NSF Lab Furnace Control System

Team 47

Client & Advisor: Dr. Gary Tuttle

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

Problem Statement

What's wrong with the current NSF Furnace control system?

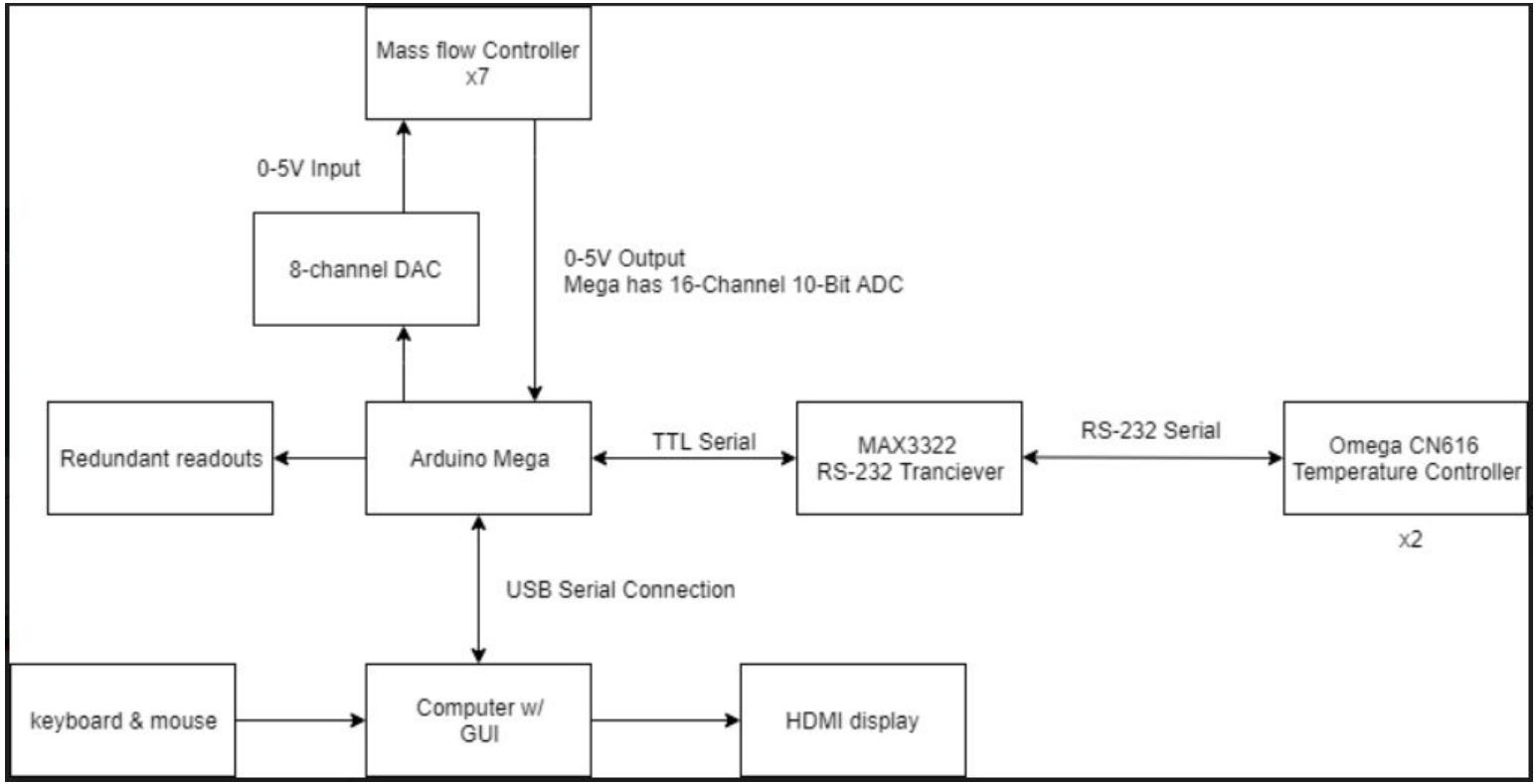
- Omega temperature controller requires long input sequences - takes 11+ inputs to change temperature setting
- Not a constant current temperature and set temperature readout
- Mass flow controllers (MFC) and Omega temperature controllers (OTC) are not controlled through the same system

Proposed Solution

Our Proposed solution has:

- One control system for both the MFCs and OTC
- Constant 24/7 information readouts
- Simple and quick setting adjustments
- Warnings and cautions
- Automated, timed controls - ramped temperatures etc.

Concept


Mass flow Controller
x7
0-5V Input
8-channel DAC
0-5V Output
Mega has 16-Channel 10-Bit ADC
Redundant readouts
Arduino Mega
TTL Serial
MAX3322
RS-232 Tranciever
RS-232 Serial
Omega CN616
Temperature Controller
x2
USB Serial Connection
keyboard & mouse
Computer w/
GUI
HDMI display

Functional Requirements

The system must be able to:

- control and display gas flow
- control and display furnace zone temperature
- automate temperature and gas flow changes according to a given profile
- run 24/7
- include a data logging feature to allow a temperature time series to be written out to the user (if requested)

Non-Functional Requirements

The system must be:

- Simple and intuitive to use
- Resistant to electronic failure
  - Reboot and reinitialize
- Robust
  - Compact and encased

Technical/Other Constraints/Considerations

- Must work with existing OTC & MFCs
- Needs to display constantly updated data
- Display should provide warnings and errors
- Must connect to every temperature and mass flow controllers simultaneously

Market Survey

OTC: Omega has existing Windows software (old)

MFC: There exist programs to control these devices

- Neither of these integrates both for creating process profiles

Semiconductor manufacture:

Advanced systems exist that are for factory level fabrication

Client wants basic, inexpensive integration with existing hardware

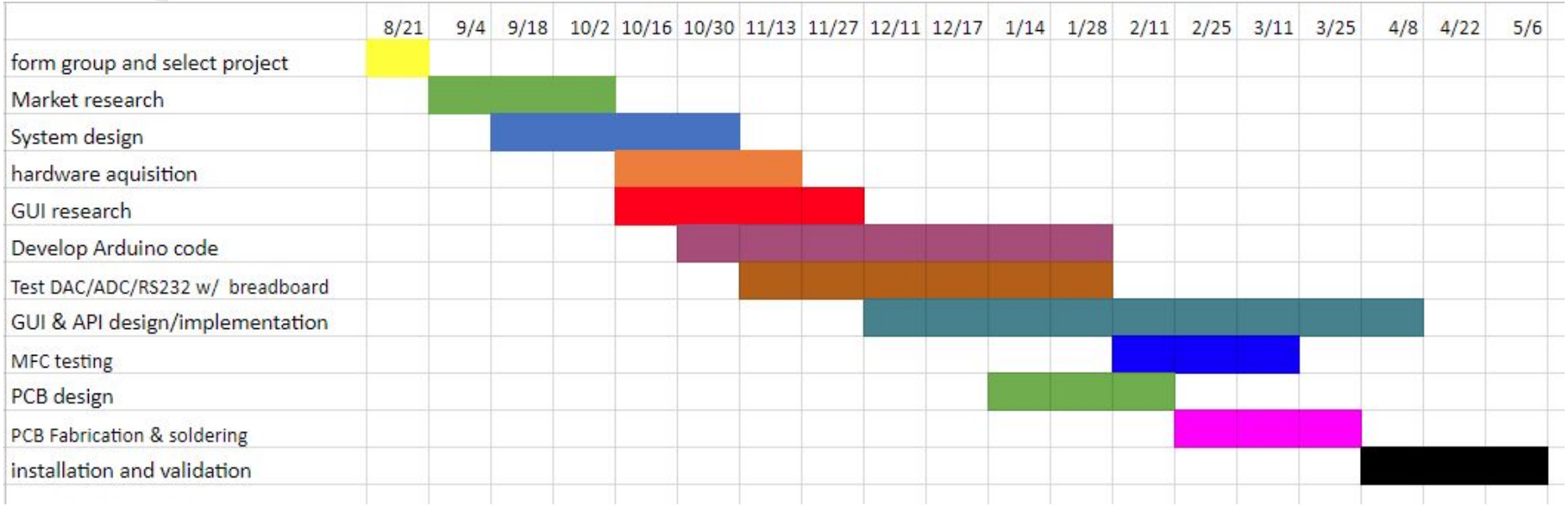
Potential Risks & Mitigation

- **Overheating the furnaces and the wafers**
  - Warnings when settings exceed limits or temperature readings exceed safety limitations
- **System failure**
  - OTC profile is managed by OTC
  - DAC is powered by the Arduino, so when the Arduino fails, the MFCs turn off
- **Imprecise settings and readings**
  - Extensive testing to ensure OTC temperature readouts are within 1% of actual furnace temperatures

Resource/Cost Estimate

Part	**cost**
Rpi Model 3	$35
Arduino Mega	$40
Arduino Shield PCB	$100
ICs (RS232 txrx and DAC)	free samples from TI and ADI
PCB connectors	$20
Display screen	$100
Total	**$295**

Project Milestones and Schedule



Functional Decomposition

Major Portions of the project include:

- The temperature controls and wiring
- DAC and gas flow controllers
- Microcontroller
- API development
- GUI and controls

Technological Platforms Used

Using Python for GUI development

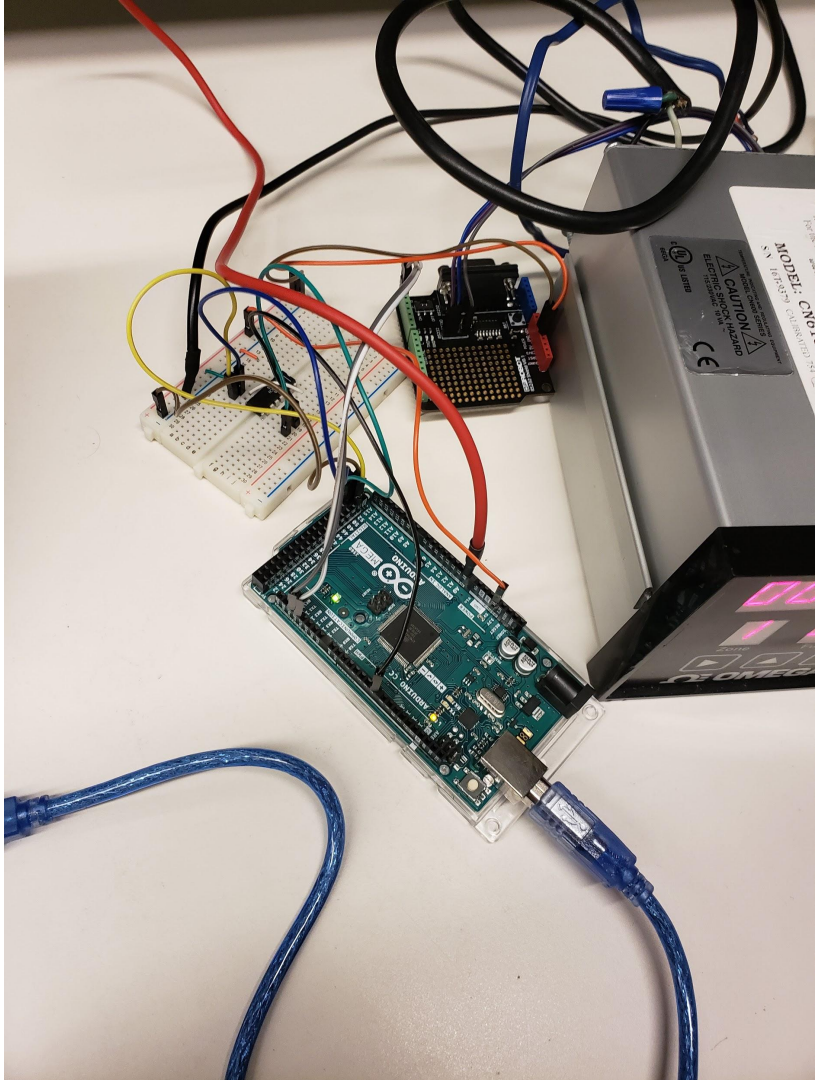
Using Arduino Mega for the Microcontroller

Omega Temperature Controllers

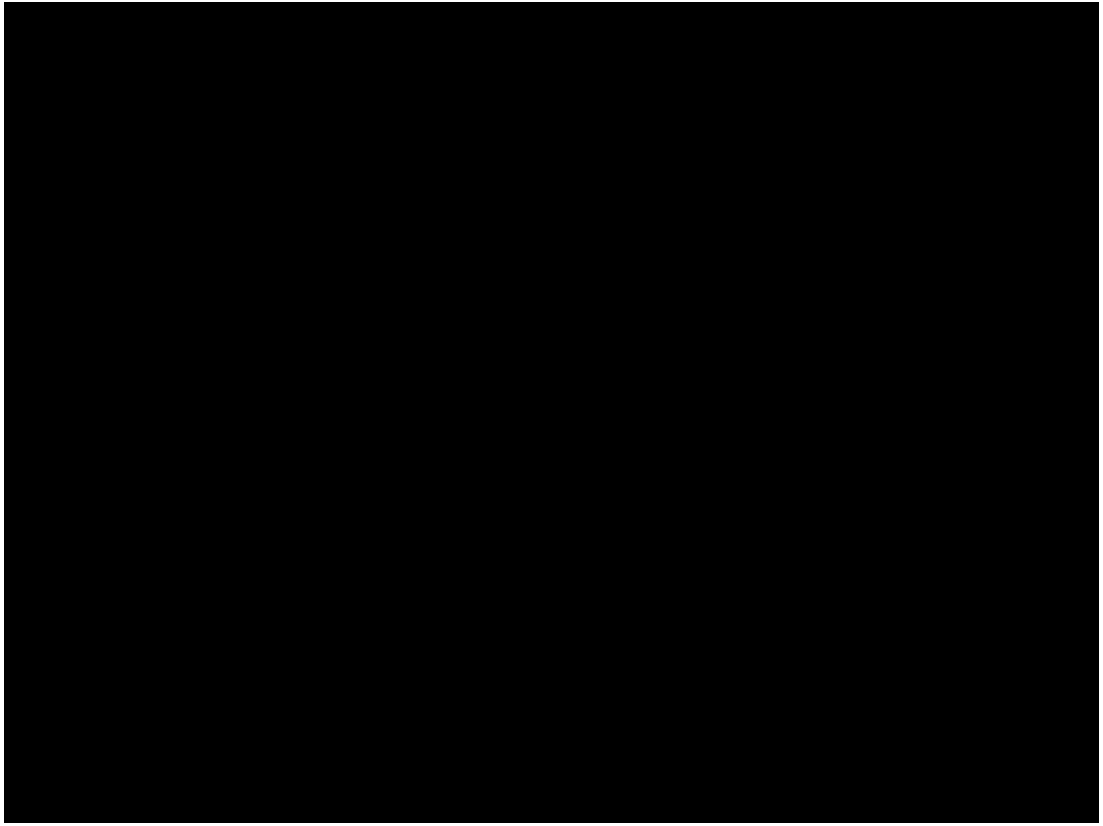
Soldered Circuit Board with DAC to Gas Flow Controllers

Test Setup

- Arduino IDE serial monitor-> Mega
- Mega-> RS232 transceiver -> temp controller
- Mega-> LTC1660 DAC -> DMM
- 0-5V Power supply -> analogread pins on Mega



DAC Test



DAC Implementation

```
val = userVal*1023/5 ;
val = (channel << 12) | (val << 2) ;


void dacWrite(unsigned int val) {

  // take the SS pin low to select the chip:
  digitalWrite(slaveSelectPin, LOW);
  delay(10);
  //  send in the address and value via SPI:
  SPI.transfer16(val);
  delay(10);
  // take the SS pin high to de-select the chip:
  digitalWrite(slaveSelectPin, HIGH);

}
```

Table 1b. LTC1660 Input Word

A3	A2	A1	A0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X1	X0
ADDRESS/CONTROL				INPUT CODE										DON'T CARE	

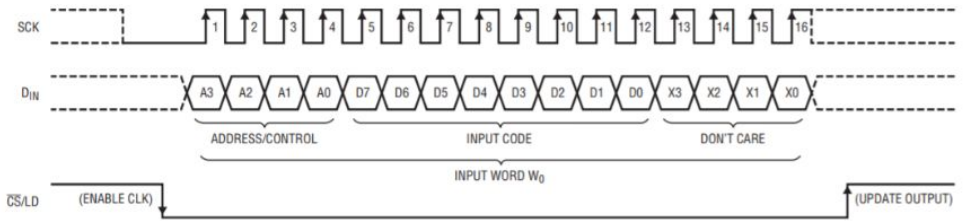
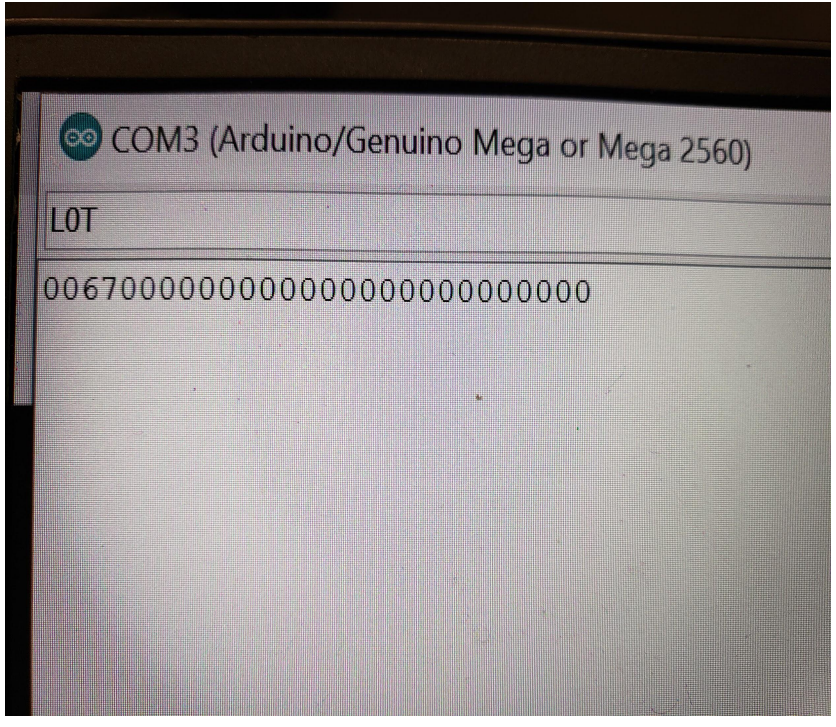
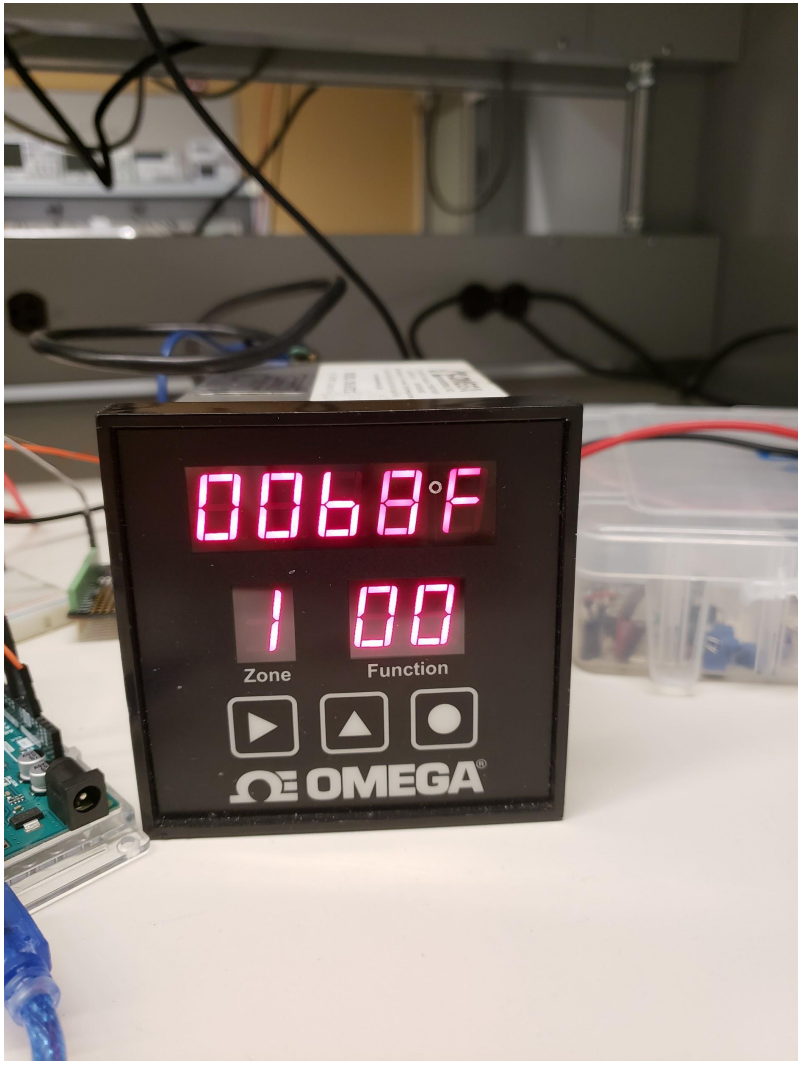


Figure 3: LTC1660 SPI DAC timing diagram

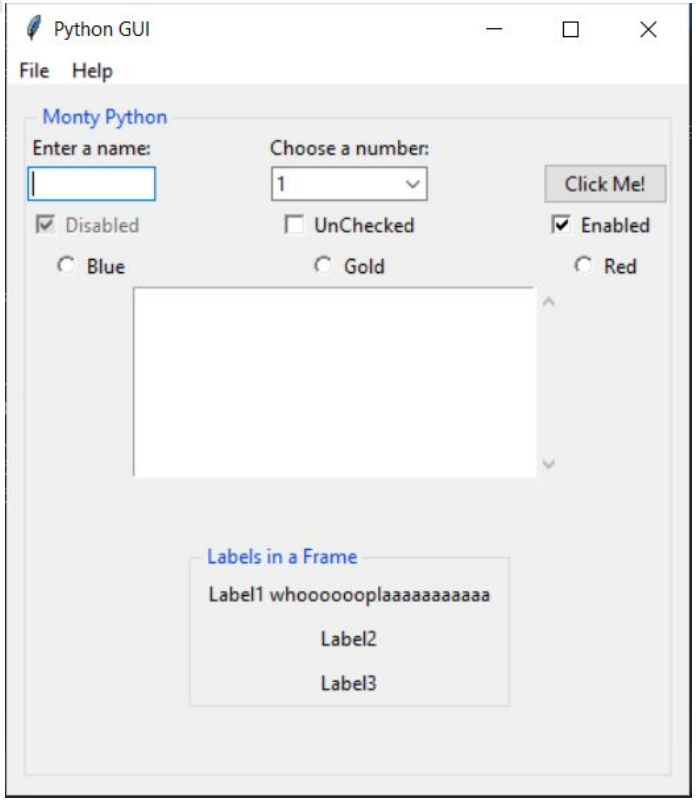
ADC Test

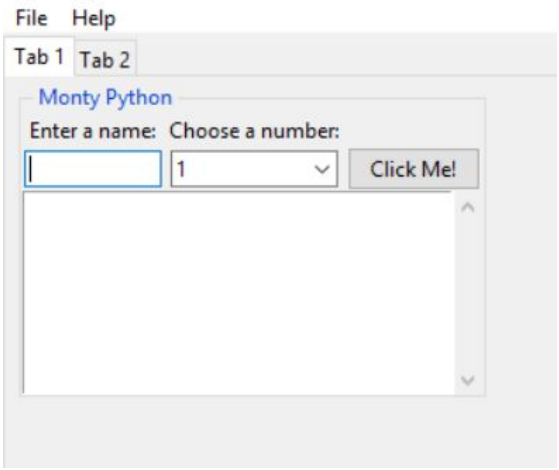
RS232 Test

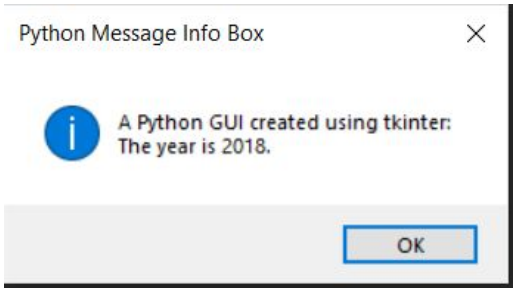

COM3 (Arduino/Genuino Mega or Mega 2560)
L0T
0067000000000000000000000000


0068°F
1
Zone
00
Function
OMEGA

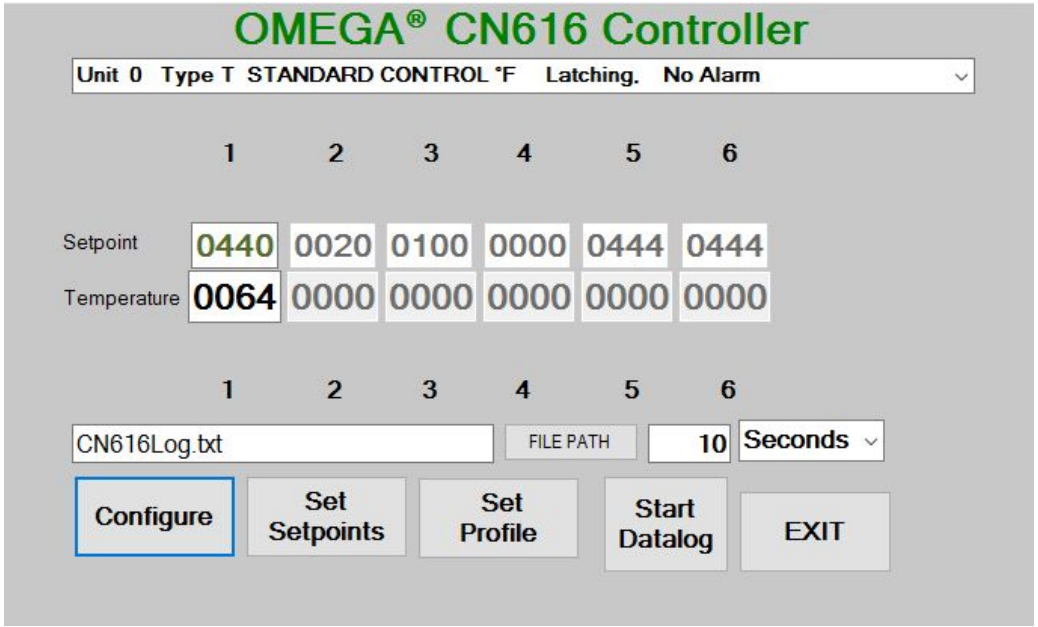
Prototype Implementations: GUI







Detailed Design



Current Project Status

We have gone past several milestones.

- Hardware, RS232 communication, most of the GUI and other research is done
- Have some examples of basic GUI building blocks.
- Basic Python API for data communication with OTC
- DAC and ADC are implemented

Plan for Next Semester

Design, fabricate, & fill PCB

GUI prototyping will begin in earnest

Building more advanced and robust API’s

Hardware testing

Connect to the furnace system

Task Responsibility

Nick: Systems engineering, Mega programming, DAC/ADC test

Jeremy: Hardware Engineer, Report Manager, GUI implementation

Adam: OTC and MFC Python API development

Chris: GUI Design

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

Questions?