



California State University, Channel Islands (CSUCI)
Department of Computer Science

**EMEC 325 Engineering Design
Lab Report
Fall 2021**

Group Project – Weather Station

Thomas Adams
003035090

11/14/2021

Abstract

The purpose of this project is to construct a weather station given the components and information regarding the purpose of a weather station. We obtained the weather station project from the NI myRIO project guide, number 58 Weather station¹. Weather stations includes several variable sensors used to measure a significant number of factors associated with the weather. Such are, pressure, humidity, temperature, lighting, wind direction, humidity, wind speed, etc. With this information in mind, we used two components from the myRIO kits to replicate our own version of a weather station using the sensors allocated to us, the NI myRIO, and LabVIEW to program our weather station. Upon incorporating such aspects into one integrated system, a weather station of demonstrating the results for temperature and light sensor value for the hour is produced. As a result of this project, we had to develop a greater understanding of the hardware interface for incorporating two sensors simultaneously to NI myRIO. While also gaining a proper interpretation of how the separate codes work for each sensor and what modifications are required to produce the desired results we seek and allow for the two codes to work together and run simultaneously.

Introduction

A weather station requires a series of complexities with a significant amount of confusion due to algorithms and circuitry. In addition, with the significant number of sensors, it can become an intimidating system that makes it difficult to approach. For those who seek to acquire data regarding the basic weather station capabilities, this project demonstrates how one can do so. At a cheaper cost, less complexity, and with great accuracy, one can create their own weather station system with sensors regard to factors they seek to measure, NI myRIO, and LabVIEW. Whether it be the same factors came to measure (light and temperature) or other factors (pressure, wind speed, etc.) or various mixes, one can measure multiple factors with greater ease as demonstrated in this project.

Formal Description

In initially pursuing this project, a decision had to be made on which measurements would be made, which kinds of methods via various kinds of sensors for each measurement. In this aspect we were provided four options, a Thermistor, temperature sensor, photocell, and an ambient light sensor. To understand which components to incorporate, we had to understand how each component operates and how this would work with our purposes.

First, the thermistor (shown in figure 1) is number 8 in the project guide previously referenced under the starter kit portion. This component is referred by the project guide as “a contraction of “thermal” and “resistor” – is a two-terminal semiconductor device whose

¹ Ni Myrio Project Essentials Guide.

https://download.ni.com/evaluation/academic/myRIO_project_essentials_guide_Feb_09_2016_optimized.pdf.

resistance varies with temperature”. The thermistor requires the use of the Steinhart-hart thermistor equation:

$$T = 1/(A + B(\ln RT) + C(\ln RT)^3),$$

a 3rd order polynomial. To obtain RT, a voltage divider would be incorporated to measure said value. The voltage divider equation being,

$$\frac{V_T}{V_{DD}} = \frac{R}{R+RT} V_{DD} = \frac{100}{1+\frac{RT}{R}} (\%FS).$$

For greater understanding and definition of this equation and the variables please refer to the project guide, under interface theory greater understanding of these equations can be obtained.

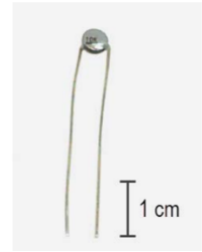


Figure 1 Thermistor

Next on our list of recommended components is the Temperature Sensor, (shown in

Figure 2). The temperature sensor is number 33 in the project guide, within the NI myRIO Embedded systems kit. In this case, we are referring to the Microchip TCN75A temperature sensor with I2C-bus serial communications. The sensor offers $\pm 1^\circ\text{C}$ accuracy over the range -40°C to $+125^\circ\text{C}$, nine to twelve bits resolution, and conversion times from 30 to 240 ms. The sensor also provides an “alert” output that triggers when the measured temperature exceeds a user-adjustable limit. Figure 3 shows the temperature sensor wiring diagram, and figure 4 a demonstration of the wiring to the NI myRIO. Upon connecting the temperature sensor to the code provided by the project guide, we obtain continuous measurements of temperature in Celsius. It is continuously obtained and

continuous variation in temperature every 250 ms as demonstrated in Figure 5. For greater ease of use and simplicity, we chose to incorporate the

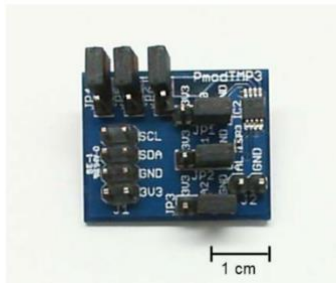


Figure 2. Temperature Sensor

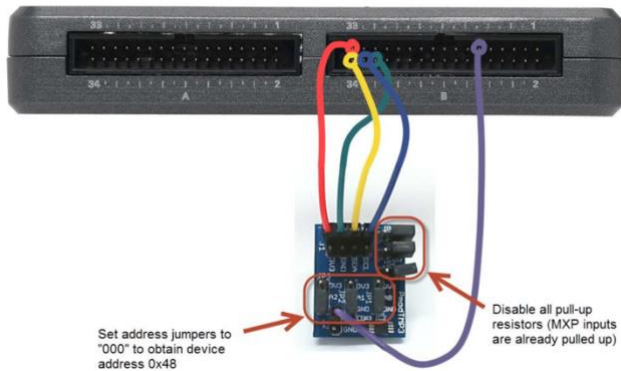


Figure 4 Temperature Sensor Wiring Diagram

temperature sensor rather than the thermistor for the purposes of measuring temperature. The temperature also provides greater accuracy and range when it comes to ability to measurement, and the continuity to measure within a short span of time.

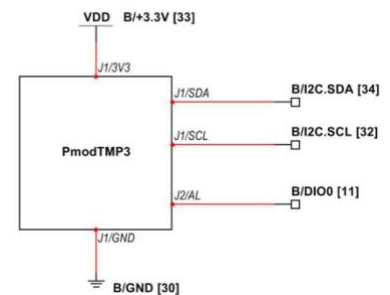


Figure 3. Temperature Sensor Wiring Diagram

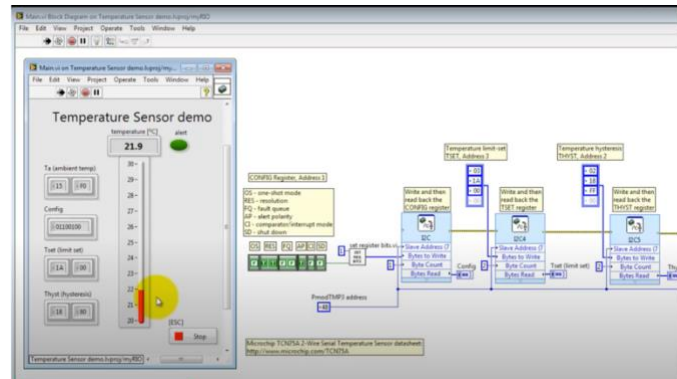


Figure 5 Temperature Sensor Demo, Front Panel and Back Panel

For the purposes of measuring light, we were provided the option of a photocell or an ambient light sensor. The photocell is number 9 in the project guide within the starter kit. The photocell (Figure 6), like the thermistor uses a variable resistance to measure the intensity of light in an environment. As stated in the project guide the photocell measures “10 k Ω at moderate illumination, less than 100 Ω at high illumination, and more than 10 M Ω in darkness”. Its counterpart, being the ambient light sensor shown in Figure 7.



Figure 6. Photocell

The PmodALS (Ambient light sensor in question) is referred to as number 24 in the project guide, under the mechatronic kit. The ambient light sensor is essentially a photocell transistor that converts the 8-bit measured value to a range of 0 to 255. 0 being none or very little light, and 255 at the brightest measurable value. For the purposes of

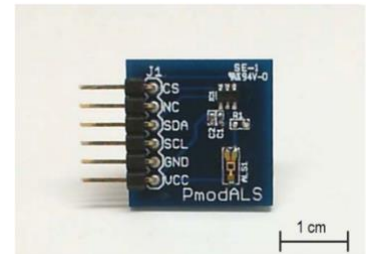


Figure 7. Ambient Light Sensor

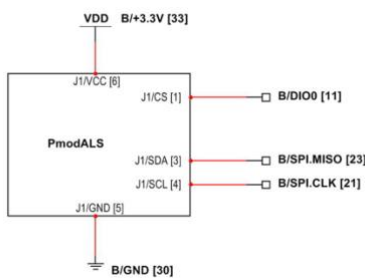


Figure 8. Ambient Light Sensor Wiring Configuration

our weather station, we will allow 0 to be cloudy to sunny, 255. As shown in figure 8 and 9, there is not great complexity in the wiring of the ambient light sensor. In addition, to ease in the wiring to the NI myRIO, we used the supplied LabVIEW program which easily enable us to measure the light sensor value continuous at every 100 ms. Figures 10 and 11

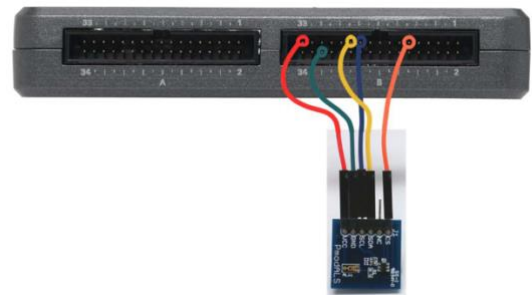


Figure 9. Ambient Light Sensor Wiring Demonstration

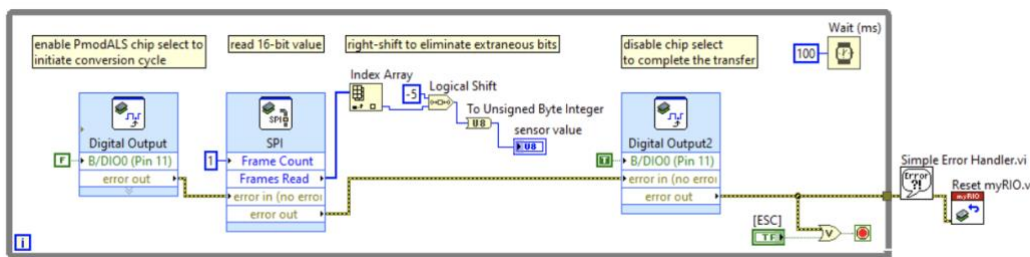


Figure 10. Ambient Light Demo VI, Back Panel

demonstrate this successful use of the LabVIEW Ambient light sensor VI program. As a result of its simplicity and ease of use, we chose to use the Ambient light sensor for the purposes of measuring light in the purpose of creating our weather station.

Upon successful selection and demonstration of our two selected components, (Ambient Light Sensor and Temperature Sensor), we were then tasked with being able to have both components connect simultaneously to the NI myRIO. To do this, we tried various configurations and ran the separate VI programs for both sensors separately to see if the configuration was successful.

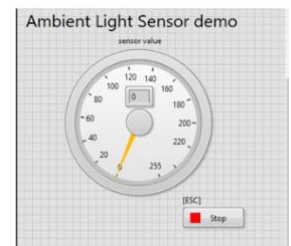


Figure 11. Ambient Light Sensor VI Demo, Front Panel

The first thought was that to leave the original connection of both sensors and place the shared connections (Voltage supply and Ground) Pins 33 and Pin 32 in Connector B, in parallel. And seeing as both sensors were connected to an arbitrary pin, pin 11, in connector B, we would merely change one of the sensors connections to any other arbitrary pin, such as pin 13 in connector B. Thus, we selected to connect the ambient light sensor to pin 13. We did have to

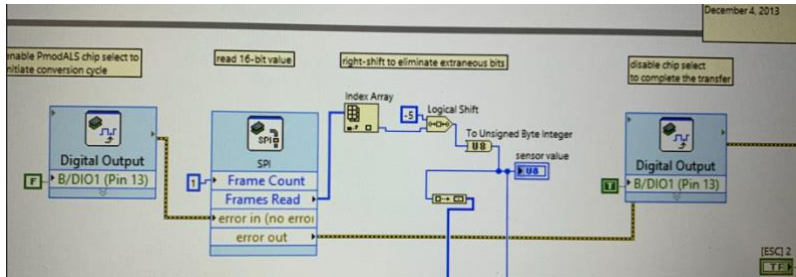


Figure 12. Pin 13 Configuration for ALS

adjust the code before testing this configuration, in which the Digital input in the ambient light sensor read from Pin 13 rather than Pin 11, because the temperature sensor Digital output was configured to Pin 11 in connector B of the NI myRIO. This change in the code can be seen in Figure 12. Upon executing this configuration, we found ourselves unsuccessful in our current pursuit. The error message had demonstrated improper wiring configuration that had not allowed for the sensors to be read from. More specifically, with greater review of our circuitry and the use of voltmeter, our Voltage supply was not working. Meaning the sensors were capable of successfully running in the current configuration except for the matter of improper voltage supply. To resolve this issue, we gave each sensor its own voltage supply to the NI myRIO. Therefore, the ambient light sensor (ALS) received pin 33 of connector B and the temperature sensor pin 33 of connector A. Leaving both sensors connected in parallel to ground via the breadboard, then into pin 32 of connector B. We also left the Digital input of the light sensor to pin 13 and the temperature sensor to pin 11 of connector B. Upon running the separate VIs, we proved successful in our wiring configuration and were able to run the separate VIs in the current wiring configuration shown in Figure 13.

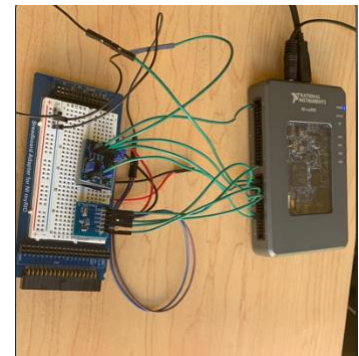


Figure 13. Temp Sensor & ALS Simultaneous Wiring Configuration

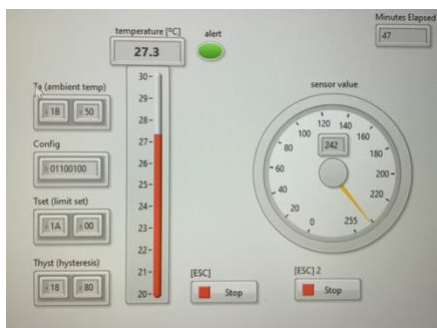


Figure 14. Temp Sensor and ALS Simultaneous Code Run Demo, Front Panel

With successful integration of the hardware, the next step was to acquire successful integration of the software in the LabVIEW code to run both sensor programs within the same VI at the same time. This was a simple task, we simply created a new Project VI and copied both modified codes into a new VI, title Weather Station. Figure 14 shows the front panel displaying the two working temperature and Light sensor codes.

Because we wanted to incorporate a waveform chart demonstrating the weather conditions for the hour, we

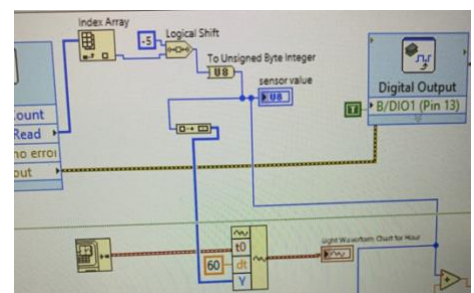


Figure 15. Waveform Builder, Back Panel (Light Sensor)

values to the Y variable in a waveform builder. The resulted waveform was displayed on the front panel. This process was incorporated individually for the ALS and the temp sensor. The

code for building the waveform is demonstrated in Figure 15 for the ALS and is identical for the Temp sensor. The front

panel waveforms for both temperature and light sensor values are shown in Figure 16 and 17. As shown, the temperature sensor can detect vary minimal changes in temperatures. The light sensor does as well. In Figure 17, a hand was waved over the light



Figure 16. Waveform for Temp Sensor, Front Panel

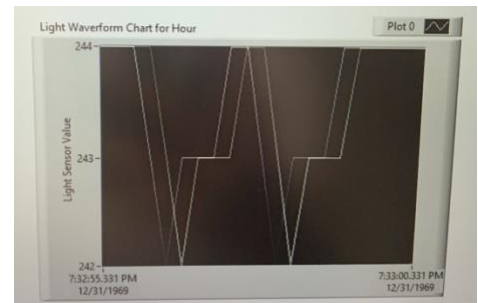


Figure 17. Waveform for Light Sensor, Front Panel

sensor to show the light sensor and the waveform to detect variance in the light, no matter how short the change occurred.

When the readings are constant within the period of 5 seconds (00:00:05:00) shown in the waveform shown at a time, a straight-line is shown in those 5 seconds, for the temp and light values where perfectly constant. With any variation in measurements in demonstrated in the waveform.

We then wanted this program to run for an hour and display the average temperature and light sensor values for the pervious hours. To account for an hour in our program, we changed the wait (ms) in each loop for the sensors to 250.

So that we can still view continuous readings at a high rate in our front panel. Since there are 3.6×10^6 ms in an hour, we used this see that our loops will iterate 14,400 times within one hour.

Meaning we will measure the light sensor values and temp 14,400 times within one hour. We simply inputted a comparison statement within each loop, that

when the iteration reaches said value it will stop. Stopping both loops at the same time, (after one hour of the program beginning), allowing the entire VI program to stop. The code is shown in Figure 18.

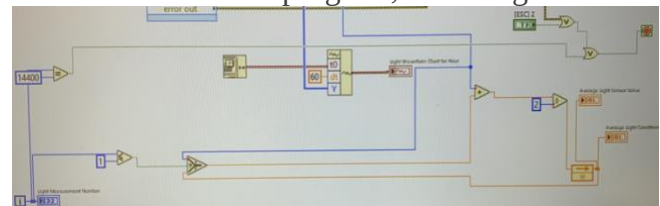


Figure 18. Hour limitation code, Back Panel

Lastly, we needed to average the values, to provide the average readings for the hour. This was done using comparison statements as shown in Figure 18. In each loop, an average is obtained, and then that average is used in the next iteration, to compose the new average. Resulting in the last iteration, the end of the hour, to have the average reading for the entire hour. This is done by adding the current reading with the previous average and then average the two to yield the new average. When the iteration is within its first iteration, where no previous average is possible, we check for this using a equal comparison and an if true or false statement. If the iteration has no previous average, we simply added the current reading to itself and divide by 2, resulting in the first average being the first read value. Thus, we can then

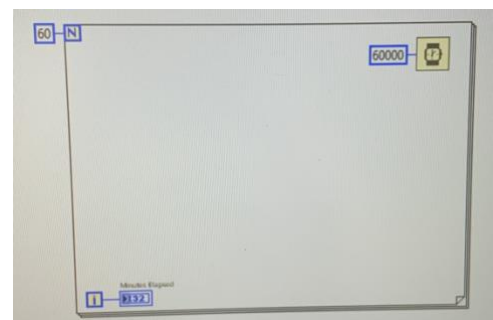


Figure 19. Elapsed Time Code, Back Panel

continue obtain the average for the current moment time as the program is running, and upon the end of the run, the program will have displayed the average reading for the entirety of the hour. In addition, for the sake of preference, we incorporated in elapsed time counter. A for loop

whose waits (ms) is 60,000 (1 minute) and repeats only 60 times, and the displayed iteration number is the number of minutes that has elapsed.

We have now completed our weather station and to attest for accuracy have run the program to measure the outside weather conditions and compared them with the predict weather stations on a local weather site. We ran the program from 11:41am to 12:41pm, here in Camarillo, CA at CSU Channel Islands, on Sunday November 14, 2021. The Weather report site (reference 2) predicted it to be 81 °F during the hour of 11am and 82 °F during the hour of 12pm.

It was sunny during both hours without a cloud in the sky during this time. As shown in Figure 20, the average temp for the previous hour in concern was 81.275 °F, and the light sensor value shows 244 out of a 255 range. Meaning it was significantly sunny. We have successfully created our weather station with accurate readings for the previous hour that the programs had ran for.

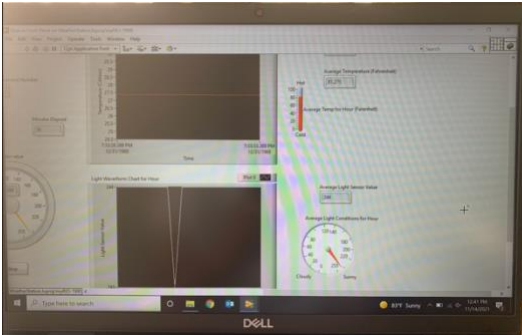


Figure 20. One hour Program Run with Displayed Results, Front Panel

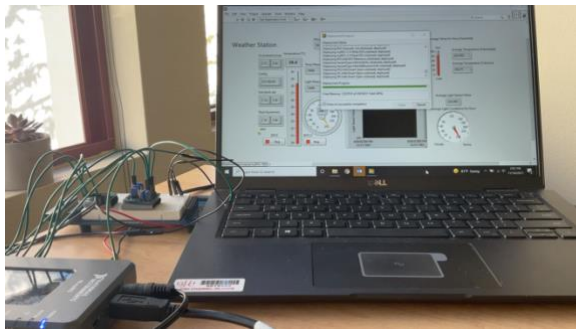


Figure 21. One hour Demo, Program Running from Start to Finish

2021, where an online weather station². Figure 22 shows that it is shown 84 °F and Sunny. As shown in Figure 21, we were precise and accurate in reading similar if not nearly exact readings regarding temperature and light values.

Conclusion

With successful completion of our weather station project we have demonstrated, with the sue of LabVIEW, NI myRIO, and selected sensors, one can replicate a weather station with accuracy and precision. In our case, we attempted to demonstrate we can find the temperature

Figure 21 shows a one-hour video of the program running in its full. If you are viewing this report as word doc, simply right click on the video in figure 21 and select play. The value shown for the average hour match those in Figure 22. As shown the program begun at 2:02pm and ended abruptly one hour later at 3:02pm. The date is Sunday November 14,



Figure 22. Weather Conditions for 2pm and 3pm

² MSN, <https://www.msn.com/en-us/weather>.

and light sensor value for an hour of time, accurately. We demonstrated such use with a Temperature Sensor (TCN75A tempera-true sensor) and Ambient Light Sensor (PmodALS). However, this could have easily been replicated with other sensors for different kinds of measurements, such as pressure, wind speed, humidity, etc. It is important to recognize that success was a result of proper hardware configuration, in which sensors receive their own Voltage supply from the NI myRIO and configuring the code to meet any Digital Output Pin changes. Modifications, to meet the circumstances we sleeked to satisfy where not difficult with the use of LabVIEW and as a result, others can meet their own criteria and desired results with modifications of their own. Lastly, our program could have easily been simplified in greater ways as well. For example, rather than check for iteration for the hour limitation, we could have simply converted the sensor while loops to for loops and set our 'n' index value to 14,400. Such menial aspects can be adjusted for greater simplicity and ease of use, and the code itself could have been approached in several other ways to achieve the same desired results or even betters results. With the use of LabVIEW, and NI myRIO there a significant number of ways to approach the task.