EEE088F 2021 Assignment 4

Temperature Sensor PiHat

ALTMIC003, DGDC001, DBYTAI001

10/06/2021

[Executive Summary [5] 1](#_Toc74343699)

[Introduction [5] 2](#_Toc74343700)

[Project Subsystems Block Diagram [5] 2](#_Toc74343701)

[2](#_Toc74343702)

[Design process [10] 3](#_Toc74343703)

[Power Supply submodule [10] 3](#_Toc74343704)

[Status Comparator-LEDs submodule [10] 5](#_Toc74343705)

[LM135 Temperature Sensor submodule [10] 7](#_Toc74343706)

[PCB [10] 9](#_Toc74343707)

[10](#_Toc74343708)

[Conclusion [10] 11](#_Toc74343709)

[Reflection on design process [5] 11](#_Toc74343710)

# Executive Summary [5]

The following report describes the design process, each submodule of the PiHat in detail, as well as the PCB design/development, and finally includes a reflection on the full design process. Various conclusions were drawn over the design process and are discussed in this report.

Each submodule - namely the power supply, the Comparator-LED’s and the LM135 temperature sensor submodules respectively - is discussed in depth. Lastly, the PCB design is discussed, and attached are screenshots of both the 3D and 2D models of the PCB design. Furthermore, hyperlinks to the git repository containing all relevant data and information for this PiHat design are listed in this report.

# Introduction [5]

Our PiHat is an analogue temperature sensor. The purpose of the Hat is to allow the user to get a visual indication as to the surrounding temperature. Temperature sensors can be used to measure ambient room temperature, or as a probe to measure the temperature of liquids or the inside temperature regulated appliances like freezers and fridges.

The circuit is powered from the 3.3V output pin from the Raspberry pi and is grounded into pin 6. The daughter board takes the 3.3V input and passes it through a buck-boost regulator IC, boosting the voltage to a steady 4.3V output to the rest of the subsystems. A physical switch is incorporated and allowing for the user to turn the Hat on and off while still being plugged in. If the circuit is closed, an LED will turn on to show that there is a voltage being outputted from the switch. This output will then run into an analogue temperature sensor that will be covered in a waterproof casing, so that it may be used as a probe. 3 LEDs display the output. 1 LED will turn on if the temperature is above 0 degrees Celsius. 2 lights will come on for a temperature of above 25 degrees Celsius, and all 3 lights will turn on for if the temperature is above 50 degrees Celsius. This was done using voltage dividers and op -amps in comparator circuit configurations. However, the voltage dividers could be exchanged for potentiometers, as this would allow for the threshold temperature of each LED to be set by the user.

## Project Subsystems Block Diagram [5]

# A piece of paper with writing on it Description automatically generated with medium confidence

*Figure 1: Subsystems Block Diagram.*

# Design process [10]

The design process began by first specifying fixed meeting to delegate work and set sprint goals for each fortnite, with a design review happening with a tutor taking place in the weeks in between where we could discuss and reflect on progress. An initial idea was developed and use cases were specified. A circuit was developed comprising of 4 separate subsystems that met each of the requirements of our use cases

(available here: <https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/Project%20Submissions/EEE088F%202021%20Assignment%201%20.pdf>)

After this submission, the team began to simulate each subsystem on LT Spice to see if the desired output would be achieved. Our first subsystem, the voltage regulator, became a source of concern as the initial design did not lie within the scope of the project design requirements. As such, this design evolved into a buck switching regulator, before finally a buck-boost voltage regulator IC component was chosen. Another point of inflection for the design came in our temperature sensor subsystem, as the LM355 sensors simulation had a different temperature to voltage relationship than the data sheets. This led to changing resistor values in order to hit the desired output.

(calculated and explained in depth here: <https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/Project%20Submissions/Assignment%202%20submission.pdf> )

## Power Supply submodule [10]

This power supply submodule is solely made up of an LTC3442 integrated circuit which acts as a buck-boost regulator, providing a stable 4.4V output from the 3.3V input supplied by the RaspberryPi. In order for this submodule to work congruently with the rest of the circuit, it has to have various specifications that fit the criteria of the following two submodules (i.e the TMP135 temperature sensor submodule and the Comparator-LED submodule). These specifications include:

* Vin (pin 1) = 3.3V (DC supply from the RaspberryPi).
* Vout = 4.4V (This value is reached as a result of the external configuration of the LTC3442 integrated circuit).

Chart

Description automatically generated

*Figure 2: LT Spice simulation of Vout of the LTC3442 IC.*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/Regulator/BuckBoost.png>

Diagram, schematic

Description automatically generated

*Figure 3: Screenshot of LTC3442 KiCad schematic*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/Regulator/BuckBoostKiCad.png>

This submodule fits seamlessly into the design of the full circuit. It reliably outputs the desired 4.4V needed by the TMP135 and the comparator-LED configurations.

## Status Comparator-LEDs submodule [10]

This subsystem is made up of a series of comparator-LED configurations used to display the current temperature threshold met and to further display whether the circuit/PiHat is on (receiving power from the Buck Switch Regulator) or not. Three of the configurations have their non-inverting input connected to the TMP335 data line, and their inverting input connected to the 4.4V line supplied by the Buck Switch Regulator. Each of the three inverting inputs implement a voltage divider which determines when the comparator will output high based on the voltage supplied by the TMP335.

For the status LEDs, 3 simulations are run showing the 3 states that our output diodes are in. The thresholds were predefined as 0 ℃, 25 ℃ and 50 ℃ .Graphical user interface, text

Description automatically generated

*Figure 4: Simulation when temperature is .*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/ComparatorLEDSub/2degrees.png>

A screenshot of a computer

Description automatically generated*Figure 5: Simulation when temperature is .*

A screenshot of a computer

Description automatically generated[*https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/ComparatorLEDSub/26degrees.png*](https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/ComparatorLEDSub/26degrees.png)

*Figure 6: Simulation when temperature is .*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/ComparatorLEDSub/52degrees.png>

Diagram, schematic

Description automatically generated

*Figure 7: Screenshot of the Comparator-LED subsystem KiCad schematic.*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/ComparatorLEDSub/ComparatorLEDKiCad.png>

According to all the theoretical conclusions drawn when designing this submodule, and from further subsequent LT Spice simulation, the status LED’s operate as they are supposed to. One LED will turn on within the first temperature range, two will turn on in the second, and finally all three will turn on in the third. It’s tough to draw a conclusion on the overall operation of this subsystem in a real-life situation as this circuit was only theoretically designed. However, if one did have to supply this subsystem with 4.4V (supplied by the LTC3442), it theoretically should work perfectly with no safety issues.

## LM135 Temperature Sensor submodule [10]

This subsystem is made up of the LM135 temperature sensor - the component being powered by the 4.4V line outputted by the Buck Switch Regulator, and further supplies the data needed by the comparator-LED configurations to output/display their temperature thresholds based on the voltage supplied by the LM335.

The LM355 temperature sensor was imported as an external library into LT Spice for the simulations. The data sheet supplied with the LM335 stated that the correlation between output voltage from the sensor and ambient temperature is given as: (eqn.1). Where T is measured in degrees Celsius.

A picture containing text, indoor, computer, screenshot

Description automatically generated

*Figure 8: Output voltage of LM135 for different temperatures (specified on the x-axis).*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/LM135Sub/tmp135Simulation.png>

Diagram

Description automatically generated with low confidence

*Figure: KiCad schematic for the LM135 submodule.*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/LM135Sub/LM135KiCad_Schematic%20.png>

In conclusion to this subsystem, it can be said with certainty that it will operate in a real-life situation safely. However, after running a sweep across the temperature from 0 to 100 (depicted below where the x axis represents temperature), it was evident that the LT spice component had a slightly different relationship to the theoretical relationship specified. Thus, a new equation was calculated for the relationship between the output voltage of the sensor and the temperature discussed above (eqn.1). For the status LED’s to accurately display which temperature threshold is met, the LM335 needs to output the correct voltage to the comparators based on their respective temperature thresholds. From the LM335 data sheet, it was seen that at 0℃, the sensor would output 2.73V, increasing by 10mV/℃. However, after testing on LT Spice, a different outcome was observed. For experimental and circuit accuracy, the simulated results for Vdata were used in place of the theoretical values. This ensured that the status LEDs turned on at the correct temperatures. Thus, the specifications for this subsystem were accurately met through empirical evidence from the simulation rather than theory.

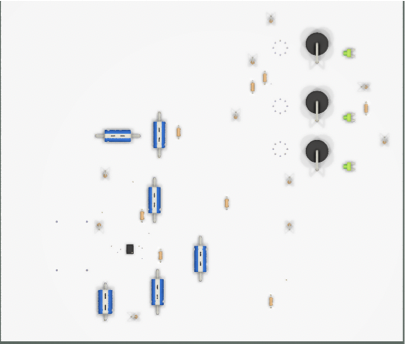
# PCB [10]

The dimensions of the PCB are: 65X30mm  
The mounting hole’s diameters are 3.5mm  
There are two connection points needed to be made to the Raspberry Pi Vin and GND these The Vin supplied should be connected to the 3.3V supply from the Raspberry Pi (pin 1) and the Ground (pin 6) Please note we have found an issue with our PCB as we have not properly formatted the size, input and ground pins. This will be adjusted before next submission.

The bill of materials can be found here :

https://github.com/TainedeBlaze/3088-PiHat- project/tree/main/PCB's%20Bill%20Of%20Materials  
A how to guide on manufacturing one’s own temperature sensor PCB can be found using the following link: https://github.com/TainedeBlaze/3088-PiHat-project/tree/main/Howto  
How to use your PiHat link can be found below (same link as above):

https://github.com/TainedeBlaze/3088-PiHat-project/tree/main/Howto



*Figure: Screenshots displaying the front(right) and back(left) of the 3D PCB*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/PCB%20and%20Gerber%20Files/3D_PCB.png>

# page1image29976320page1image29976112

*Figure: Screenshots of 2D PCB.*

<https://github.com/TainedeBlaze/3088-PiHat-project/blob/main/PCB%20and%20Gerber%20Files/2D_PCB.png>

# Conclusion [10]

During the initial stages of the design process, each subsystem of the PiHat was designed using theoretically calculated values that would at a later stage be used when designing the simulated version of the full circuit (all subsystems combined). Issues were encountered when calculating comparator resistor values as the temperature coefficient provided by the LM135 datasheet did not match the simulated results. A new temperature coefficient was then determined for temperature and voltage based on the LT Spice components. The threshold temperatures selected are  0℃, 25℃ and 50℃ and the resistor voltage divider values of each comparator configuration had to be edited until the desired output was achieved.

As mentioned above It’s tough to draw a conclusion on the overall operation of this system as a whole in a real-life situation as this circuit was only theoretically designed. However, if one did have to supply this ssystem with 3.3V from the Raspberry Pi to the LTC3442 which subsequently provides 4.4V line to the rest of the circuit, it theoretically should work perfectly with no safety issues.

During the PCB design stage we had to create our own component for LM135 temperature sensor and build the schematic for the sensor. This sensor was then placed into our larger schematic for the PiHat. The next step was to generate the netlist and build the PCB. This followed by the timely process of connecting nodes and ensuring all rule checks held zero errors.

Furthermore, the implementation of an agile approach to this project enabled the group to work as efficiently as possible. Particularly the use of SCRUM by short sprint periods in the form of fortnightly meetings and submissions, with the specifications changing according to new developments in the project.

The design process of this PiHat gave the group a great deal of insight into the components that make up this circuit, as well as how LT Spice and KiCad operate. Specifically the creation of a library component for the LM135 on LT Spice gave the group intuition on the workings of the software needed to design this system. However, there were various aspects of this assignment that resulted in unanswered questions that had to be solved using manipulation of results and component values rather than known calculations - although this was a direct result of the imported temperature sensor not fitting its datasheet, this assignment has been useful in conveying the complexities in design processes.

# Reflection on design process [5]

In retrospect, this project has taught us a lot about the design process and challenges one should expect to face in developing an idea. The course provided a streamlined process, which allowed for compartmentalization of the project. Breaking the circuit down into subsystems and developing block diagrams was an excellent way to allow for simultaneous development. The use of regular design reviews with the tutor was also great as it allowed the team time to set short term goals, and then reflect over the outcome of each goal at a later stage. This kept us focussed and maintained clarity.

SPICE was a great application to design and test our ideas, its hotkeys made it fast and easy to edit and change designs. However, the simulation did prove challenging when it came to simulating nonstandard components such as our temperature sensor, as the simulated component had slightly different characteristics as compared to the ones specified by its data sheet. As a result, we needed to change other components to match the output of our simulated values rather than our theoretical ones. This was important in teaching us about the need to adapt to the tools available to us rather than trying to stick with theory that does not work in practice.

The team all agreed that GitHub was an extremely useful tool in working collaboratively on the assignment. It allowed for fast transfer of information and served as a useful tool for version control and documenting our progress.

Looking forward, the team would recommend a faster feedback process, as there were moments where time to fix errors was limited in between submissions. However, we understand that this is a difficult request given the nature of online learning and the Pandemic. We are grateful for the experience and have a far better understanding of the process for designs.