High Precision Temperature Controller for Infrared Systems Development

Authors: Ashley Desiongco, Stacy Glass, Martin Trang and Cara Waterbury

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816

***Abstract* – The following paper details the design methodology and motivations for the High Precision Temperature Controller for Infrared Systems Development. The device described is customized to control Infrared Systems Development's blackbody heat sources via a nested PID algorithm. This controller is capable of interfacing with two different heat sources, which vary in temperature range and temperature sensor types. Users are able to determine a temperature set point on the controller's interactive touch panel display, and also are in constant view of the current heater temperature. In addition to the touch panel user interface, users are able to communicate with the controller via RS232 communication protocol. Using the RS232 communications, users are able to monitor the heater status remotely through a computer user interface which has also been custom developed for Infrared Systems Development.**

***Index Terms* – Master-slave, pulse width modulation.**

I. INTRODUCTION

The High Precision Temperature Controller Project is a senior design project sponsored by Infrared Systems Development. Infrared Systems Development is a privately owned company located in Winter Park, Florida. The company's specialty is mainly in development of black body heat sources used for scientific research and military applications. Infrared Systems Development currently uses a commercial, off-the-shelf controller to control their black body sources, but was intending to develop their own. The company approached one of their interns, Martin Trang, to come up with a design for the controller and to implement it with new hardware.

The specifications for the controller were to read the temperature from several types of sensors and send it to the display. The readings had to stabilize within +/- .5 degrees Celsius, and the display had to show the temperature at all times. The display must also take user inputs to control the set point of the heater. Our controller must have the heater reach its set point without much overshoot, and to reach it in a reasonable amount of time. This design also had to be cautious of the ramping up process to avoid damage to the heaters. A computer user interface was also added to be able to monitor the heaters from a remote location, which includes RS232 communications to connect the controller to a network.

This project was a huge undertaking, especially for a group of full time students with part time jobs. The workload was split into 4 major subsystems: analog hardware, digital hardware, display and PID/software. The analog hardware encompasses all temperature sensors, amplifiers, voltage references and analog to digital converters. The digital hardware includes the microcontrollers and all communication devices. The display section entails integrating the touch panel display and rendering all the graphics used for our project. Finally, the PID/software section encompasses writing the PID algorithm and tuning the parameters. All group members were responsible for developing code for each subsystem and integrating them with other systems.

A. Heater Types

Infrared Systems' black body sources come in two different configurations: cavity and extended area. The cavity sources, shown in Fig. 1, use either type S or type T thermocouples to obtain its temperature readings and the heat is confined to a circular cavity in the center of the source. The range of these cavity sources are from 0 – 1300 degrees Celsius. The extended area source, shown in Fig. 2, use platinum RTDs to obtain its readings, and the heat is spread across a flat surface. Since the extended area sources have distributed heat, the device uses four RTDs placed in different locations across the surface of the heater. The range of these sources is from -30 to 700 degrees Celsius; therefore, bipolar configurations for the rest of our analog hardware were considered when designing the subsystem.

Fig. 1 Cavity heat source by Infrared Systems Development.

Fig. 2 Extended area heat source by Infrared Systems Development.

II. Analog Hardware

The analog hardware subsystem encompasses all temperature sensors, amplification devices, voltage references, solid state relays and analog to digital conversions. The subsystem must also account for the interface between the heater and the controller, as well as the chassis. The requirements of the analog hardware subsystem are that the temperature readings must be accurate within .5 degrees Celsius and must be compatible with both heater types listed in section A of the introduction. Shown in Fig. 3 is a simple block diagram of the analog hardware.

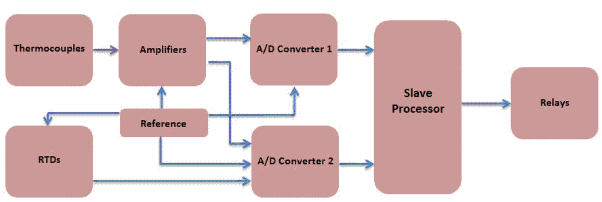


Fig. 3 Simple block diagram of analog hardware.

A. Heater-controller Interface

Mounted on the chassis of the controller is a 12 pin circular, female connector. The input pins to the controller are 4 differential sensor readings (8 pins) and the output pins going into the heater are a ground pin and 2 PWM pins. The input pins could contain either thermocouple or RTD readings; therefore, precautions were taken to use gold plated pins and thermocouple extension wire to avoid creating unwanted junctions for the thermocouple readings. The chassis is a piece of hardware that was manufactured by Infrared Systems Development to the desired size of the controller. The completed chassis is shown below in Fig.4. The touch panel chosen for this project fits perfectly in the allotted space the chassis provides for a display. Fig. 5 shows the front panel of the chassis with the display installed. Because of the slim and compact design of the display, which is elaborated on further in the Display section, the entire hardware contents of this controller is much smaller than the commercial off-the-shelf controller that Infrared Systems Development Corporation currently uses.

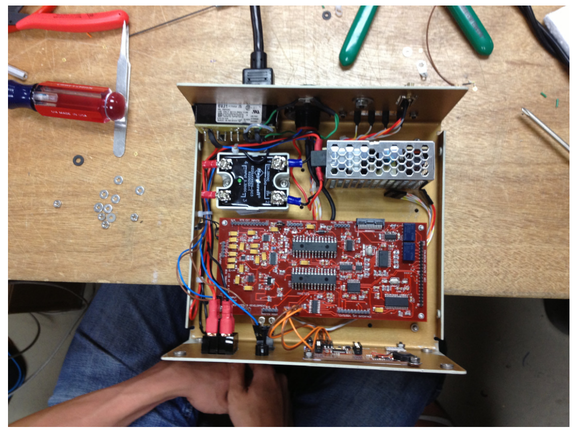


Fig. 4: Completed chassis

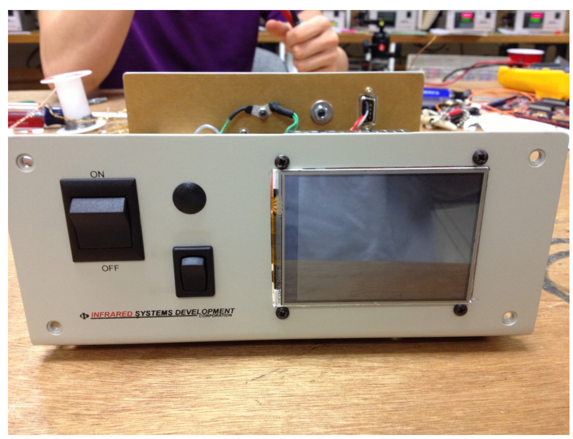


Fig. 5 Chassis with display installed

B. Analog to Digital Conversions

There are 2 analog to digital converters integrated into this device. The first is the AD7797, by Analog Devices, which is a 24-bit, sigma-delta converter with enough input channels for 1 differential input. This converter is used for the thermocouple reading from the cavity source heaters. The other is the AD7718, also by Analog Devices, which is also a 24-bit, sigma-delta converter, but has availability for 5 differential inputs. This converter will be used for a secondary thermocouple reading, the 4 RTD readings from the extended area sources and the cold junction compensation RTD. These multiple inputs are multiplexed by the internal 10:2 multiplexer on the AD7718 and then passed through a programmable gain amplifier before reaching the logic of the converter. Both the AD7797 and AD7718 have an analog input range of +/- 300 mV. Considering that the thermocouple readings are in the mV range, amplification is required so that there is no loss in data on the input of the converters. A differential op amp, the AD8476 by Analog Devices, is included in the design in order to maintain a true differential reading and also boost the signal to meet the input requirements of the analog to digital converters. The op amp and both analog to digital converters are all referenced to a 2.5 V precision reference voltage, the ADR431 by Analog Devices.

Since the controller must interpret data from thermocouples, the design must take into consideration cold junction compensation. This is realized in the design by an RTD mounted onto the chassis. The microcontroller code will subtract the voltage value of the RTD reading from the voltage value of the thermocouple reading in order to negate any ambient temperature differences.

The differential operational amplifier used in this design is the AD8476 by Analog Devices, which was mentioned previously. This device has a common voltage input that centers the differential reading about that common voltage. This design implements this common voltage by using the reference provided by the 2.5 V part, the ADR431 by Analog Devices.

C. Controller Output

The output of this device is a PWM pulse train outputted from the slave PIC32MX150F128B and inputted to the 2 solid state relays mounted on the chassis. These relays control the AC power into the heater, and can turn off when the heater needs to cool down. This PWM control is implemented in the software for the slave microcontroller.

III. DIGITAL HARDWARE

The digital hardware subsystem encompasses the microcontroller, all communication devices and any multiplexers needed for the communication devices. The requirements are that there is a minimum of RS232 communication and that the microprocessor is capable of the following:

-Processing 24-bit, signed, floating point math
-Communication with 4 UART devices
-Communication with 4 SPI devices
-Including enough chip select lines for all the peripheral devices
-Storing all of the thermocouple and RTD lookup tables.

The explanation for the capability of communication with 4 UART and 4 SPI devices is for future implementations of this controller. Eventually, Infrared Systems Development would like to have RS232 as well as Ethernet, USB and RS485 communications abilities with the controller. These desires were made known at the beginning of the design phase, and although there were not enough resources to implement them into this project's design, the hardware is present for future software development.

A. Microcontroller

The microcontroller used in the High Precision Temperature Controller design is the PIC32MX150F128B by Microchip. This microcontroller is a 32-bit processor in a 28 pin DIP chip. This model of the PIC32 features 2 SPI and 2 UART lines and allows full-featured ANSI-Compliant C. Since the design requires communication with 4 of each SPI and UART devices, a master-slave configuration implementing two PIC32 chips is utilized in the design. By using a general master microcontroller combined with a specialized slave microcontroller, the design lends itself for future development by having the ability to add additional slave devices that can perform tasks beyond blackbody temperature controller. The two microprocessor chips talk to each other via SPI interface, and each must be configured in their appropriate mode; the master in master mode, and the slave in slave mode. This configuration is realized through the device configuration registers.

The PIC32MX150F128B has 32K bytes of data memory, which stores all of the temperature tables required for the sensors. It also holds the calculations made by the PID algorithm to be referenced in the future when tuning the PID parameters. More details on these calculations are provided in the Software section of this paper.

The code written for this device is in C and edited and compiled by Microchip's MPLABX IDE. Initially, a Microchip development board was used to build and test on, and the PICKIT3 was used to debug and program the code once the PIC32 was placed on the PCB. A pin assignment table for the master and slave of the PIC32MX150F128B is shown in Tables 1 and 2

B. RS232

The part to realize the RS232 communication in this design is the MAX232 by Maxim Integrated Circuits. The

MAX232 is a dual driver/receiver although this design only utilizes one of each. This device is used to communicate between the controller and a computer, of which the user can monitor the current temperature or change the current set point via the computer user interface.

IV. Display

The display subsystem considers all the components and software utilized to realize a touch panel display with custom user menus. The requirements for the display are that the temperature must be displayed at all times and have a custom user menus. It must also fit in the existing chassis mentioned in the analog hardware section above. The High Precision Temperature Controller utilizes the µLCD32GFX by 4D-Systems, which is a slim 3.2 inch resistive touch panel display as shown in Fig. 6.

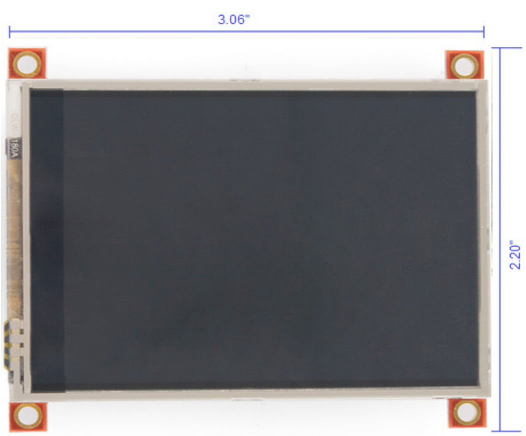


Fig. 6 uLCD32GFX display by 4D Systems

A. Display Features

The µLCD32GFX has a resolution of 480x272 and has physical dimensions of 3.06x2.64 inches, which meet the requirement to fit in the existing chassis. The device also has 15K bytes of flash memory for user code and 14K byes of SRAM for any variables. It also has the ability to read and write to an SD card, of which graphics, videos and audio may be stored and used to output to the display. This display communicates with the master microcontroller via a UART interface and a RESET line. To program the device, the µLCD32GFX has an easy 5 pin interface which includes:

-VCC
-TX
-RX
-GND
-RESET.

This feature alone is a marvel, considering many other displays of this nature require a 30+ pin interface. Another feature which significantly simplified the controller's design is the PICASO-GFX2 Processor which functions as a custom graphics controller. This processor includes all functionality of the code, including the high level commands needed to communicate to the display itself. Since the uLCD32GFX comes with this PICASO-GFX processor we will not need to have an external graphics controller or backlight led driver which simplifies our design.

B. Display Software

The software tools included with the purchase of the display included an IDE, PmmC loader, graphics composer and font tool. The font tool is used to generate the font shown in all of the menus. It is compatible with any Windows based fonts and converts the fonts into bitmap fonts. The PmmC loader allows the PICASO-GFX2 processor to load a PmmC file which contains all of the low level microcode information which defines the functionality of the device [1]. The graphics composer is used to compose the images, animations, and movie clips which can then be downloaded onto the micro-SD card. The IDE is used to program in proprietary 4DGL language, compile and download to the device. The IDE also provides another valuable feature called 4D-ViSi. 4D-ViSi is a software tool that is used to see instant results of a desired graphical layout for the display. It has variety of premade buttons, gauges, digits etc. that can be simply dragged and dropped onto the simulated display. When using the 4D-ViSi the workshop generates a base code for what we put on the simulated display. This features reduces the development time required for programming the 4D-ViSi.

One constraint of the µLCD32GFX was the limited program memory to 14400 bytes. This lead to a more modularized software design relying on smaller pointer based programming instead alternative, lengthier methods.

C. User Menus

Shown in Figures 7 and 8 are screen shots of the user menus. Fig. 7 shows the main screen with constant view of the heater temperature. In this screen, the "Change Set Point" button allows the user to continue to the screen in Fig. 8 to update the desired set point value. The user will use the up and down arrows the get the desired set point. Once the user presses the "Save" button they will be brought back to the screen in Fig. 7.

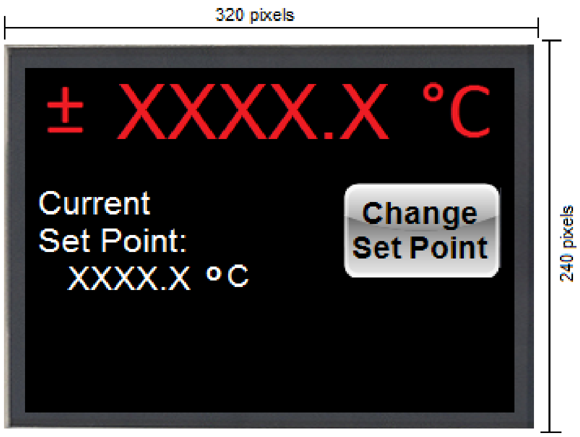


Fig. 7 Screenshot of main user screen

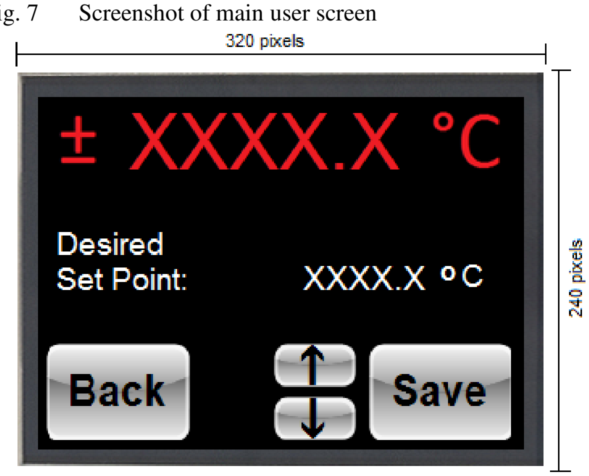


Fig. 8 Screen for changing the set point

V. Power

To power the High Precision Temperature Controller, 5V and 3.3V supplies are needed. The design accommodates these requirements by using a 5V power supply. The input to the LS25-5 power supply is 90-240V (AC) from a standard wall outlet, and the output is a 5V (DC) signal. The 5V line is heavily decoupled before reaching any analog VCC pins, and feeds directly into any digital VCC pins. We will need 5V to power the ADC, op-amp, reference, buffer, RS232, display and the USB. We will need 3.3V to power the Ethernet, microcontroller, 4:1 MUX and the ADC. For devices requiring 3.3V power, the design implements a linear drop-out regulator that steps the power down from 5 to 3.3V. The LT1129-3.3 is utilized for this purpose. Again, any analog VCC pin has a decoupling circuit before receiving any power and the digital pins are directly fed from the LDO. A simple power diagram showing the components that we will power is shown below in Fig. 9.

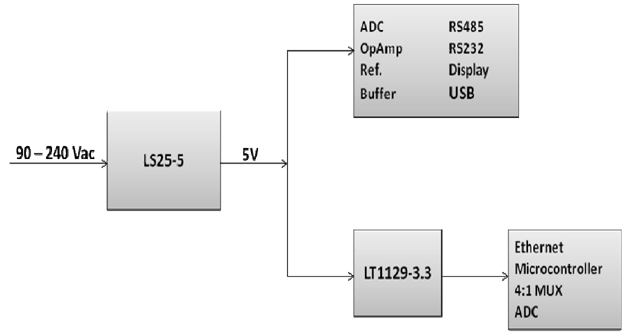


Fig. 9 Power diagram

VI. Software

A. PID Algorithm

The control method used for the device is an adaptation of a nested PID. The PID algorithm takes the difference between the set point of what the desired temperature is and the current temperature that is read through the thermocouple to determine the error. From here the error is read into the closed control and through one of the two ranges within the nested loops displayed in Figure 10.

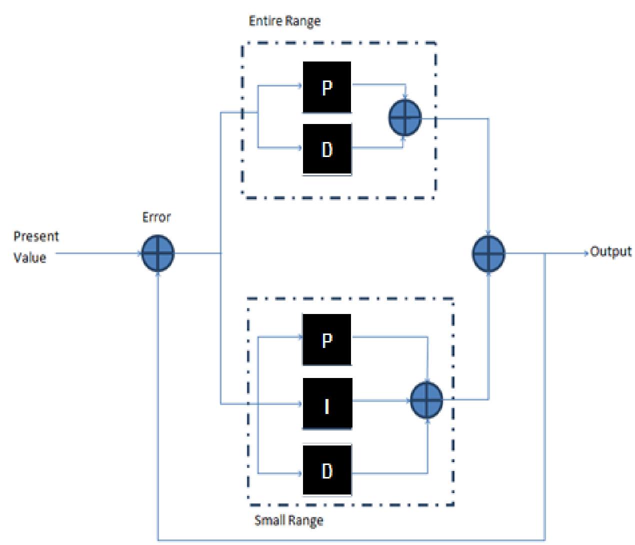


Fig. 10: Nested PID

Through the nested PID For two variations of PID juxtaposed to reach our final control method. The first loop contains only the P and D parameters encompassing a vast temperature range in order to cut down on overshoot and zoom into the desired value. This loop will initially kick in when the device is first turned on and continue until it comes within 20% of the set point. This is done so that the Integral does not get saturated before the present value gets anywhere near the desired value. The P and D values will kick on the heater and begin the regulating process. Once it within the smaller range, the second loop consisting of all three PID parameters will kick in. Should the temperature deviate to high above or below outside of the 20% range it will kick back to the outer PD loop. It is a high hope that by combining multiple loops the two variations will work together to create a more efficient control process. This method will in turn making our device faster and more precise than older models.

B. Microcontroller Code

The code written for this device is very robust. It encompasses the following functions:

- Initializing each peripheral device
- Set up of SPI and UART communications
- Loading of temperature tables into memory
- Computing signed, floating point math to average temperatures and output to the display
- Communicating to the display, RS232 and A/D converters
- Delivering PWM output schema to solid state relays.

To compute the appropriate temperature values to output to the display, the software for this device utilizes several methods. From the A/D converter, the PIC32MX150F128B will utilize the 32 bit SPI buffer to clock-in the data from the ADC. Since the first byte of the 32 bit buffer contains the command, the remaining three bytes will contain the converted value. After acquiring the data, the value will be converted back to millivolts. To obtain the temperature from the millivolt reading, two look-up tables will be used, one for the Type S thermocouple and the other for the PT-100 RTD. Since the look-up tables have one degree C of resolution, linear interpolation will determine the intermediate temperature not directly listed in the look-up tables. As mentioned in the Analog Hardware Section, cold junction compensation will be used to deal with the dissimilar metal junction. Since the temperature of a Type S thermocouple is not accurately represented using linear models, the software must determine the temperature of the dissimilar metal junction by using the resistance of the PT-100 RTD. After acquiring the temperature, it must be converted it to a voltage reading, according to the Type S thermocouple table. The voltage from this dissimilar metal junction will be subtracted from the perceived thermocouple voltage. After performing these math functions, the final voltage reading will be converted to a temperature. Lastly, the software continuously averages the temperature to eliminate as much possibility for error as seen fit.

To communicate to the display, the controller's software must break up the temperature data into three packets of eight bits before the display can correctly format it onto the screen. The first packet that is sent contains the value in the tenths and ones place, with the tenths place occupying the upper four bits. The second packet that is sent contains the tens and hundreds place, with the tens place occupying the upper four bits. The final packet that is sent contains the thousands place and the sign information. The upper four bits contains the thousands place. The lower four bits of the third packet contains the sign information and a check bit. For a negative value, the lower four bits will equal 0xE. For a positive value, the lower four bits will equal 0x6. The fourth bit of the third packet will be used a check bit. This value must be equal to one, otherwise the display will reject all three packets. To achieve this, the software encodes some formatting data along with the actual data into these packets. The sign of the temperature is also encoded into these packets.

C. Remote User Interface

The remote user interface is utilized primarily to monitor the current heater temperature. This interface is realized by communications through the RS232 port on the controller. In the future, Infrared Systems Development plans to develop this interface further to allow for adjustment of PID parameters through the administrator menu and change the current set point. This interface is designed in Lab View and the GUI is shown below in Fig. 11.

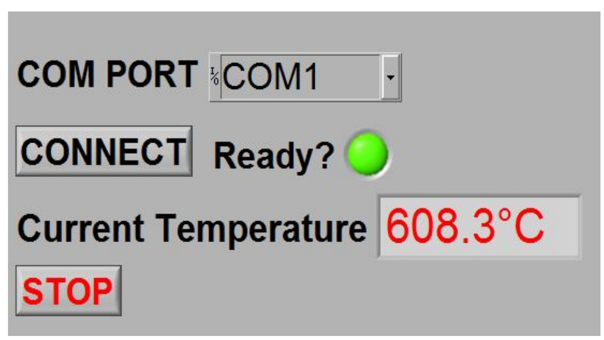


Fig. 11 Remote User Interface GUI

VII. Conclusion

This project helped each group member become more familiar with designing and implementing a real world engineering project. It also provided the group with insight on working in a group to produce a final product. This will help each member of the group when going out into their respective jobs.

This project provided the group with hands-on experience with circuitry, hardware, programming and communication between all of the components. In addition to adding experience to the group's knowledge, this project also provided the sponsor, Infrared Systems Development Corporation, with an initial prototype to replace the existing commercial off-the-shelf temperature controller.

For the continual development of this product, all the communications mentioned in the Digital Hardware section will be realized through software. Also, a better computer user interface will be developed to have more capabilities and control of the controller remotely.

Pin #	**Port Name**	**Function**
1	MCLR	Master clear
2	RPA0	Ready – 7797
3	RPA1 [SDO2]	Master to slave
4	RPB0 [OC3]	PWM2
5	RPB1 [SDO1]	ADC
6	RDY7718	Ready – 7718
7	NC	No Connect
8	VSS	Ground
9	RPA2 [SDI2]	Master to slave
10	RPA3 [SS2]	Master to slave
11	NC	No Connect
12	RPA4	Addr line 2
13	VDD	3.3V
14	RPB5 [SDI1]	ADC
15	RPB6 [T4CK]	AC input
16	RPB7 [OC1]	PWM1
17	RPB8	Addr line 1
18	NC	No Connect
19	VSS	Ground
20	VCAP	1.91V
21	PGED2	Debug 1
22	PGEC2	Debug 2
23	NC	No Connect
24	NC	No Connect
25	RPB14 [SCK1]	ADC
26	RPB15 [SCK2]	M-S Clock
27	AVSS	AREF- [AGND]
28	AVDD	AREF+ [3.3V]

Table 1 Pin assignments for slave PIC32MX150F128B.

Pin #	**Port Name**	**Function**
1	MCLR	Master clear
2	RPA0 [U1TX]	Display transmit line
3	RPA1 [SD02M-S]	Master-slave
4	RPB0 [SS2]	Slave select 2
5	RPB1 [SDO1]	Ethernet
6	UART Select 1	UART select line 1
7	UART Select 2	UART select line 2
8	VSS	Ground
9	RPA2 (U1RX)	Display receive line
10	RPA3 [SPISEL1]	SPI select - ethernet
11	RPB4	Reset display
12	RPA4 [SDI2]	Master-slave
13	VDD	3.3V
14	RPB5 [SDI1]	Ethernet
15	RPB6 [SPISEL3]	SPI select – ethernet
16	RPB7 [SS1]	Slave select – ethernet
17	RPB8 [U2RX]	COMM Receive line
18	RPB9 [U2TX]	COMM transmit line
19	VSS	Ground
20	VCAP	1.91V
21	PGED2	Programming pin
22	PGEC2	Programming pin
23	RB12	Driver, receive – MAX481
24	RPB13	SPI select line 2
25	SCK 1	Ethernet clock
26	SCK2	Master-slave clock
27	AVSS	AREF- [AGND]
28	AVDD	AREF+ [3.3V]

Table 2 Pin assignments for master PIC32MX150F128B.

VIII. Test/Verification

To verify the optimal operation of the high precision temperature controller, several test procedures were done to ensure that all subsystems were performing as expected.

For the analog subsystem, a high precision millivolt reference source was used to simulate the thermocouple that was sent to the AD7797 24 bit Sigma-Delta ADC. A high precision resistor was used to test the operation of the CJC ADC (AD7718 24 bit Sigma-Delta).

For the digital subsystems, various test values served as place holders for the outputs of the various ADC. These values originated from the Slave PIC and were sent to the Master then to the Display using SPI and UART, respectively. To ensure that all the math functions, data encoding, and data encoding was correct, the desired display output was determined and was verified to appear on the display after running the code. The computer interface was tested using a LABVIEW module that would send information to the Master PIC and would wait for an ACK of the same command.

During the manufacturing processes, several steps were done to ensure that the circuit was implemented per the PCB design. After hand soldering each individual 1206 resistor, 1206 capacitor, 6416 capacitor, 16-TSSOP, 8-MSOP, 16-SOIC, 28-SOIC, 8-SOIC, SOT-223, 28 DIP to SPDIP, 0.1 inch pin headers, and 0.05 inch pin headers, continuity was checked between each of the terminals. By doing so, any troubleshooting occurred after attaching the individual surface mount components and any errors did not propagate.

BIOGRAPHY

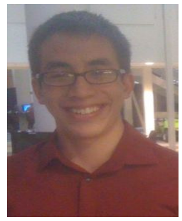
Ashley Desiongco

Ashley will be graduating from UCF with a Bachelor's degree in Electrical Engineering. She is graduating with 3 internships under her belt, and has accepted a position with Microsoft in Seattle, WA.

Stacy Glass

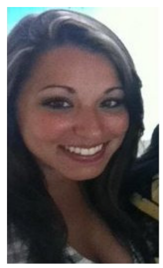
Stacy will be graduating from UCF with a Bachelor's degree in Electrical Engineering. She is currently interning at L3 Communications-Coleman Aerospace, where she plans to continue full time as an Electrical Engineer after graduation.

Martin Trang



Martin will be graduating from UCF with a Bachelor's degree in Electrical Engineering and a pristine 4.0 GPA. He currently interns at Infrared Systems Development, where he has accepted a full time position as an Electrical Engineer.

Cara Waterbury



Cara will be graduating from UCF with a Bachelor's degree in Electrical Engineering. She is graduating with 2 internships under her belt, and has accepted a position with Northrop Grumman in Orlando, FL.

ACKNOWLEDGEMENTS

We would like to acknowledge and thank the following people for their contributions to this project.
- Andy Duran CEO, Infrared Systems Development
- Rob Waterbury, Alakai Defense Systems
- Dr. Samuel Richie, University of Central Florida
- Dr. Kalpathy Sundaram, University of Central Florida
- Dr. W. Linwood Jones, University of Central Florida
- Dr. Chung Yong Chan, University of Central Florida

REFERENCES

[1] Display data sheet [http://www.4dsystems.com.au/downloads/4DGL-Display-Modules/uLCD-32-PT(GFX)/Docs/uLCD-32PT-GFX-DS-rev8.pdf]