BEE 508 Final Project: Design of Temperature Measurement and Display

Gwen Montague and Olivia Killingsworth

June 10, 2022

Abstract

In this final project we use a While Loop in Keil IDE to design and implement an embedded system that senses temperature and outputs that data onto an LED display. The sensor is an integrated circuit (I^2C) and the LED display is a UART (receiving). Both these devices are communicating via a microcontroller. We also use an Analog Discovery 2 and its accompanying WaveForms software to perform real time logical analysis and confirm the UART and I^2C are sending and receive dynamic data. The program in Keil includes some added features such as indication of a Celsius temperature, and clearing of leading zeros on the LED display for any temperature greater or less than 0 degrees.

Introduction

This project enables us to demonstrate the knowledge we gained in 508 of embedded systems. Specifically, we implement an Embedded C program that controls a UART receiver and an I^2C via a microcontroller. To do this we will modify a provided starter code. We must also connect all hardware components correctly and use an Analog Discovery 2 to retrieve and analyze logical data from both the UART and I^2C devices.

Materials

TIVA™ EK-TM4C123G Launchpad
SparkFun 4-digit 7-segment display – UART
SparkFun Digital Temperature Sensor Breakout - TMP102 – I^2C
Analog Discovery 2 (AD2)
Jumper wires
Mini grabber hook to female jumper wire test leads
Keil IDE
WaveFroms from Diligent
Freezer

Methodology

Procedures

Procedure 1: Hardware Assembly

The TIVA™ Launchpad must be connected to the UART, which serves as the receiver, and the I^2C temperature sensor. First, the TX port on the TIVA™ is connected to the RX port on the UART, the power pins are connected to each other (+3.3V on TIVA™ to positive terminal on UART), and ground is connected to ground. This configuration allows the UART to receive information from the TIVA™ board. Second, the serial data (SDA) and serial clock (SCL) on the I^2C are connected to PD1 and PDO on the TIVA™ Launchpad, respectively. The power and ground pins of both the TIVA™ and ^{I2C} must also be connected. This allows the I^2C to send temperature data to the TIVA™ Launchpad.

Next, the logic analyzer (AD2) needs to be connect to the TIVA™ Launchpad so we can analyze the data that is being collected by the I^2C, sent to the TIVA™ board, and then received and displayed by the UART. To do this two of the AD2 analyzer channels (DIO0 and DIO1) are connected to the serial clock

(SCL) and serial data (SDA) pins on the TIVA™, respectively. We also need to set up a connection between the UART and the AD2 via the TIVA™ board. We use the DIO1 channel to do this and connect it to the TX port on the TIVA™ Launchpad. The DIO1 channel will have to be manually switched between the UART and I^2C connections as needed. A ground-to-ground connection is also established for the logic analyzer.

Procedure 3: Verify Hardware Assembly
We plug the hardware into the computer via USB and run two test codes – one for the UART and the other for the I^2C. For the UART, we run program LED_module_UART_test_code_final.c in Keil, build target, download it to the TIVA™, and press the reset button on the Launchpad. The LED display correctly outputs "EE88."

Next, a new project was created in the Keil IDE and program Temperature_sensor_I2C_test_code_final.c was run. After building the target, downloading it to the TIVA™, and pressing the reset button, a blinking green light appeared on the Launchpad as expected.

Procedure 3: Analog Discovery 2 Setup
To set up the UART logical analysis the RX protocol is selected in WaveForms and the parameters are set as described in the provided directions. The trigger is set to idle and the base setting is 1 ms/div. This process is repeated for the I^2C logical analysis set up. We select the I^2C protocol, follow the provided directions and set the necessary parameters, set the trigger to normal, and set the base setting at 20 µs/div.

Procedure 4: Programing the Device
After downloading the provided starter program Read_and_display_for students.c, we set out to modify the starter program in order to run the temperature sensing and display device.

Inside the provided While Loop, we added the following features and code to the program: (See source code file for code)

And If/Else statement is used to establish a feature where if no data is received from I^2C sensor, the TIVA™ Launchpad will read in a red LED light to indicate no data is being received.

Since the I^2C transmits 2 bytes of data, we convert that 16-bit binary to a 12-bit binary. We do this by using a shift to ignore the lower 4 bits of byte 2 and then combining byte 1 and byte 2.

The 12-bit binary value is then converting to a Celsius value (in binary) by multiplying it by 0.0625.

Next, the following operators are using to convert the 12-bit binary Celsius value to a string in decimal values:
% (modulo)
/ (division)

The decimal value was then sent to the UART LED display. When sending the data to the receiver, we clear leading zeros of decimal values and indicate the temperature is in Celsius using IF/Else statements. There are 3 numerical digits, and a 4^{th} that is constant "C" to indicate Celsius. Leading zeros only appear in digit 3 or digit 2. Digit 1 can display as 0 so that the LED display can output a temperature of 0 degrees.

Test Plan

To confirm the procedures outlined above were successfully implemented the following Test Plan was run:

1. Verify the hardware connections
    a. Compile the test code LED_module_UART_test_code_final.c in Keil and download to the TIVA^{TM} board
        i. Look for build errors
        ii. Press reset and visually inspect LED display to make sure the expected “EE88” is displayed
    b. Compile the test code Temperature_sensor_I2C_test_code_final.c in Keil and download to the TIVA^{TM} board
        i. Look for build errors
        ii. Press reset and visually inspect TIVA^{TM} board to make sure green light is blinking as expected
2. Verify starter code in Keil
    a. Compile the starter program Read_and_display_for students.c in Keil
        i. Look for build errors
3. Verify modified program
    a. Compile the *modified* Read_and_display_for students.c program in Keil and build target
        i. Look for build errors
    b. Download the *modified* Read_and_display_for students.c program to the TIVA^{TM} board
        i. Press rest button
        ii. Visually confirm UART displays decimal temperature values within $\pm 7^{o}$ C. of ambient room temperature (20^{o} C.) and the “C” to indicate Celsius measurement
4. Test temperature sensing and display device by subjecting device to dynamic temperature
    a. Increase ambient temperature
        i. Apply body heat (finger) to sensor
        ii. Visually inspect LED display and confirm Celsius value increases
    b. Decrease ambient temperature
        i. Place sensor on towel and put in freezer
        ii. Visually inspect LED display and confirm Celsius value decreases and leading zeros are cleared properly
5. Verify output data with Logic Analyzer
    a. Inspect logic analyzer in WaveForms

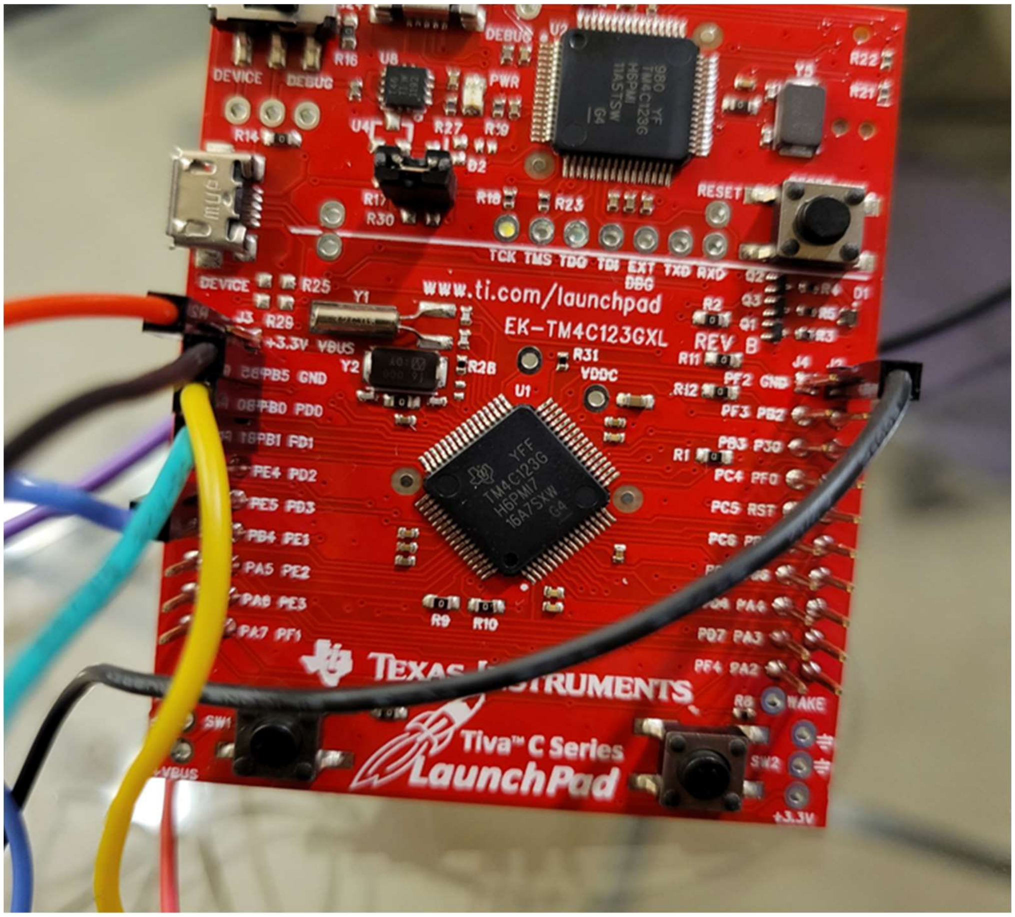


Figure 1.1: Hardware connections on TIVA™ for UART & I^2C

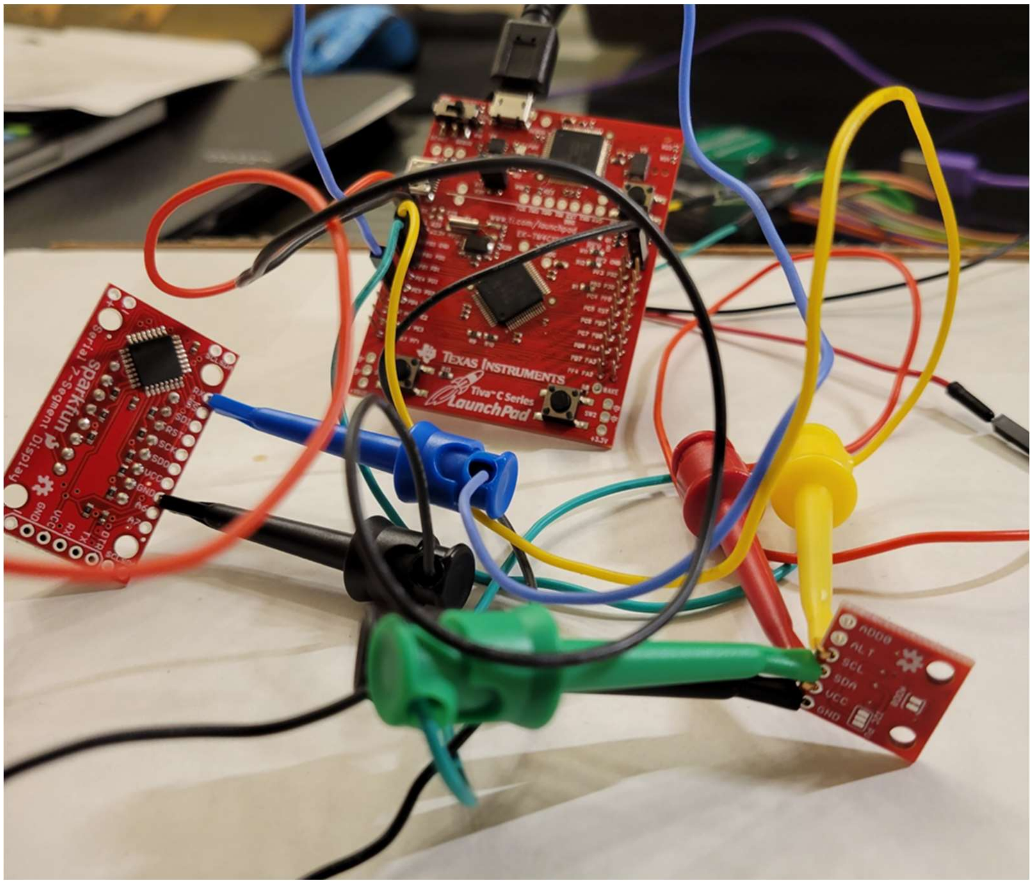


Figure 1.2: Hardware connections between TIVA^{TM}, UART, & I^{2}C

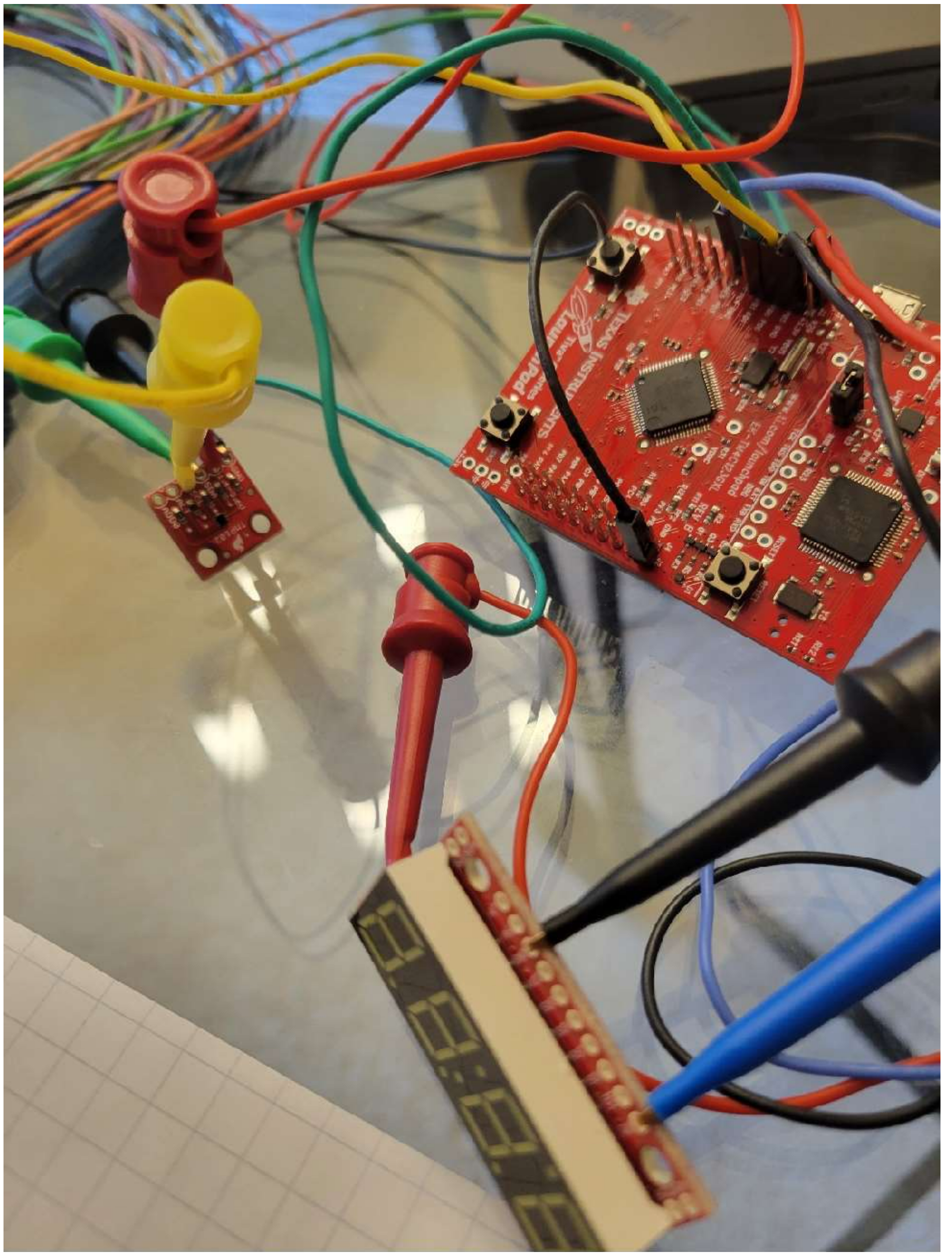


Figure 1.3: Hardware connection between TIVA™ & UART

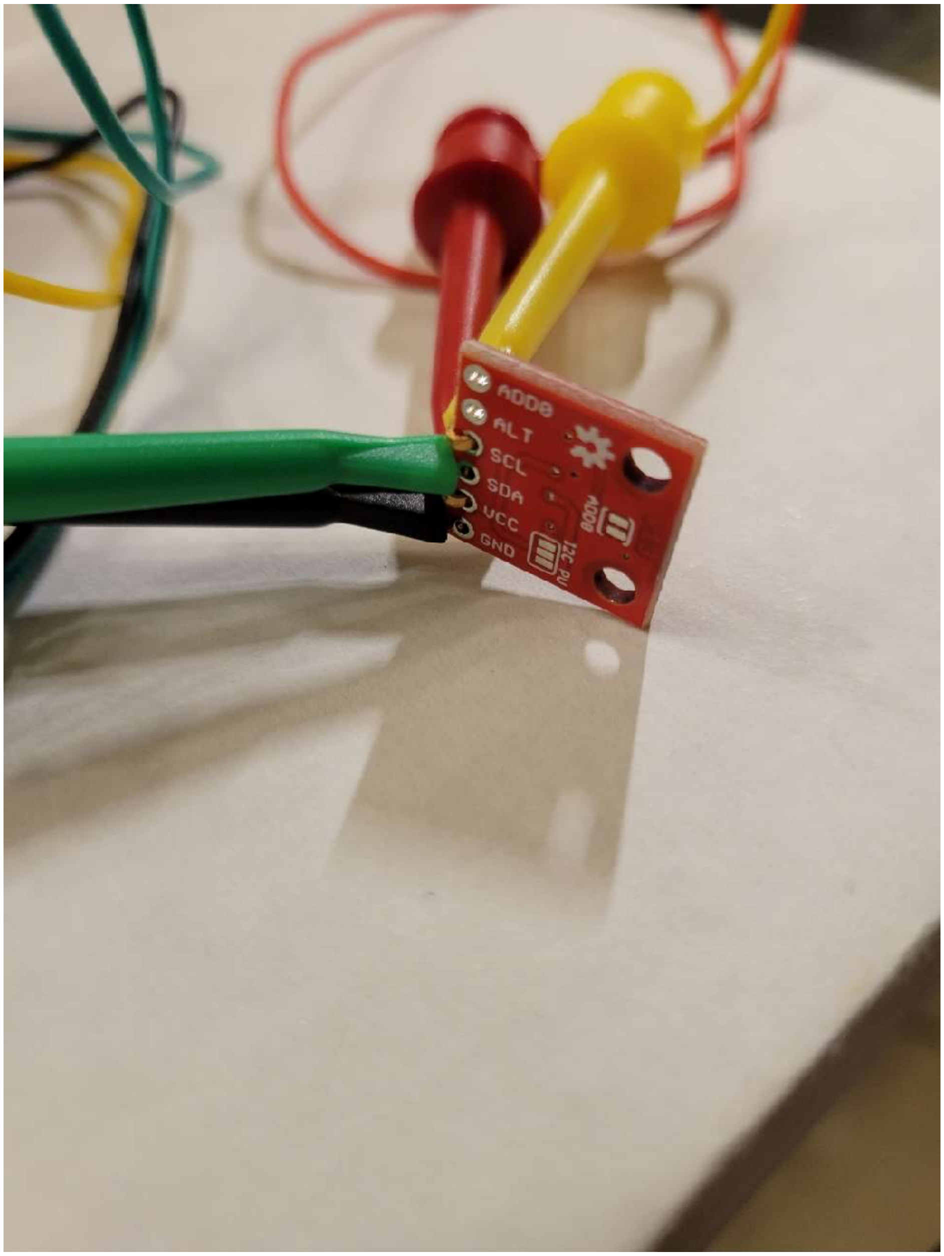


Figure 1.4: Hardware connection on I^2C

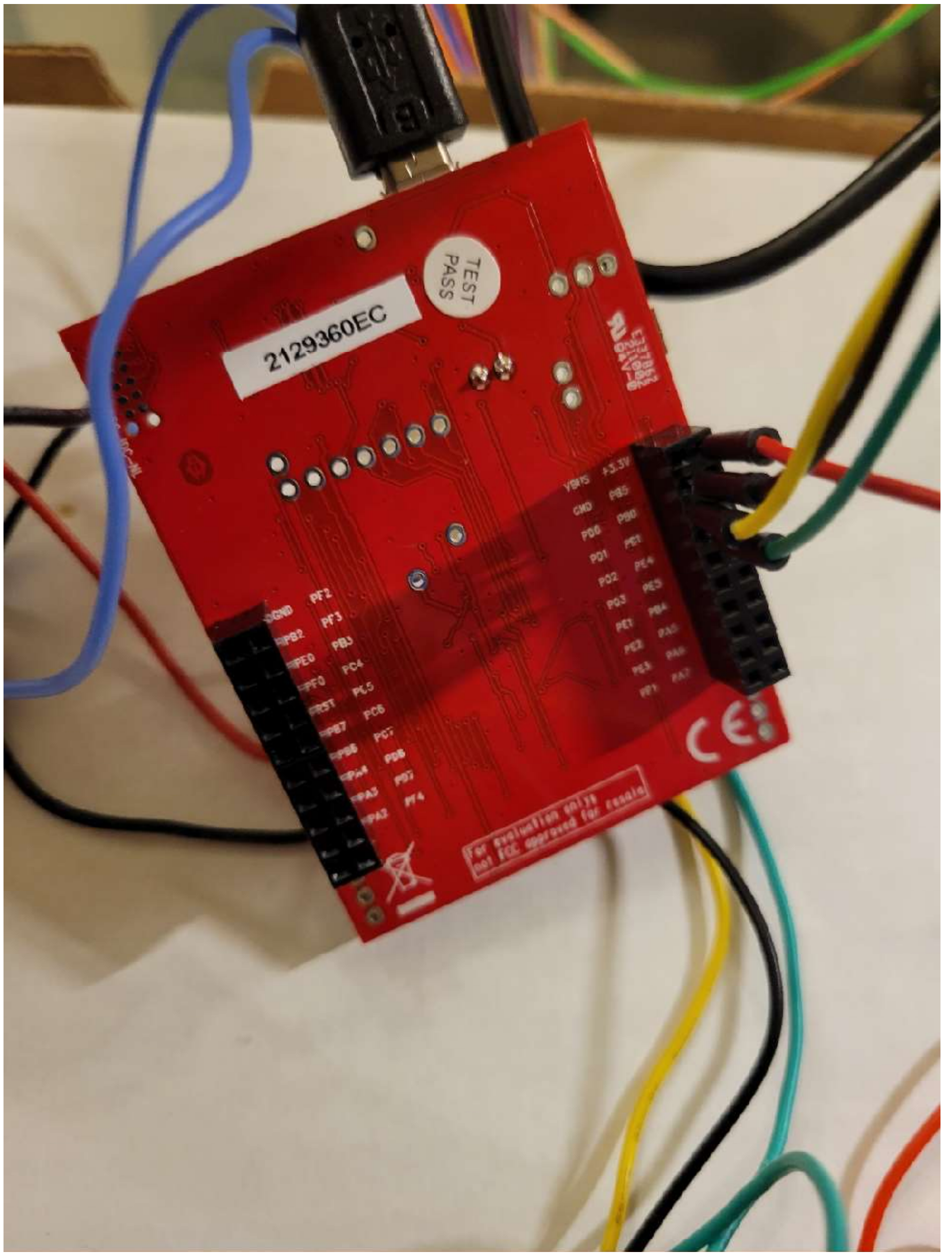


Figure 1.5: Hardware connection for TIVA™ to AD2

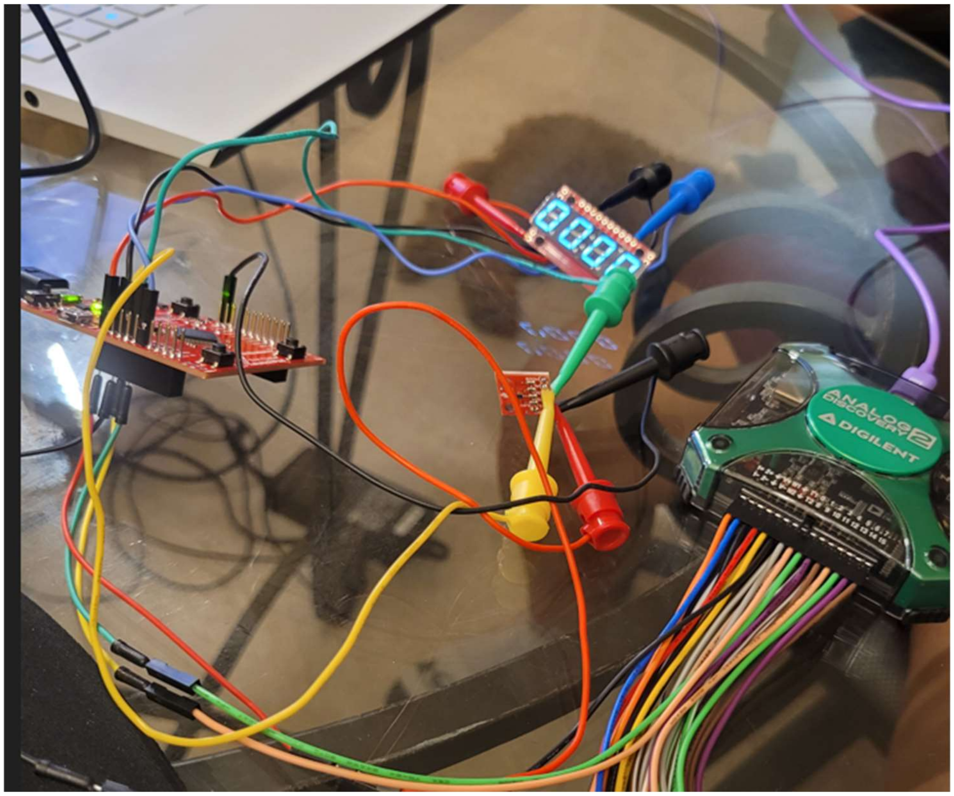


Figure 1.5: Hardware connection between TIVA™ & AD2; Visual display/output in process

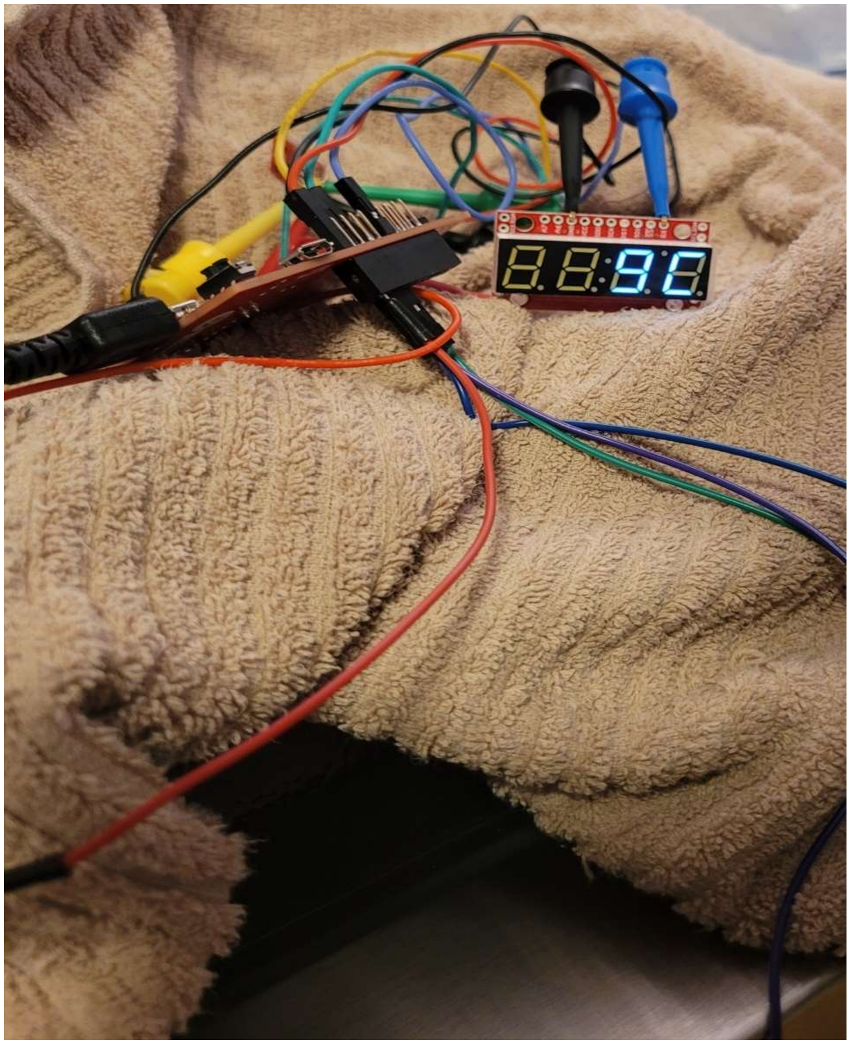


Figure 1.7: Final display output confirmation of correct temperature & leading zeros cleared

Analysis

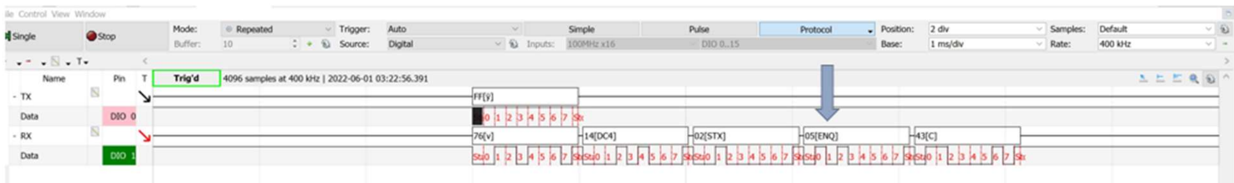
Following our test plan, we were able to verify that all hardware components are assembled correctly. We are then able to confirm our modified program worked as intended by seeing the LED display correctly increase or decrease its output decimal value in Celsius depending on whether we increased or decreased the ambient temperature near the sensor.

By using WAVEFORMS to run our real time temperature data through a logic analyzer we were able to confirm that our sensor was operating correctly and that our program was outputting the correct data to the LED display. While running the data through the analyzer we saw an immediate change in our waveform once we exposed the sensor to a temperature change.

The UART logical data allowed us to see the bit change happen in real time. Between ambient room and the increased temperature using a finger we saw the bit 03ETX change to 05ENQ. We also so a similar response with our I2C logical data. The bit h17 at ambient room changed to h20 with the sensor exposed to increased heat.

UART Logical Data

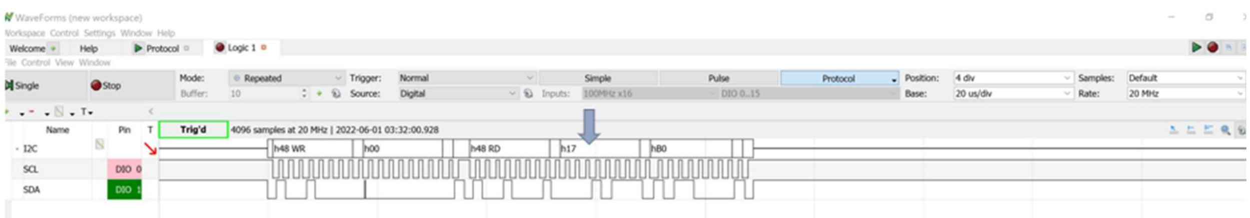
Ambient room temp data



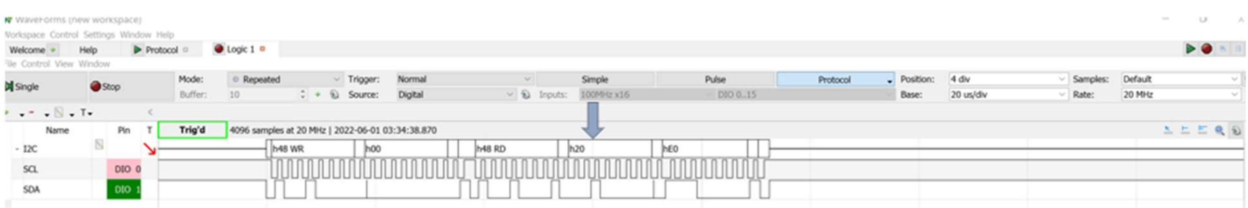
Increased temp using finger

Figure 2.1: UART data from AD2 before & after temperature increase

I2C Logical Data



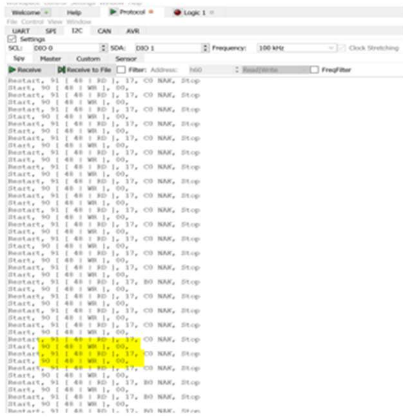
Ambient Room Temp Data



Increased Temp Using Finger

Figure 2.2: I²C data from AD2 before & after temperature increase

Ambient Room Temp Data



Increased Temp using finger

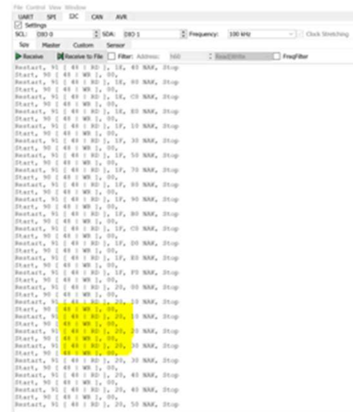


Figure 2.3: I²C Logical data from AD2 – Ambien v Increased temperature

Conclusion

By successfully completing the build and programming of our temperate measurement display system, we are more familiar with the hardware assembly of the TIVA board, and the programming of the TIVA board by way of software like the KEIL IDE. Between building and analyzing the data we also learned the importance of knowing which ports are your data input and output and being able to go back and test your connections if you aren't getting the correct data response. This project also allowed us to use WAVEFORMS to verify our real time temperature data with a logic analyzer. This allowed us to see the actual change in the sensor data all the way down to the actual bit change in real time. Lastly, we achieved a better understanding of how important both hardware and software are for a microcontroller, UART, and I^{2}C to transmit and receive data correctly.