INCUBABY (IB03) DESIGN OF SIMULATION

END CONTROLLER FOR A PREMATURE INFANT

INCUBATOR CONNECTED TO NODE-RED ON A

PC FOR INTERFACING THROUGH WEB DASHBOARD.

EASTERN MEDITERRANEAN UNIVERSITY

CMPE320 EMBEDDED SYSTEM DESIGN

TEAM-5 FIRST REPORT Revision 0

Date of Submission: **11.06.2022**

TEAM LEADER:

5.1 Doğukan Dağtekin, 19330998

TEAM MEMBERS:

5.2 Saifuldeen Ali Majeed, 18701388

5.3 Thokozani Elliot Mupundu, 17700455

5.4 Fatima Sada, 19701653

5.5 Uğur Gençer, 19000044

5.6 Ekemadu Orisa Ekpette, 19700662

Ethical Declaration:

“We are aware of the Principles of Research Ethics as should be obeyed and we declare

that this report which we submit to the instructor will be the result of our own

independent work and that in all cases, material from the work of others will be fully

cited and referenced as required by the academic rules and ethical conduct. We

understand that if any kind of plagiarism is detected in our written work, the instructor

will set our whole homework-project grade to zero, and take the case to the ‘Disciplinary

Committee’ for necessary action.”

### 

# TABLE OF CONTENTS

### 

**[DESIGN OBJECTIVES](#_shsyeib2cfm5) 3**

[Infant Incubator Unit Structure](#_tpwlpmsbkpfa) 4

[Cyber-physical modeling requirements:](#_sim4riryo4zd) 4

[Low-end embedded control unit requirements:](#_vp3fxo7zovhm) 5

[High-end embedded PC unit:](#_bcfrycfc4ltj) 5

**[MODELING AND ANALYSIS FOR CONCEPTUAL DESIGN.](#_56czowvgpfo6) 6**

[Simulation of Physical part (Design)](#_9ujfdjzegudg) 8

[Simulation of Cyber physical system](#_cs2eg49ckhzl) 8

**[DigiTerm Design, Modifications, and Tests](#_r7v4hpjrfn9d) 11**

[Modifications](#_8sqex5z2op35) 11

[Implementations and Testings](#_89fb6nzeyo1v) 12

[LCD Display](#_w4ivs369772i) 16

**[Node-Red Dashboard Design and testing](#_f0pgbu984sb1) 19**

**[Installation of Node-Red](#_87glbsk5t561) 19**

**[Setup VSPE](#_ug8hk7q6hg6c) 19**

**[PUTTY](#_jq1yqeltjjsz) 21**

**[Running Node-Red](#_995o38bxkvb5) 21**

**[Conclusions](#_8s69jminil1e) 24**

# DESIGN OBJECTIVES

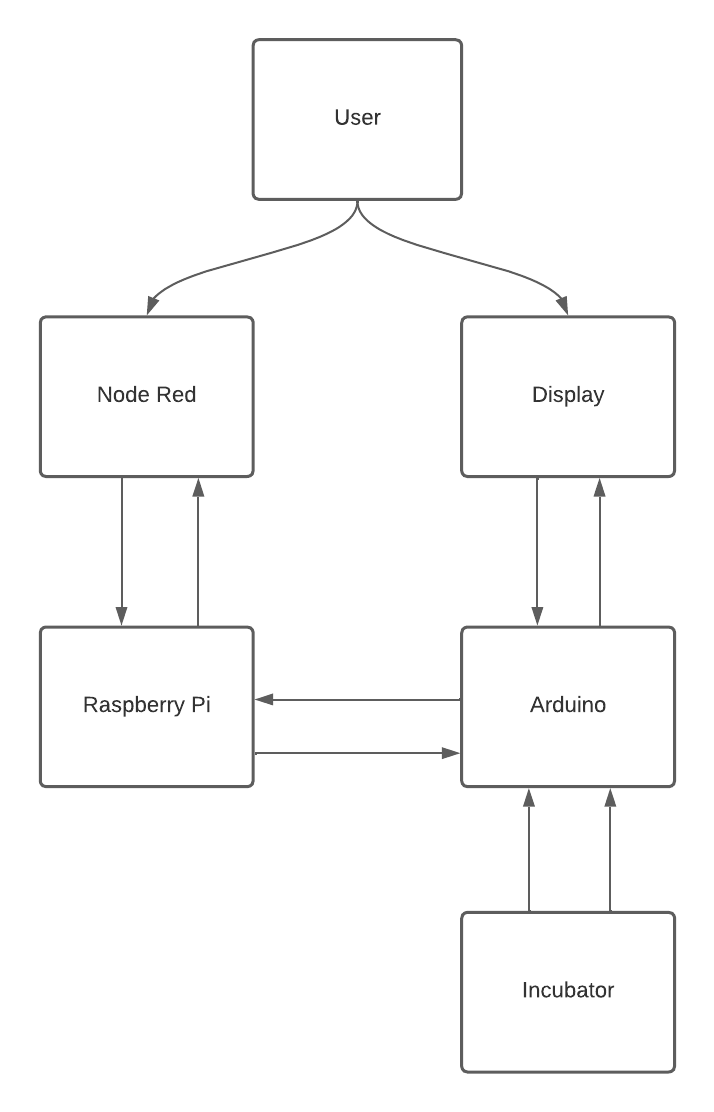
(i53 d52 t51-56 v54 e55)

The project's goal is to create an Arduino-Uno simulation in Proteus that measures the voltage of two NTC thermistors, ThA for heater air temperature and ThB for baby skin temperature, and computes the temperature in Celsius centigrade. If the heater air temperature exceeds 37 degrees Celsius, the system will sound an alarm until the baby's skin temperature falls below 35. A 2-line by 16-char LCD display will show the current temperature, the desired temperature, and more detailed components. Temperature will be controlled using proportional control action by PWM switching the heater output.

The serial connection is made by connecting COM10 on the Proteus to COM11 on the node-red with the VSPE software. The dashboard components were used with node-red to display the temperatures graphically and the power percentage on a gauge. When reading the values from the proteus setup, a split function is used.

# OVERALL SYSTEM

The devices are intended to serve as a low-cost sensor data provider for a higher-end device. The sensor is connected to the Node-Red on PC dashboard through an Arduino UNO, a 2x16 LCD display, a timekeeping counter to display how long the incubator has been operating, and a comparator that receives and displays data.



# 

# INTRODUCTION

(i51 d53 t54-55 v52 e56)

In this embedded system, the measured ambient temperature is transmitted to the PC via a series USB converter at 9600 baud and displayed on an LCD module. Our company is utilizing a Digiterm device in this project. Controlling baby body temperature (ThB) by powering an air heater at one of three predetermined desired ThBs temperatures, ThD: 30, 32, 34 C. If the infant's temperature exceeds 37 degrees, the system will issue a warning.

The most significant benefit of this project is the protection of human health and the prevention of potential health problems. The temperatures of the babies and the environment will be examined and recorded by this embedded system. The system will measure and calculate the babies' and the ambient temperatures on a regular basis, record them, and issue a warning if necessary. It will, for example, prevent health problems by alerting the parent if the baby's body temperature rises above 37 degrees. When necessary, data for up to 30 days can be displayed. It benefits people in terms of both health and safety in this way.

We display the graphical representation of the data values using node-red. As a result, we can monitor and adjust the temperatures when the alarm goes off.

## Social Impacts

(i51 d53 t54-55 v52 e56)

The key goal of this process is that it will protect the health and safety of babies by using an embedded system to monitor the temperatures that keep them warm. It will be quite useful in a wide range of temperatures and weather situations. The community will be safer if illnesses are prevented. Using such dependable technologies does not raise any ethical issues. Automation in infant incubators, for example, ensures that the incubator maintains a precise temperature for a set amount of time. Our safety will improve when we have healthy babies, and no newborns will be sick. We shall not have to move about manually because of this embedded system with cyber and physical components that will do it for us. Furthermore, the equipment used will not consume a lot of resources.

# DESIGN RESOURCES AND REQUIREMENTS

## Infant Incubator Unit Structure

The physical infant incubator system consists of two major components:

**-**a transparent outer air jacket with a heater-fan system to heat the air inside the jacket. The air blown by a fan is heated by a simple 24V - 50 W heater.

**-**The transparent double-wall cover that protects the heat in the incubator's inner chamber.

A controller and power inhibit logic at excess ThA provide temperature control for desired-ThB to prevent harmful temperature rise.

**-**The time constant in the incubation chamber (including the infant) is approximately 30 minutes. Depending on the infant's weight, it could be twice as high.

## Cyber-physical modeling requirements:

* Convert a Digiterm device into an infant incubator that regulates the infant body temperature (ThB) by activating an air heater at one of three predetermined desired-ThB temperatures, ThD: 32, 34, 36. Temperatures ThA > 37 and ThB > 37 are known to be hazardous.
* Overall body temperature modeling in an incubator with a temperature control to verify the control law can keep body temperature ThB at ThD.
* The temperature dynamics of the air-jacket and the incubation chamber, as measured by ThA and ThB sensor signals.
* Control over the air-percent heater's extent in the air-jacket.
* The desired ThB signal.
* When ThB>37, the ThB sensor's action is inhibited.

## Low-end embedded control unit requirements:

* The thermistor circuit must be modified to measure temperatures up to 50 degrees Celsius.
* A second thermistor circuit must be added to the design in order to read both ThA and ThB temperatures.
* Control action necessitates a proportional gain Kp of 100.
* Display the values on the 16x2 LCD display in an easily recognizable format.

## High-end embedded PC unit:

* NodeRed canvass must perform the following functions:

It must be connected to the serial port of the simulated low-end system.

It must run NodeRed with a dashboard that includes the following items.

Slide the slider to the desired temperature of 32, 34, or 36. When the slide settings are changed, action should be sent to the low-end.

Textual alarm status indicators. The alarm status received from the low end must be displayed on the dashboard in plain English.

* Temperature charts are used to record ThA, ThB, and ThD.

# MODELING AND ANALYSIS FOR CONCEPTUAL DESIGN.

(i-51 d-56 t-56 w-51)

## 1.1. Test A: Simulation of Physical part

A critical component of embedded system design is complete simulation of the designed cyber physical system. We will need a good Scilab XCOS model to model the physical environment and possible system performance in order to build a good cyber physical system. The following are some test and simulation examples.

➔ We used a Gainblock for effective temperature of heater:

* Gain:30
* Do on overflow:0

➔ We used a Step Function for the environment temperature and set its values:

* Step Time: 0
* Initial Value: 0
* Final Value: 70

➔ We added both temperatures with a summation block:

* Datatype: 1
* No. of inputs: [1;1]
* Do on Overflow: 0 for nothing

➔ We used a CLR block to set the air heater:

* Numerator: 1
* Denominator: 1 + 2\*s

➔ We used a pulse\_sc block to generate a signal - 10C to simulate cover opening:

* Phase Delay(s):200
* Pulse Width:5
* Period(s):400
* Amplitude:-10

➔ We added CLR for air heater and PULSE\_SC block with a summation block:

* Datatype: 1
* No. of inputs: [1;1]
* Do on Overflow: 0 for nothing

➔ We used another CLR block to set the air heater chamber:

* Numerator: 1
* Denominator: 1 + 30\*s

➔ We changed the project settings to set the simulation main used parameters:

* Final Integration time: 400 (4E02)

➔We used MUX for ThA-ThB

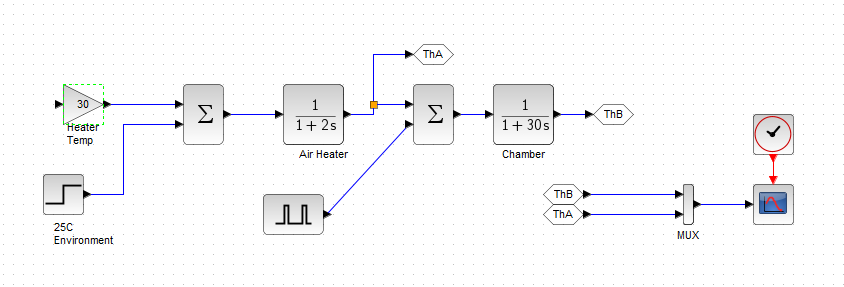
* Number of input ports:2

➔ We used CScope for ThA -ThB:

* Ymin:0
* Ymax:70
* Refresh Period:400
* Buffer Size:400

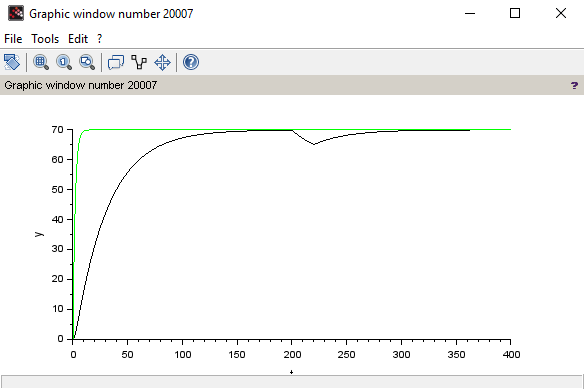
➔We connected the CScope to a Clock\_c block with parameters:

* Period: 0.01
* Initialization Time: 0



We plotted some points on the graph after running the simulation to see how the system behaved.

## Simulation of Physical part (Design)



Explanation of Results: The system container cover opened for 5 minutes at t = 200 and system stabilizes the Results: The system when the cover is closed. This demonstrates that the physical system is fully functional and operational.

## Simulation of Cyber physical system

Steps followed:

➔ We added CONST for 37C:

* Value:37

➔ We added RELATIONALOP for ThA < 37:

* Operator:2
* Use zero crossing:0
* Datatype:1

➔ We added PROD\_f

➔ We added GAINBLK for better visibility of CSCOPE:

◆Value:0.95

➔ We added MUX for Htr and UBound:

➔We addedCSCOPE for Htr-UBound: (for 400 minutes display)

* Ymin:0
* Ymax:1
* Refresh:409
* Buffer: 400

We added CLOCK\_c for ThA-ThB cscope:

Period Time:0.01

Initialisation Time:0

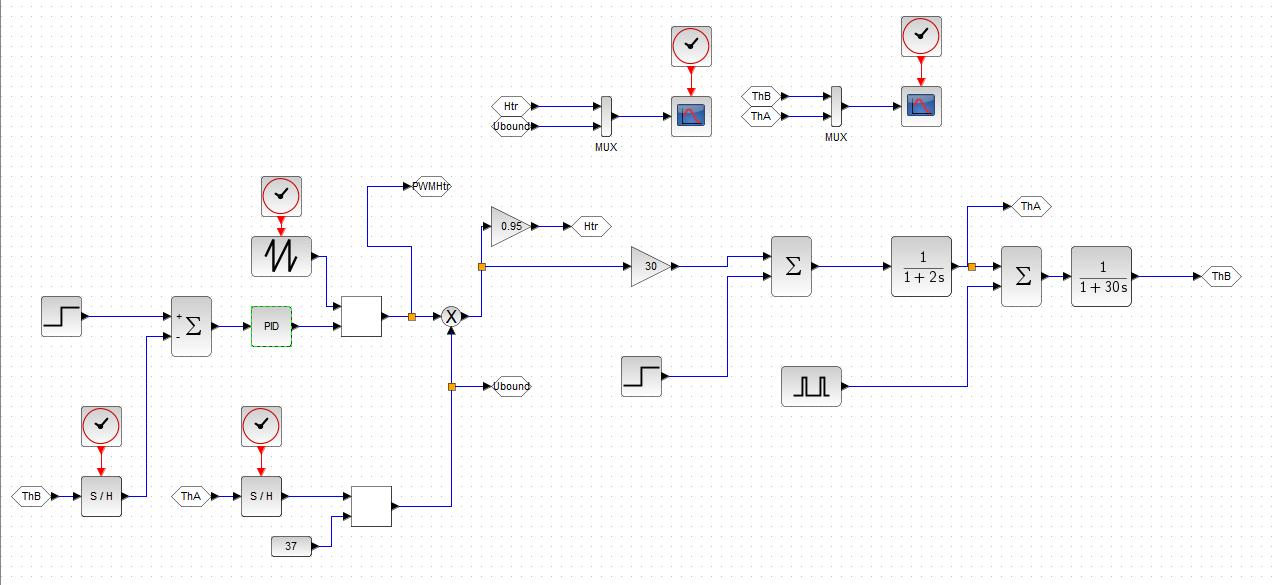
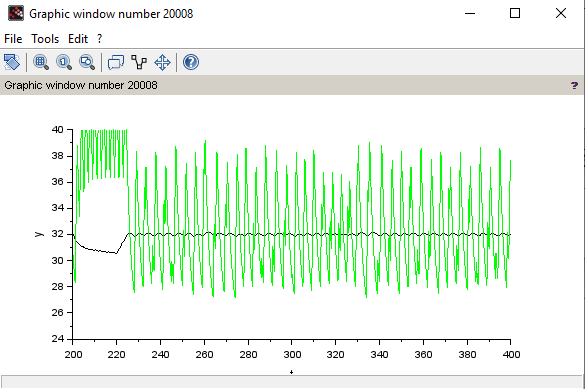


Figure 3: Full cyber part for the embedded system

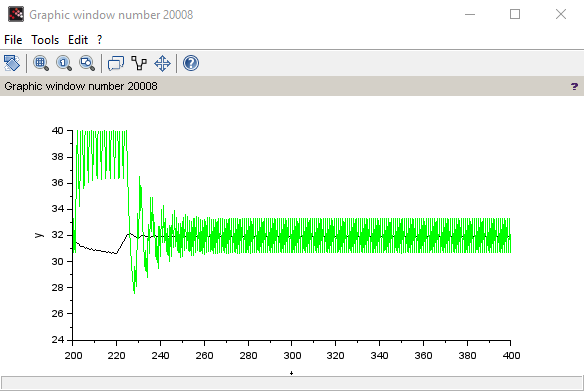
## Simulation results

**Simulation with PID control with Proportional control value 10.**



Simulation explanation: This simulation with proportional control value 10 was not getting the desired temperature (32C). This was due to our PID control parameters; we tested the PID control for various parameters until we found the appropriate parameters.

**Simulation with PID control with Proportional control value 3.**



Simulation explanation: This test demonstrates that the system heater will stabilize the container's desired temperature when the proportional control values are set correctly. When

the container cover is opened, the temperature of the system falls, and the heater manages to raise it to the desired level when the cover is closed.

# DigiTerm Design, Modifications, and Tests

## Modifications

(i-52 d-52 t-55 w-55)

We have modified the Digi-Term component as follows:

* This component is developed using Arduino 328
* Two 4k7 NTC thermistor sensors are used, with a 2700-ohm linearization resistor, both read temperatures in the range of 30…40 degrees that’s sufficient for our need. With 0.1C precision.
* Analog to Digital Conversion (ADC) formula parameters are modified to T1=30; N1=425; T2=40; N2=530; Tcr=16.5 (To compensate for the decimal digit) and
  + ThA = 10\*T1 – Tcr + (10\*T2-10\*T1) \*(Ns0-N1\*1.)/(N2-N1\*1.)
  + ThB = 10\*T1 – Tcr +(10\*T2-10\*T1) \*(Ns1-N1\*1.)/(N2-N1\*1.)
  + Voltage deliverables are as follows:
    - Vs(30C) = 5\* 2700 / (2700+3788) = 2.081 V
    - Vs(40C) = 5\* 2700 / (2700+2512) = 2.590

With above voltages, the NTC sensors readings are expected to be:

* + - Ns(30C) = 1024\*2.081 /5 = 426
    - Ns(40C) = 1024\*2.590 /5 = 530

Thus, Dynamic Quantization error of our ADC and NTC sensors with the above configuration:

* + - DQ = (530– 426) /1 = 104, DQ-dB = 40.34 dB
    - Ds = (40 - 30) / 0.2 = 10/0.2 = 50, Ds-dB = 33.98 dB

As can be seen that DQ > DS, therefore, indicating that the quantization error of our ADC is better. However, very close with respect to the sensor precision.

* The component is set to transmit temperatures through UART at 9600 (bps) baud.
* ThD (Desired Temperature) and Kp (Gain) can be set through the terminal by inputting decimal integers in the following form:
  + - \*#d for ThD, where # represents an integer
    - \*#p for Kp, where # represents an integer
* Temperatures and other data such as Kp, PP (Power Percent), H (Heater Status), AA (Air Alarm), and AB (Baby Alarm) values are displayed in a 4x20 LCD display.

## Implementations and Testing

Implementation:(i-55 d-52 t-55 w-52)

Testing:(i-52 d-52 t-55 w-55)

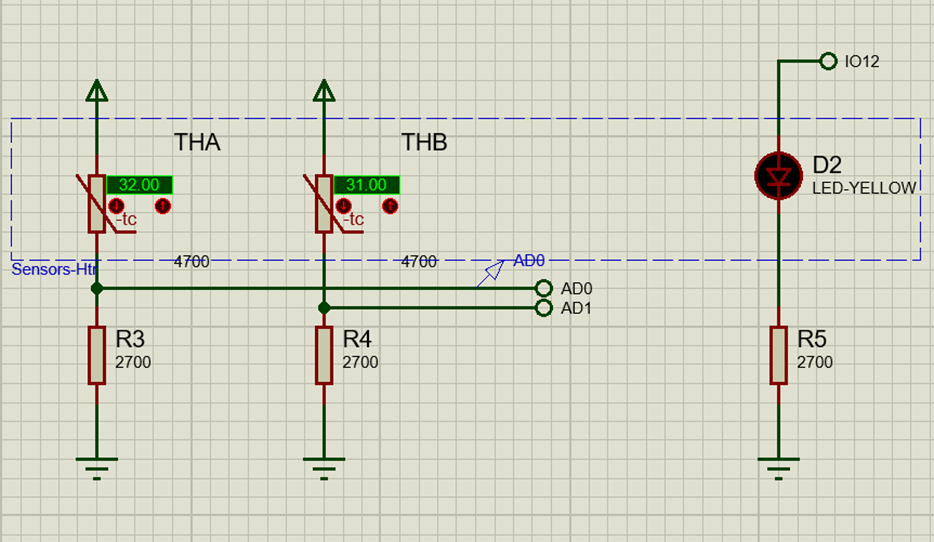


Figure 6 - NTC sensors implementation

The above figure shows the implementation of the system NTC sensors as well as an LED that’s activated when the heater is.

We implemented all three components with 2700 ohms resistors, both NTCs are 4700 ohms. After testing, the temperature values computed almost accurately match the actual temperatures.

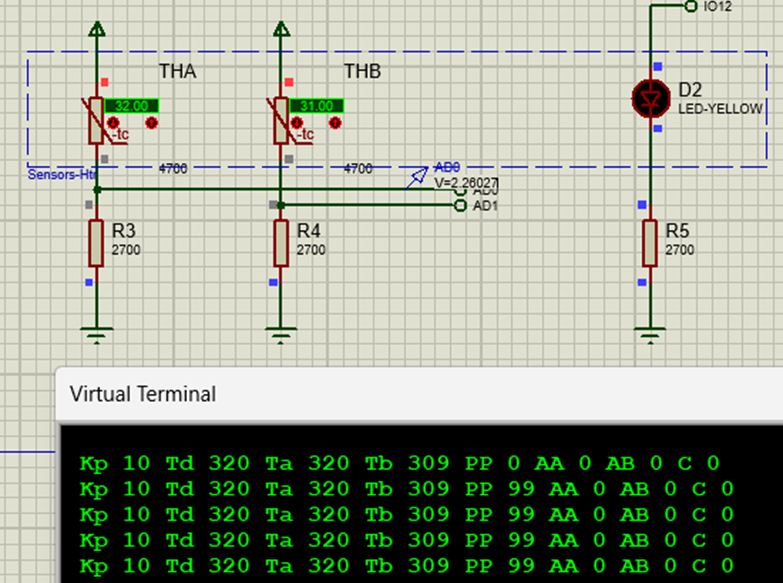


Figure 7 - Testing readings

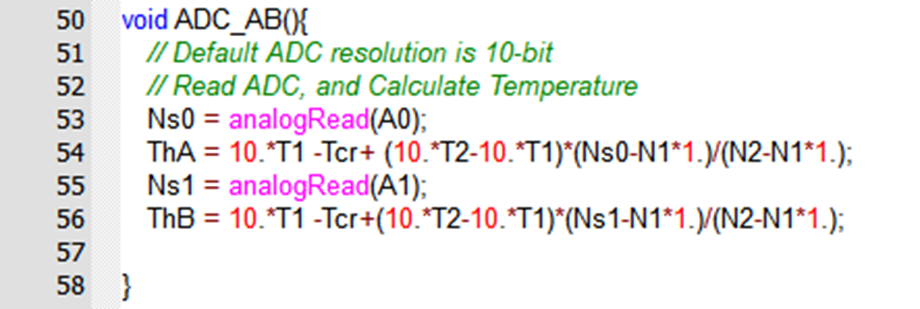


Figure 8 - ADC Readers and Temperatures Calculations

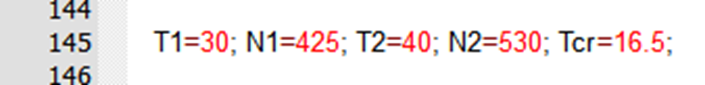


Figure 9 - Calibration Parameters

The above two figures represent the codes for reading analog values from NTC sensors and converting those values to temperatures using the predefined configuration.

To regulate and maintain a steady temperature with no swing in the control temperature, we employed proportional temperature controllers. As the incubator temperatures need to remain within 30 and 37 degrees and abnormal readings are considered harmful and thus an alarm is triggered. Thus, we used proportional temperature controllers to ensure that the temperature remains constant, every 0.1s reading, regardless of ambient temperature changes.

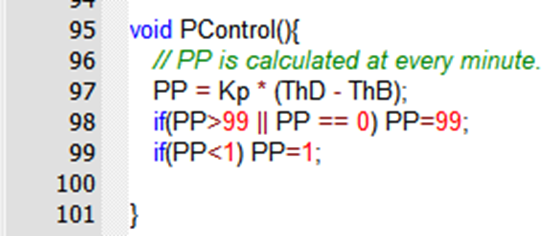


Figure 10 - Power Percent Calculator and Controller

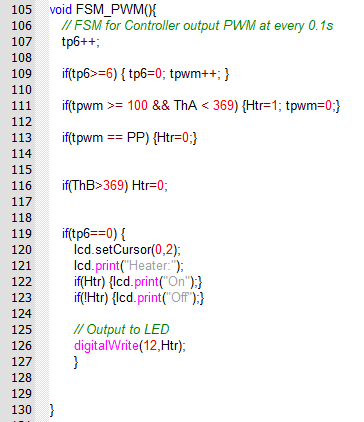


Figure 11 - Temperature and Heater Controller

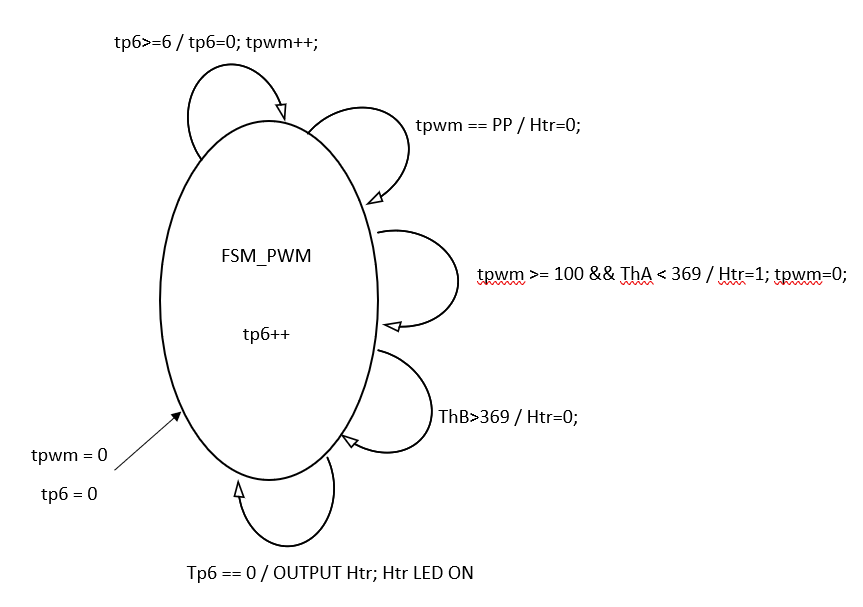


Figure 12 - FSM Control Diagram

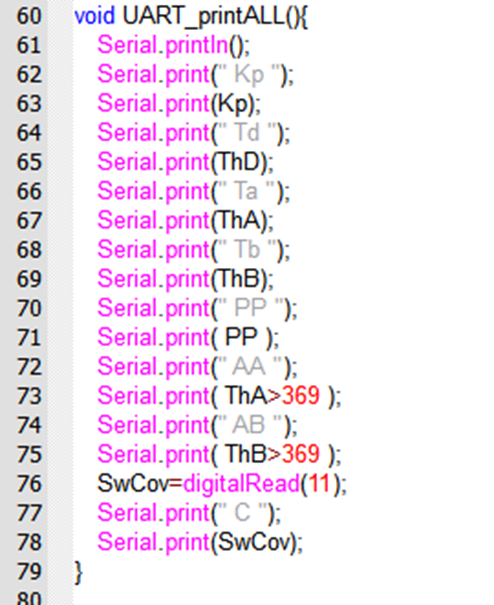


Figure 13 - UART Data Transmission

The above figure shows what the UART functions transmits data to the terminal. ThA/ThB>369 corresponds to the state or value of the temperatures, if they’re greater than 36.9 degrees which means 37 degrees, we output AA 1 an alarm or AB 1 accordingly and AA 0 or AB 0 otherwise.

## LCD Display

(i-52 d-52 t-55 w-55)

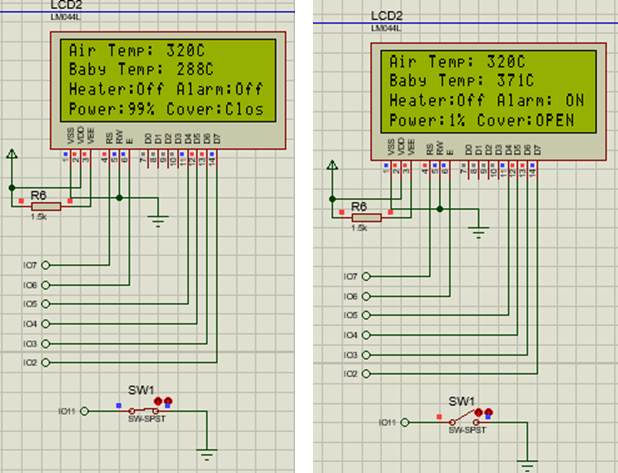


Figure 14 - LCD display configuration

In order to show the temperatures, Kp, AA alarm, AB alarm, PP values, H (Htr/Heater) status, as well as C indicating if the cover is open or closed. We’ve implemented an LCD that receives data through a function (shown below), such LCD is implemented alongside a 1.5k ohms resistor as a precaution.

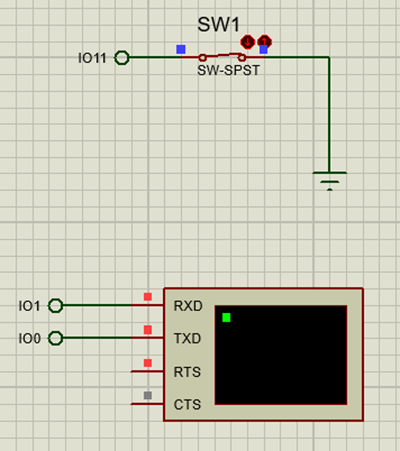


Figure 15 - Switch and Terminal implementation

As shown in the figure, a switch is implemented to imitate the opening and closing of the incubator cover. The figure also shows the terminal in which we display related data.

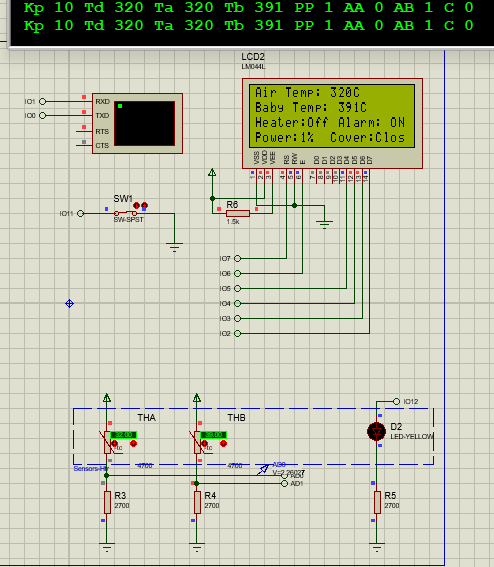


Figure 16 - Exceeding Temperature Limit Alarm Demonstration

# 

# Node-Red Dashboard Design and testing

(i71,d71,v73,v75,v76,v72,v74,w737576)

Node-Red is a flow-based development tool for visual programming developed by IBM. We can use Node-Red as part of the IoT to connect hardware devices, APIs, and online services. Node-Red provides a web browser-based flow editor, which can be used to create JavaScript functions.

## Installation of Node-Red

Firstly, we need Node.js on our computer to install the Node-Red environment. Then open the terminal and write the “npm install –g –unsafe-perm Node-Red" command.

After the processing is done, we will be able to run Node-Red. We need to add two more useful packages which we will use on this project. The commands to install these packages are:

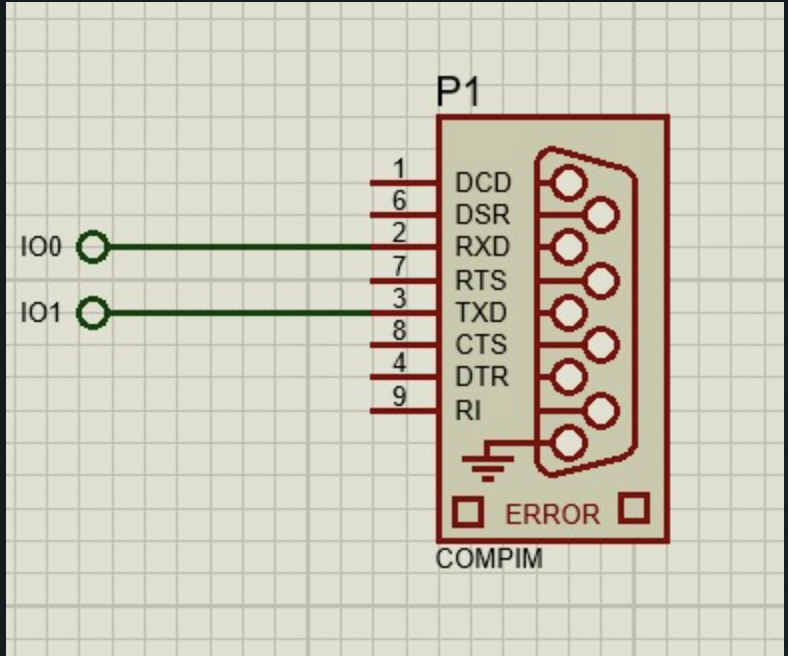
● *npm i Node-Red-dashboard*

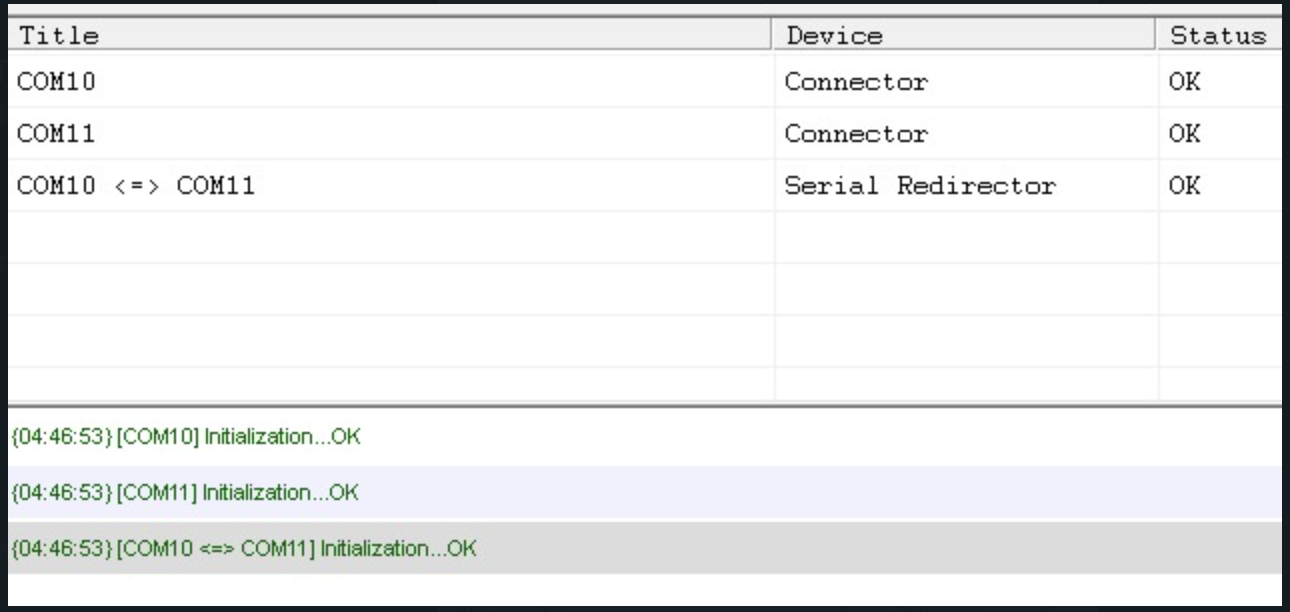
● *npm i Node-Red-node-serialport*

Installation of the Node-Red part is done. Since we downloaded the serial port package. Now we can add serial-in, serial-out, and serial request components into the palette on Node-Red.

## Setup VSPE

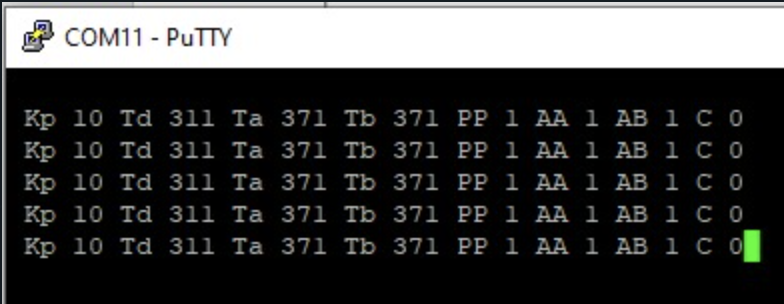
We should install Virtual Serial Ports Emulator to create a serial port emulator connection between Raspi and the Node-Red dashboard. On the VSPE program, we need to create two new connectors and a serial redirector to connect these ports.





## PUTTY

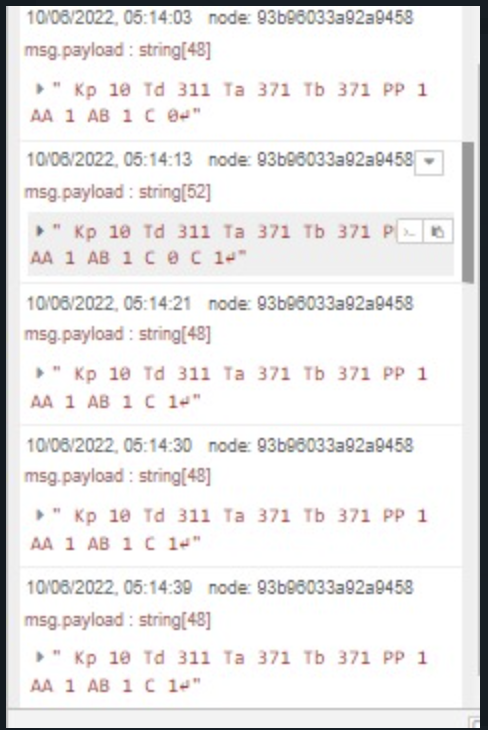
We use PuTTY to verify that the serial ports are functioning. In the Session tab we will select Serial and set the Serial Line to COM11

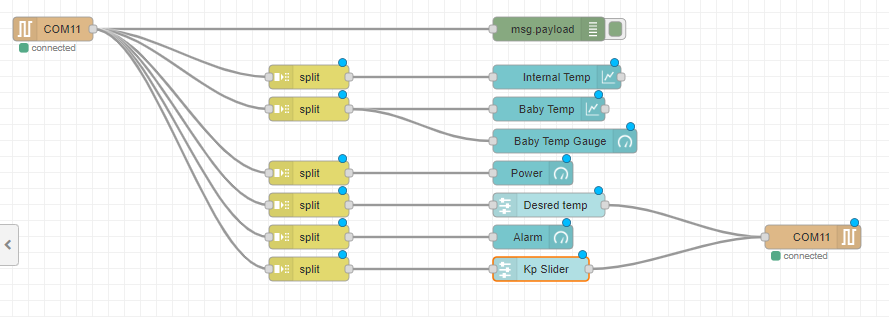


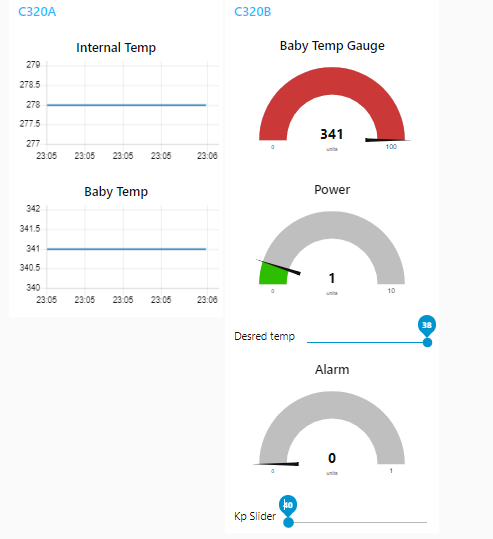
## Running Node-Red

Dashboard design deployed, VSPE running and proteus file running. Now we should see the outputs on the debug console for Node-Red.









# 

# Conclusions

**Thokozani Elliot Mupundu & Fatima Sada**

We have come to the finalization of this project and we’d say it was a great learning experience. Our team understood that every part of each process was crucial. One misstep would land us in a totally different result. That is why we took absolute care and we were very meticulous in regards to this project. First thing we did was we divided our team into subgroups of two and worked on different aspects of the project, after that we compiled every part and merged it into this final report. We worked on the Node-Red part. From our perspective building such systems using these softwares and tools is an essential part of the cycle of designing a real project in the real world. Our technology is developing every day so it is important to keep up with it and always use the tools that will help us observe reality efficiently. Some personal experiences that we had gained from this project includes how well we work and communicate with our team-mates and how efficiently we delegate tasks. Another thing is that we had excellent co-operation and we were all so eager to help each other out. Overall, it was a pretty challenging task but with the help of our amazing team players, we thrived!

**Saifuldeen Ali Majeed**

To conclude, the project shows clear challenges as it’s an integration of the real-world application with a modern, and user-friendly interactive interface. Thus, we collaborated on completing the project in a well-organized manner. Each team was responsible on a section, however, all teams worked interchangeably and helped each other to produce an accurate and well-designed system. As for my team, Uğur Gençer and I, and our section, we implemented the DigiTerm component of this project, we divided the work into implementors, testers, and writers. We demonstrated all modules using Proteus 8.9, tests were carried properly to ensure a complete, stable, and working system. Finally, the project has posed real challenges in managing such problems, nevertheless, with team collaboration we managed to handle them effectively. In return, I have learned a lot and enjoyed working with the rest of the team. This project has given me a chance to dive deeper into embedded systems in general and outcome a better understanding, it was a challenging process but worth it.

**Doğukan Dağtekin**

Overall it was a great experience being a part of designing an incubaby project. We had some issues with the simulation of the parts but with the help of teamwork, we managed to overcome all the problems. We worked synchronously we divided each part of the task into a group of two people. In my opinion, this project taught me a lot about embedded system design and how to work planning should be done when developing such a system. My task on this project was to create the physical and cyber part of the system and I worked with Orisa to design such a system and test it with different parameters. Thanks to all of my group friends it was so much fun working with them.

**Uğur Gençer**

In conclusion, I'd like to express my gratitude to Assoc Prof Dr. MEHMET BODUR for allowing us to work on such a wonderful project. I believe we were successful in reporting the project by effectively distributing tasks with our teammates. Our team was divided into two divisions, with each division consisting of two members. I worked with Saifuldeen Ali Majeed on Digiterm components. We also helped our teammates in other divisions when needed. Working on this project has significantly assisted me in reinforcing what we've learned in class, improving my ability to utilize the Proteus app, and teaching me how to work more efficiently in a group environment. I would like to thank all my teammates one by one, it was a pleasure working with them.

**Ekemadu Orisa Ekpette**

My team and I worked very hard on this project to complete the design and various parts of this term project. We took charge of the project and oversaw everything. We worked synchronously, and the work was assigned to each member of the group separately. This project has taught me to work more efficiently, dexterously, proficiently, and effectively as a team member. Technically, I was able to delve into embedded systems and learn how to design and simulate a project. It was an extremely difficult process. But, in the end, we triumphed and learned a lot, not only about embedded systems, but also about dealing with groups of people you don't know and getting to know them individually.