Online Summer Student Program at CERN
Research Internship Report:
Environmental Expander (EnviE)

Submitted by: Amani Alrumaih
Supervised by: Vagelis Gkougkousis

October 7th, 2021

Introduction

The goal of this internship is to write a code using Lua programming language that measures the signals from both the humidity and the temperature sensors. The environmental board, which is based on ESP8266, is equipped with platinum thin film Resistance Temperature Detector (RTD) and a Honeywell HIH-4021-003 calibrated and filtered sensor. To measure the signals, the analog-to-digital (ADC) response is converted using a voltage-temperature relationship. The code is broken into three main parts: connecting the device to Wi-Fi, converting ADC values, and creating a website to display the results. Due to time restrictions and the remote setting of the internship, only the solver that does the calculations for converting ADC responses was written.

Background Information

The temperature sensor has a range from -55 ^{o}C to 175 ^{o}C, while the humidity sensor has an operational range 0 %-100 % humidity at -40 ^{o}C to 85 ^{o}C. Temperature resolution is estimated to be 0.80 ^{o}C per ADC level. Humidity sensors take into account temperature accuracy and has a precision of 0.1-0.2%

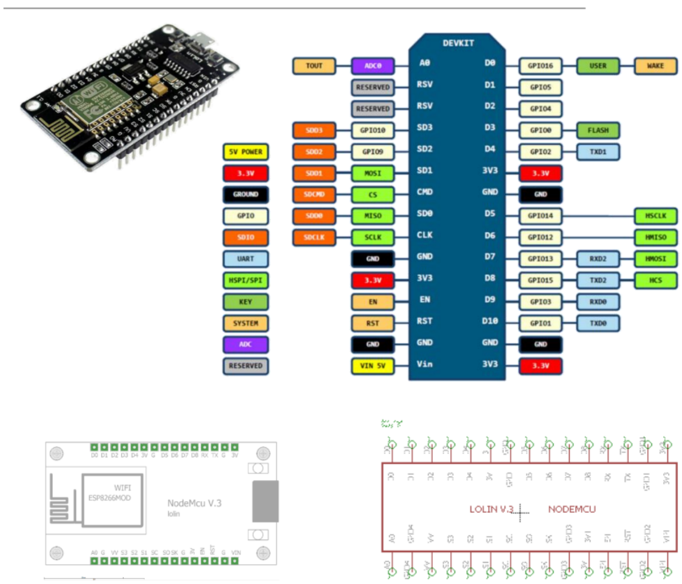
For RTD, resistivity varies linearly with temperature (Steinhart-Hart equation). To convert ADC response for RTD, there is no need for voltage divider since RTD levels $2.03 < V_{out} < 2.96$ for 5 V V_{in} and 115 Ω reference resistor. For a 10-bit ADC (1024 levels) between 0 - 3.3 V, 289 levels correspond to the temperature range, assuming full linearity:

$$L_{ADC} = \begin{cases} \dfrac{100 \times (1 + A \times T + B \times T^2 + C \times (T - 100) \times T^3)}{115 + \left(100 \times (1 + A \times T + B \times T^2 + C \times (T - 100) \times T^3)\right)} \times \dfrac{5 \times 1023}{3.3} & -55 \le T \le 0 \\[2ex] \dfrac{100 \times (1 + A \times T + B \times T^2)}{115 + \left(100 \times (1 + A \times T + B \times T^2)\right)} \times \dfrac{5 \times 1023}{3.3} & 0 \le T \le 175 \end{cases}$$

For the humidity sensor, there is a temperature-dependent behavior, the variation in V_{out} increases with humidity for different temperatures. Typical expected voltages between $0.96 \text{ V} < V_{out} < 4.57 \text{ V}$ for the range of interest, while the ADC range is only up to 3.3 V. Therefore, a voltage divider is needed. ADC level correspondence can be described by:

$$L_{ADC} = \frac{R_2}{R_1 + R_2} \times (\mathrm{R}H \times Slp \times (1{,}0546 + 0{,}00216 \times T) + \ Off) \times \frac{V_{cc} \ \times 3.3}{1023}$$

Figure 1 and 2 below illustrate the schematics of the environmental board and its properties.



- Single board microcontroller based on ESP8266
  - ✓ 80MHz RISC SoC
  - ✓ 64KBytes of instruction RAM
  - ✓ 96KBytes of data RAM
  - ✓ 64KBytes boot ROM
  - ✓ Winbond W25Q40BVNIG SPI flash
- Integrated single channel 10 bit ADC, 0 - 3.3 V range
- Integrated WiFi 802.11 b/g module
- SPI to USB interface for programming and power
- Integrated I2C interface
- 13 GPIO digital lines
- On-board voltage regulation with 5 V and 3.3 V output
- Extremely low cost (< 5 $ per module)
- Use of 8:1 analog multiplexer with 3 control lines to provide 8-sensor readout
- Arduino Core / Lua programming libraries

Figure 1: Environmental Board Properties and Description.

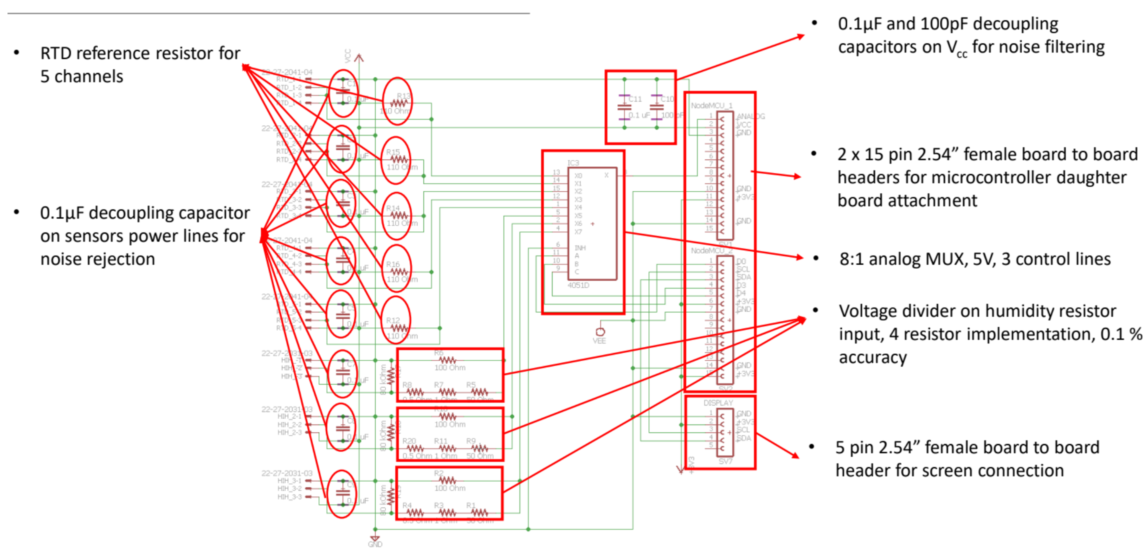
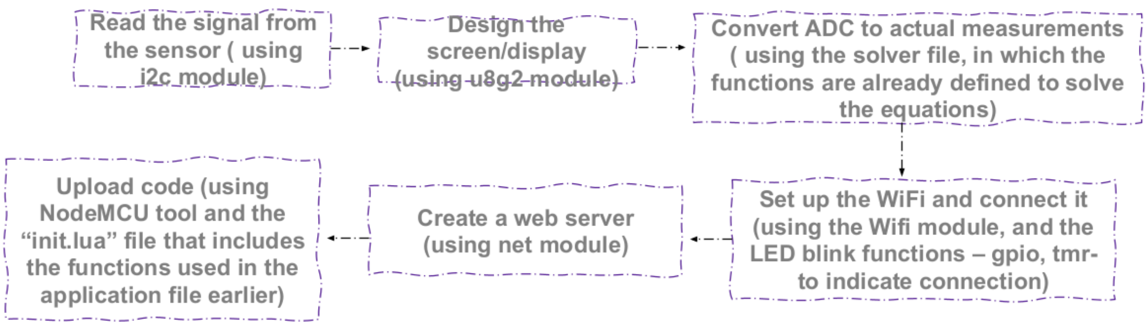


Figure 2: Environmental Board Schematics.

Overview of Internship Experience

After being introduced to the project concepts and the theoretical part, the next step was to get familiar with NodeMCU software and to install the necessary tools and packages and understand the electronics details of the board. Then, the code diagram below was created to start planning for writing the code



The Wi-Fi part has already been completed, so the focus of the work was to convert the ADC response values, which requires creating a solver file that solves polynomial functions. Also, the display screen features including the colors, the font, and the size of the text was manipulated as well. In addition to reading the ADC values in the first place and verifying that they are accurate.

Conclusion

As mentioned before, the duration of the internship was not sufficient to finish all these tasks. However, the solver file now has the quadratic and the cubic solver functions working. So, it can be used to convert ADC response values by calling the function and making sure that the sensors are connected. To finish the work on the ADC measurements code, the quartic function needs to be completed in the solver file.