IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 2, June 2024, pp. 1817~1837
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i2.pp1817-1837

Neobots: an open-source platform for a low-cost neonatal incubator with internet of things approach

I Komang Agus Ady Aryanto[1], Dechrit Maneetham[1], Evi Triandini[2]
[1]Department of Mechatronics Engineering, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand
[2]Department of Information Systems, Faculty of Informatics and Computers, Institute of Technology and Business STIKOM Bali, Bali, Indonesia

Article Info

Article history:

Received May 20, 2023
Revised Nov 8, 2023
Accepted Dec 3, 2023

Keywords:

Fuzzy logic
Internet of things
Low-cost
Neonatal incubator
Open source
Web

ABSTRACT

A baby incubator implements the internet of things (IoT) with an architectural design combining several scientific fields, such as networks, software, and hardware. Furthermore, this research develops an open-source platform called Neobots, including open-source program code to create a baby incubator. Then an overview of the system includes sending sensor data to the IoT Broker with the message queuing telemetry transport (MQTT) protocol and automatically storing data in the database. The results of the comparison value on each temperature sensor with a temperature sensor at the midpoint with an error of less than 0.7˚C. Then testing the fuzzy between the Neobots program and the simulation in MATLAB got an error rate of 0-28.27%. In addition, in less than 10 minutes, the system response can adjust the temperature conditions to a setpoint value of 34˚C from 29˚C, and the average error value is 0.35˚C during 1 hour of the Fuzzy implementation on the incubator. Then transfer data from the incubator to the database in a room without noise and full noise to get results for lost data less than 16.41% and 42.14%, delay rates between 0-6 seconds and 0-7 seconds with testing for 1 hour at every 1 second.

Corresponding Author:

Dechrit Mannetham
Department of Mechatronics Engineering, Faculty of Technical Education
Rajamangala University of Technology Thanyaburi
Tambon Khlong Hok, Amphoe Khlong Luang, Chang Wat Pathum Thani 12110, Thailand
Email: dechrit_m@rmutt.ac.th

1. INTRODUCTION

Neonatal are compassionate creatures with delicate skin; the growth of a baby's life will depend on the birth process and its care. Normal baby newborns have a gestational age of 37 weeks to 42 weeks, a baby weight at newborn is 2,500 grams to 3,500 grams, and a body length of 48 cm to 50 cm. On the other hand, babies born unprepared (premature) or taken with complications, called high-risk babies, quickly get sick because of limitations and impaired organ function [1]. These high-risk babies must receive high-level care and be appropriately handled. This is necessary to prevent the baby's condition from worsening, and it is hoped that the baby can be saved in optimal health conditions. Good care requires adequate medical facilities and equipment, including a baby incubator. A baby incubator is a medical device that is used to optimize humidity and temperature within normal limits and is helpful in minimizing the risk of direct contact with other people and the environment that has the potential to transmit the disease because the baby's organ functions are still not good. The incubator is one of the medical supports commonly used in the emergency room or neonate intensive care unit (NICU) [2]. In general, the working principle of a baby incubator is to manipulate several parameters, namely temperature, humidity, heartbeat, weight, and lighting. This manipulation process must constantly be monitored until the baby's condition meets normal requirements. The need for this baby incubator is very large, while in the hospital there is still little availability, this is due to the high price. Therefore, a study is needed to find a new model of a baby incubator at an affordable price that can monitor and control from anywhere, anytime, regarding the baby's condition.

The Internet of Things is a concept to maximize the use of hardware such as sensors and actuators through an internet connection [3], [4]. The meaning of the word "A Things" in the Internet of Things can be associated with an object that can communicate with one another or can be described by communication such as machine-to-machine (M2M) [5], [6]. Each object has a unique identity to be able to communicate with other things automatically without the need for human assistance [7], [8]. The main challenge in internet of things (IoT) is connecting objects in the real world with digital information technology, such as managing sensor data on things and then converting them into a digital format to become parameters to act on the control process. Then the formatted data is exchanged with other objects through the communication network [9].

Therefore, in this study, a baby incubator with the Internet of Things was developed with an architectural design that combines several fields of knowledge, such as network, hardware, and software [10]. The hardware section uses a microcontroller and various types of sensors and actuator modules. The sensor modules include temperature, humidity, heart rate, weight, and lighting. The Arduino Mega has a microcontroller chip that can be programmed. The microcontroller can also be programmed repeatedly, the last installed program will function, and the programming language is based on C [11], [12]. The Arduino Mega board uses a microcontroller of the ATmega2560 type, a flash memory capacity of 256KB, a required input voltage is 7-12V, and 54 I/O pins [13]–[15]. In addition, a Raspberry Pi microcomputer is used as an IoT Broker server for the message queuing telemetry transport (MQTT) protocol. The Raspberry Pi has a USB port, LAN port, HDMI port, micro-SD slot, camera port, and GPIO pin. Raspberry pi can be turned on with 5V 2A power which means it is very energy efficient [16], [17].

Then the software developed a platform to perform data processing to produce a decision in carrying out control actions on the incubator. In addition, this software design includes storing data in a database. Each sensor module can send data to an IoT Broker automatically then the data is handled by Socket as a service to store data in the database. Technically, this module is server software (IoT Broker) for the loop function with the scheduling process between objects. Then the software interface was developed web-based by applying the latest technologies such as the JavaScript (JS) framework, connected secure systems (CSS) framework, and internet of things tools that support using the MQTT protocol to communicate objects with IoT Broker. In addition, the Asynchronous Javascript and XML (AJAX) technology implemented in web applications makes the application work indirectly if it processes data to the server. The workings of AJAX technology start from the web browser sending HTTP requests to the server. Then the server processes client requests, and the results are returned to the client and displayed on the web page without further interaction. So that it can shorten processing time when accessing the web and simplify the user experience [18], [19]. Finally, this research uses network technology Wi-Fi called IEEE 802.11 with the work of using radio networks to communicate between devices. Wi-Fi has different types that affect the bandwidth of communication [20], [21]. The sensor values in the incubator will be sent to an IoT Broker with a Wi-Fi module connected to the internet. This network Wi-Fi is installed to the controller in the incubator and configured according to service set identifier (SSID) and password.

Then to, adjusting the temperature in the baby incubator is automatically designed as a control system with the fuzzy method [22]. Using the fuzzy method, guidelines are needed in regulating the temperature based on plant parameters that aim to get a rule evaluation. The fuzzy control system applied uses a microcontroller as the core processor to read the sensor and control values based on the output values obtained by fuzzy. Previously, this fuzzy design was made using the MATLAB application to see directly the output results obtained in the simulation in MATLAB.

Furthermore, this research creates hardware and software and expands to create an open-source platform. Researchers are trying to develop a platform for a baby incubator machine called Neobots, which can monitor and control automatically with fuzzy method and the internet of things concept. This platform includes designing hardware and software with open-source code to create a baby incubator machine. In addition, the required hardware, such as sensor and actuator modules, are easy to find on the market at relatively affordable prices. Hopefully, this platform will enable everyone to make incubator machines quickly and effectively.

2. RESEARCH METHOD

Figure 1 shows the research methodology; the initial process begins by looking for references regarding related research. This reference can be in the form of journals, scientific papers, books, or other information that can support research activities. The next step is to analyze the system requirements. The research to be carried out is designed with an architecture combining several scientific fields consisting of architecture for hardware, software architecture for software, and internet network communication technology.

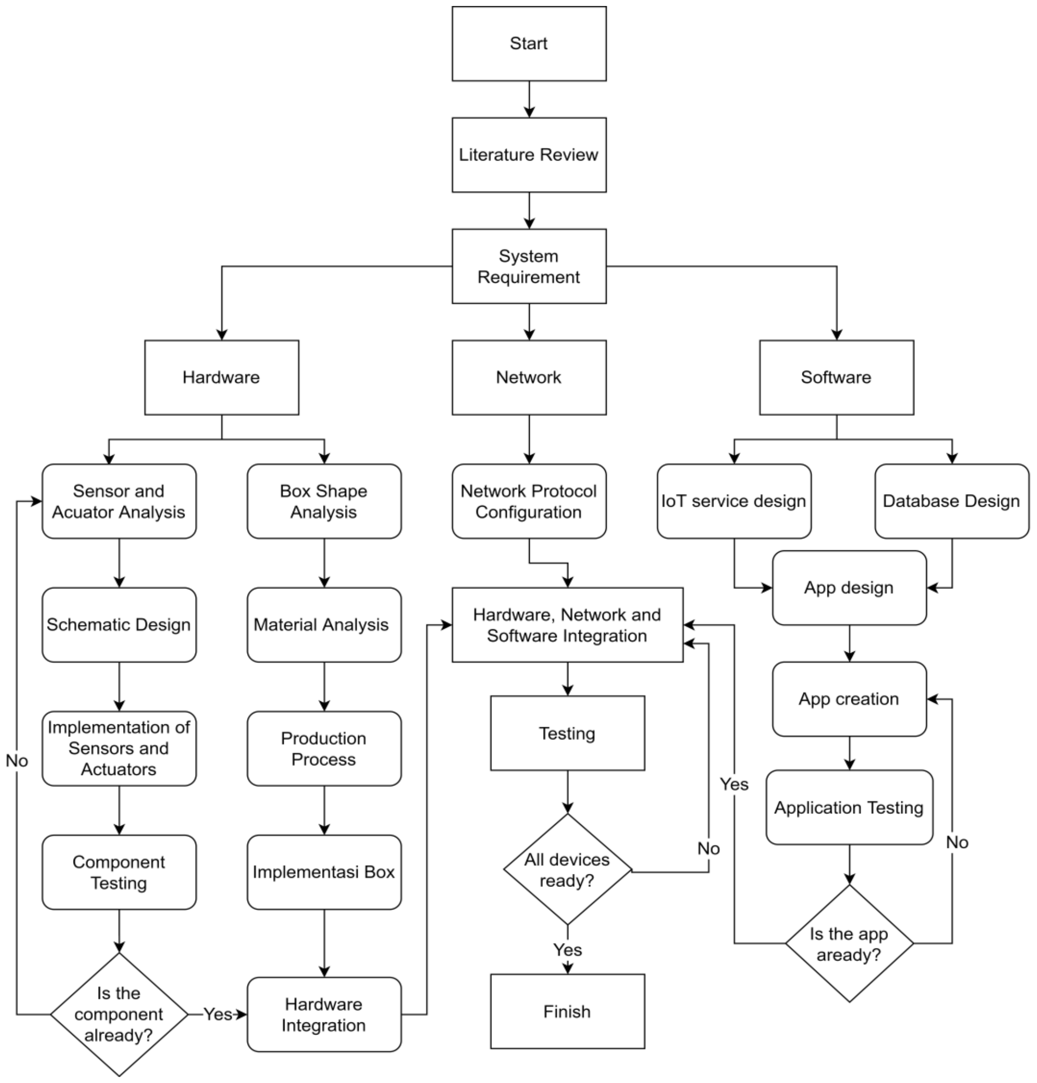


Figure 1. The research method

Hardware design architecture is separated between architecture for incubator prototypes and electronic devices for incubators consisting of microcontrollers, sensors, and actuators. The first incubator prototype was made using a 3D model. After that, making a simulation design for electronic devices using the Proteus application. If the electronic simulation has been completed, assemble electronic devices using sensor and actuator modules connected to the microcontroller. Then making program syntax using the C programming language embedded in the microcontroller. This hardware architecture's last stage is integrating the electronic module with the incubator prototype.

In the software section, starting from the system requirements analysis process, the next step is the software design process. The analysis process is divided into two stages: Internet of things service analysis and database design analysis. These two processes are carried out together, and when finished, they are integrated into an application. However, at this stage, only the software design process must proceed to the stage of making the application with the program syntax using the appropriate programming language. After the application development is complete, it is continued by conducting the application testing process before integrating it with the hardware. Integration of these devices requires a network that can connect hardware and software. This network process is needed to configure the protocol and SSID of the network. The communication protocol used is MQTT which supports communication between hardware and software.

Then, if the network configuration process is complete, the software and hardware can be integrated. Finally, system testing will continue to be carried out until the hardware and software can run properly, and there are no errors.

2.1. System overview

Figure 2 is an architectural overview of the system in a baby incubator. A microcontroller is equipped with a network module and a real-time clock (RTC) on the hardware. Then each sensor and actuator module is connected to the microcontroller and assigned a unique identifier to facilitate the communication process to the IoT Broker. In addition, this identity also reduces the occurrence of failures in the communication process in each sensor module.

Using the MQTT protocol, each sensor measurement value can be sent (publish) to the IoT Broker or vice versa; the microcontroller can receive data (subscribe) from the IoT Broker. That means with the implementation of IoT, all objects can communicate with each other between other things through a network [23]. Therefore, the device will continuously request and send data. The data that has been sent will be entered into an IoT Broker before being stored in the database. This storage process uses a Websocket with output data in JSON format, which is then handled by the Webservice to be stored in the database. The data stored in the database will be analyzed using programming algorithms to be used as parameters in the control system. The results can be monitored on a web application page.

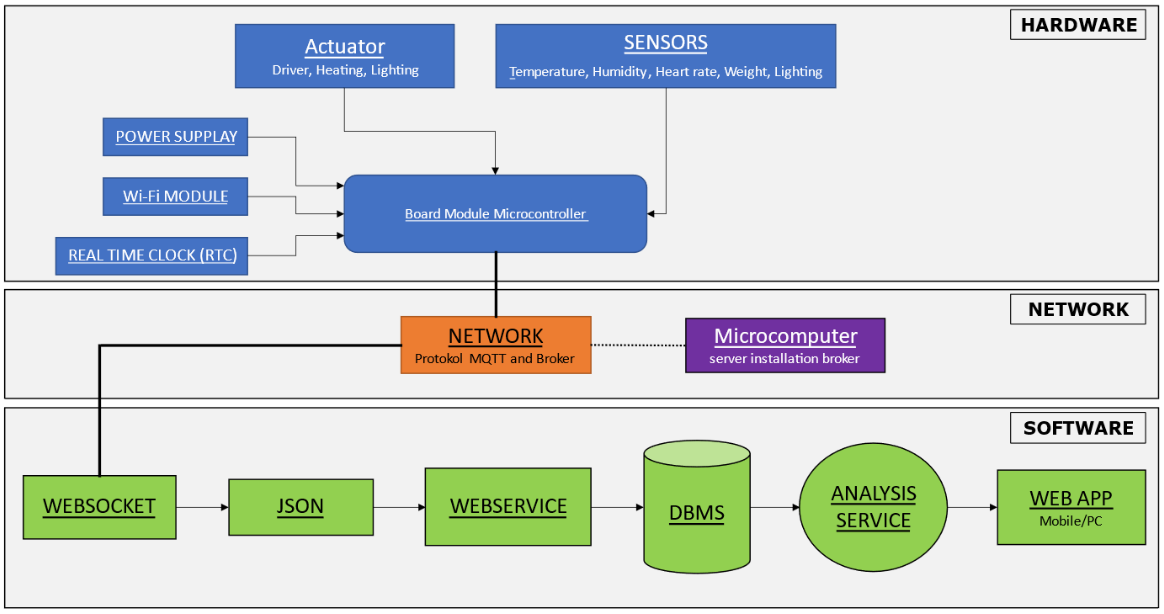


Figure 2. The architecture system baby incubator

2.2. The mechanical design and development

A neonatal incubator is a medical device designed to provide special care to babies who are born prematurely. Infant incubators provide a controlled and isolated environment, thus helping to maintain a stable condition according to the baby's needs. The use of an infant incubator can help increase the chances of survival for babies who are born prematurely, as well as reduce the risk of medical complications that may occur.

Figure 3 shows the design of the baby incubator used in this study. In general, the size of an incubator is 90×45×122 cm3. The bottom of the baby incubator is also equipped with wheels to make it easier to use and move from one room to another. On the side is a panel as a container for electronic incubator devices such as microcontrollers, microcomputers, chip drivers for sensors or actuators, and power supply adapters. In the middle of the incubator, there is a panel box as a heating controller to produce hot air, which is then channeled through the ventilation system. A heater and a fan are inside this panel used to regulate air circulation shown in Figure 4. The heat generated is then channeled by the blower fan into the baby incubator room through the air ducts. There is also a channel inside this panel to release air if the air temperature inside the incubator exceeds normal limits or the air quality level is terrible.

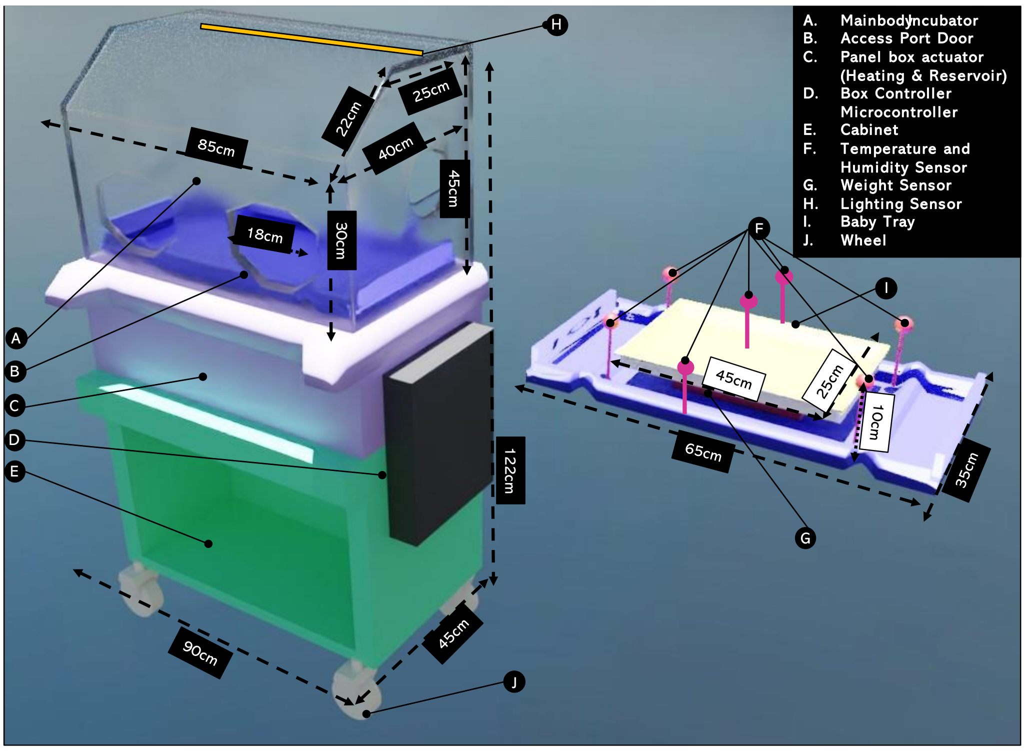


Figure 3. The mechanical design and development

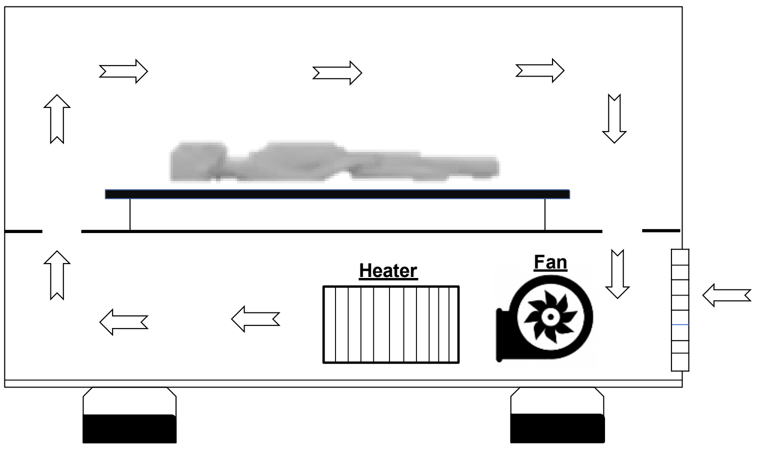


Figure 4. The temperature flow process

Then the central part of the incubator is made of transparent plastic, allowing nurses or doctors to monitor the baby's condition without having to open the central part of the incubator. In addition, the baby incubator is also equipped with various medical devices such as heart monitors, temperature, humidity, weight, and lighting to help monitor and maintain the baby's health. The number of temperature and humidity sensors installed in the incubator is four arranged around the incubator's main panel. This allows the system to measure or monitor temperature conditions evenly at each incubator point (T_1, T_2, T_3, T_4, T_5, T_6, T_7). Then the light in the baby's incubator is also measured using a sensor installed. This light sensor adjusts the light conditions to provide enough light for baby care. In addition, the baby's weight is also measured directly. This weight sensor is installed under the mattress of the baby. Then the rules for using the incubator are shown in Table 1.

Table 1. The rule on the incubator

Weight	Incubator temperature by age			
	35°C	34°C	33°C	32°C
< 1.5 kg	1-10 day	11day – 3 week	3-5 week	> 5 week
1.5 – 2.0 kg		1 – 10 day	11day – 4 week	> 4 week
2.1 – 2.5 kg		1 – 2 day	3 day – 3 week	> 3 week
> 2.5 kg			1 – 2 day	> 2 day

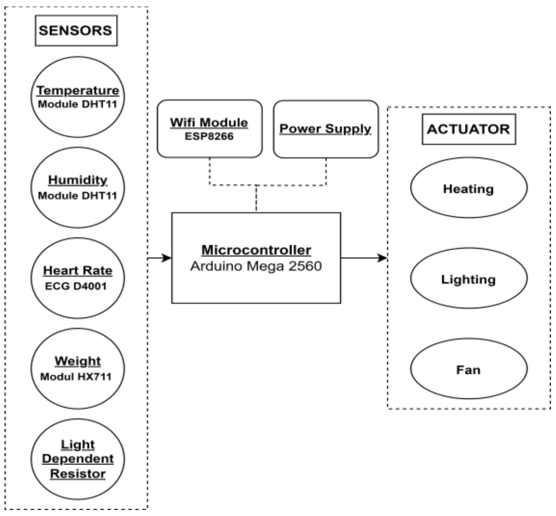
2.3. The electrical hardware designs

In this study, the hardware used consists of a controller (actuator) and a data processor (sensors). All this hardware is integrated into one piece, as is the microcontroller board, mechanical drive, relay, Wi-Fi, sensors, actuator, and power supply by 5V 2A. This hardware is designed to work efficiently, with the microcontroller as the brain of this system, as shown in Figure 5(a). This microcontroller controls actuators to manipulate environmental conditions based on data collected by sensors. While the Wi-Fi module is used to facilitate wireless communication between the system and the database, so that data transfer can be carried out.

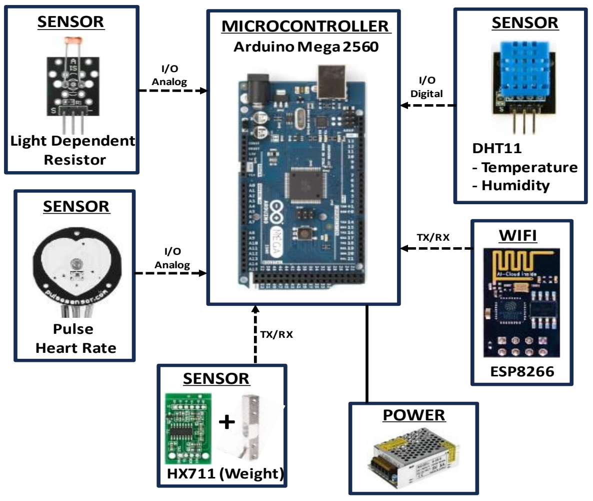
The sensors installed function to measure environmental conditions in the incubator. This sensor module is installed in the incubator and placed at each point in the incubator. The process of measuring the environmental conditions of the incubator by the sensor is processed by the microcontroller with a time delay that has been set in the program. The sensor measurement values obtained are used for input in the control system. Five types of sensor value parameters are measured in this study: temperature, humidity, heart rate, weight, and lighting (Light dependent resistor).

The schematic circuit design is shown in Figure 5(b) core processing uses the ATmega 256 microcontroller, which is installed on the Mega version of the Arduino Board platform. Arduino Mega is a board product from the Arduino company. The specifications of the Arduino Mega are the flash memory capacity of 256kb; the required input voltage is 7-12V; it uses an ATmega2560 microcontroller chip with 54 I/O pins [24].

The sensor device uses a DHT11 module to measure temperature and humidity, a light dependent resistant (LDR) sensor, a heart rate sensor, and a weight sensor. The sensors used are connected to the microcontroller via analog, and digital I/O pins, and in the microcontroller program syntax, each sensor is given a unique id as its identity. DHT11 module is connected with digital pin 12, the heartbeat sensor with analog pin 1, the LDR sensor with analog pin 0, and the weight sensor driver module are connected with serial communication (TX/RX) number 2 in board microcontroller.



(a)



(b)

Figure 5. The hardware design for, (a) The wiring flow diagram and (b) The schematic board circuit

According to the procedures for other electro-medical devices, the baby incubator device should be calibrated periodically to prevent malfunctions that could compromise the baby's health. Based on the standard of IEC60601219 stipulates technical specifications for the safe operation of incubator devices for newborns. One of the requirements is that the temperature sensor value at point middle (T_7) cannot be more than 0.5ºC with the setpoint temperature (T_{set}) value in the incubator(ΔT) [25].

$$|\Delta T| = \Delta T_7 - T_{\text{set}},\ |\Delta T| \leq 0.5 \tag{1}$$

Hardware module for the communication process with IoT Broker via the internet network using module Wi-Fi ESP8266 makes it easier to communicate between objects (sensor) in the incubator [26], [27]. Then the energy source used in the sensor electronic device is a 5V 2A power supply. This power supply is the standard used in mobile devices. Thus, it can be applied quickly and does not use too much energy.

The device used is an Arduino board programmed with the wiring programming language with an editor and compiler using the Arduino IDE [28]. Program development in this device is divided into several sub-sections that regulate the looping process in the microcontroller. The initial sub-process is an initialization, which is the first to run when the microcontroller is on. In this initialization program, there is a library calling process and variable declaration for the address of the sensor pins used. Then the following process is the setup process, which is run only once when the microcontroller is on. Then the last one is a looping process where all the sensor measurement processes, the communication process through the network with the MQTT protocol for sending/receiving data, and the control process of the accumulator in the baby incubator. The following in Figure 6 is a design form of the existing microcontroller or hardware firmware and an explanation of the program syntax flow with pseudocode as shown in Algorithm 1. The microcomputer module is a mini computer connecting the microprocessor and main memory as an input-output interface [29]. This study uses a Raspberry Pi Model 4 microcomputer; this computer board is installed with a Linux-based operating system and is used as a core processing device that serves as an IoT Broker and Apache web service. The IoT Broker used is Mosquitto which is specially designed for communication with the MQTT protocol.

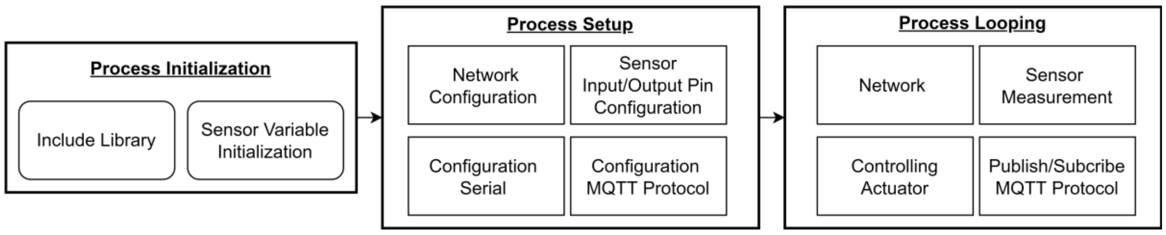


Figure 6. The design firmware microcontroller

Algorithm 1: Hardware firmware

```
1.  DESCRIPTION :
2.      VAR sensor, output: FLOAT
3.      VAR last send: LONG
4.      VAR setPoint: INT
5.  PROGRAM :
6.     INITIALIZATION
7.        SET pinNumberSensor = pinDigitalAnalog
8.        SET broker_mqtt_account = server, port, username, password
9.        SET wifi_account = username, password
10.    END
11.    SETUP
12.       wifi = wifi_account
13.       mqtt = broker_mqtt_account
14.       pinNumberSensor:PINMODEOUTPUTINPUT
15.       fuzzy = MembershipInputFuzzy()
16.       fuzzy = MembershipOutputFuzzy()
17.       fuzzy = MembershipRuleFuzzy()
18.    END
19.    CALLBACK(topic:char, paylod:byte, length:int)
20.       IF(topic == topicSetpoint):
      Setpoint = payload
      PRINT length
    END
  END
  LOOP:
    IF(!wifi.status ):
      WHILE (!Wi-Fi.status):
        Wi-Fi.status = Wi-Fi.begin(username, password)
      END
    END
    IF(wifi.status AND !mqtt.status):
      mqtt.status = mqtt.connect (id,username,password):
        IF(mqtt.status):
          PRINT mqtt.status
          SET topicSetpoint = subscribe("topic")
          mqtt.CALLBACK()
        END
    END
    IF (milis() – lastsend > time AND mqtt.status):
        value = ReadSensor(pinNumberSensor)
        mqtt.publish("topic", ConvertToString(value))
        error = setpoint – ValueSensor(value)
        error_speed = error_speed - error
        fuzzy = setInput(error, error_speed)
        output = fuzzy->defuzzify(1)
        PRINT output
        Lastsend = milis()
    END
  END
END
```

2.4. The design of software

Figure 7 is a design for the baby incubator system software. This software design's work process starts with a module such as sensors, microcontrollers, and actuators reading data or controlling an object by the program's syntax. Each of these objects has a unique code to identify the object's identity. The things in this study are temperature, humidity, lighting, baby weight, and heart rate. The unique code that exists in these objects has been set at the time of coding the program on the microcontroller. The application of syntax on the microcontroller refers to the type of object that matches the object's function, whether used for monitoring or controlling. In addition, the program created must also be able to make objects on the device able to communicate data (TX/RX) over a Wi-Fi network with the MQTT protocol.

The data obtained from the microcontroller is sent via the internet network with the Wi-Fi module installed on the object. The data transfer process uses the term publish/subscribe in the MQTT protocol. This transferred data has a topic according to the data communicated on the device. Then this data is entered into the IoT Broker application, a container accommodating temporary data. This IoT Broker program is installed on the Raspberry PI microcomputer by the IoT Broker application Eclipse Mosquitto. In addition, the raspberry pi is also a server to install Apache as a web server. The IoT Broker configuration steps are carried out as follows: first, the mosquito IoT Broker installation process. After installation, continue with the Broker port configuration and Broker authentication settings for security.

Furthermore, the data on the IoT Broker is stored in a database with several stages of the process. The process starts by retrieving data from the IoT Broker using a socket service custom-made with the python programming language. This socket service is equipped with features for account verification by the IoT Broker to make the data communication process more secure. In addition, any data taken with the socket service must subscribe (*topic*) to the topic so that the IoT Broker can publish data. Then these data are stored in the database with the web service program automatically. The process of storing in the database according to the tables and columns of each data according to the object of the baby incubator. Then the web application can request data in the database based on its access rights. The server then processes the requested data, and the results are sent back and displayed on the web with a responsive interface page according to the user experience. This web application development is made with a combination of front-end and back-end programming languages such as Javascript, cascading style sheets (CSS), Hypertext Preprocessor (PHP), and Python programming languages.

The data is displayed on the web application page with an interface in graphs, tables, and numbers to make it easier for the operator to process all activities in the baby incubator. The web application interface design uses responsive modern web technology to be displayed on various screen sizes. Then the process of program syntax in the software is explained by pseudocode as shown in Algorithm 2.

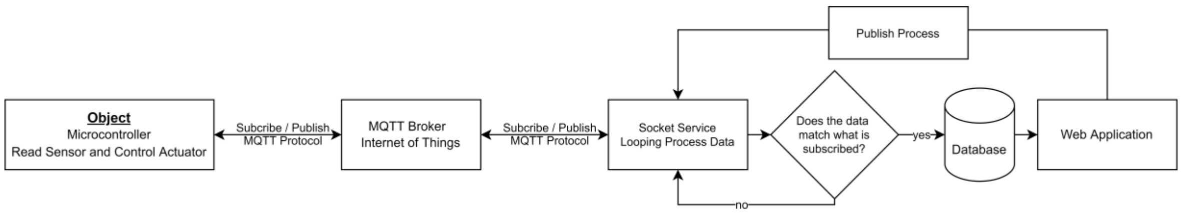


Figure 7. The flow diagram of software design

Algorithm 2: Software firmware

```
1.  PROGRAM :
2.      VAR valueSensor[]:STRING
3.      SET valueSensor[] <- ReadValueSensorMicro()
4.      IF(Publish(valueSensor[]):
5.          LOOP:
6.              STORE MQTTBroker <- valueSensor[]
7.              VAR data:STRING
8.              SUBSCRIBE data <- MQTTBroker
9.              STORE database <- data
10.             SET JSON <- Database
11.             DISPLAY WebApp <- JSON
12.             SET setpoint <- InputSetpointOnWebApp
13.             PUBLISH value <- setPoint
14.         END
15.     END
16. END
```

2.4.1. The data description

The data obtained during the research is described in Table 2, which displays the data structure of the sensor devices used in this study. Each row and column display certain variables using five different sensor device types and having variations in variables and values. For example, the DHT11 sensor device has variables named Temperature and Humidity, where the value of the temperature variable has a range between 0 – 50°C while humidity has a value range between 0-1023 %RH. This type of data is included in the numerical or quantitative data category. Sensor measurement results are stored according to the table structure in the database automatically. This database consists of seven types of tables to store each data obtained from the sensor. The database design structure can be seen in Figure 8. Data stored in the database can be exported in .csv or .xml format. In addition, the time of the sensor value measurement process is recorded in the DateTime format; this time is obtained from the hardware module named RTC Real-time on the incubator. So, every sensor measures the value at the incubator; the measurement time is also processed to be stored in the database.

Through the data that has been described, it will be able to explain the form of the data in a systematic and detailed manner, making it easier to understand the characteristics of the data, including size, value, variation, data type, and pattern contained therein. This helps prepare the data for further analysis and makes it easier to understand the results of the resulting study in the future. In addition, a detailed description of the data helps in identifying data patterns.

Table 2. Data description captured by the logger device

Device	Data Variabel	Example Value	Unit
DHT11	Temperature	36	°C
	Humidity	60	%RH
RTC Real-time	Date-Time	2023-03-22 11:12:20	yyyy-mm-dd hh:mm:ss
Light Dependent Resistors	LDR	821	Lux
HX711 Weight	Weight	2.4	Kg
SEN-11574 pulse sensor	Heart rate	64	BPM

2.4.2. The data entity relationship

Figure 8 is the database structure relationship contained in the system. The data stored in this database results from measurements from the sensors installed in the baby incubator. The stored data is adjusted to its type, which aims to classify the data so that it is easy to manage in the controlling process. In this study, MySQL is one of the usual database management systems used to collect data. This uses SQL as a programming language between application software and a database server [30], [31].

Then the table relations in the database consist of seven types: *humidity tables, temperature tables, ldr tables, weight tables, heartrate tables, operator tables, and incubator tables*. All these tables have a relationship to the incubator table. In this database design, all tables have a connection to the incubator table with a one-to-many relationship, which means that one incubator can have many temperature data. In contrast, one temperature data can only have one incubator data. The details of each database table's columns are shown in Table 3.

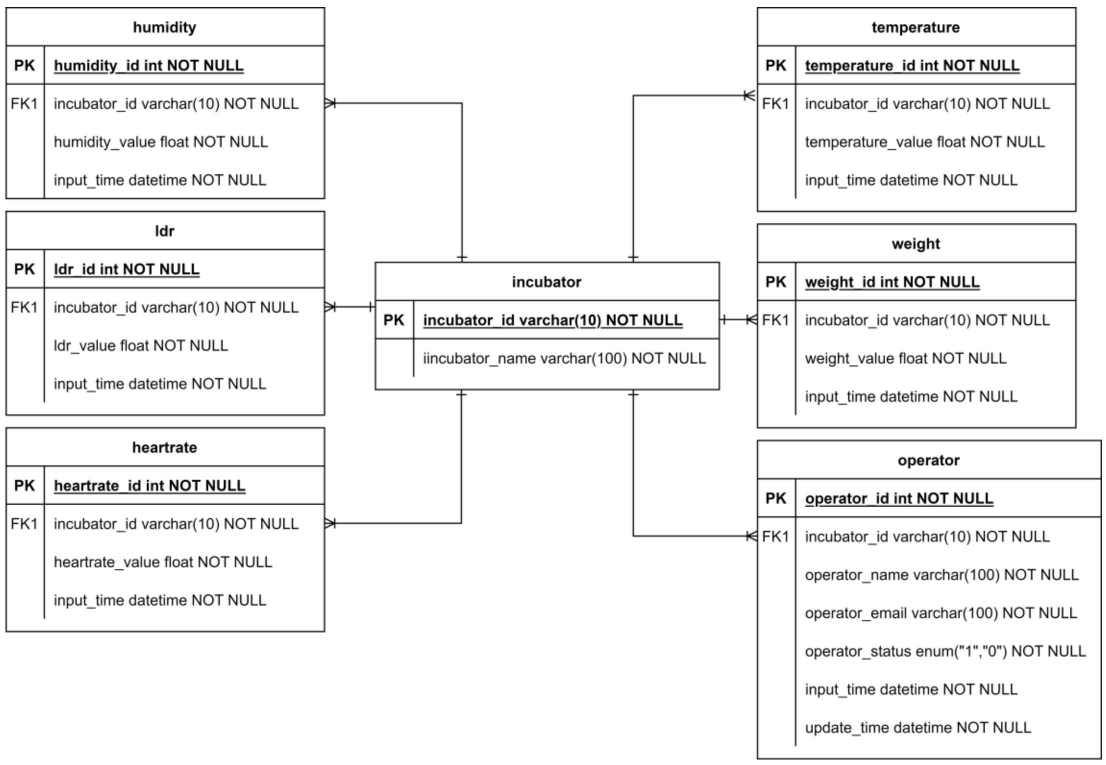

humidity
PK humidity_id int NOT NULL
FK1 incubator_id varchar(10) NOT NULL
humidity_value float NOT NULL
input_time datetime NOT NULL
ldr
PK ldr_id int NOT NULL
FK1 incubator_id varchar(10) NOT NULL
ldr_value float NOT NULL
input_time datetime NOT NULL
heartrate
PK heartrate_id int NOT NULL
FK1 incubator_id varchar(10) NOT NULL
heartrate_value float NOT NULL
input_time datetime NOT NULL
incubator
PK incubator_id varchar(10) NOT NULL
iincubator_name varchar(100) NOT NULL
temperature
PK temperature_id int NOT NULL
FK1 incubator_id varchar(10) NOT NULL
temperature_value float NOT NULL
input_time datetime NOT NULL
weight
PK weight_id int NOT NULL
FK1 incubator_id varchar(10) NOT NULL
weight_value float NOT NULL
input_time datetime NOT NULL
operator
PK operator_id int NOT NULL
FK1 incubator_id varchar(10) NOT NULL
operator_name varchar(100) NOT NULL
operator_email varchar(100) NOT NULL
operator_status enum("1","0") NOT NULL
input_time datetime NOT NULL
update_time datetime NOT NULL

Figure 8. The data entity relationship diagram

Table 3. Table and column in the database

Table name	Colum name
incubator	{incubator_id, incubator_name}
temperature	{temperature_id, incubator_id, temperature_value, input_time}
humidity	{humidity_id, incubator_id, humidity_value, input_time}
weight	{weight_id, incubator_id, weight_value, input_time}
ldr	{ldr_id, incubator_id, ldr_value, input_time}
heartrate	{heartrate_id, incubator_id, heartrate_value, input_time}
operator	{operator_id, incubator_id, operator_name, operator_email, operator_status, input_time, update_time}

2.4.3 The temperature control system with fuzzy

The system works by the user temperature input as a setpoint value, and then the temperature sensor will detect the value in the incubator to get the latest value. The difference between the setpoint and the current temperature is called an *error*. This *error* value will be entered into the fuzzy system to be calculated. Furthermore, the results of this fuzzy calculation are used as the value of the motor speed on the blower to control the temperature airflow. And so on, this activity is carried out until the error has reached zero or is close to zero.

The characteristics of the temperature sensor used have a measurement range of 0 – 50ºC, in which the coldest conditions have a value of 0ºC and the hottest conditions have a value of 50ºC. To determine the difference in the input error value as follows,

Error (e)= setPoint - ΔT
Negative error (-e) = condition is overheated
setPoint < temperature (2)
Positive error (+e) = condition is cold
setPoint > temperature

This study covers all possible results of calculating the *error* value from the range of -50 to 50 according to the characteristics of the sensor. Then the development of this *error* value is mapped to -10 to 10 with the program syntax. So, the input error variable is made from -10 to +10 with the number of fuzzy function memberships, namely five (*VL, L, M, H, VH*). A value set like this ensures that no system error values are outside this range. Furthermore, the membership for the input error speed value is formulated as follows,

$$e_{speed} = (e_{previous} - e) / \Delta_t \quad (2)$$

The error rate is the difference between the previous error minus the current error divided by the time it took to find the two errors. In this study, 1 second (Δt=1) was used. Thus, regarding the fuzzy membership function, the value between the error and the error speed can be considered the same. Then the output has a value range of 0 to 100. The result of this value will be used as input to adjust the motor speed (PWM). The following is a fuzzy output function with a membership number of five (VL, L, M, H, VH). After completing the membership fuzzy input and output program syntax on the microcontroller, the next step is to make the rules. Figure 9 shows the input Figures (9a) and (9b) and output Figure (9c) of the membership function for the fuzzy design and the fuzzy rules in Table 4.

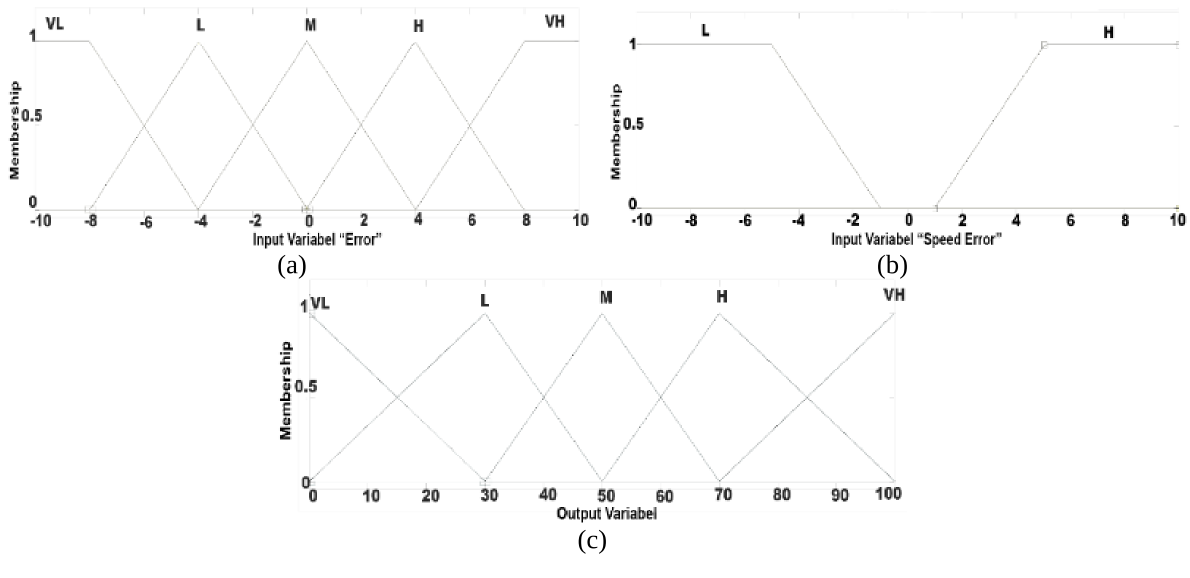


Figure 9. The fuzzy membership, (a) Input error, (b) Input speed error, and (c) Output

Table 4. The fuzzy rules

Fuzzy Rules	
1	IF(error is VL) and (error_speed is L) then (output_speed is VL)
2	IF(error is L) and (error_speed is L) then (output_speed is L)
3	IF(error is M) then (output_speed is M)
4	IF(error is H) and (error_speed is H) then (output_speed is H)
5	IF(error is VH) and (error_speed is H) then (output_speed is VH)

3. RESULTS AND DISCUSSION

The implementation process for Neobots combines several technology platforms, such as platforms for hardware (microcontrollers, sensors, actuators, drivers) and platforms for software on web applications, servers, and IoT Broker. This documentation is prepared with the hope of providing clear and comprehensive understanding to all parties involved in the Neobots project [32]. In general, the summary of the implementation of an open-source platform for incubators is as follows in Tabel 5.

3.1. The software implementation

Figure 10 shows the work process of the software application. The implementation process of Neobots can be like a client-server architecture that can communicate directly between objects without human intervention through a network with the MQTT protocol with a server called IoT Broker. The developed system can manage multiple incubators connected to IoT Brokers, such as the client-server concept in this context. Each incubator must have a unique address or identity to communicate with the IoT Broker system. Next, the sensor results obtained from each incubator are sent to the IoT Broker based on the topic and uniqueness of the incubator. After that, the data in the IoT Broker is processed to be sent to the database using a service program that works in a loop made in the Python programming language.

Table 5. The implementation of the Neobots IoT platform technology

IoT Architecture	Function	Device	Description
Hardware	Controllers	Microcontroller	Board Arduino Mega 2560 R3
	Computer	Microcomputer	Board Raspberry Pi
	Sensor	Sensor temperature	DHT11
		Sensor humidity	DHT11
		Sensor heart rate	SEN-11574
		Sensor weight	HX711
		Sensor light dependet resistor	LDR
	Power	Power supply	5V 2A
	Actuator	Heating	Incubator heating
		Lighting	LED
		Realtime clock module	RTC DS3231 AT24C32
		Wi-Fi module	ESP8266
Software	Server	Operating systems	Debian 11 Bullseye
	Control system	Library	Fuzzy algorithm
	Broker IoT	Framework	Mosquitto
	Data storage	Service	MySQL
	User inferface	Web application	Javascript, python, PHP, CSS, DataTables, jQuery, Bootstrap, Chartjs, Web browser, Eclipse Paho
	Notification	Framework	Email
	API	Framework	HTTP-POST, JSON
Network	MQTT	Library communication	Protokol MQTT
	Network interface	Network support	Wi-Fi

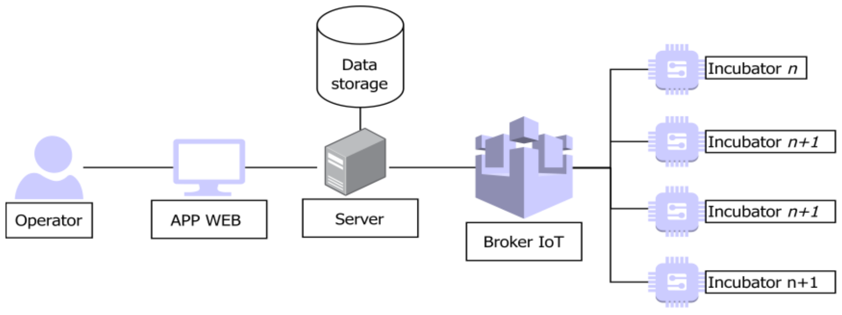
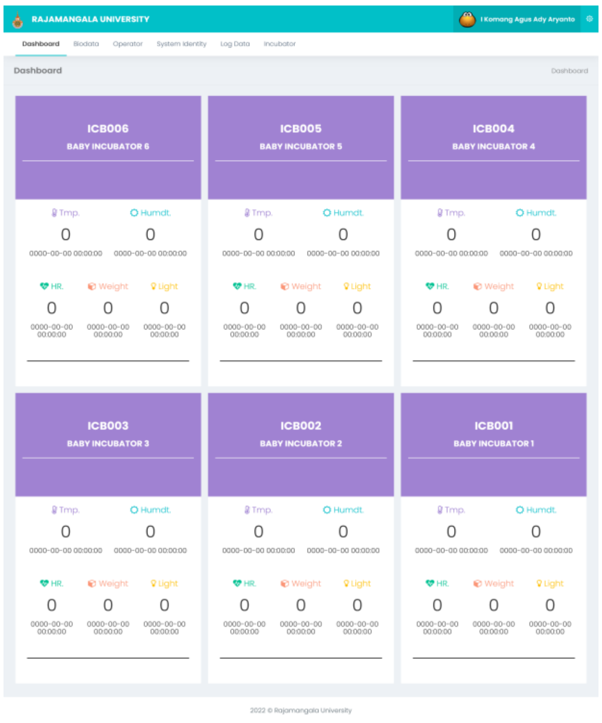


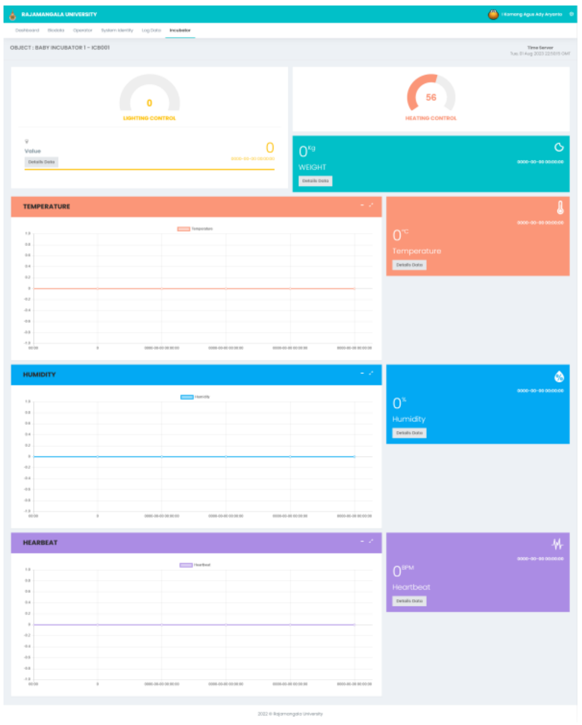
Figure 10. The implementation of software work processes

The data stored in the database is then accessed through a web application using a web service installed on the web server. The web server used is Apache which is installed on the microcomputer board. Then the information displayed on the web page is based on the identity of each incubator registered in the system. In addition, users who will see data from each incubator must also have access rights to the incubator. Meanwhile, if the user does not have access rights, the user cannot see any information from the incubator. Features so that users can view or manage multiple incubators in one account are also programmed in the system. Vice versa, one incubator can also be controlled by many users. To support ease of management, the system is equipped with program features that can dynamically manage user and incubator data through web pages which users can only do with access rights at the super admin level. The features of the software platform are shown in Table 6.

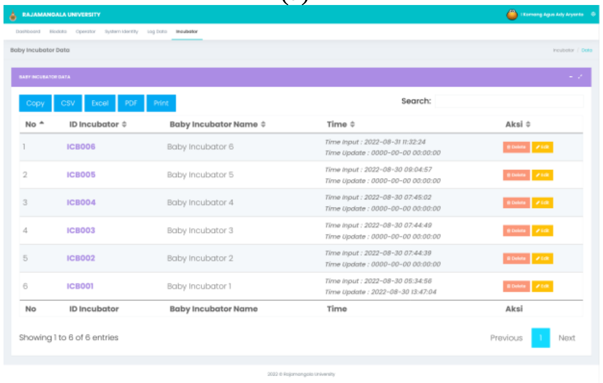
Figure 11(a) at Appendix shows the dashboard page on the website, which displays all of the incubators. Meanwhile, Figure 11(b) at Appendix displays the detail page for each incubator, which users can access if they select an incubator on the dashboard page. Figure 11(c) at Appendix is the master page used to manage incubator data in the system. Finally, Figure 11(d) at Appendix shows the page used to manage user access, in this case medical staff, doctors and parents, so they can monitor or control each incubator in the system. The implementation of a web-based application to make it easier for users to monitor and control IoT devices in baby incubators. To access this application, the user must have an account or access rights to log in so that the system can be more secure and supervised by every user who will access it. This login process has two input data as parameters for validating existing accounts in the database. The input data are username and password; password data is authenticated using MD5 encryption so that the character length becomes 32, producing a 128-bit ciphertext [33], [34]. If the data is successfully verified during the login process, the system will grant access rights to the main page (Dashboard), and if it fails, the system will not grant access rights.



(a)



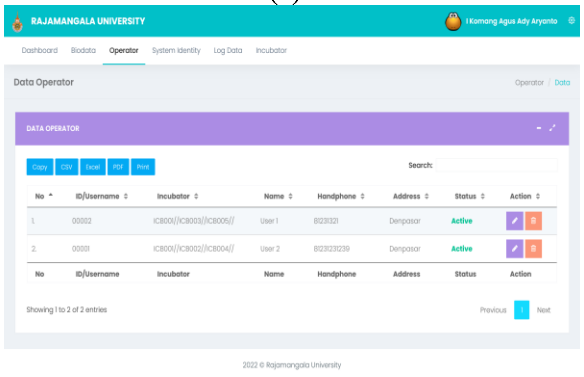
(b)



No	ID Incubator	Baby Incubator Name	Time	Aksi
1	ICB006	Baby Incubator 6	Time Input : 2022-08-31 11:32:24 Time Update : 0000-00-00 00:00:00	Delete Edit
2	ICB005	Baby Incubator 5	Time Input : 2022-08-30 09:04:57 Time Update : 0000-00-00 00:00:00	Delete Edit
3	ICB004	Baby Incubator 4	Time Input : 2022-08-30 07:45:02 Time Update : 0000-00-00 00:00:00	Delete Edit
4	ICB003	Baby Incubator 3	Time Input : 2022-08-30 07:44:49 Time Update : 0000-00-00 00:00:00	Delete Edit
5	ICB002	Baby Incubator 2	Time Input : 2022-08-30 07:44:39 Time Update : 0000-00-00 00:00:00	Delete Edit
6	ICB001	Baby Incubator 1	Time Input : 2022-08-30 05:34:56 Time Update : 2022-08-30 13:47:04	Delete Edit
No	ID Incubator	Baby Incubator Name	Time	Aksi

Showing 1 to 6 of 6 entries

(c)



No	ID/Username	Incubator	Name	Handphone	Address	Status	Action
1.	00002	ICB001//ICB003//ICB005//	User 1	81231321	Denpasar	Active	
2.	00001	ICB001//ICB002//ICB004//	User 2	81231231239	Denpasar	Active	
No	ID/Username	Incubator	Name	Handphone	Address	Status	Action

Showing 1 to 2 of 2 entries

(d)

Figure 11. The implementation web (a) displays all incubator data on the dashboard page, (b) incubator information detail page, (c) incubator management page, and (d) operator management page

Table 6. Web app features

No	Web app features
1	Login process
2	Displays temperature sensor values in the form of graphs and numbers as well as historical values of temperature data and can be exported to CSV format
3	Displays humidity sensor values in the form of graphs and numbers as well as historical values of humidity data and can be exported to CSV format
4	Displays heart rate sensor values in the form of graphs and numbers as well as historical values of heart rate data and can be exported to CSV format
5	Displays light sensor values in the form of graphs and numbers as well as historical values of light data and be exported to CSV format
6	Displays weight sensor values in the form of graphs and numbers as well as historical values of light data and can be exported to CSV format
7	Input setpoint value
8	Manage incubator data
9	Manage operator data
10	Manage IoT Broker data

Then the dashboard page displays all the measurement values of each sensor. This page was created using AJAX which is a web development technique that is used so that web applications work asynchronously (indirectly), which means that data sent and received on the server can be shortened to simplify the process for user experience. That way AJAX will continue to send requests to the server to request new data. After that, the server responds with new data and then updates the existing data on the web page.

The information displayed on the application is in the form of incubator identity and the value of each object (sensor) in the incubator. The value of this sensor device is obtained from a tool that has embedded the firmware program. This value is then sent to the IoT Broker to be processed according to the client's (operator) request. The value request process between the client and the IoT Broker uses a specially designed websocket app based on the topic id of each object (incubator). Or it can be said that this websocket is a liaison between IoT Broker and clients who carry out the monitoring and controlling process.

3.2. The hardware and network implementation

The implementation of this hardware platform uses sensor modules, actuators, microcontrollers, and microcomputers shown in Figure 12(a) shows the incubator body, Figure 12(b) shows the heater, Figure 12(c) shows the sensor, microcontroller, and microcomputer modules, while Figure 12(d) shows the main parts of the incubator, and the cost of each module is shown in Table 7. Price information in Table 7 is obtained based on one of the e-commerce sites on the internet, which was accessed in February 2023. The integration of hardware components into a single unit can be used to perform monitoring and control in an incubator through an application.

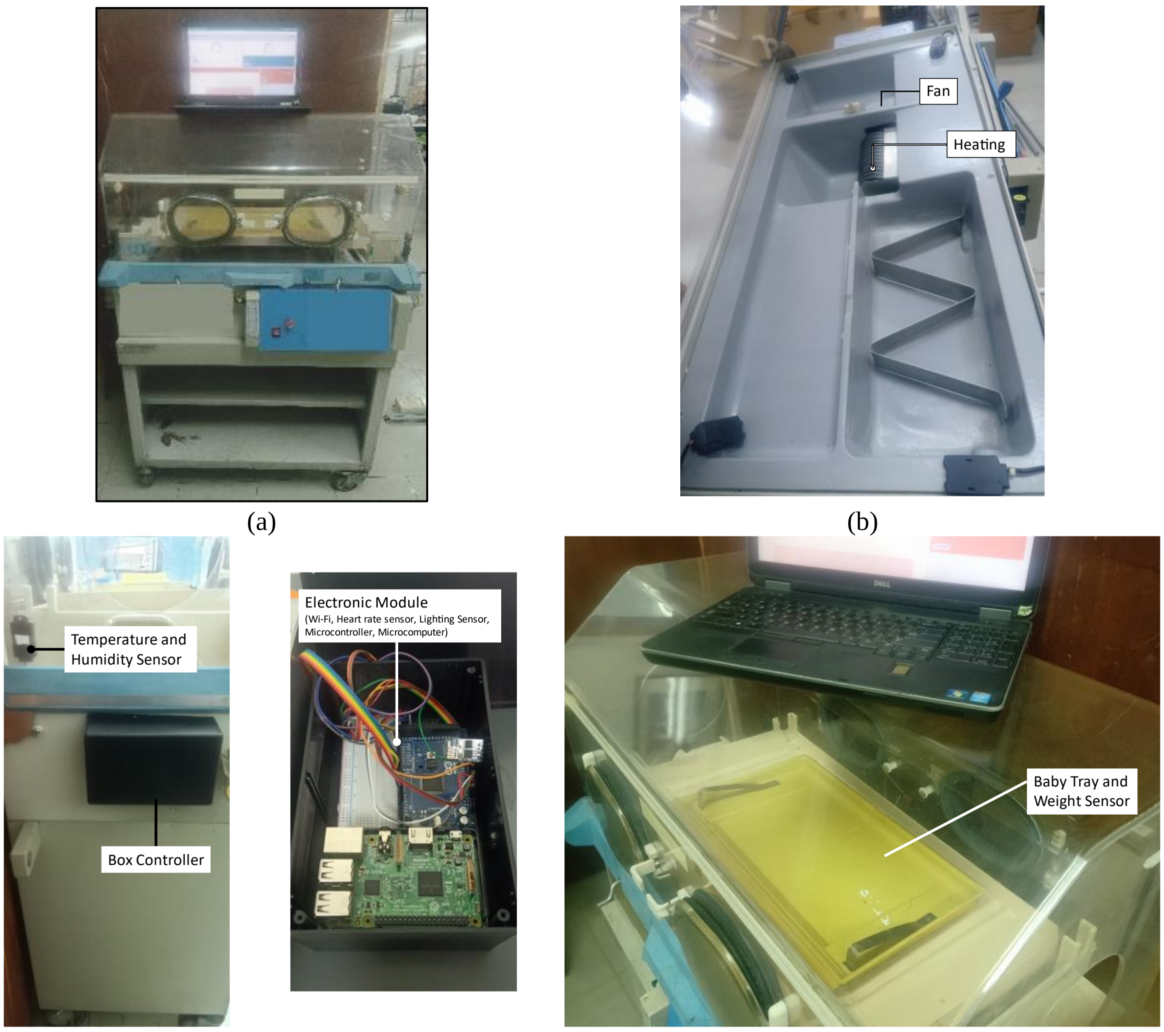
The sensor module device has several variants connected to the microcontroller via I/O pins according to the analog or digital signal. Each sensor module has a unique identity for communication with other module devices. In general, the workflow of each module is the same, namely to read sensor devices and be able to configure networks to get communication access rights so that they can send data to IoT Broker.

The IoT Broker application is installed on the Raspberry Pi microcomputer, and Raspberry is a minicomputer board with a Linux-based operating system. This study uses Raspberry 4, a processor Broadcom BCM2711B0, a clock speed of 1.5 GHz, and 4 GB memory. Then the power supply needed for electronic devices such as sensors, microcontrollers, and microcomputers is 5V 2A, which means that these devices require little electrical energy at a low cost.

Then the communication network uses a wireless network with the IEEE802.11 standard. The sensor modules will automatically connect to the Access Point according to the configuration parameters predefined by the user. The configuration performed on the Wi-Fi module only requires two main configurations, namely the service set identifier (SSID) and the password, making it easier to implement.

3.3. The experimental results

The testing process is the last stage in this research. This test aims to find errors from the combined hardware, software, and network, which is then expected to find test data that can be used to correct errors more thoroughly and quickly. The testing scheme is based on the modules that have been developed, such as temperature sensor modules, light sensors, sensors, and weight and heart rate sensors. Each result obtained from this testing activity is recorded, and then an evaluation is carried out on each module in hardware and software.


Fan
Heating
Temperature and Humidity Sensor
Box Controller
Electronic Module
(Wi-Fi, Heart rate sensor, Lighting Sensor, Microcontroller, Microcomputer)
Baby Tray and Weight Sensor

(a) (b)

(c) (d)

Figure 12. The hardware implementation, (a) Incubator, (b) Actuator heating, (c) Electronic device, and (d) Incubator main body

Table 7. The Neobots electronic cost

Product/Module	Qty	Price ($)	Final price ($)
Board Microcontroller	1	20	20
DHT11	4	2.5	10
ESP8266	1	3	3
RTC DS3231 AT24C32	1	3.6	3.6
Light Dependent Resistors	1	0.45	0.45
HX711 Weight	1	9	9
SEN-11574 Pulse rate	1	8	8
Power Suplay 5V 2A	1	7	7
Total			61.05

Then the test results are shown in Tables 8 and 9. The test in Table 8 is a test to determine whether the sensor module can work adequately in measuring the state of the baby's incubator and the suitability of the data obtained from the sensor. In addition, this testing process also ensures that the measurement results are transferred or sent to the IoT Broker with the MQTT protocol and stored in the database whether they are appropriate. Furthermore, Table 9 shows the results of testing web application modules whether they can work as expected. In the testing process, it is necessary to pay attention to the use of a web browser in accessing the web application because the system developed uses the latest Javascript technology, so it requires a web browser engine that supports Javascript.

Table 8. Testing sensor value

Temperature (ºC)	Humidity (% (RH))	Light (Lux)	Weight (Kg)	Heart Rate (bpm)
33.5	57	257	1.5	71
33.8	58	314	1.6	75
33.7	57	293	1.7	74
33.8	58	295	1.6	78
33.8	57	312	1.4	79
33.4	57	315	1.5	80
33.2	57	327	1.8	72
33.2	57	301	1.9	75
33.5	57	304	2.0	78
33.7	58	267	1.8	78

Table 9. Testing app web and device electronic

Test scenario	Expected target	Result
The sensor module device, when activated, automatically connects to Wi-Fi.	The sensor module automatically connects with Wi-Fi	Valid
The module device performs value measurements, and the results are automatically sent to the IoT Broker	The sensor module takes measurements and is visible in the data log on the IoT Broker	Valid
The value on the IoT Broker is processed automatically by the Socket to be stored in the database	The Socket application can process the values on the IoT Broker to be stored in the database	Valid
The web page displays the measured values automatically in the form of graphs, numbers, and tables with a responsive web design	Web pages can be accessed, and measurement value results are displayed on web pages automatically with responsive web design.	Valid

Fuzzy logic testing also compares the output values obtained in the MATLAB application with the program syntax created on the microcontroller. In this test, the value obtained from the sensor is processed with a setpoint to get an error value and an error speed value. Then these two values are used as a reference in the input values for the fuzzy logic program on the microcontroller and the input values on MATLAB. The output results obtained on the microcontroller can be seen on the serial monitor and web page, while the output results on MATLAB can be seen directly on the rule interface MATLAB page. The data from this test can be seen in Table 10. The input value entered in the microcontroller program must be the same as the input value entered in the MATLAB application, and the output results obtained from the microcontroller and MATLAB must have the same accuracy.

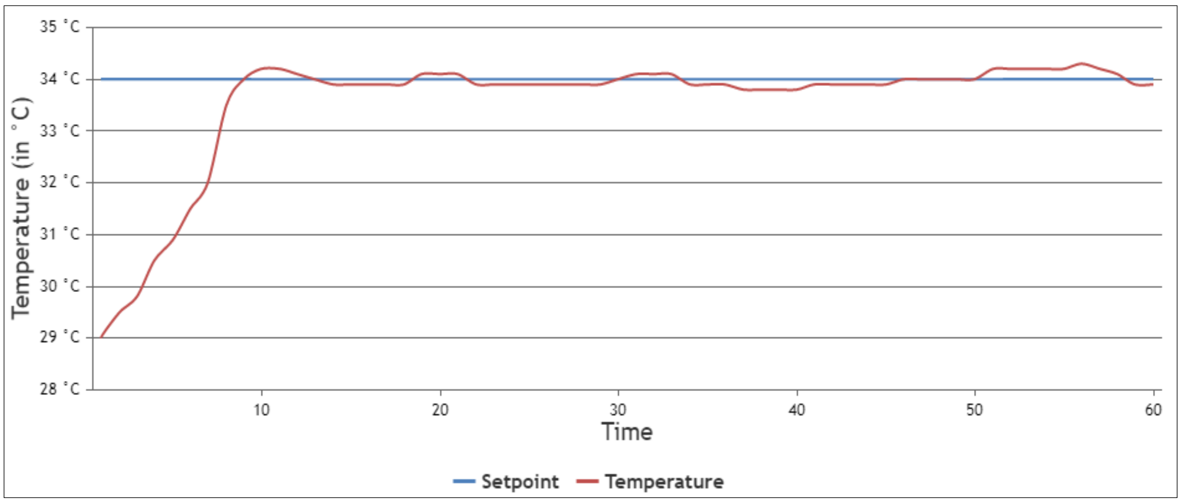
Table 10. Fuzzy testing results

Input Error Temperature	Input Speed Error	Output Neobots	Output MATLAB	Error (%)
1	1	50	50	0
2	3	74.1	63.8	13.90
4	4	75.0	73.5	2.00
6	7	70.4	77.7	10.37
10	10	70.4	90.3	28.27
-2	-2	42.4	39.7	6.37
-4	-4	30.1	26.5	11.96
-6	-6	22.7	22.3	1.76
-8	-8	10.0	9.67	3.30
-10	-10	10.0	9.67	3.30

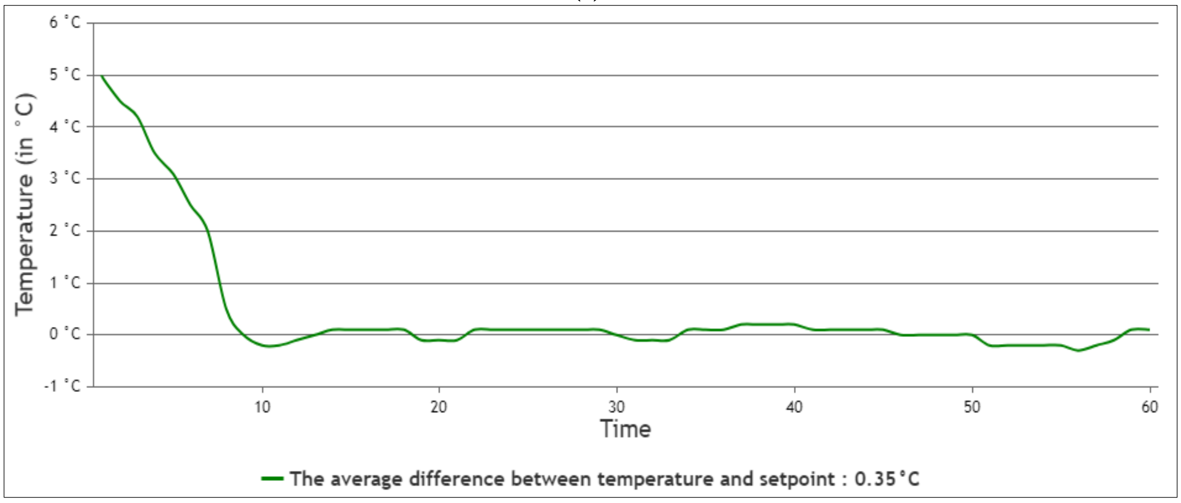
Table 10 shows that the output results between the comparison of the Neobots and MATLAB get an error rate of 0 - 28.27 %. This means that programs on the Neobots platform get results close to testing on applications on MATLAB. The error value is obtained from the formula as follows:

$$error = (Out_{mikrokontroller} - Out_{MATLAB}) / Out_{mikrokontroller} * 100 \quad (3)$$

Next is the fuzzy logic test to see the system response by entering the set point value. The setpoint value entered is 34ºC, and the actual temperature is 29ºC. Then the system responds to carry out the control process, and a temperature change occurs to the setpoint value for 10 minutes and the average error value with setpoint is 0.35 ^{O}C during 1 hour, as shown in Figure 13(a) displays a comparison between the setpoint values and the actual values as seen on the web graph, while Figure 13(b) presents the error values. The following results explain the error values at each sensor point compared to the central sensor value. Figure 14(a) shows the temperature error value (T_1), Figure 14(b) shows the temperature error value (T_2), Figure 14(c) displays the temperature error value (T_3), Figure 14(d) displays the temperature error value (T_4), Figure 14(e) displays the temperature error value (T_5), and Figure 14(f) displays the temperature error value (T_6). Then the results of the temperature difference at each point are less than $0.7^{O}C$ with the temperature at the midpoint (T_7).



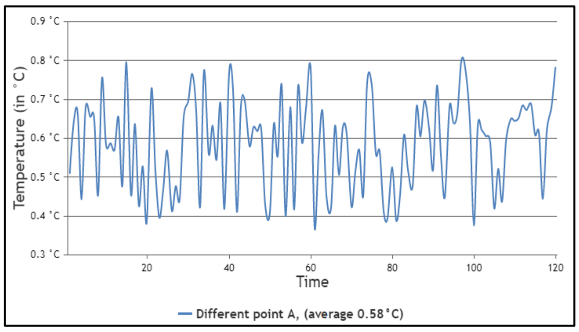
(a)



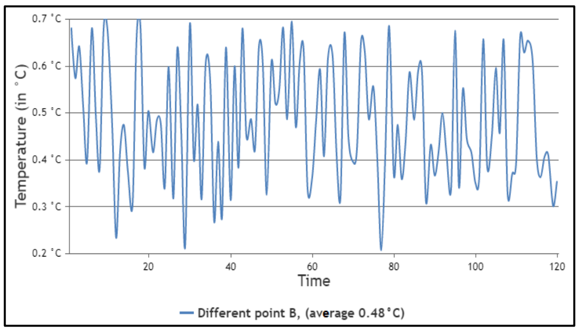
(b)

Figure 13. System response, (a) Temperature with setpoint and (b) Error temperature with setpoint

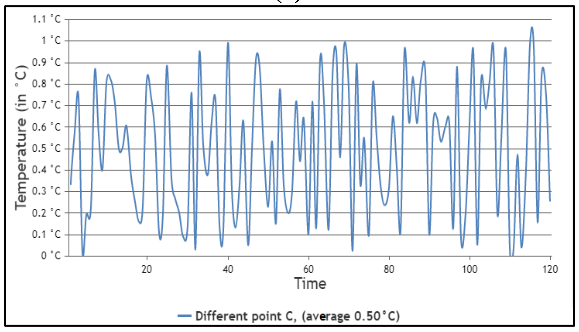
Test scenarios for the data transfer process with the MQTT protocol are carried out to see if the data sent to the incubator can be received in the database. The testing process is carried out for 1 hour, and data is sent every 1 second, so the total data must be sent to 3600 for each sensor. In addition, the rate time delay is also seen from the data transmitted by the incubator until it is received and stored in the database. This time lag is obtained by calculating the time from sending data from the incubator to the time received by the system. Furthermore, the lost data is also calculated during the transfer process with the MQTT protocol by comparing the amount of data received in the database with the amount of data that should be received for 1 hour if the data transfer time is set every 1 second (3600 data/hour). Data transfer testing from the incubator to the database using the MQTT protocol with the ESP8266 module Wi-Fi network was conducted in two different environmental conditions. The first condition is testing the indoors with the environmental conditions without noise disturbance. The second condition is carried out outdoors with environmental conditions full of noise. The following are the test results shown in Table 11 for indoor testing and Table 12 for outdoor testing.



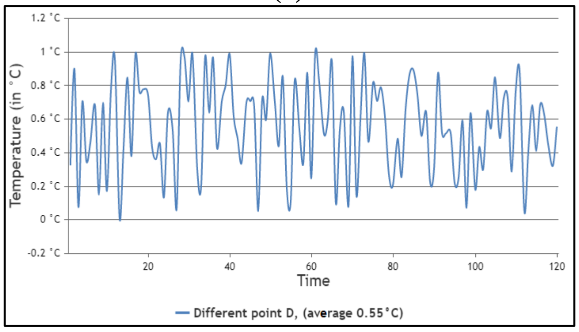
(a)



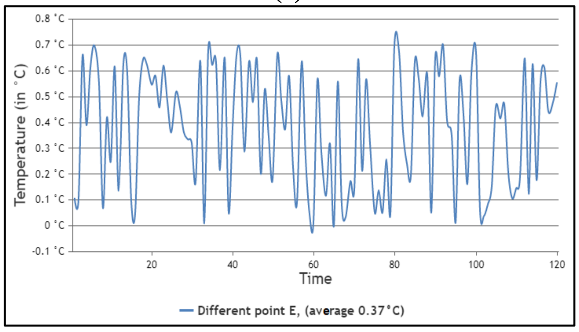
(b)



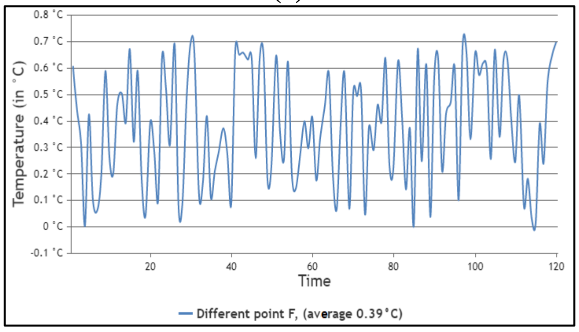
(c)



(d)



(e)



(f)

Figure 14. The difference in temperature error values at each point, (a) Point temperature (T1), (b) Point temperature (T2), (c) Point temperature (T3), (d) Point temperature (T4), (e) Point temperature (T5) and, (f) Point temperature (T6)

Table 11. Data transfer from the incubator to the database on indoors (without noise)

Object	Time Incubator	Time Database	Delay Rate (Second)	Data Lost (%)	Total Data (3600/hour)
Temperature	10:18:09 - 11:18:09	10:18:09 - 11:18:09	0 - 5	9.91	3243
Humidity	11:33:11 - 12:33:11	11:33:11 - 12:33:11	0 - 6	16.41	3009
Weight	15:52:16 - 16:52:16	15:52:16 - 16:52:16	0 - 5	13.58	3111
Heartrate	14:40:31 - 15:40:31	14:40:31 - 15:40:31	0 - 2	9.08	3273
LDR	13:23:25 - 14:23:25	13:23:25 - 14:23:25	0 - 3	9.27	3266

Table 12. Data transfer from the incubator to the database on outdoors (with a lot of noise)

Object	Time Incubator	Time Database	Delay Rate (Second)	Data Lost (%)	Total Data (3600/hour)
Temperature	14:16:17 - 15:16:17	14:16:17 - 15:16:17	0 - 6	37.11	2264
Humidity	14:10:15 - 15:10:15	14:10:15 - 15:10:15	0 - 7	37.20	2261
Weight	14:16:00 - 15:16:00	14:16:00 - 15:16:00	0 - 7	38.72	2206
Heartrate	14:17:00 - 15:17:00	14:17:00 - 15:17:00	0 - 7	37.97	2233
LDR	14:20:59 - 15:20:59	14:20:59 - 15:20:59	0 - 7	42.14	2083

Following the facts from the results of the tests carried out, it can be concluded that the results of data transfers carried out indoors with environmental conditions without noise and other disturbances get better results than the results of tests carried out the outdoors. This is because the Wi-Fi connection from the esp8266 can be more stable if there is no interference or noise. And based on the test results in Table 11, the average delay rate in the data transfer process from the incubator to the system for each sensor is 0 - 6 seconds, and the data lost during the data transfer process for each sensor is below 16.41%. While in Table 12, the average delay rate for each sensor is 0 - 7 seconds, and the data lost during the data transfer process for each sensor is below 42.14%. This data loss is caused because the Wi-Fi module device on the incubator cannot work stably, or the signal is often intermittent. Hence, the incubator stops sending data until the Wi-Fi is reconnected to the network.

The results of the evaluation comparing the Neobots IoT platform with existing research on baby incubators are presented in Table 13. This evaluation shows a comparison between the Neobots IoT platform and the techniques and methodologies from previous research on incubators. Through these results, the features of the Neobots IoT platform in the field of incubator technology become clearer.

Table 13. State of the art comparison between the existing research

Researcher Reference	Microcontroller	Microcomputer	Sensor	Communication Protocol	Database	Application Interface	Control System	Network	Programming
[35]	ESP32	Raspberry	Temperature	MQTT				Wi-Fi	Node-Red
[36]	ATMega 16		Temperature, Humidity			Desktop	Fuzzy-PID		BASIC, VB.NET
[37]	ATMega 328		Temperature, Humidity, Fingerprint, Lighting, Heart Rate, Camera	HTTP	MySQL	Mobile Android		GSM	C++, PHP, Java
[38]	ATMega 2560		Heart Rate, Weight			Web		Ethernet	C++
Neobots	ATMega 2560	Raspberry	Temperature, Humidity, Heart Rate, Weight, Lighting	MQTT & HTTP	MySQL	Web	Fuzzy	Wi-Fi	Python, PHP, JavaScript, CSS, C++

4. CONCLUSION

Based on the research and results of the tests that have been carried out, it is concluded that the hardware consisting of electronic sensors can work well to measure the environmental conditions of the incubator. Each sensor electronic device or object in the incubator is implanted with a unique identity as identity so that each of these objects can communicate with each other with other things. The communication process on this object uses the MQTT protocol with data transfer to the IoT Broker using the term publish/subscribe. The system also creates a Back-End Socket Program so that the data in the IoT Broker is automatically stored in the database. The structure of the database system is related to one another and is divided into seven tables to store data from each sensor or object. Data stored in the database can be displayed on web applications in real-time using the webscoket program. Apart from displaying data stored in the database, web applications can also display data directly from the IoT Broker according to the subscribed topics. This data is displayed on web pages in graphs, tables, and figures that are easy to understand and a responsive web display, which means that it can be accessed using various types of screen sizes such as mobile phones or personal computers. Furthermore, the conclusion from testing the temperature control system in an incubator with fuzzy logic shows that the system response can stably adjust the temperature according to the setpoint value inputted by the user through the web application. The temperature value inputted by the user is used as the setpoint value in the system; according to the test results, the initial incubator temperature value is 29^{o}C, and the setpoint value entered is 34^{o}C. In less than 10 minutes, the system can adjust the temperature conditions to the setpoint value and the average error value is 0.35^{O}C during 1 hour. In addition, the values obtained from each temperature sensor placed at six points in the incubator get an error of less than 0.7^{O}C of the temperature sensor value at the midpoint. Then the error comparison results from the output values obtained in the fuzzy logic programmed on the Neobots with simulations with the MATLAB application get an error rate of 0 - 28.27%. Furthermore, the conclusions for testing data transfer from the incubator to the database with the Wi-Fi module via the MQTT protocol get different results in tests carried out with indoor and outdoor environmental conditions. In testing the indoors with conditions without any noise disturbance, the value for lost data is less than 16.41%, and the delay rate is between 0 - 6 seconds. As for the outdoor test value with lots of noise disturbance, the results of lost data are 42.41%, and the delay rate is between 0 - 7 seconds with testing for 1 hour with data transfer every 1 second. These results show that the data transfer process with the Wi-Fi module can be more stable if the surrounding conditions in the incubator are not disturbed by noise.

ACKNOWLEDGEMENTS

Appreciation is given, and thank you to all parties who have provided support and prayers to complete this research, especially for the Rajamangala University of Technology Thanyaburi and the ITB STIKOM Bali.

REFERENCES

[1] B. J. Stoll *et al.*, "Trends in Care Practices, Morbidity, and Mortality of Extremely Preterm Neonates, 1993-2012," *JAMA*, vol. 314, no. 10, pp. 1039–1051, 2015, doi: 10.1001/jama.2015.10244.

[2] T. Restin, M. Gaspar, D. Bassler, V. Kurtcuoglu, F. Scholkmann, and F. B. Haslbeck, "Newborn Incubators Do Not Protect from High Noise Levels in the Neonatal Intensive Care Unit and Are Relevant Noise Sources by Themselves," *Children*, vol. 8, no. 8, 2021, doi: 10.3390/children8080704.

[3] T. Sanchez Lopez, D. C. Ranasinghe, M. Harrison, and D. McFarlane, "Adding sense to the Internet of Things," *Pers. Ubiquitous Comput.*, vol. 16, no. 3, pp. 291–308, 2012, doi: 10.1007/s00779-011-0399-8.

[4] W. Zhang, M. Kumar, J. Yu, and J. Yang, "Medical long-distance monitoring system based on internet of things," *EURASIP J. Wirel. Commun. Netw.*, vol. 2018, no. 1, p. 176, 2018, doi: 10.1186/s13638-018-1178-2.

[5] P. Thota and Y. Kim, "Implementation and Comparison of M2M Protocols for Internet of Things," in *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)*, 2016, pp. 43–48. doi: 10.1109/ACIT-CSII-BCD.2016.021.

[6] P. Rawat, K. D. Singh, and J. M. Bonnin, "Cognitive radio for M2M and Internet of Things: A survey," *Comput. Commun.*, vol. 94, pp. 1–29, 2016, doi: https://doi.org/10.1016/j.comcom.2016.07.012.

[7] H.-S. Kim, J.-H. Kang, J.-Y. Hwang, and U. S. Shin, "Wearable CNTs-based humidity sensors with high sensitivity and flexibility for real-time multiple respiratory monitoring," *Nano Converg.*, vol. 9, no. 1, p. 35, 2022, doi: 10.1186/s40580-022-00326-6.

[8] H. H. Alshammari, "The internet of things healthcare monitoring system based on MQTT protocol," *Alexandria Eng. J.*, vol. 69, pp. 275–287, 2023, doi: https://doi.org/10.1016/j.aej.2023.01.065.

[9] S. Iranpak, A. Shahbahrami, and H. Shakeri, "Remote patient monitoring and classifying using the internet of things platform combined with cloud computing," *J. Big Data*, vol. 8, no. 1, p. 120, 2021, doi: 10.1186/s40537-021-00507-w.

[10] N. S. Al-Blihed, N. F. Al-Mufadi, N. T. Al-Harbi, I. A. Al-Omari, and M. A. Al-Hagery, "Blockchain and machine learning in the internet of things: a review of smart healthcare," *IAES Int. J. Artif. Intell.*, vol. 12, no. 3, pp. 995–1006, 2023, doi: 10.11591/ijai.v12.i3.pp995-1006.

[11] A. Hussain, M. Hamad, K. Hafeez, and T. Zainab, "Programming a Microcontroller," *Int. J. Comput. Appl.*, vol. 155, no. 5, pp. 21–26, Dec. 2016, doi: 10.5120/ijca2016912310.

[12] A. Y. Al-Rawashdeh *et al.*, "Experimental Investigation of Microcontroller-Based Acoustic Temperature Transducer Systems," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020884.

[13] S. Azfar *et al.*, "An IoT-Based System for Efficient Detection of Cotton Pest," *Appl. Sci.*, vol. 13, no. 5, 2023, doi: 10.3390/app13052921.

[14] S.-G. Racz *et al.*, "Mobile Robots&mdash;AHP-Based Actuation Solution Selection and Comparison between Mecanum Wheel Drive and Differential Drive with Regard to Dynamic Loads," *Machines*, vol. 10, no. 10, 2022, doi: 10.3390/machines10100886.

[15] G. S. Sharath, N. Hiremath, and G. Manjunatha, "Design and analysis of gantry robot for pick and place mechanism with Arduino Mega 2560 microcontroller and processed using pythons," *Mater. Today Proc.*, vol. 45, pp. 377–384, 2021, doi: https://doi.org/10.1016/j.matpr.2020.11.965.

[16] D. Bishop and J. G. Chase, "Development of a Low-Cost Luminance Imaging Device with Minimal Equipment Calibration Procedures for Absolute and Relative Luminance," *Buildings*, vol. 13, no. 5, 2023, doi: 10.3390/buildings13051266.

[17] A. R. Mead and K. M. Mosalam, "Ubiquitous luminance sensing using the Raspberry Pi and Camera Module system," *Light. Res. \& Technol.*, vol. 49, no. 7, pp. 904–921, 2017, doi: 10.1177/1477153516649229.

[18] T. Turc, "AJAX Technology for Internet of Things," *Procedia Manuf.*, vol. 32, pp. 613–618, 2019, doi: https://doi.org/10.1016/j.promfg.2019.02.260.

[19] J. Wang, J. Zhou, R. Gu, M. Chen, and P. Li, "Manage system for internet of things of greenhouse based on GWT," *Inf. Process. Agric.*, vol. 5, no. 2, pp. 269–278, 2018, doi: https://doi.org/10.1016/j.inpa.2018.01.002.

[20] L. Verma, M. Fakharzadeh, and S. Choi, "Wifi on steroids: 802.11AC and 802.11AD," *IEEE Wirel. Commun.*, vol. 20, no. 6, pp. 30–35, 2013, doi: 10.1109/MWC.2013.6704471.

[21] C. Deng *et al.*, "IEEE 802.11be Wi-Fi 7: New Challenges and Opportunities," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 4, pp. 2136–2166, 2020, doi: 10.1109/COMST.2020.3012715.

[22] H. Pranoto, A. Marwanto, S. Alifah, and L. A. Fatah, "Valve control system on a venturi to control FiO2 a portable ventilator with fuzzy logic method based on microcontroller," *IAES Int. J. Artif. Intell.*, vol. 12, no. 4, pp. 1593–1602, 2023, doi: 10.11591/ijai.v12.i4.pp1593-1602.

[23] B. Clerckx, Z. Popović, and R. Murch, "Future Networks With Wireless Power Transfer and Energy Harvesting [Scanning the Issue]," *Proc. IEEE*, vol. 110, no. 1, pp. 3–7, 2022, doi: 10.1109/JPROC.2021.3133676.

[24] S. A. Omidi, M. J. A. Baig, and M. T. Iqbal, "Design and Implementation of Node-Red Based Open-Source SCADA Architecture for a Hybrid Power System," *Energies*, vol. 16, no. 5, 2023, doi: 10.3390/en16052092.

[25] IEC, *BSI standards publication medical electrical equipment part 2-19: Particular requirements for the basic safety and essential performance of infant incubators*. IEC Standard, 2011.

[26] R. Garg and B. D. Reddy, "IoT Smart Plug based on ESP8266 Wi-Fi Chip," in *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, 2022, pp. 550–555. doi: 10.1109/ICOSEC54921.2022.9952001.

[27] S. El Himer, M. Ouaissa, M. Ouaissa, M. Krichen, M. Alswailim, and M. Almutiq, "Energy Consumption Monitoring System

Based on IoT for Residential Rooftops," *Computation*, vol. 11, no. 4, 2023, doi: 10.3390/computation11040078.
[28] A. B. Paredes-Baños, A. Molina-Garcia, A. Mateo-Aroca, and J. J. López-Cascales, "Scalable and Multi-Channel Real-Time Low Cost Monitoring System for PEM Electrolyzers Based on IoT Applications," *Electronics*, vol. 13, no. 2, 2024, doi: 10.3390/electronics13020296.
[29] Z. Lima, H. García-Vázquez, R. Rodríguez, S. L. Khemchandani, F. Dualibe, and J. Del Pino, "A System for Controlling and Monitoring IoT Applications," *Appl. Syst. Innov.*, vol. 1, no. 3, 2018, doi: 10.3390/asi1030026.
[30] B. Jose and S. Abraham, "Performance analysis of NoSQL and relational databases with MongoDB and MySQL," *Mater. Today Proc.*, vol. 24, pp. 2036–2043, 2020, doi: https://doi.org/10.1016/j.matpr.2020.03.634.
[31] G. Ongo and G. P. Kusuma, "Hybrid Database System of MySQL and MongoDB in Web Application Development," in *2018 International Conference on Information Management and Technology (ICIMTech)*, 2018, pp. 256–260. doi: 10.1109/ICIMTech.2018.8528120.
[32] Komangady, "NEOBOTS," *Github*, 2023. https://github.com/komangady/NEOBOTS (accessed Feb. 15, 2024).
[33] D. Rachmawati, J. T. Tarigan, and A. B. C. Ginting, "A comparative study of Message Digest 5(MD5) and SHA256 algorithm," *J. Phys. Conf. Ser.*, vol. 978, no. 1, p. 12116, Mar. 2018, doi: 10.1088/1742-6596/978/1/012116.
[34] K. Shahbazi, M. Eshghi, and R. Faghih Mirzaee, "Design and implementation of an ASIP-based cryptography processor for AES, IDEA, and MD5," *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 4, pp. 1308–1317, 2017, doi: https://doi.org/10.1016/j.jestch.2017.07.002.
[35] I. Sukma *et al.*, "Real-time wireless temperature measurement system of infant incubator," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 1, pp. 1152–1160, 2023, doi: 10.11591/ijece.v13i1.pp1152-1160.
[36] A. Alimuddin, R. Arafiyah, I. Saraswati, R. Alfanz, P. Hasudungan, and T. Taufik, "Development and performance study of temperature and humidity regulator in baby incubator using fuzzy-pid hybrid controller," *Energies*, vol. 14, no. 20, 2021, doi: 10.3390/en14206505.
[37] P. T. Kapen, Y. Mohamadou, F. Momo, D. K. Jauspin, N. Kanmagne, and D. D. Jordan, "Development of a neonatal incubator with phototherapy, biometric fingerprint reader, remote monitoring, and heart rate control adapted for developing countries hospitals," *J. Neonatal Nurs.*, vol. 25, no. 6, pp. 298–303, 2019, doi: https://doi.org/10.1016/j.jnn.2019.07.011.
[38] M. Irmansyah, E. Madona, and A. Nasution, "Design and application of portable heart rate and weight measuring tool for premature baby with microcontroller base," *Int. J. GEOMATE*, vol. 17, no. 61, pp. 195–201, 2019, doi: 10.21660/2019.61.ICEE12.

BIOGRAPHIES OF AUTHORS

I Komang Agus Ady Aryanto Completed a computer systems undergraduate program at the ITB STIKOM Bali in 2015. After that, continued his master's degree in computer systems at the Ganesha University of Education and finished his education in 2018. I lecturer on the ITB STIKOM Bali since 2018 until now and teaching web programming, Internet of Things, sensors and transducers, microcontroller programming, data structures, and databases. The research topics occupied are the Internet of Things, information systems, computer science, robotics, and programming. In addition, he also works as a software developer to develop web-based and mobile applications that have been implemented in the fields of government, transportation, industry, academics, hospitals, and others. Contacts that can be via email: komang.aryanto@gmail.com and i_komang@mail.rmutt.ac.th.

Dechrit Maneetham is a lecturer in the Faculty of Engineering Education at Rajamangala University, Thailand for more than 15 years. Fields taught by lecturers are mechatronics, robotics, Internet of Things. Education in Computer Engineering with M.S and B.Tech degrees at the Raja Mongkut University of Technology in Thailand. D.Eng doctoral education in the field of mechatronics at the Asian Institute of Technology. Then the second Ph.D. doctoral education at Mahasarakham University in the field of Computer Engineering. Author of books on PLC, Microcontroller, Pneumatics, and Robotics topics. Research interest in the fields of mechatronics, automation, Internet of Things, and robotics and has published more than 30 international papers. For further information, please contact us via email at: dechrit_m@rmutt.ac.th.

Evi Triandini as a lecturer at the ITB STIKOM Bali in the information systems study program. Teach courses related to software engineering, object-oriented modeling, and information systems design. Undergraduate education was taken by the University of Brawijaya in Indonesia in the field of Agricultural Cultivation. Master's degree at Asia Institute of Technology Thailand with M.Eng. Doctoral education at the Institute of Technology Sepuluh November Surabaya Indonesia in the department of computer science with the title Dr. Published research has topics that include information systems, software, and electronic commerce. Further information can be called via email: evi@stikom-bali.ac.id.