Aaron Fischer, Margarete Graßl
January 2022
Germany

Astro - Air

Astro Pi Challenge

Part 1: Initial experiment idea and improvements

Our initial ideas for phase 2 all focused on quality of life onboard the ISS and we considered several different approaches. The idea that appealed to us most was the comparison of the air quality onboard the ISS with data from Earth found in every-day situations.

Our final concept is the following: We will use the humidity, temperature and pressure sensors onboard the AstroPi to measure air quality onboard the ISS and represent this data live on the LED-board. After the completion of the experiment on the ISS, we will compare and analyse our logged data to measurements taken on earth.

We have already commenced coding the AstroPi for this experiment and decided to make some additions: First of all, we will compare the data not only to that on earth, but also to the optimal default values already defined for the ISS. Secondly, we will make use of the PIR sensor to detect motion (e.g. from an astronaut) and relate that motion to changes in the data.

Part 2: Phase 2 - Our process

In order to work efficiently, we split the code into different sections, such as databanks, sensor measurements, LED display etc. We worked intensely with the web and Pi application “Sense Hat emulator” to write and test our code before trying it out on the AstroPi itself. After testing each individual script, we compiled and structured the scripts into one. We assembled the AstroPi and improved our script to account for unexpected problems, for instance by adding a log file which we found especially useful for testing. After we felt confident everything worked we performed test runs of 1 up to 5 minutes and a run over 3 hours, which all went very smoothly. One of our main concerns is the heat generated inside the AstroPi itself as the temperature sensor is located within the AstroPi's chassis. While performing our test runs, we observed severe increases of temperature, apparently due to internal heating. At first glance it seemed to be about a 10 °Celsius difference in temperature, but since we suspect the temperature of the AstroPi would consistently be high the increase could be more. If we see the same temperature effects during our measurements onboard the ISS, we will need to factor out constant heat and mainly evaluate the changes in temperature corresponding to changes in other measurements. In the meantime, we will perform practical tests using precise, calibrated temperature testing equipment using Pt -100 temperature sensors with a high accuracy which were made available to us to quantify the effects of internal heating.

In response to the feedback we got for our experiment idea - mainly centred around how we would obtain the optimal reference values and values from Earth, we decided to use researched optimal parameters from the ISS and data from measurements on earth which we performed ourselves as reference to the measured data. We would go on and evaluate these datasets after phase 3. To get values from earth, we came up with a second experiment: In order to make our data from earth comparable to the data from the ISS, we needed to simulate a surrounding similar to the surroundings on the ISS. Thus, we thought of using a classroom of comparable size with 3 people inside, some moving around and some remaining at one place for longer periods. The AstroPi will be placed in the classroom, running the same script as on the ISS. After a period of three hours of recording data, we will calculate the mean values and standard deviation for temperature, pressure and humidity.

Experiment results:

Over a period of 3 hours the AstroPi took measurements from the classroom. The values changed over time since we opened windows a couple of times and walked around to simulate normal conditions. We performed this test during end of January, so naturally the humidity was quite low. several different tests at home etc. had other results, and humidity is a parameter that changes drastically over weather, time of year, location etc. The mean humidity of the classroom was 21 % r.H. and other test runs had results up to 45 to 50 % r.H.

Next up was pressure. Throughout the experiment the data showed an evolution of 1023.5 hPa. up to 1025.5 hPa. setting the average pressure of earth to 1024 hPa.

Lastly we measured temperature. These values varied a lot, since opening the windows carried away a lot of heat. The average temperature we measured for our values from earth was exactly 19° Celsius. Again, these values will vary drastically by location - in our case in Germany - throughout the year, but we will continue with 19° Celsius as low and 25° - 26° Celsius as very high temperature.

Overall this experiment was a success, testing the capability and accuracy of our program once again.

Part 3: Technicality

Our script makes use of the following libraries: “sense_hat”, “datetime”, “time”, “csv”, “pathlib”, “logzero” and “gpiozero”. Three of our four sensors are integrated in the Sense Hat, for the fourth one we used pin 12 as sensor output. A large amount of the script is made up of variable definition, such as the optimal readings (target_temp, target_pressure, target_humidity) for all sensors - except, of course, for the motion sensor. After some research, we set the optimal temperature onboard the ISS to be 22° Celsius, the optimal pressure to be 1013 Pascal and the optimal humidity to be 60% r.H. We also introduced a set of variables to count how often we measured a specific parameter and a set to add all measurements of one parameter together. These will be used to create mean values after the main loop has finished. The next section of code defines the LED display for good, medium and bad air quality. We chose the easily recognisable colours green for good, yellow for medium and red for bad. To give the AstroPi a face, we additionally made those colours appear in the form of happy (green_led), neutral (yellow_led) or bad (red_led) smileys, depending on the air quality. These pictures will appear on the Sense Hat LED display. We thought that resembling actual data in an easily - graspable way was a good idea, so not only will the AstroPi show the air quality, it will also cycle through different sensors and show their current measurements: For temperature red, for pressure green and for humidity blue. We gave the remaining pixels a white colour to create contrast and a better recognisability.

The next larger section of code is made up of function definition. The first and one of the most important function updates the Sense Hat’s screen and sets the according amount of coloured pixels for temperature, pressure and humidity. The measurements are scaled to fit the frame of an 8 by 8 matrix, but the actual value can be calculated from this table, since each pixel corresponds to the same value.

Sensor	LED - pixels	Sensor data
Temperature	n amount of LED - pixel	n * 0.3125 + 10 °Celsius
Pressure	n amount of LED - pixel	n * 3.125 + 900 hPa
Humidity	n amount of LED - pixel	(n / 64) * 100 % r.H.

The scale for temperature was chosen to represent expected values in the centre of the LED display. Thus, because 0 pixels equate to (0 pixels) * 0.3125 + 10 = 10 °Celsius and all 64 pixels equate to (64 pixels) * 0.3125 + 10 = 30° Celsius, the scale reaches from 10° Celsius to 30° Celsius. For the pressure values we simply scaled the expected values to be somewhat in the centre of the LED matrix. The scale reaches from (0 pixels) * 3.125 + 900 = 900 hPa to (64 pixels) * 3.125 + 900 = 1100 hPa. Lastly, the humidity values are simply scaled so that 100% r.H. corresponds to all 64 pixels and 0% r.H. to no pixels. Thus the equation is formed: (64 pixels / 64) * 100 % = 100%

The next function is called diff and takes two values: The optimal value and the value that the sensors measured. The function calculates the difference between the two: a greater difference between target and measured values will roughly correspond to worse air quality.

Next up is the function for drawing the smileys. For each separate sensor, there is a separate function, since each sensor has a different threshold for good, medium and bad quality.

Lastly, there are two functions to create and write into a databank. We decided to use cvs files for this. The first function creates a cvs file and the second automates the writing process, storing date and time, temperature, humidity, pressure and movement (as a boolean: movement or no movement). Through the "pathlib" library we managed to keep general paths for the data_file and log file as not to store these in absolute directories.

Main loop:

First, we call the function create_csv to create a csv file in the current folder. Next, define the starting time to be at the start of the main loop to compare a separate time with this starting time and check if their difference has exceeded 179 minutes as to make sure we don't overstep the time limit with the end animation. This statement will be the condition for the main loop. This loop is expressed as a try - except - loop in oder to store any exceptions and their type into the log file. The main part of the try function is made up of cycling the displays of the measurements and the corresponding smileys. Additionally wether or not movement is detected will be printed through the terminal. Every two seconds, the LED display switches from temperature to pressure and finally to humidity displaying first the graphic followed by the corresponding smiley for each parameter. After six seconds, when each cycle is finished, all sensor measurements, including movement, are written into the generated csv file. Each time this happens, a message saying a new line has been started will be saved into the log file, too.

After the main loop has ended (because the experiment time has exceeded 179 minutes) the LED display will read "EXPERIMENT ENDED". The last lines calculate the mean values for temperature, pressure and humidity and append those into the csv file.

Part 4: After completion of the experiment and phase 4

After the completion of phase 3 - if we make it :) - we plan to use the library "pygal" to present and evaluate the data. A line diagram would be especially helpful to emphasise correlation between different sets of measurements, e.g. a correlation of the parameters temperature and movement. We could also use these diagrams to easily compare the differences to earth's approximate data. To present our results in an understandable way, we would use both the pygal file and a written documentation and evaluation of the experiment referring to the measurements we took onboard the ISS.