." [2]*

Since IoT becomes more and more widespread, it would soon turn into a technological standard that humanity needs. So, the company aims to make their own first IoT system. The first, easiest prototype which TBN plans to make is an IoT system to control air conditioners in a building. Their first target customers (for prototyping/testing) are LG brand AC users, though the scope will be expanded in the near future.

3.2 Software requirement specification

1. **User Requirements**
   a. Functional
      i. Users can toggle power/set temperature of a specific brand of air conditioner via a webpage remotely.
      ii. Users are able to view the air conditioner's status and its updates in a webpage.
      iii. There is a button to re-read sensor data and simultaneously update the status. (Phase 1 only)
      iv. Users are notified whether the system is connected or not.
      v. Users can track the temperature and current log via a webpage.
   b. Non-functional
      i. The AC should respond to the command promptly.
      ii. Web control page contains only basic controls such as power and temperature setting.
      iii. The temperature/current history is updated every 1 minute.
      iv. The control, status and log interface will be presented in English.
      v. The application currently works with LG-brand air conditioners only.
2. **System Requirements**
   a. Functional
      i. Raspberry Pi can send commands to Si7021 temperature sensor and ADS1115 current sensor and get data from their result registers.
      ii. The control in HTML/JS page sends and receives MQTT messages to communicate with the main system in the Raspberry Pi. (Phase 1 only)
      iii. The control in HTML/JS page sends and receives sensor data via PHP to get data from the log inside MySQL. (Phase 2 only)
      iv. Main system receives a specific formatted MQTT messages and parses them.
      v. The received message in the main system is then converted them into LIRC commands, sensor reading commands and ultimately sending back an acknowledgement response.

vi. Raspberry Pi can send sensor reading through a C-Qt code to the online MySQL database, using PHP GET mechanism. (Phase 1 only)

b. Non-functional

i. Mosquitto MQTT is used as the protocol for Server-Raspberry Pi communication.

ii. Air conditioner status is shown and controlled via a HTML/JS webpage.

iii. MQTT and HTML are bridged with Websocket. (Phase 1 only)

iv. The controller programs inside Raspberry Pi is written in C and C++ language, Qt library and embedded Linux commands.

v. The server is written in C, Qt Library and MQTT standalone library.

vi. The web application is written in HTML (Phase 1~2), JS and PHP (Phase 1).

vii. Sensor data is kept inside the MySQL database of the website. (Phase 1 only)

viii. The website is temporarily hosted at 000webhost.com. (Phase 1 only)

ix. The website is temporarily hosted at localhost port 8080 using Qt WebApp. (Phase 2 only)

x. The system is considered disconnected if the system is unresponsive for 5 seconds. Phase 1 only)

xi. Sensor readings are sent every 1 minute inside the Qt timer event function.

3.3 Development team and job allocation

I developed the planning, most of software parts, hardware-software communication and some hardware systems, e.g. connecting modules to Pi. Other hardware works (such as circuit assembly, PCB/board design and manufacture) were provided by the staffs of the company. Some guidances, some software codes and some modules for the second phase of internship were provided by the president of the company.

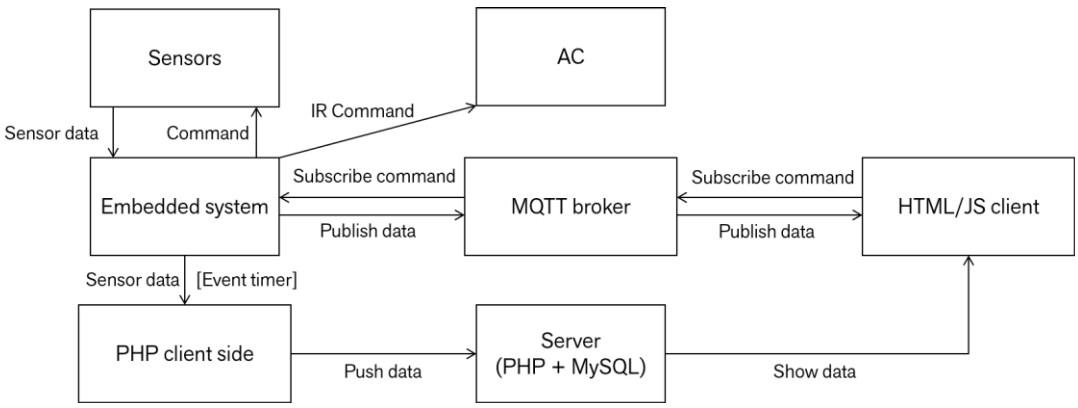
3.4 Project plan

The project plan was quite flexible, I was given a month (June 22 to July 22) to finish the first prototyping, and proceed to the second phase of prototyping afterwards. The deadline was met just exactly. I planned to finish the internship on August 9, but stuffing work days to Saturdays and Sundays, I could finish it on August 2.

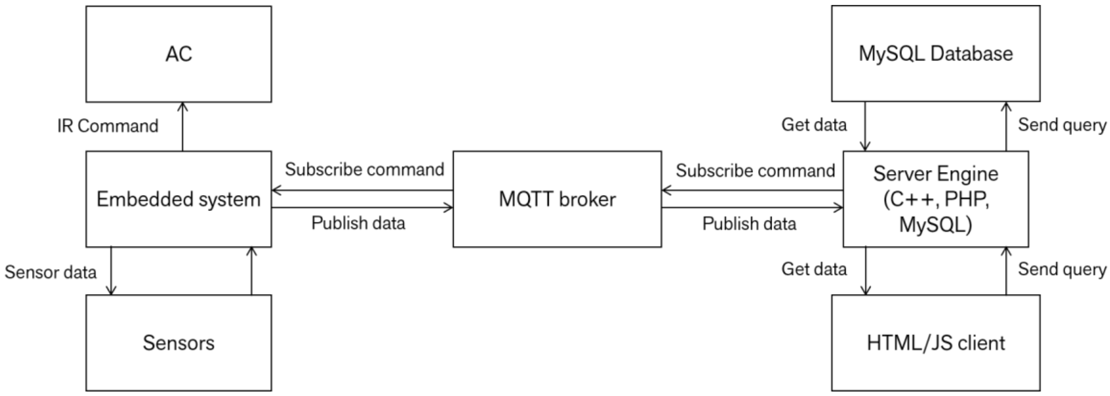
3.5 Software design

3.5.1 Data flow

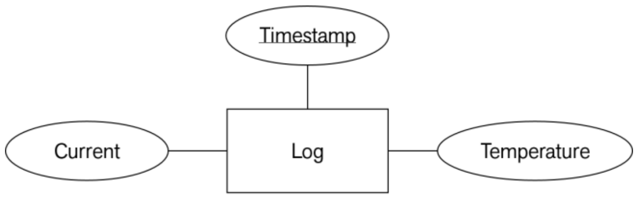
3.5.2.1 Original data flow:



3.5.2.2 Revised data flow:



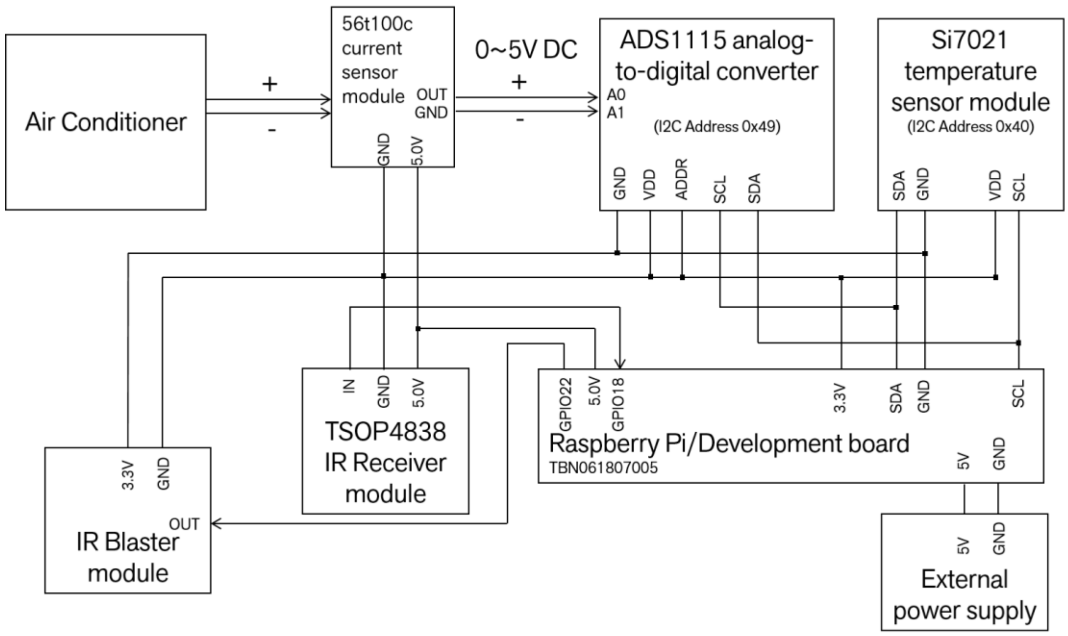
3.5.2 Database design (Phase 1 only)



Log

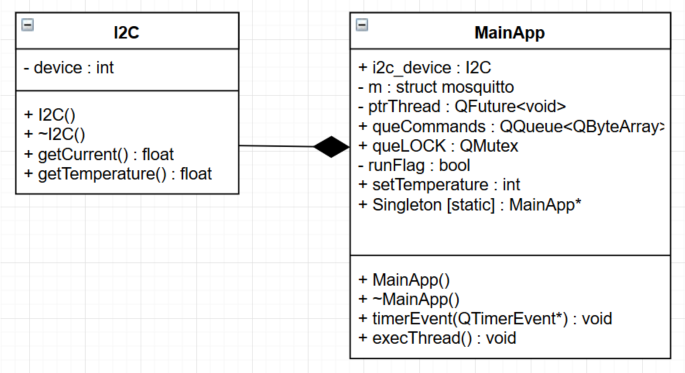
Timestamp int	Current float	Temperature float

3.5.3 Hardware design

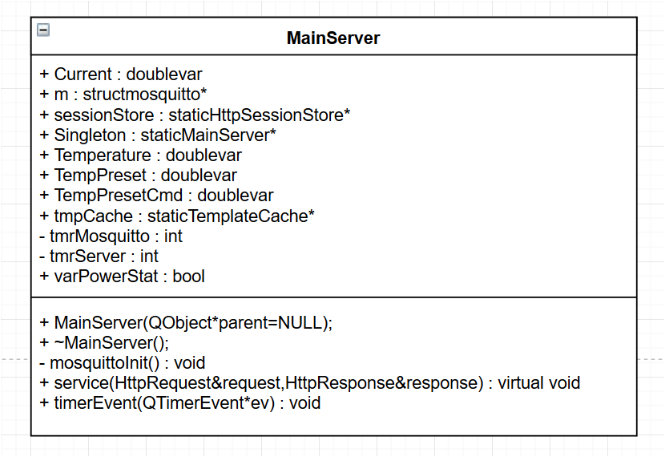


3.5.4 Class diagram

Raspberry side:



Server side:



3.5.5 Definitions: *See Appendix II and III*

3.6 Implementation

3.6.1 Development tools

1. Hardware
    1. Main module: Raspberry Pi 3 model B+ (2017) for the main system host and processor for reading sensors and sending sensor data;
    2. Module: IR receiver containing TSOP4838 IR receiver for testing and mapping real remote keys into binary codes of infrared pattern;
    3. Module: generic IR blaster to blast infrared signals into the AC's infrared receiver;
    4. Module: SI7021 humidity and temperature sensor with I2C port address 0x48;
    5. Module: AD7124 Analog-to-digital converter with SPI port (discontinued);
    6. Module: ADS1115 Analog-to-digital converter with I2C port address 0x48/49/4A/4B;
    7. Module: SCT-013 current sensor (discontinued)
    8. Module: 5600t100c current sensor + testing cable and power supply socket;
    9. Development boards: TBN061807005/06/07;
    10. Appliance: LG air conditioner model F05NXA;
    11. Connector: LAN + LAN-to-USB connector (as a Raspberry Pi display);
    12. Power supplies: (2 pcs. 5V/2A power supplies) for Raspberry Pi and stable external pin power supply

2. Software
    1. OS: UNIX Raspbian Stretch
    2. IDE/Editor: Vim Text editor
    3. IDE/Editor: Text Editor
    4. IDE/Editor: Sublime Text 3
    5. IDE/Editor: Nano Text Editor
    6. IDE/Editor: Qt Creator 4.2.0, 4.5.2
    7. Library: mosquitto MQTT (Message Queueing Telemetry Transport) for Javascript, C++ and C
    8. Library: LIRC (Linux Infrared Remote Control)
    9. Library: WiringPi for controlling GPIO (for debugging)
    10. Library: LibCURL for internet connection via C code
    11. Library: QWebApp by Stefan Frings
    12. Library: Qt 5.7.1
    13. Application: XAMPP for building a server in localhost (Phase 2 only)
    14. Web hosting: 000webhost (Phase 1 only)
    15. Database manager: PHPMyAdmin + MySQL
    16. Git Repository: GitHub via GitKraken
    17. Documentation: Draw.io
    18. Documentation: Microsoft Office 2016
    19. Display for Raspberry Pi: VNC Viewer
    20. Raspberry Pi Job scheduler: Crontab (Phase 1 only)
3. Programming languages and protocols
    1. C, C++
    2. HTML5
    3. JavaScript (Phase 1)
    4. PHP 5.5 (Phase 1)
    5. MySQL (Phase 1)
    6. Python 3 (for testing)
4. Protocols
    1. MQTT protocol
    2. LIRC infrared protocol
    3. Serial Peripheral Interface (SPI) protocol (discontinued)
    4. I^2C protocol
    5. HTTP, TCP

3.6.2 Techniques

Hardware techniques

1. Interfacing – commanding sensors through a controller (Raspberry Pi) and getting data from them via some communication protocols such as SPI and I2C
2. Basic electrical engineering concepts – connecting modules together based on the current flow, voltage levels, logic gates, et cetera.
3. Linear Regression (scaling) – used in rescaling outputted binary values from sensors to be calibrated to the real world scenario.

Software techniques

1. Multithreading – running more than one process at once
2. Synchronization – ensuring only one process accesses one resource at a time
3. Object-oriented programming – treating entities in an application as objects
4. Web programming – building web applications and integrating them to remote systems
5. Communication via MQTT, an IoT protocol – subscribing and receiving IoT control and acknowledgement messages
6. Relational database – storing data as a table, and modify the data using queries

3.7 Source Code

The source code is available at: https://github.com/cdsw/tbnintn

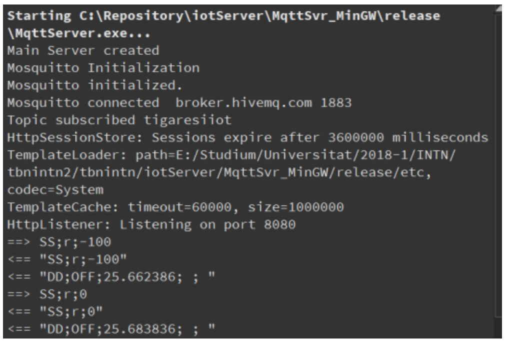
3.8 Testing

Hardware: Some hardware debugging needed multimeters to measure the outputs or voltages between two important buffers/outputs.

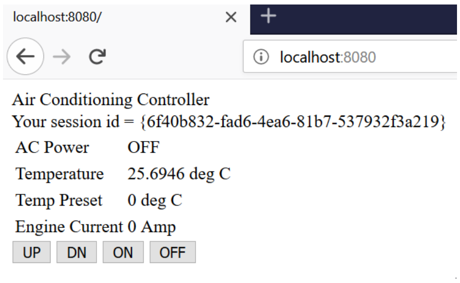
Software: We used a simple testing technique by tracing the logic of the code, putting the code to use, then comparing the real output with the desired output.

3.9 Screenshots and Images

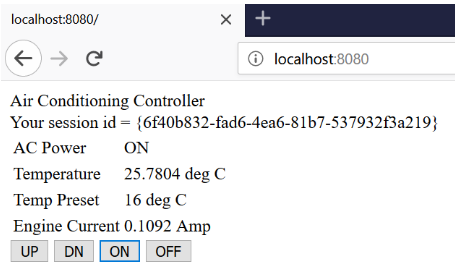
- Phase 2/Final prototype



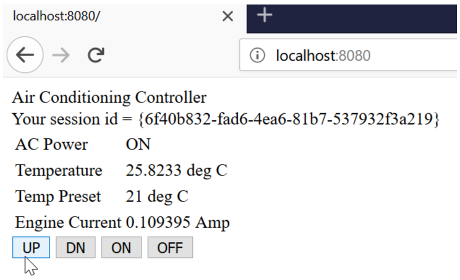
Console view for the server side



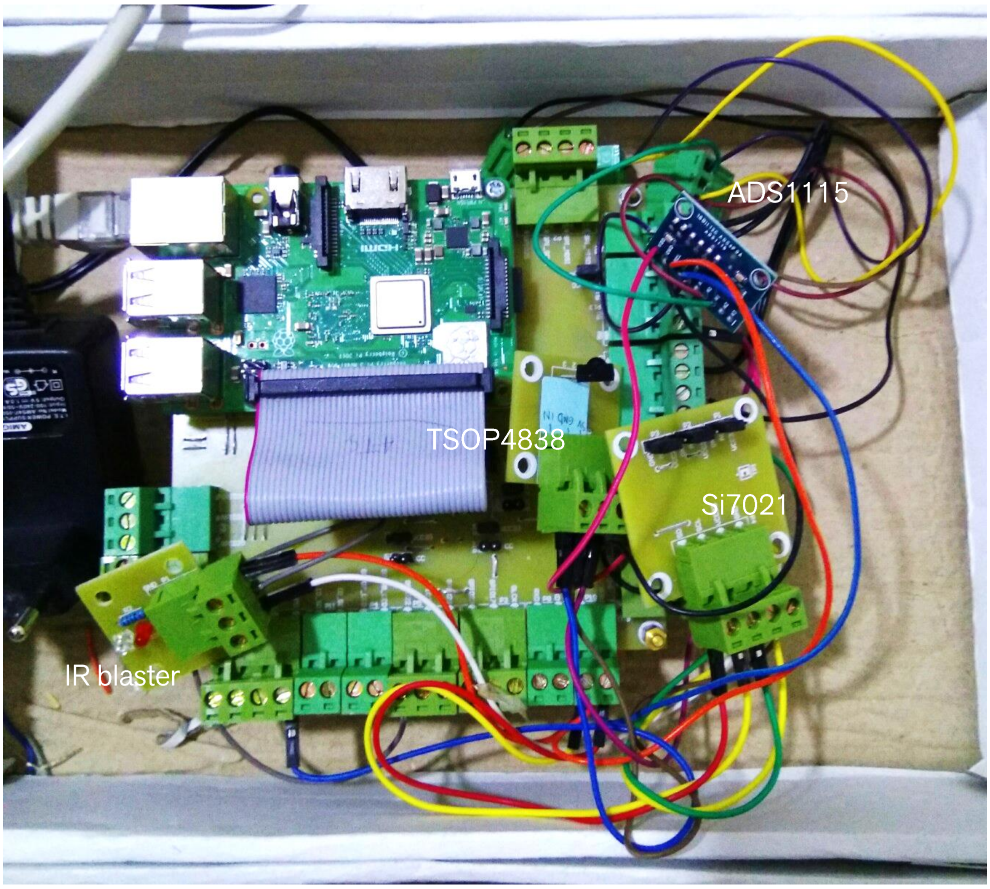
Browser HTML controller/monitor view: Initial screen (AC turned off)



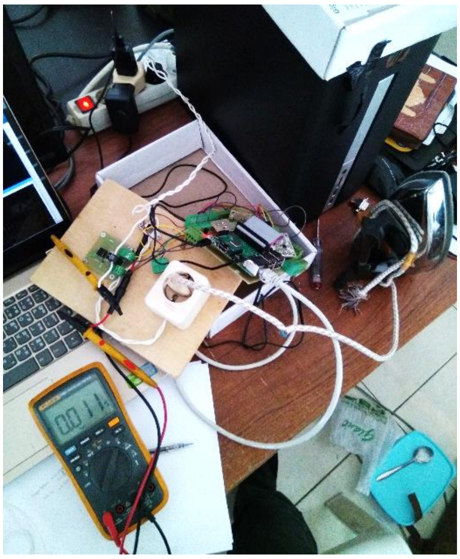
Browser HTML controller/monitor view: AC is turned on



Browser HTML controller/monitor view: set temperature up

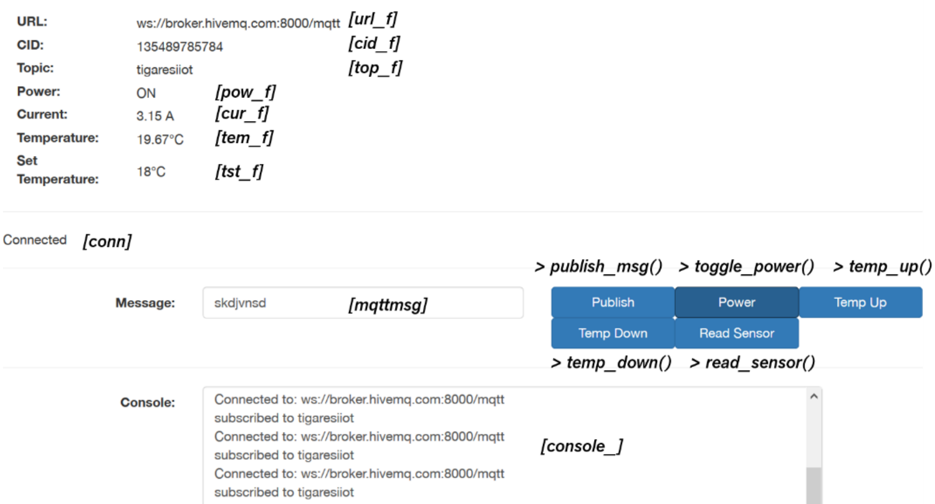


Development board for this project and some components

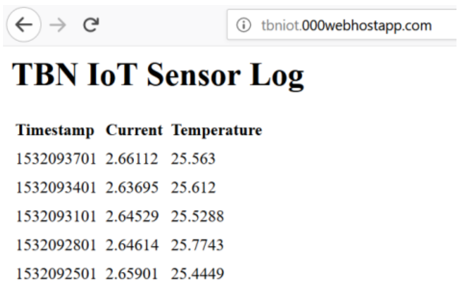


Development board and 56t100c current sensor plus a 220v power socket

- Phase 1/Deprecated version



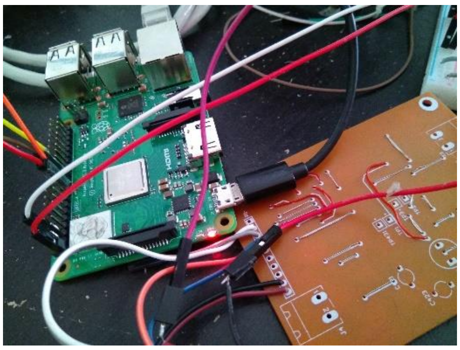
Old HTML controller/monitor page and the component names


tbniot.000webhostapp.com

TBN IoT Sensor Log

Timestamp	Current	Temperature
1532093701	2.66112	25.563
1532093401	2.63695	25.612
1532093101	2.64529	25.5288
1532092801	2.64614	25.7743
1532092501	2.65901	25.4449

Old online database view



AD7124 SPI Analog-to-Digital module

4. Achievement

4.1 Summary of work done and Evaluation

Phase 1 – The system in Phase 1 has already worked fully (100%): the web control can get data from the Raspberry Pi sensors and can control the main system in Raspberry Pi through its interactive buttons. Also, Raspberry Pi can send log data per 2 or 5 minutes to a remote online server. Yet, the system is not too secure and inefficient. I wrote approximately 400 lines of C code, 300 lines of HTML/JS code excluding content, 45 lines of PHP code and ~100 lines of Python testing codes.

Phase 2 – Most of the codes were given by the company's president, mainly in Qt and C++. The old phase was transformed into OOP and heavily modified. I myself transferred the code from C to C++ with much adjustments equivalent to around 200 lines of C++ code and 100 lines of HTML/JS code. The system started to work on July 31, in which the HTML page could control the Raspberry Pi's main system, and Raspberry Pi main system could respond in time. The prototype was completed, except that there has not been any database storing or logger.

Despite having finished Phase 1 prototype and got it to work, the program was not really well-structured and still have a large potential to get a bug. That was the reason why we developed the application more in Phase 2. Overall, everything worked smoothly and the system could run as defined in the software requirements.

5. Conclusion

5.1 Skills and knowledge learned

Having studied thoroughly at International College KMITL, most of the courses that I took helped me to be more prepared when I started working at Tigaresi Bangun Nusaperdana, especially Object-Oriented Programming classes, *Computer Organization and Assembly Language, Operating Systems* and *Microprocessors and Interfacing.* Those knowledge are coherent to the project assigned to me, which involves Linux on Raspberry Pi, with some interfacing jobs in C++.

There are sorts of new skills which I learned, including web design and programming, IoT and server communication, threading and synchronization. But, the most challenging job that I did was all tasks using electrical equipments. Hardware works dominated most the entire first 10 work days, from soldering components (sensors) for testing, connecting wires via a breadboard, designing hardware connections, to debugging faulty hardwares/connections.

Besides technical aspects, I learned several non-technical skills too. I was "forced by nature" to be always curious and to ask questions to as many staffs as I can, in order to thrive in my internship. Not only curious, but the internship taught me how to work independently with less guidance and breaking communication challenges. However, all off these new knowledge do nothing other than improving my personal qualities.

As for the resources which I took during the internship, most of them were taken from various sources from the web, such as Raspberry forums, Qt forums, some technical blogs and definitely stackoverflow. Moreover, I also had to search for the datasheets of every hardware modules (sensors) and software libraries, in their respective producers' website. Without combining all knowledge from aforementioned materials, the project would not have been as successful as it is now.

5.2 Problems and possible improvement

There were moments where problems arose, and finding solutions required much time. The main problem encountered was about the requirement. The job description did not specify how and using which way the project will be carried out, so I had to bring my own initiative to start and work on the project, with some consultation with the staffs. Moreover, most of the time I was working by myself, especially in Phase 1's software side, which I believe that I was too independent.

The second major problem was hardware connection faults (disconnection or voltage inconsistence) which were hard to debug. But fortunately, the staffs at TIgaresi are very experienced in detecting hardware faults. The biggest among such problem was SPI dysfunctionality, in which the staffs and I used up around 9 days to figure it out, without any result. We decided not to continue using AD7124 module, eventually.

Though, the mentioned problems did not significantly affect the project.

6. Acknowledgement

I would like to express my greatest gratitude to Mr. Raymon Mudrig, the president of PT. Tigaresi Bangun Nusaperdana for having accepted me to get involved in one of TBN's project for 35 days and for supporting me very much in the process of the IoT development. Furthermore, many thanks to Mr. Victor P. Limbong for being my caring supervisor. Last but not least, I thank the International College, especially Dr. Natthapong Jungteerapanich for guiding, preparing and arranging this internship program so that students could know how to live the life in the real software industry.

7. Reference

[1] tigaresi.wordpress.com/about/ (2018/07/16)

[2] internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT (2018/07/28)