

# NSF Lab Furnace Control System (SDMAY19-47)



By: Nick Brylski, Adam Matthews, Christopher Pohlen, Jeremy Hartl and Lingkai Lang  
 Client/Advisor: Professor Gary Tuttle, Iowa State University

## Summary

The NSF furnace is used in EE 432: Semiconductor Fabrication to perform doping and oxidations on silicon wafers.

The current method for controlling the temperature of the furnaces is unintuitive and prone to error.

Our project is a hardware & software solution that makes controlling the furnace easier, while also allowing for control of the mass-flow controllers in the system.

## System Implementation

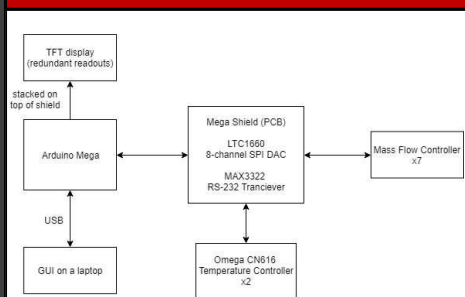
- Python GUI running in windows
- GUI using Tkinter framework
- GUI & API were packaged into an executable using PyInstaller.
- GUI talks with arduino stack via USB
- Arduino handles hardware interfacing
  - SPI DAC -> MFC
  - RS-232 transceiver -> OTC
- Shield sits on top of Arduino Mega
- Redundant display on top of shield

<- OTC



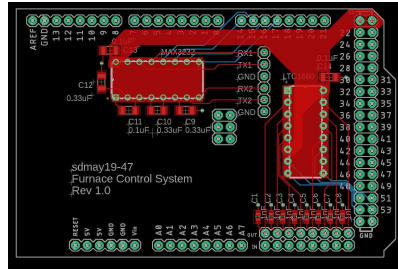
<- MFC

## Concept Diagram

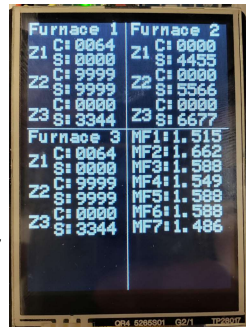


## Work Completed

- **Arduino PCB shield design and soldering of parts using Eagle**

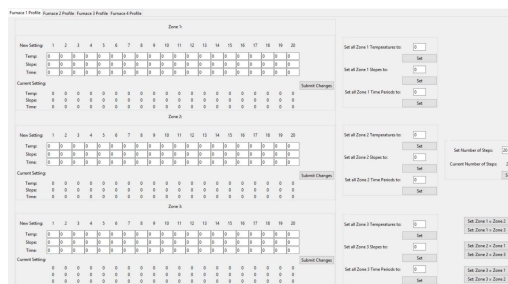


- **C programming of Arduino**
  - External Serial DAC control
  - Relay of OTC commands over RS-232
  - Redundant TFT display



- **Functional and Simple GUI**
  - Readable with quick and convenient inputs
- **API Design**
  - Maps RS232 OTC commands and SPI DAC commands to easy to use Python functions

	Temperature			Gas Flow		
	Zone 1	Zone 2	Zone 3	Oxygen	Nitrogen	Hydrogen/ Water Vapor
Current Furnace 1:	1	2	3			25
Furnace 1 Setting:	13	14	15			33
Current Furnace 2:	4	5	6	26	27	28
Furnace 2 Setting:	16	17	18	34	35	36
Current Furnace 3:	7	8	9	29	30	31
Furnace 3 Setting:	19	20	21	37	38	39
Current Furnace 4:	10	11	12			32
Furnace 4 Setting:	22	23	24			40



## Requirements

### Functional

- Set temperature and gas flow rates
- all-in-one GUI
- combine the temperature and gas controls into a single unit
- Constant Setpoint and Data value readouts
- redundant display

### Non-Functional

- Must be useable 24/7
- Cannot crash
- Capable of communicating with OTC and MFC
- User friendly
- Automatic Failsafe reconnection

## Standards

- RS232 communication (OTC)
- USB communication
- SPI communication (DAC)
- Tkinter Python GUI Library
- ASCII (OTC)

## Testing and Evaluation

### Hardware

- OTC communication
  - RS232 to USBconversion
  - Analog to digital conversion
- Continuous screen readouts from from Arduino

### Software

- API functions tested with OTC and DAC, properly acquire and send commands
- GUI tested with API, properly updates and sends new user designated data

## Conclusion

The project meets its goals very well. The current system is much easier to use than its predecessor, is more precise, combines everything into one platform, and it helps prevent errors. In the future, the GUI could be expanded to include more of the functions present on the temperature controllers among other things.