NSF Lab Furnace Control System

Team 47

Client & Advisor: Dr. Gary Tuttle

Nick Brylski, Jeremy Hartl, Adam Matthews,
Christopher Pohlen, Lingkai Lang

Table of Contents

Functional Requirements

1. Must be able to view current values of temperatures and gas flow, on two separate temperature controllers for a total of 4 furnaces with 3 zones per furnace and variable number of gases per furnace.
2. Must be able to view and change setpoints for the same fields.
3. Must be able to use the built in profiling functions on the temperature controllers.
4. Must display all this data in a succinct format for easy viewing.

Non-Functional Requirements

- The Graphical User Interface (GUI) must be able to display information in a way conducive to use.
- Must simplify the process to make changes to settings and temperature values
- Improve the precision of entering gas flow rates
- Help prevent errors

Use Cases

The NSF Lab Furnace Control System will be used primarily used student's projects, class work, and some research.

It will be used for a range of functionality, from simple value setting to more complex time based profiling.

Design Plan

Utilize RS232 commands to control Omega temperature controllers (OTC)

Implement ADC/DAC to control mass flow controllers (MFC) for gas flow

Build a GUI and API that combines all the important information and control systems of the mass flow controllers and temperature controllers.

Use a microcontroller to act as a gateway between the various pieces.

Package the code into an executable for simple usage.

Constraints, Trade-offs

Garbage data is sent by the temperature controllers if a set command is sent

Unable to test on the actual furnaces and with mass flow controllers.

Mass flow controllers running off 5V inputs

Technologies Used

Microcontroller: Arduino Mega

PCB: Mega Shield

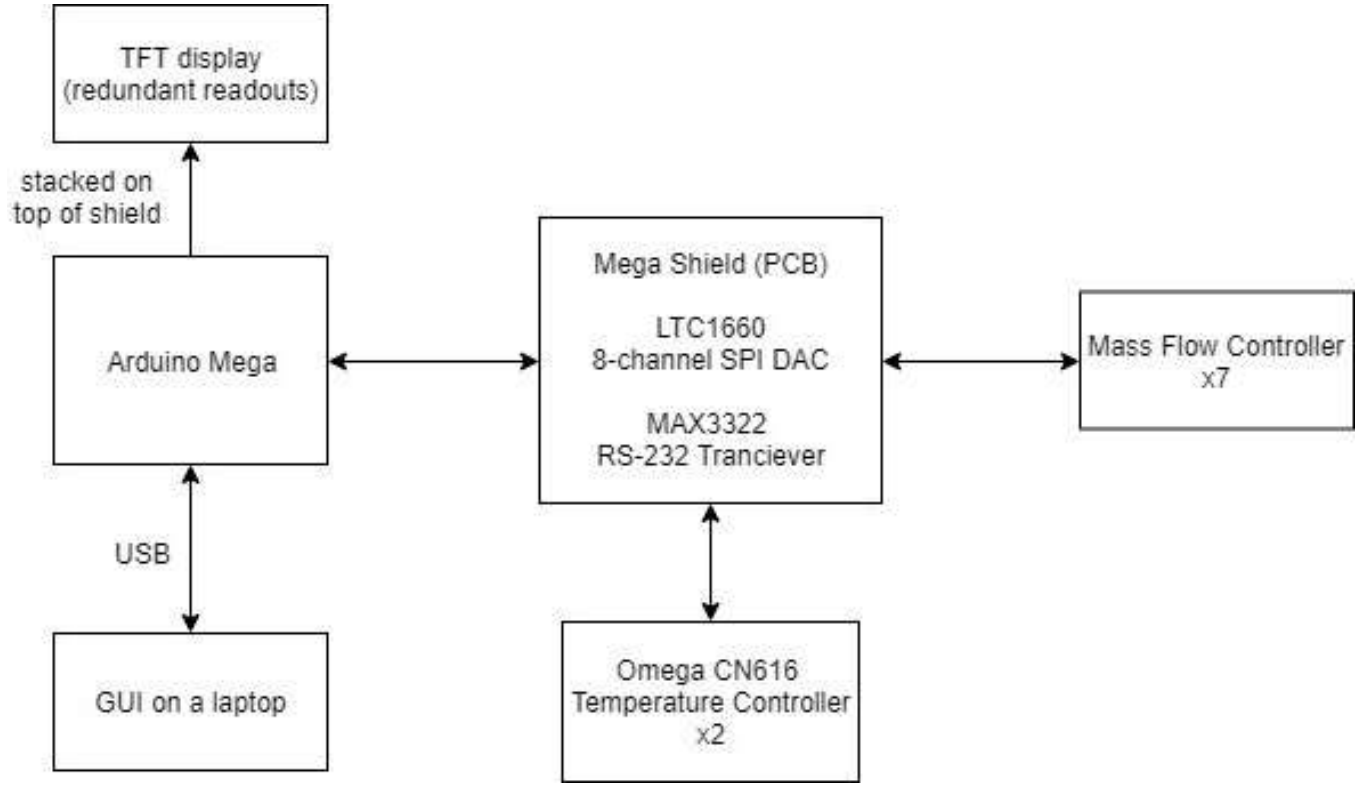
ADC: Arduino Mega 10-bit 16-channel

DAC: LTC1660 SPI programmable 10-bit 8 channel

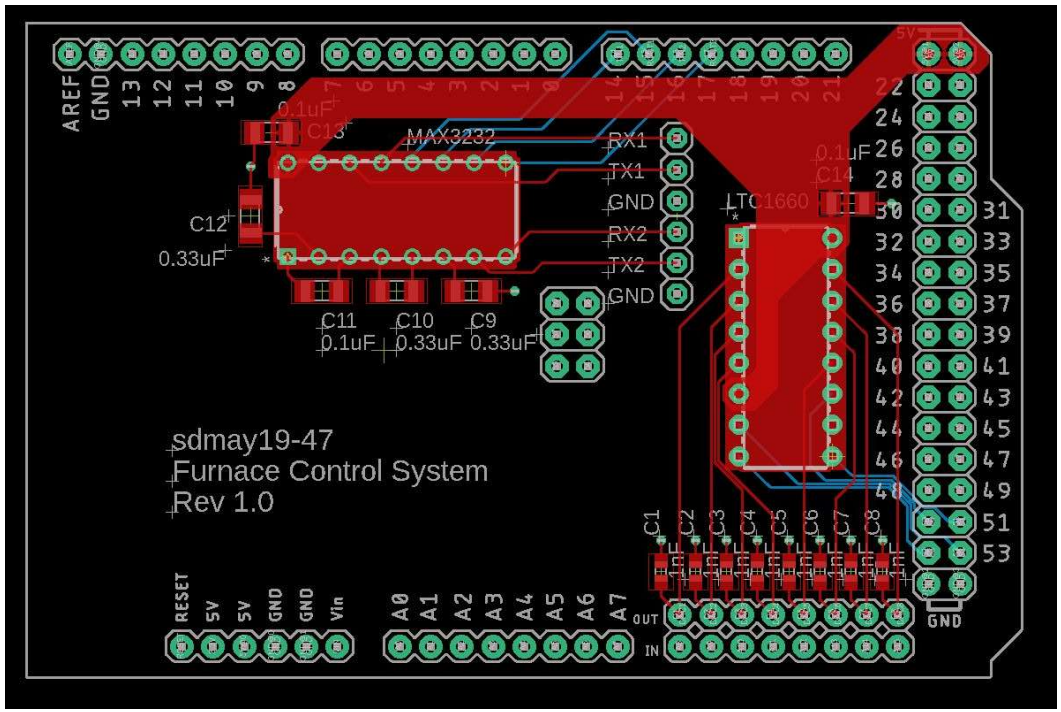
GUI: Python(TKinter library)) and pyinstaller for packaging into an executable

API: Python

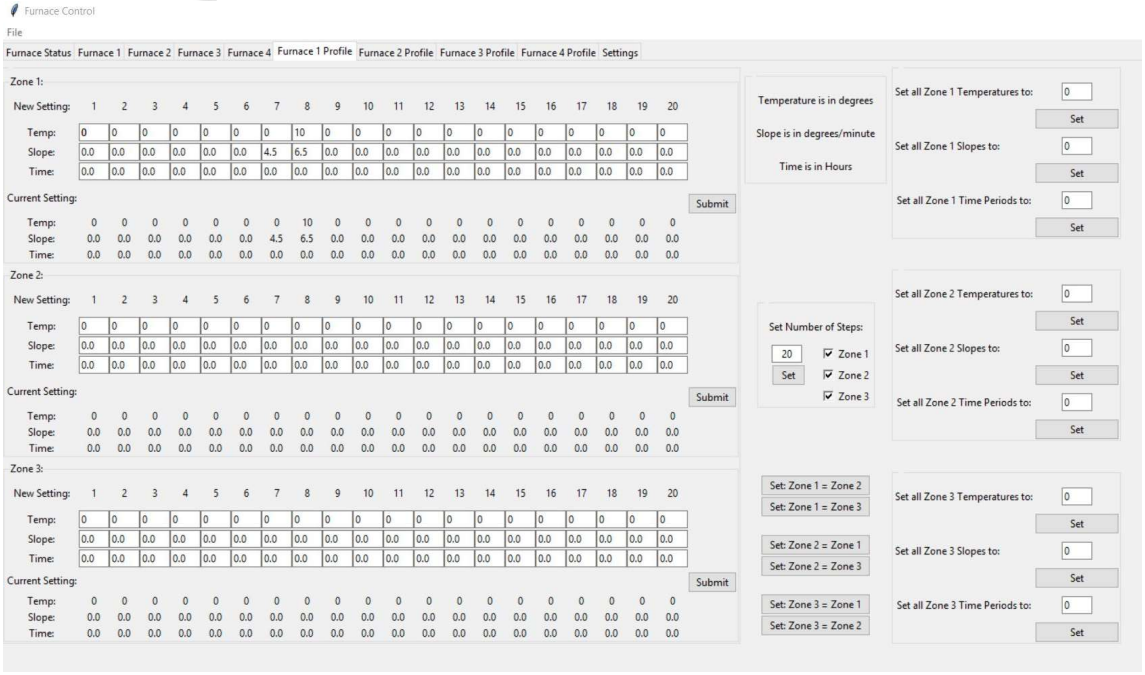
Block Diagram



PCB Layout

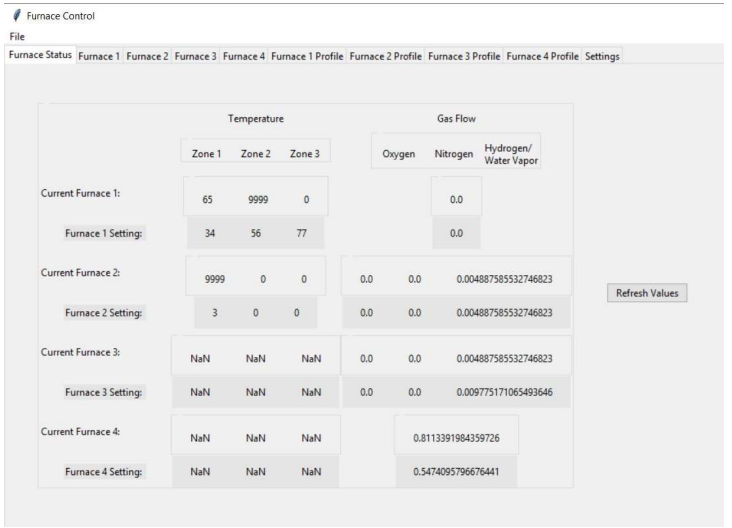

AREF
GND
13
12
11
10
9
8
0.1uF
C13
MAX3232
C12
0.33uF
C11
0.1uF
C10
0.33uF
C9
0.33uF
RX1
TX1
GND
RX2
TX2
GND
LTC1660
0.1uF
C14
5V
22
24
26
28
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
51
53
GND
C1
C2
C3
C4
C5
C6
C7
C8
1nF
sdmay19-47
Furnace Control System
Rev 1.0
RESET
5V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
A6
A7
OUT
IN

GUI Snapshots



Profiles Setting Page

Homescreen: Shows Settings and Current Read Values



Rationale

Microcontroller/ADC: Arduino's are relatively cheap and Professor Tuttle already had some materials for them

DAC: A reprogrammable DAC makes testing and development easier

API: Python would be able to hook into both the GUI and the Arduino easily

GUI: Python comes with Tkinter built in and pyinstaller was compared with other solutions and was deemed both easy to use and effective.

Test Plans

Hardware:

- DAC
- Separate components after soldering
- (note) Future testing required with the MFC and actual working furnace

Software:

- The GUI is mainly tested as if from a user perspective with a checklist of functionality to work through.
- The API was tested through a number of automated test cases in addition to manual testing with the GUI.

Interface Testing

The checklist for the GUI involves:

- Making sure the status screen remains updated despite changes
- Checking its ability to set values
- Change settings
- Prevent expected kinds of misuse (i.e letters instead of numbers, too large of inputs)

Integration Testing

We made a set of test cases of messages and data that needs to be sent back and forth from various pieces of the project.

Of particular importance is the microcontroller’s ability to pass information between all pieces of the project without getting anything confused.

The always-on display and the GUI should be able to make requests for updates from the controller without conflict

Roles and Responsibilities

Nick: Systems engineering, Arduino, DAC/ADC test

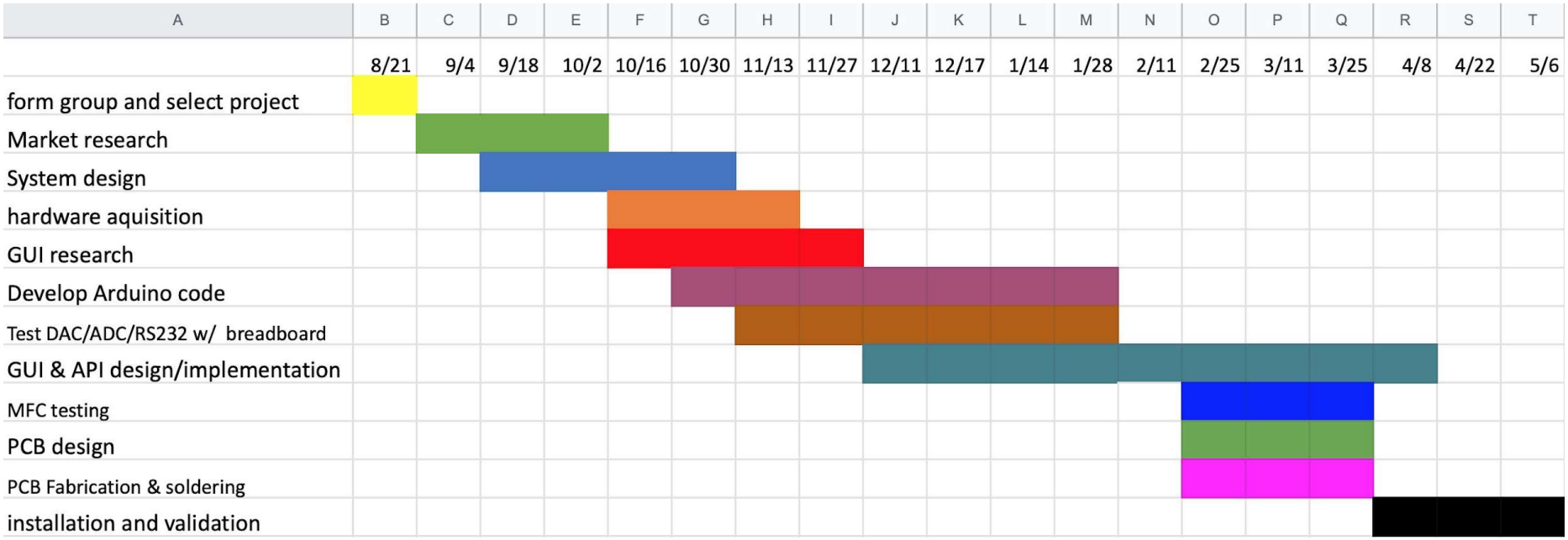
Jeremy: Hardware Engineer, Report Manager, GUI implementation

Adam: OTC and MFC Python API development, Arduino

Chris: GUI Design and Implementation

Kevin: OTC and DAC research and implementation. PCB design and soldering

Project Schedule Proposed



Project Schedule Actual

Risks and Mitigations

Risk that we've accounted for.

- Serial port can easily reconnect if disconnected with a retry error message
- The OTC might not have been able to communicate with the arduino fast enough. Because we were having trouble getting reliable comms even with a fast refresh rate so we reverted the GUI refresh rate to be user click based.

Lessons Learned

As a group we learned a lot about project management and taking individual initiative.

We learned a lot about the aspects of the project with which we had little prior experience. Examples would include Tkinter, serial communication, microcontroller usage, and reading datasheets.

Closing Remarks

We believe that we performed well on this project and delivered a rather robust and useful system.

We enjoyed being able to work with Professor Tuttle and with systems we have not used very much previously.

Future Directions

Add even more functionality present in the temperature controllers to the GUI. This might make rarely used but convenient functionality to become more available.

Another possible direction is replacing the older, temperature controllers with something more modern that has more built in performance.

List of References

1. *Lecture Number - 23: Oxidation*, Electronic Materials, Devices and Fabrication, Dr. S. Parasuraman, Department of Metallurgical and Materials Engineering: Indian Institute of Technology, Madras, 2012

Team Information

Nicholas Brylski: Senior in Electrical Engineering, nbrylski@iastate.edu

Jeremy Hartl: Senior in Electrical Engineering, jmhartl@iastate.edu

Lingkai Lang: Senior in Electrical Engineering, llang@iastate.edu

Adam Matthews: Senior in CprE and Spanish, ajrmatt@iastate.edu

Christopher Pohlen: Senior in CprE and Psych, cmpohlen@iastate.edu

Questions?