Project Work Summary

Students: Upload the signed sheet as a first page of your final report → w

Name	Adrian Brink
Project Title	RAS Autoclave Inlet Test Controls
A Brief Description:	system will test flow and heating control using thermocouples to monitor temperature, and an inline heater to warm the air flowing through the system. Two LEDs are also controlled to allow visual confirmation to the user that specific operating are taking place

Equipment to be used	Lab Computer		Servo Arm	
	Microprocessor		Robotic Kit	
	PLC		Ultrasonic Rangefinder	
	Coupled Tank		Power Supply	
	Items to be purchased			

	Slot Label	Student Signature	Lab Instructor Confirmation
Time Slot 1			
Time Slot 2			
Extra Time Slot			

Option

Notes on Project Work and Completion by Instructor:

- Heating element
- K~~type~~type thermo co-ple
- Relay solenoid
- solid state Relay
- valve

Works

12/5/22

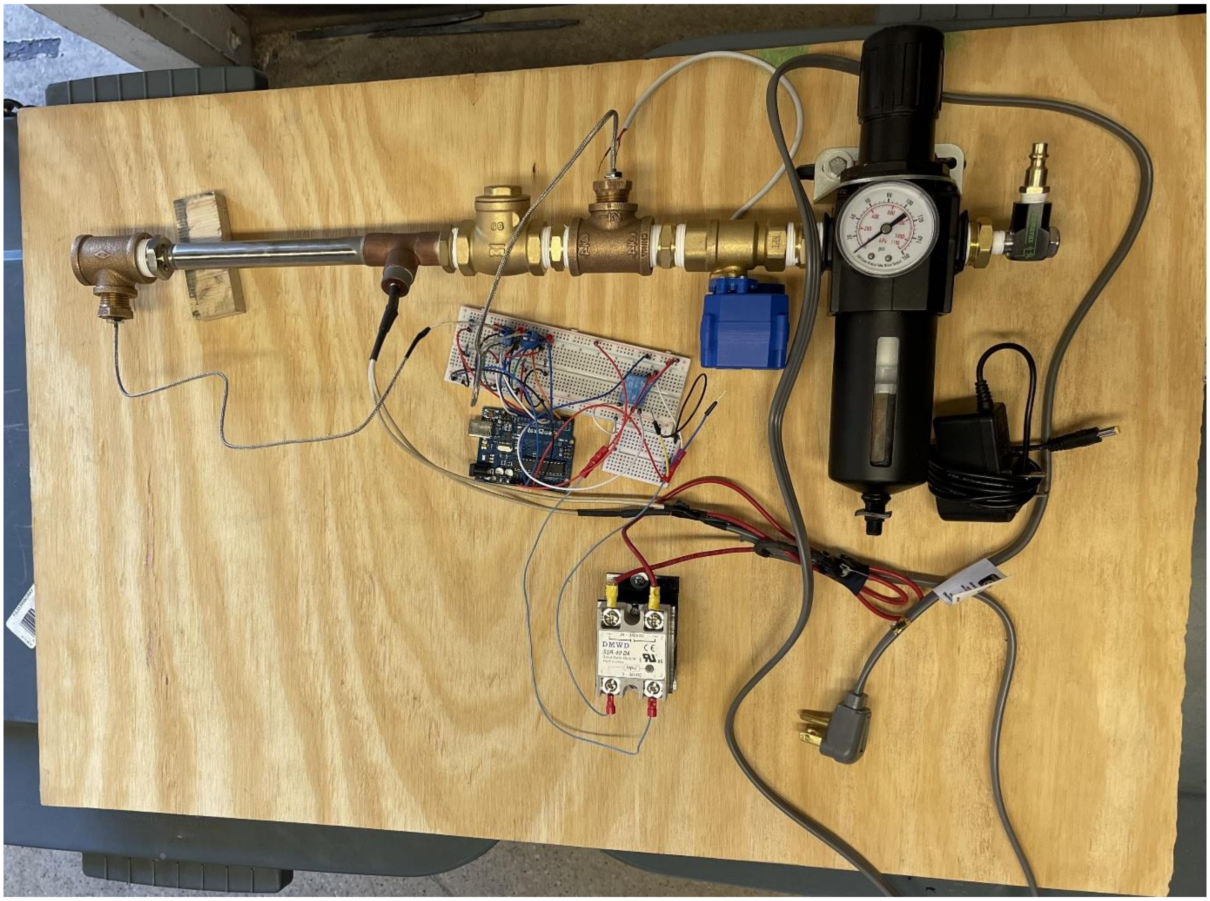
A+

↑ Instructors / TAs: ① Sign only after successful demo.

② Keep record of whose paper you signed (picture or write down name)

Student Name: Adrian Brink

Project Title: RAS Autoclave Inlet Test Controls



Description

At Texas Tech University there is a need for infrastructure that can be used for the manufacture of large-scale, high-performance composites that is not met by our facilities. The Raider Aerospace Society has been developing an autoclave to fill this need and this autoclave must have a versatile control system to accommodate a range of part geometries and polymer types, so the hardware has been designed to stay simple yet also achieve this goal, and this project specifically is to test these components and their interaction to ensure compliance with design specifications. This benefits from automation as it will allow for the collection of data (not implemented in code however it will be expanded to export to Excel so we can use the data to develop performance reports) and prevent the components from being overloaded when run. I will be simplifying the controls from their designed state by running the heater at full power exclusively, and not modulating flow rate so I do not have to implement the calculations to tune the outlet temperature to a specific steady state.

Inputs

Thermocouple 1 and 2 will be reading the inlet and outlet temperature of the flow, respectively, continuously throughout operation and returning that data to be assessed. If the outlet temperature rises above the set point the heater is turned off and cycled down to a cool temperature by continuing flow through the system. The signal is digital and required a set of libraries to decode and receive meaningful data.

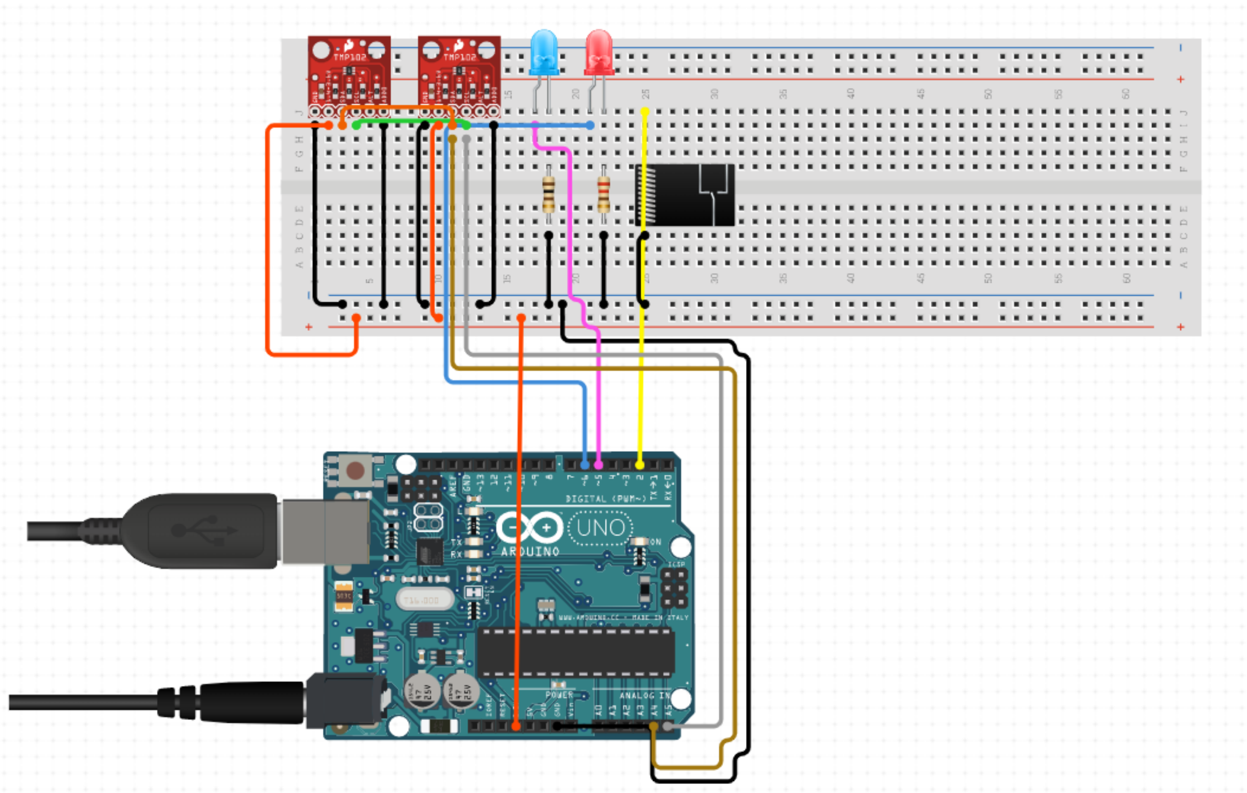
Outputs

heatOffLED (blue LED) is turned on at the same time that the heater is turned off to serve as a visual indication that the heater is off for the operator and the type of output signal is a 5V analog DC voltage. heatOnLED (red LED) is turned on at the same time the heater is turned on to serve as a visual indication that the heater is on for the operator and the type of output signal is a 5V analog DC voltage. The heaterRelay output is a 5V analog signal to energize an electromechanical relay which then steps up to a higher load for the 120V heater circuit which sits on a solid state relay energized by a 3.3V load switched on by the electromechanical relay. This was done because the power running through the heater circuit is too high for the breadboard to handle so it had to be taken off the board and controlled using a secondary relay.

Functions

1. Detect inlet and outlet temperature
2. Process temperature signal and determine if there needs to be an increase or decrease to maintain target
3. Energize or de-energize relays to turn on/off heater, respectively
4. Turn appropriate LED on as visual indication of what heater is doing
5. Read more data and cycle through logic continuously

Design



Measurements and Calculations

Resistors for LED function and powering k-type thermocouple boards are all 220 Ohms as this is what allows the LEDs to operate where they will not burn out and is what was called for in the data sheet of the MAX6675 thermocouple processing board I used.

Results

To test and validate the system various target temperatures were input so the logic could be tested for when the heating circuit should be closed or open. Full flow tests were also conducted to determine that the system works in working conditions it will experience in operation, and it performed exactly as expected, managing the temperature of output flow below the 100 F setpoint. Thermocouples are accurate to within +/- 0.25 degrees and were tested with calibration solution (ice saline solution) to verify that they were within specifications, which they passed.

Discussion

The system worked exactly as intended and demonstrated that the components function properly together and do not have defects which may affect their performance in the full autoclave system. The control system allowed for the monitoring and operation of the system, as well as performing its job of adequately safeguarding the heater from over loading and burning out.

Improvements and Extensions

The controls need to be expanded to monitor flowrate and modulate power delivered to the heater to get accurate and consistent control of temperature at the outlet continuously. More sensors will have to be added to accomplish this goal. When this assembly is incorporated into the pressure vessel at the core of this project zoned temperature data will also be added to ensure even heating throughout which will require more in-depth controls as well.

References

Rob Tillaart from GitHub's MAX6675 libraries: https://github.com/RobTillaart/MAX6675

\

Appendix

```
#include "max6675.h"

//test code for verifying flow and heating conditions of an autoclave
//inlet assembly for the Raider Aerospace Society
//Includes code for controlled visual indicators, monitoring thermocouples, and cycling a full flow heater


int heatOnLED = 8;//red LED to visually show heater is on
int heatOffLED = 9;//blue LED to visually show heater is off
int heaterRelay = 6;//pin used to energize

//input pins for 1st thermocouple
int thermoDO1 = 2;
int thermoCS1 = 3;
//input pins for 2nd thermocouple
int thermoDO2 = 4;
int thermoCS2 = 5;
//shared timing pin for both thermocouples, wired together on board
int thermoCLK = 13;

int T_desired = 100;//target temperature, set low for demo

MAX6675 thermocouple1(thermoCLK, thermoCS1, thermoDO1);
MAX6675 thermocouple2(thermoCLK, thermoCS2, thermoDO2);


void setup() {
  // startup code for visual checks and pin allocation
    Serial.begin(9600);
    //set digital pins used as analog to correct type
    pinMode(heatOnLED,OUTPUT);
    pinMode(heatOffLED,OUTPUT);
    pinMode(heaterRelay,OUTPUT);
    //test lights on startup, confirm they are working
    digitalWrite(heatOnLED,HIGH);
    digitalWrite(heatOffLED,HIGH);
}
```

```
void loop() {
  //setup serial monitor to display current temperatures for calibration and troubleshooting
  Serial.print("T1: F = ");
  Serial.println(thermocouple1.readFahrenheit());
  Serial.print("T2: F = ");
  Serial.println(thermocouple2.readFahrenheit());
  //read thermocouple data using library and store it in a usable form
  int T1 = thermocouple1.readFahrenheit();
  int T2 = thermocouple2.readFahrenheit();
  //delta will be the foundation of logic on whether to turn heater on/off
  int delta = thermocouple2.readFahrenheit() - T_desired;

  //if outlet is hotter than desired temperature turn heater off and cycle air for 10 seconds to cool off
  if (delta > 0){
    Serial.println("Heater turning off");
    digitalWrite(heatOnLED,LOW);
    digitalWrite(heatOffLED,HIGH);
    digitalWrite(heaterRelay,LOW);
    delay(10000);

  }

  //if outlet is cooler than desired temperature turn heater on and run for 400 ms then check again
  if (delta < 0);{
    Serial.print("Increase Still Needed: ");
    Serial.println(delta*(-1));
    Serial.println("Keeping heater on / turning on again");
    digitalWrite(heatOnLED,HIGH);
    digitalWrite(heatOffLED,LOW);
    digitalWrite(heaterRelay,HIGH);
    delay(400);

  }
}
```