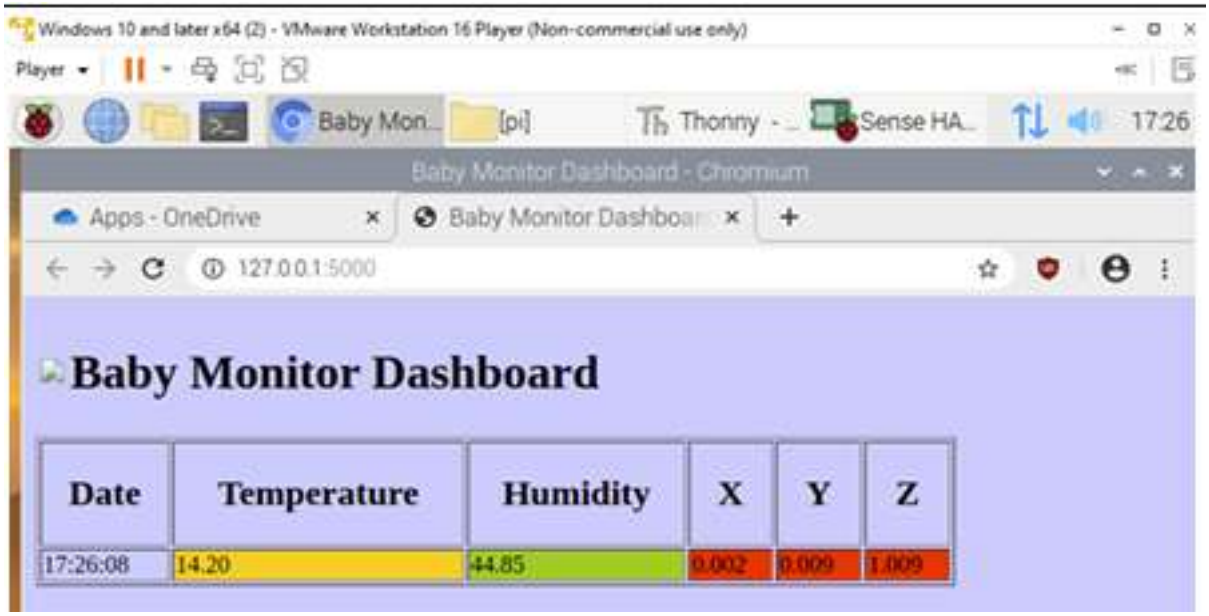
Baby Monitor

24th May 2024



Background:

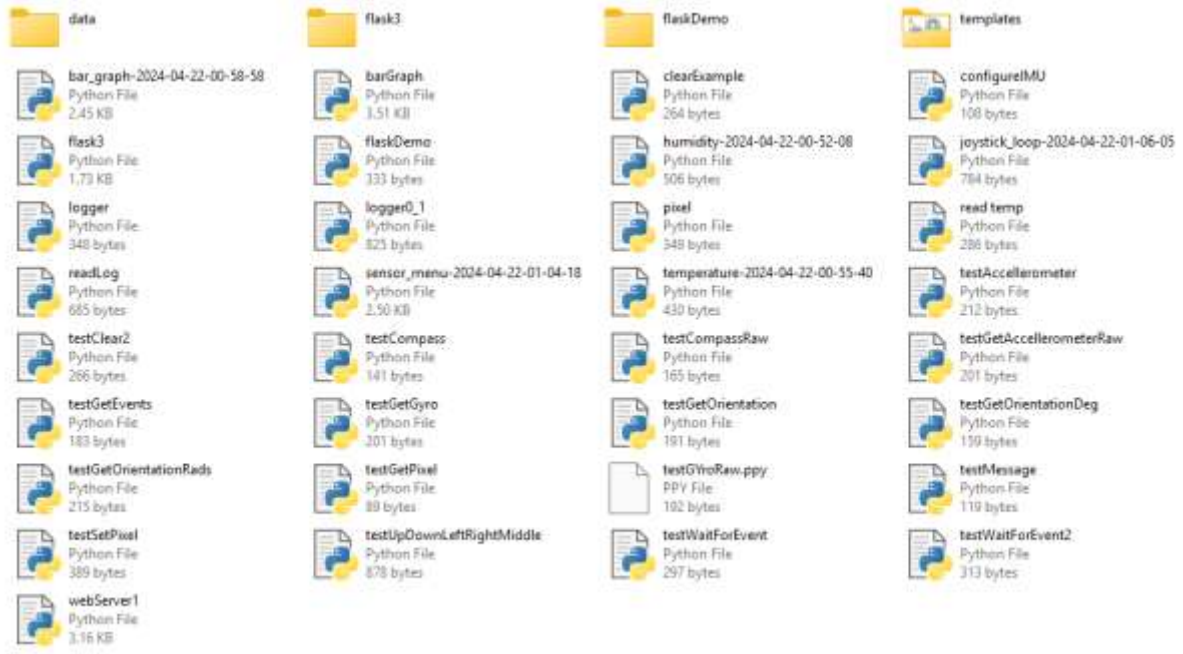
In January of 2024 I was set out to create a Baby Monitor system using a Raspberry PI to measure the room temperature, room humidity, and the baby's movement. I will be coding the Raspberry Pi using Phyton and coding my web page for displaying the monitors data using HTML. The baby monitor will give alerts to the user when the temperature is below or above the minimum or maximum temperature set on the monitor, with the ability to change the set temperature limit based on the time of year like summer or winter conditions. The background of the data will be green if everything is ok, yellow if it starting to change and is close to be bad, and red for bad conditions and will alert the parent about the environment not being good enough for the baby.

Planning Stage:

Before I create the baby monitor, I first set out a list of requirements that they baby monitor should have which include:

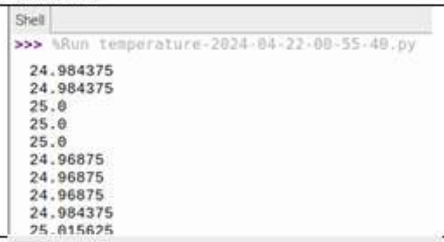
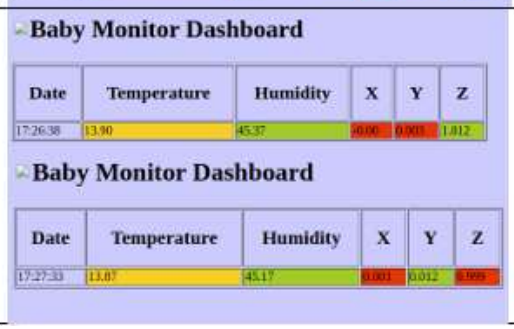
- The temperature sensor must be able to read the correct temperature in the room
- The humidity sensor must be able to read the correct air humidity levels in the room
- The motion sensor must be able to track the baby's x, y, and z axis of movement to make sure that baby is still moving a small amount to prove that they are alive but alert the parent if the baby moves a lot to show that there something wrong with them
- The web page must be able to display the data of the baby monitor from the Raspberry Pi
- The data on the web page must show colour indicates to show the progress of the baby monitors data to show if the baby is doing good or starting to show change, or is doing bad and needs attended to and alert the parent
- The web page must have a life time counted to show when it last updated
- The web page update timer should update every 5 seconds
- The Raspberry Pi's grid must should a percentage of the temperature of it in red to show the user what the room temperature is by just looking at the grid, and being able to switch to the humidity levels grid by clicking the button on the Raspberry Pi.
- The Raspberry Pi's grid must should a percentage of the humidity levels of it in green to show the user what the room humidity levels are by just looking at the grid, and being able to switch to the temperature grid by clicking the button on the Raspberry Pi.
- The device must save the data and log it on a file to save the data from the Raspberry Pi toa file that will be used to display the data on the web page.

Developing Stage:

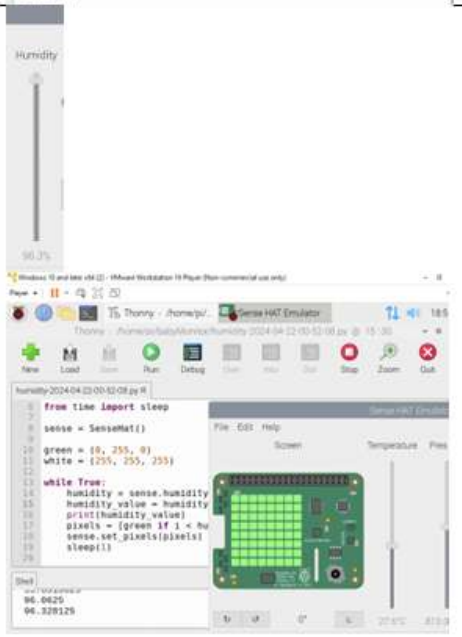


When developing the functions of the Raspberry Pi I created and coded a file for each function for the Baby Monitor such as the temperature sensor for reading the temperature, the grid to display the number of the degrees the temperature setting is reading, a logger to save the data of the Raspberry Pi in, a read logger function to read the logger file, and more. All these files and functions working together helps the Raspberry Pi to work as Baby Monitor by making the sensors record the correct data and being able to save and display it helps to make the Raspberry Pi into a working Baby Monitor.

Testing Stage:

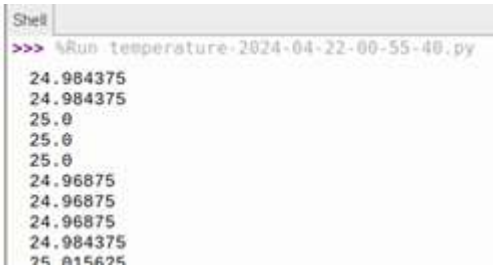
Description	Expected Result	Actual Result	Evidence
Measure room temperature	The programme should display the room temperature	The programme displayed the room temperature which was around 25 degrees	
Measure room humidity	The programme should display the humidity levels	The programme displayed the humidity levels which was about 28	
Measure baby breathing movements	The programme should display the x, y, and z location of the device	The programme displayed the x, y, and z location of the device	
Allow for sensitivity to be adjusted for the movement monitor	The programme should display new x, y, and z when moved	The programme displayed new x, y, and z location after being moved	

After I created all the files and function of the Raspberry Pi, I tested all the files and functions and recording them in a table to document each feature to make sure that they are working as they should. From the screenshot above you can see that the first two tests show that the Raspberry Pi is working correct in Python, and in the bottom two test shows that the Raspberry Pi has saved / logged the data and is also connected to the HTML file and is displaying the data on the web page I created / coded in HTML to display the data of the Raspberry PI. After seeing this data from testing I was happy to see that the Raspberry Pi I coded is working correctly as a working Baby Monitor system.

What I am Testing	What went wrong	How I fixed it	Evidence
I am testing to see if the temperature programme will read the temperature and display it on my sense hat.	Sense hat giving an error of not working as it importing the sense hat as a physical sense hat instead of the one that's being used for the emulator.	Because I am using an emulator for my sense hat, I forgot to change my code from sense hat to sense emu, as I originally made this code on a physical sense hat and forgot to change it to the emulator when testing my code.	
I am testing to see if the humidity reader works and to see if the grid can full display it without going passed the limit, so that if the number goes above 64, the grid will still display an amount that represents it without covering the whole grid.	The sense had gone over 64 for this testing and the grid was fully max, which meant that the grid value was invalid cause it only shows up to 64 rather than the actual amount which was 96 in this test.	I fixed this by coding a line of with an equation that would dived the humidity value by 100 and times it by 64 so that the result on the grid of the sense hat would display a matching value percentage that will show what the humidity is. In the first test the grid was full, but in the second one with new code it had 62 out 64 covered which shows that it is about 96% humidity levels which is easier to understand thanks to this equation.	

When testing I found a few bugs with my code that was making the data of the Raspberry Pi not working as intended. In this screenshot you can see two of these issues that I had. The first one I had was an error a simple error of working with an emulator I was using to work from home when coding for the Raspberry Pi and I forgot to change the SenseHat code to read from the emulator instead of a physical Raspberry Pi. Another issue I had was with the calculations with the humidity level read on the grid. My original calculations went up to 100 but the grid can only go up to 64 so any number above 64 would all display the same number of grid lights, to fix this I used a different calculation to make the grid be a percentage of 64 and not all a singular unit. This makes it easier and more accurate to read the grid for the humidity unit.

Temperature Sensor:



When creating the temperature reader for the baby monitor, I coded the temperature sensor of the Raspberry Pi to record the data of the temperature of the room and display the data in the python code and then save it into a log file that can be read in the web page for the user to be able to see that data and understand it. I also coded the HTML file for the web page for the temperature colour in the table to be green when the temperature is safe for the baby, yellow when it's still ok but is getting close to being too hot or cold, and then red if its too hot or cold for the baby so that the user can be alerted and can come and monitor their baby to make sure that they are ok thanks to the alert from the baby monitor.

Humidity Sensor:

When creating the humidity reader for the baby monitor, I coded the humidity sensor of the Raspberry Pi to record the data of the humidity levels of the room and display the data in the python code and then save it into a log file that can be read in the web page for the user to be able to see that data and understand it. I also coded the HTML file for the web page for the humidity colour in the table to be green when the humidity levels are safe for the baby, yellow when it's still ok but the humidity levels are not too high or low, and then red if the humidity levels are too high or low for the baby so that the user can be alerted and can come and monitor their baby to make sure that they are ok thanks to the alert from the baby monitor.

Motion Sensor:

When creating the motion reader for the baby monitor, I coded the motion sensor of the Raspberry Pi to record the data of the axis of the Raspberry Pi to monitor the baby's movement to make sure they are still moving ok and not in a position that could harm them, and display the data in the python code and then save it into a log file that can be read in the web page for the user to be able to see that data and understand it. I also coded the HTML file for the web page for the axis colour in the table to be green when the baby is moving a safe amount and is in a good position, yellow when the baby seems to be ok but is starting to show signs of a lot of movement to show that they are uncomfortable or starting to go into a dangerous position but is still ok, and then red if the baby is moving a lot which would be a sign that they are upset or is in a dangerous position and needs to alert the user to check to see if they are ok.

HTML Web Page:

I coded a HTML web page to display the data from a log file I used to record each data of the Raspberry PI and display that data onto the web page for users to easier read and understand. I also coded features for each data with colours so show a visual indication of each data to show at a glance and make it stand out more if the baby is ok or not. I also had a timer update to show the last reading of the data to help show users how up to date the data is to help them get a better understanding with how up to data the data is and how accurate it is.