

Component Oven Temperature Control

by

Stephen John Hoelzer and Alan Carl Brooks

Final Report

Submitted to Dr. Gary Dempsey

EE 451 Senior Laboratory I

October 31, 2000

The objective of this mini project is to design and implement three different control algorithms in a microcontroller based component oven. A component oven is used to keep temperature sensitive circuitry at a known temperature for reliable operation. To build this system in lab, an EMAC 80535 microcontroller board and Kiel development software will be used. The project has a six lab period time frame with a two-engineer team. The given specifications for this project are as follows:

- 1) Display Format: TEMP=xxx°F
 where xxx=oven temperature in degrees F
 CONTROL=y(xxx)
 where y=0 bang-bang, y=1 analog, y=2 PWM, y=3 proportional
 and xxx=BANG, ANALOG, PWM, or PRO
- 2) Display Refresh: display update time = 0.5 seconds
- 3) Oven Temperature: 160°F with accuracy $\pm 5^\circ\text{F}$
- 4) Power Supply: 24VDC (nominal), 20VDC (minimum) for heating element
- 5) Temperature Scan: 5 milliseconds
- 6) Controller Outputs: updated every 5 milliseconds
- 7) PWM timing: Fixed period waveform (5 milliseconds), variable duty cycle in
 200 microsecond steps.
 Minimum duty cycle=4%, Maximum duty cycle=96%.
- 8) Port 4.0: update an external LED at LCD refresh rate,
 ON=temp in spec, OFF=out of spec
- 9) Port 4.1: flip pin to show 5 ms timing
- 10) Port 4.2: flip pin to show 200us timing
- 11) Port 4.3: flip pin to show 0.5 sec timing
- 12) D/A channel: use same channel for simple analog and proportional controllers
- 13) General port pin: same port for bang-bang and PWM controllers

HARDWARE DESIGN AND TESTING

The hardware portion of the component oven consists of two main blocks. First, the power supply block converts 120V AC to 24V DC. Second, the oven uses this power supply to maintain 160°F as regulated by a microcontroller. Temperature readings are sent to a microcontroller where command signals are generated to control oven heating. While designing these blocks, each was broken down further to better describe its function. The complete hardware block diagram in Figure 1 shows all of these sub-blocks. Each block and sub-block will be examined to show the circuit design, related theory, and testing.

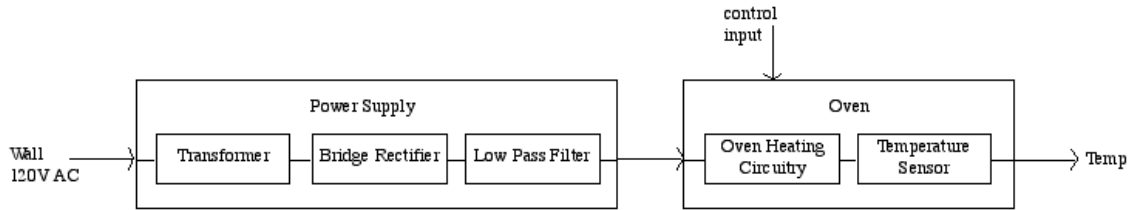


Figure 1: Hardware Block Diagram

Power Supply

The power supply consists of a transformer, bridge rectifier, and low pass filter (LPF) as shown in Figure 1. Dr. Dempsey selected a wall outlet transformer rated at 18V AC and 1A. The transformer's rated voltage is RMS, so the peak output is expected to be $18 \cdot \sqrt{2} = 25.5\text{V}$. The specifications of this project's power supply require 24V DC, so the transformer output had to be rectified and filtered. These are the other two blocks in the power supply diagram. The bridge rectifier converts the \pm sine wave to positive voltage. Then the LPF's low cutoff frequency generates a DC output with a small ripple voltage. Since this is a thermal system based on limited current and has a slow time constant, a small ripple voltage should not adversely affect the heating. However, the ripple should be small enough to keep the output above the 20V minimum specification.

In his design, Steve calculated the R and C values corresponding to a cutoff frequency, f_c , of 10 Hz for a first iteration on page 7 of his lab notebook. Initial testing in PSpice with $f_c=10\text{Hz}$ produced a plot with a 6V ripple. Assuming that the Tip 120 heating element (in Figure 4) is fully on, its emitter voltage will be about 0.7 volts which corresponds to $I_c=0.7\text{A}$. Since $V_{ce}=24\text{V}-0.7\text{V}=23.3\text{V}$, the resistance of the heating element will be a minimum of $23.3\text{V}/0.7\text{A}=32 \ \Omega$. To further reduce the ripple, f_c was lowered to approximately 1Hz by using this minimum resistance $R = 32 \ \Omega$ and choosing $C = 4400\mu\text{F}$. In simulation, the ripple shrank to only 1V which is an allowable distortion in the power supply output. This can be seen in Figure 2. Since $R = 32 \ \Omega$ is the worst case expected from the oven circuitry, the ripple should never be more than 1V.

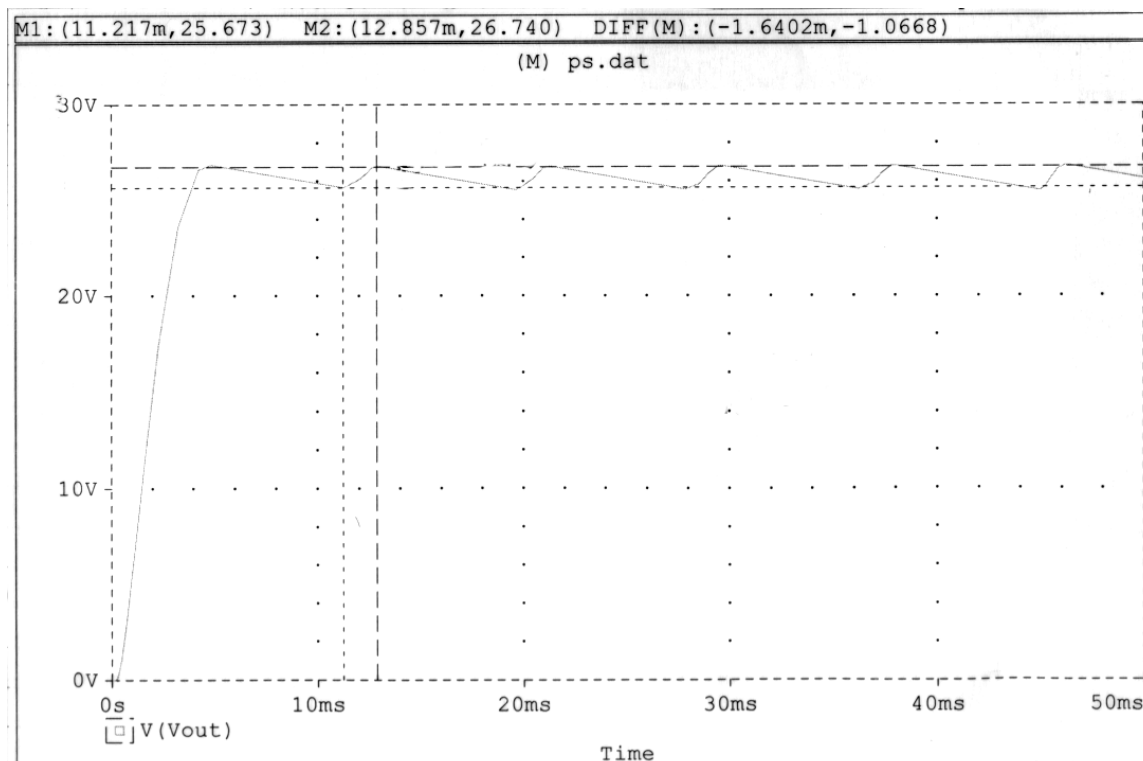


Figure 2: Power Supply PSpice Simulation

The last addition to the power supply was a voltage divider that generates a 5V output used to power the CMOS LMC6482 op amp and the LM34 temperature sensor. Dividing by four ($R_2/(R_1+R_2)=1k/(3k+1k)=1/4$), the Vout range of 20-28V was stepped down to an output ranging from 5-7V. This range was within the data sheet power specifications of the LM34 and LMC6482 devices. Figure 3 shows the final design of the Power Supply.

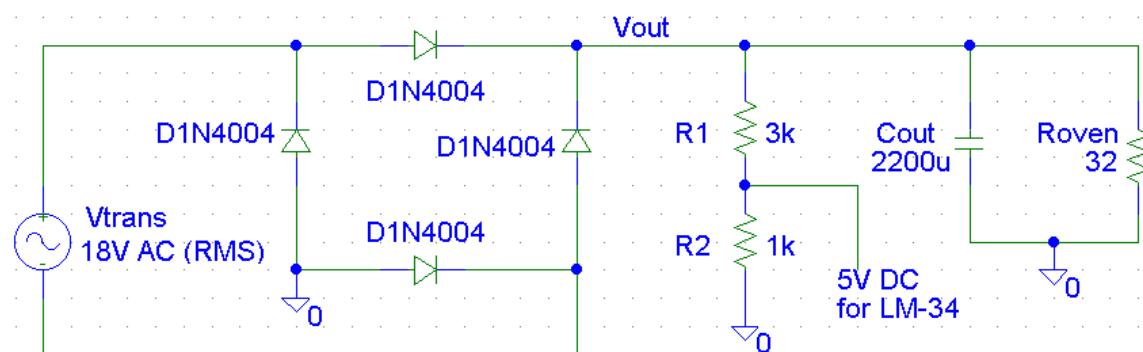


Figure 3: Power Supply

Oven Circuitry

The oven portion of the hardware consists of the heating circuitry and temperature sensor components of the diagram shown in Figure 1. The lab assistant, Jarrod Luker, constructed the physical oven containing the TIP 120 heating element and temperature sensor.

The LM34 temperature sensor gives a voltage output corresponding to its temperature in degrees Fahrenheit. A comparison was made with other group's temperature readings in lab to ensure that our sensor's output was accurate. Further along in the development cycle, testing revealed that the sensor had output noise of 250mV amplitude corresponding to +/- 12.5°F of variation. Therefore, this output was smoothed by connecting a 1000µf capacitor from the LM34 output to ground. Further examination of this value is included in Table 3.

The oven circuitry in Figure 4 was given to us by Dr. Dempsey. The LMC6482 op amp is a buffer for the microcontroller input to provide base current to the TIP 120, the power transistor heating element. The 2N2222A transistor limits I_e of the TIP 120 to a maximum of 0.7A. To ensure that we can reach a maximum current of 0.7A, we determined that a 2.4k resistor should replace the 22k resistor to allow for the worst case β of the TIP 120. The calculations related to this change can be found in Alan's notebook on page 9-10.

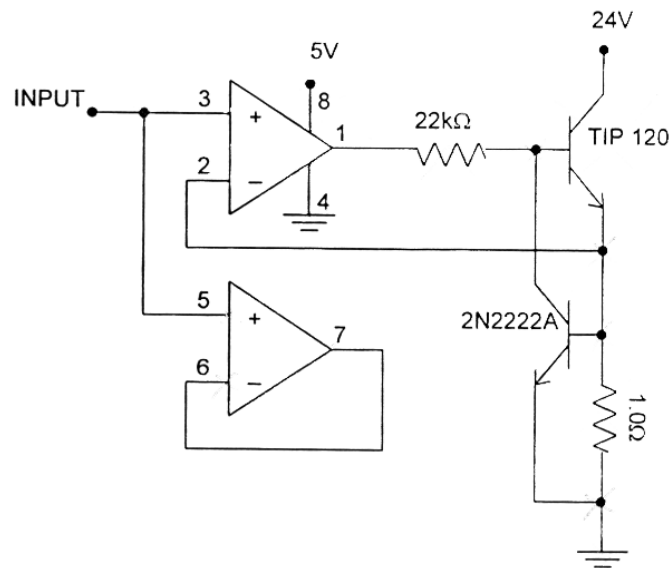


Figure 4: Oven Circuitry from Dr. Dempsey

Additional hardware in the design includes an in-range light-emitting diode (LED) and some microcontroller interface circuitry. Steve chose a standard LED with an assumed 20mA forward current. This was driven with P4.0 of the microcontroller, a high current output. With a low P4.0 output, the measured voltage drop across the LED was 2.8V. Therefore, the limiting resistor equals 110 to set current at 20mA (calculation shown on

page 31 of Steve's lab notebook). Also, 10k pull-up resistors were used on P4.1 through P4.3 to test the timing.

Part	# used	Part	# used
D1N4004 Diode	4	10k Resistor	3
LED	1	3k Resistor	1
2200 μ F Capacitor	1	2.4k Resistor	1
10 μ F Capacitor	1	1k Resistor	1
Wall Transformer (120V-18V)	1	110 Resistor	1
TIP 120 Power Transistor	1	1 Resistor	1
LM 34 Temperature Sensor	1	Aluminum Enclosure (L102-ND)	1
LMC 6482 CMOS Op Amp	1	Hole Grommet (RP453-ND)	1
2N2222A Transistor	1	3 Position Terminal Block (ED1612-ND)	2
EMAC Evaluation Board	1	Wrap-IT Ties (RP101-ND)	1
Breadboards	2		

Table 1: Parts List

Table 1 lists each part used in the final design. Figure 10 shows a complete schematic containing all hardware components from Table 1 and their locations and values in the final design.

SOFTWARE DEVELOPMENT

The microcontroller software closes the loop in this control problem. A closed loop controller can respond to a changing environment. Specifically, this component oven maintains an oven temperature of 160^aF with varying outside air temperature by using software controllers. The implemented controllers are bang-bang, analog, pulse width modulation (PWM), and proportional.

In addition to these controllers, the complete oven project includes timing and interface functionality to meet the given specifications (see page 1). To organize the high level interactions between components of the software, a flowchart was designed (see Figure 5). The input to the software is the oven *temperature* and the output is the *control signal*, which adjusts temperature.

The central component is the *Timing* block. It sequences the operation of the system by generating 0.5s, 5ms, and 200 μ s timings. The *Display* is updated every 0.5 seconds. The *A/D* samples a new temperature reading and the current controller (*Bang*, *Analog*, or *PWM*) updates its output every 5ms. If the *PWM* controller is chosen, it uses the 200 μ s timing to generate a PWM signal.

Finally, the *Keypad* block continuously checks for user input to set the current control type.

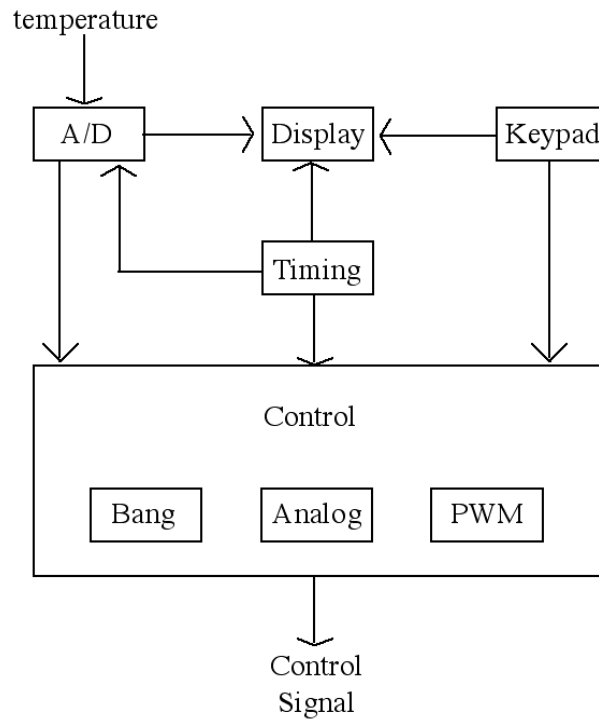


Figure 5: Software Flowchart

In the lab, the two engineer team worked together to develop the initialization code, timing, file structure, and the bang controller. Once these modules were designed, each engineer was able to work independently while maintaining a congruent overall code structure. Steve worked on the analog controller and display while Alan designed the PWM controller and keypad. Also, as an extra feature, a proportional controller was implemented. Table 2 details the specific software modules implemented in the project.

Module	External Data	Functions Called	Description
startup.a51	-	main_lop, adin, bang, analog, pwm, pwm_out, prop, display	Initializes microcontroller, defines public variables, generates timing, and calls controllers
main_lop.a51	-	lcdinit, keypad, display	Initializes LCD, then checks keypad in each loop and displays new data every 0.5 seconds.
adin.a51	-	-	Returns digital temperature value in acc from A/D channel 0 with A/D range from 0 to 2.1875 volts.
bang.a51	T_SET	-	Compares T (in acc) to set point (T_SET) and controls by turning P1.1 on and off.
analog.a51	T_SET	-	Compares T (in acc) to set point (T_SET) and controls by adjusting D/A channel 0.
pwm.a51	T_SET, duty_low	-	Compares T (in acc) to set point (T_SET) and controls by adjusting duty_low. (every 5ms)
pwm_out.a51	COUNT_TEMP_SCAN, duty_low	-	Called every 200us. Outputs the current PWM output on P1.1 according to duty_low.
prop.a51	T_SET	-	Compares T (in acc) to set point (T_SET) and controls by adjusting the A/D output proportionally.
keypad.a51	CONTROL_TYPE	-	Checks for key pressed and sets appropriate CONTROL_TYPE.
display.a51	CONTROL_TYPE, T_VALUE, T_SPEC, T_SET	LCDOUT DLAYA	Updates display every 0.5 seconds showing temperature and control type. Also, lights in-spec LED
lcd_gld.a51	-	-	Unchanged. Outputs ASCII Acc to LCD.

Table 2: Software Module Descriptions

This well-defined software framework afforded smooth integration of each engineer's code modules into the project. After small amounts of debugging, the project seemed to meet all of the specifications.

A final review and demo with Dr. Dempsey confirmed that the product was fully functional according to specifications. The project also included additional features to enhance the quality of the temperature sensor signal, the resolution of the A/D conversion, and the accuracy of the display. Furthermore, an advanced proportional controller was implemented. Combined, these extra features led to a Demo grade of 110%.

PRODUCT ANALYSIS

Design Considerations

Even with a grade of 110% on the demo, there is still some room for improvement in the design. The power supply ripple voltage and the proportional controller need further thought.

When the oven is fully on, the designed power supply drops down to 20.1V at the bottom of the 1.4V voltage ripple. This comes very close to the minimum specification of 20V. For volume production, a larger margin would be necessary so that most units built meet the specification. An appropriate way to improve this would be to increase the capacitor size in the power supply's LPF. Simulation showed that with $C=4400\mu\text{F}$, the ripple should be about 1V, making the specification margin about 0.5V. Also, to avoid such high capacitor values, we could consider more complex filter designs.

A review of the proportional controller code revealed that Alan's design is not an actual proportional controller. His design adjusts the previous A/D output with a proportional gain. A true proportional controller defines the control signal as the error signal multiplied by a constant.

Final Testing

To evaluate the finished design, accuracy testing was performed for each controller. Figures 6-9 show the steady-state outputs of each controller along with the LM 34 Temperature sensor output. The operation of each controller is described in Table 2. The accuracy of each controller along with the temperature sensor capacitor size are shown in Table 3.

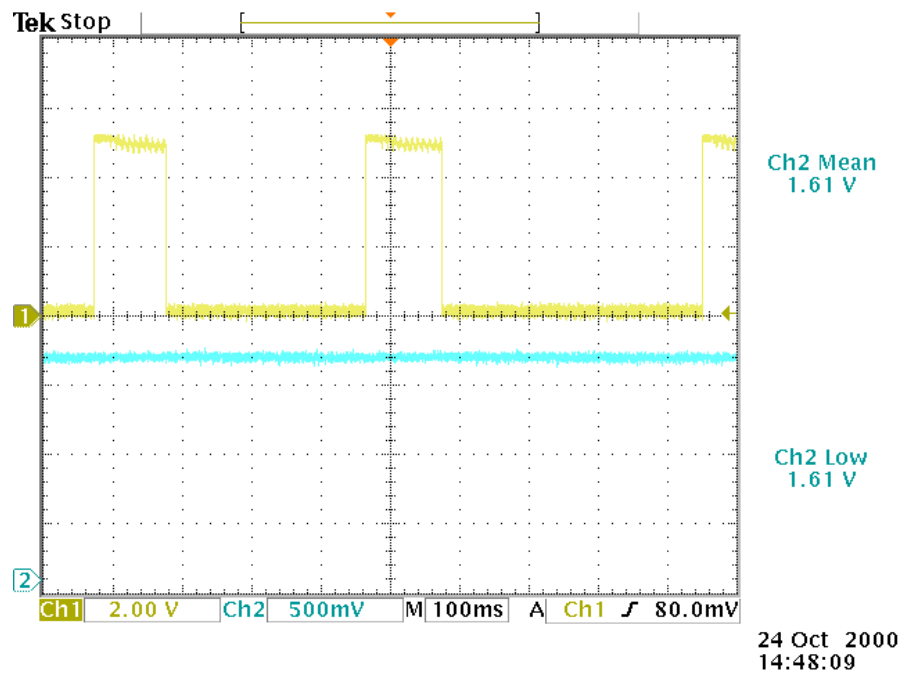


Figure 6: Bang Controller

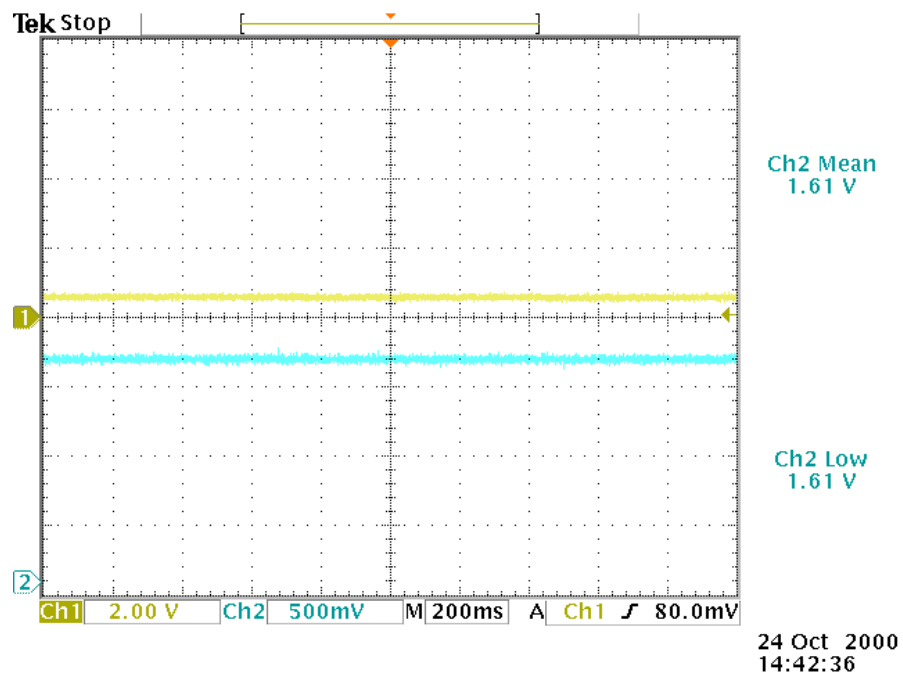


Figure 7: Analog Controller

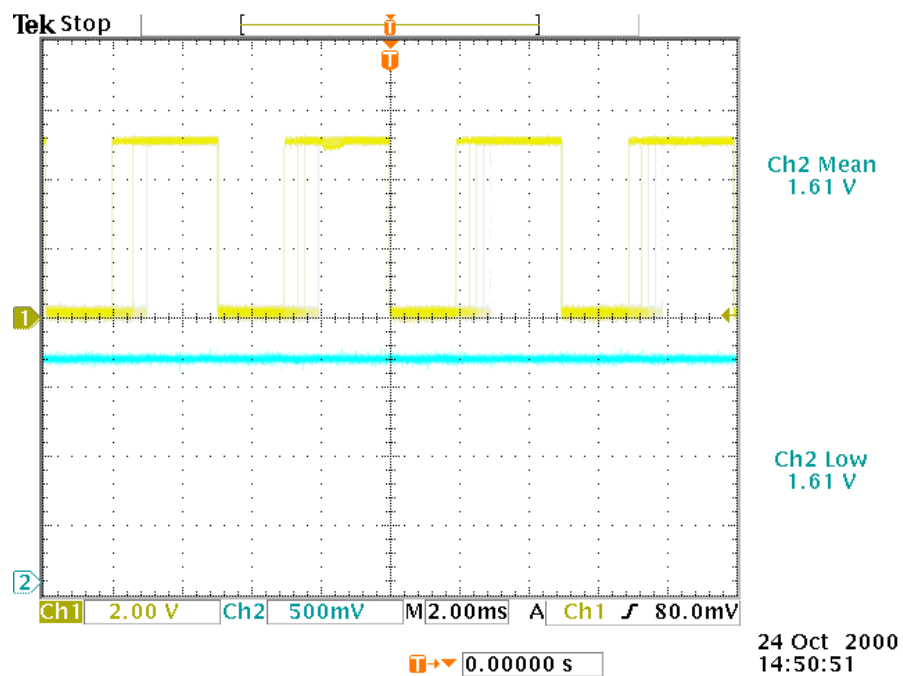


Figure 8: PWM Controller

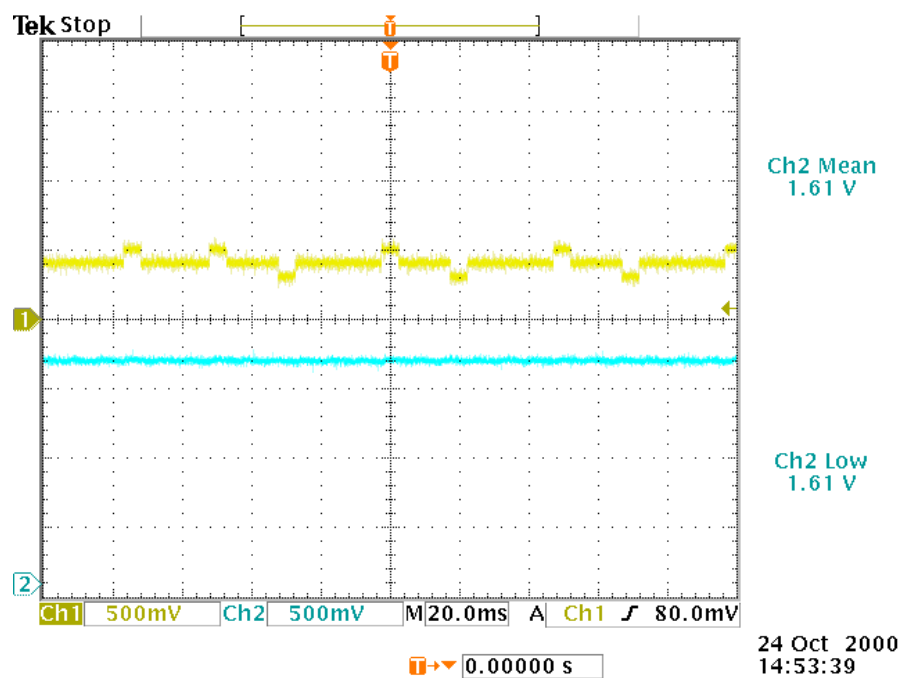


Figure 9: Proportional Controller

Controller	Temperature Range (°F)		
	1000 μ F Capacitor	10 μ F Capacitor	No Capacitor
Bang	160-162	159-162	158-165
Analog	161-162	160-162	158-165
PWM	160-162	160-162	158-165
Proportional	160-162	160-162	158-165

Table 3: Controller Accuracy Testing

The results in Table 3 show the effect of filtering the temperature sensor output. In summary, the 1000 μ F, 10 μ F, and no capacitors produce average temperature ranges of 1.75°F, 2.25°F, and 7°F, respectively. For a high volume design, a 10 μ F capacitor would produce accurate results with low cost.

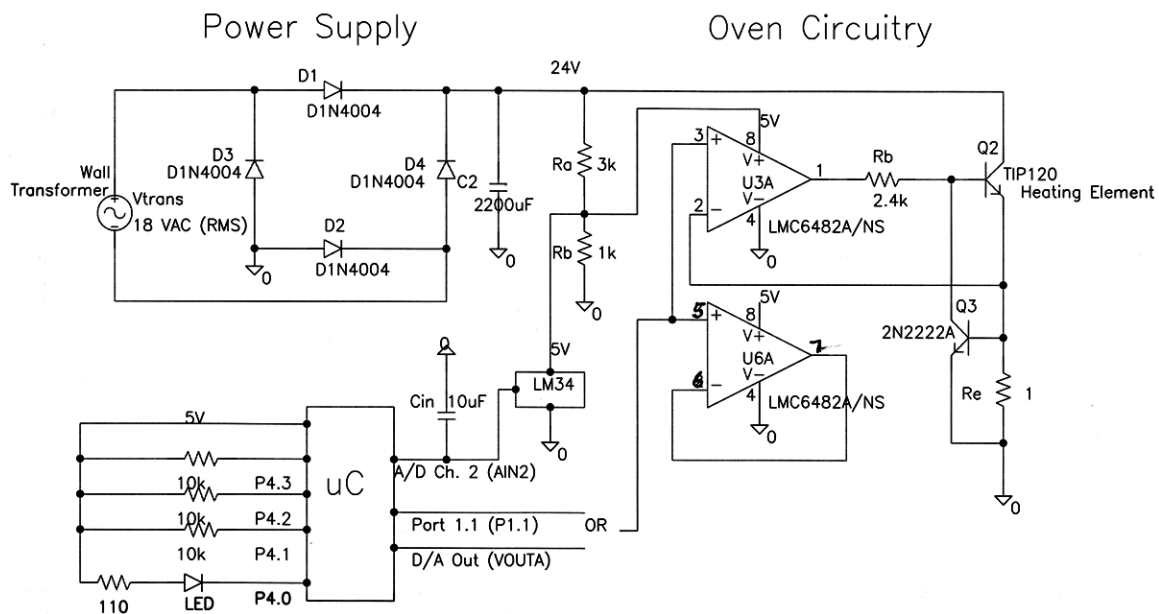


Figure 10: Full Schematic

As mentioned earlier, Figure 10 shows the full schematic of our design used in the demo and in testing. Also, a full code listing of each file described in Table 2 is attached.

Conclusion

Over the course of the six-week development cycle, a total of 105 hours were spent on the project. The time division was even with each engineer contributing slightly more than 50 hours (a detailed breakdown of the time division is shown in the lab notebooks).

The key factor that made this project a success was the initial top down design. By laying down a clear foundation, the two-engineer team was able to work independently while maintaining consistency and avoiding conflicts in the system design.