To: Robert Severinghaus
From: Jumping Jacks - Team 4
Date: March 26, 2021
Subject: Testing Results Report

Introduction

This report will focus on the tests that were performed to ensure our final product is complete. In this memo there will be an introduction to the entire system, an explanation of the system architecture, requirements, testing types, major tests, analysis of the results, and lessons learned. The memo will go over in detail the areas of the project that were tested and the associated requirements that each test meets for the client. The lessons learned section will go over the important takeaways from testing the system that can be applied to future projects and teams to ensure similar mistakes do not occur.

EE486C
Team 4
Gas Reactor Sensors
March 26, 2021

Introduction to the System

The client for this project was Schuur Lab - Christopher Ebert. The client needed a new temperature control system as his old one was outdated and very close to not working any longer. The client also mentioned the systems being independent of one another which was not beneficial for him as he would like to run reactions overnight but did not have the ability to turn it off or automatically turn it off. The team came up with a plan that would utilize two Arduino controlled systems that would communicate with one another to control both temperature and pressure of the system. Our project contains a proportional, integral, derivate (PID) temperature control system. The system is controlled by an Arduino Mega 2560 microcontroller. The microcontroller has different components such as a temperature sensor, a solid-state relay (SSR), an oven heater, and a power supply unit. The system was intended to communicate with the pressure sensor system (another Arduino-based system) and keep the temperature at a setpoint (set by the user) by constantly correcting itself. The system is very user friendly as the user would use push buttons to set the wanted temperature press the “Start” button and then be able to go about their daily life as the system would turn off by itself.

EE486C
Team 4
Gas Reactor Sensors
March 26, 2021

System Architecture Showing Areas that were Tested

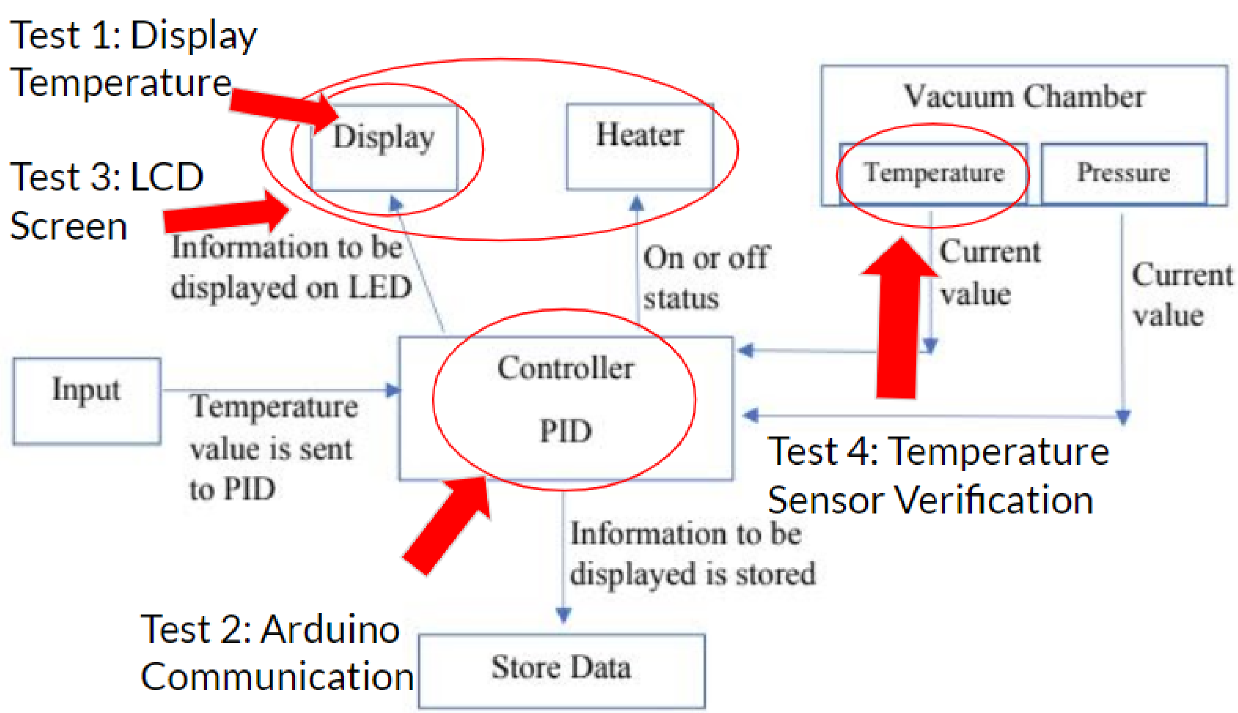


Figure 1: System Architecture

EE486C
Team 4
Gas Reactor Sensors
March 26, 2021

Requirements, status, type of test (Requirements tab on one page, color)

Instructions: List all of your requirements, and use a numbering system.

Type of Test	Status	Req #	Requirement
Integrate	*	1	LCD Screen
UTM	*	1.1	Temperature
UTM		1.2	Pressure
UTM		1.3	Time of reaction to nearest second
UTM		1.4	Heater on/off status
		2	Input
UTS		2.1	User selects temperature between 300-400 degrees Celsius in 10 degree steps
		3	Performance of PID controller
UTS		3.1	Monitor the reaction
UTS		3.2	Check pressure value to decide if heater should be on or off
		4	Store Data
Integrate		4.1	Information that needs to be sent to the Display
UTS		4.1.1	Temperature
UTS		4.1.2	Pressure
UTS		4.1.3	Time of reaction to nearest second
UTS		4.1.4	Heater on/off status
		5	Vacuum Chamber
UTS	*	5.1	Temperature
Inspect		5.1.1	Current value of temperature will be sent to controller every 0.5 - 5 seconds
UTS		5.2	Pressure
Inspect		5.2.1	Current value of pressure will be sent to controller every 0.5 - 5 seconds
		6	Heater
UTS	*	6.1	Status of either on or off

Figure 2: Requirements

Most Important Requirements

There are four different requirements that we deemed most important. The first is the LCD Screen, where we needed to ensure that the LCD Screen was working properly and displayed the required information. This is important to the client because it is where he will be able to see the temperature and desired temperature of the system immediately. If this requirement was not met there would not be a suitable way to display the current temperature of the system. Another important requirement was the temperature, where we needed to determine whether or not the temperature sensor was working properly. This is important to make sure that the oven reaches the proper temperature and stops at the correct temperature as well. If these values are not correct, the entire system and experiment will be incorrect. The third test is the status of the oven, on or off. This is important because if the pressure is not measured correctly and shared with the oven system, the oven will not turn off. This action is important to our client because it will ensure that the various experiments he has running at a time will shut off when they are complete.

Types of Tests

There are four different test types. The first is UTM, a unit matrix test. This test is used when the inputs are structurally the same and only differ in their value. We chose to use this test to determine the functionality of our PID, ensuring that the input temperature that we set will be the temperature that the oven goes to. The second test is the UTS, a unit step-by-step test. This test is used for instructions for generating the test and checking the results. We used this in two places, to test for Arduino to Arduino communication and to ensure that our temperature sensor was working correctly. The third test is an Integration Test which checks that the major modules of the overall system operate correctly together. We used this test to make sure all of the correct data we need displays on the LCD Screen. The last test is Inspection which is done by simply looking at the system. We used this test to make sure that the current value of the temperature displays and is sent to the controller every 0.5 to 5 seconds.

Major Tests

The first major test that was done was the verification of the temperature system and the K-type thermocouple that is used to read high temperatures. This test was done to verify the accuracy of not only the temperature sensor itself but help the team find out what adjustments needed to be made in the PID controller manual. For this test it was clear the K-type thermocouple was not the most accurate device but when researched and discussed with the client this was found to be normal and the client gave us the green light to continue with this thermocouple. In the chart below it shows the accuracy of the max6675 temperature sensor and thermocouple compared to a laser temperature sensor. The chart plots the difference between the laser temperature reader and the temperature sensor (k-type thermocouple) temperatures.

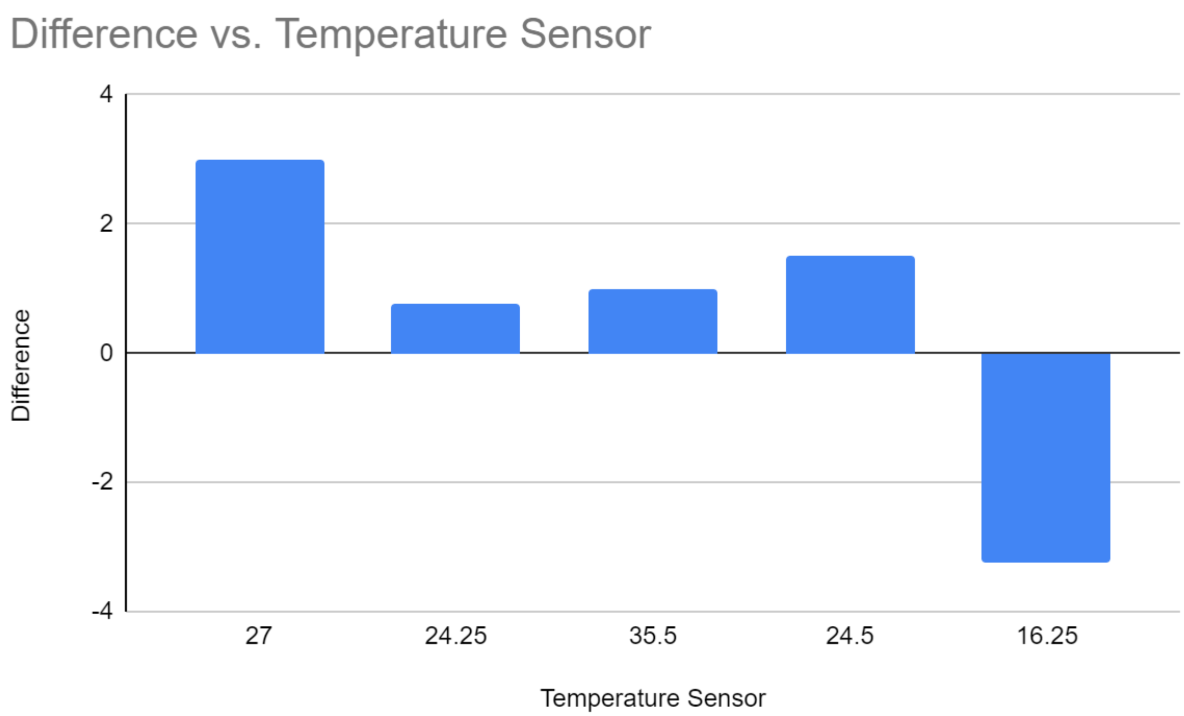


Figure 3: Difference in Temperature Sensor and real temperature

As you can see in the chart it is inconsistent in its ability to detect temperature at times. The chart shows massive differences while being cooled and slightly heated.

The second major test was the Arduino to Arduino communication between the pressure and temperature systems. In this test we did a number of sub-tests from sending a number from one Arduino to another and printing it out to the serial monitor to trying to display the received number or character to a liquid crystal display (LCD) screen. The image below shows the characters being sent from Arduino master to Arduino slave. The code below has the master send a value of x to the slave arduino which sends back the word “hello” every time the “x” value changes.

EE486C
Team 4
Gas Reactor Sensors
March 26, 2021

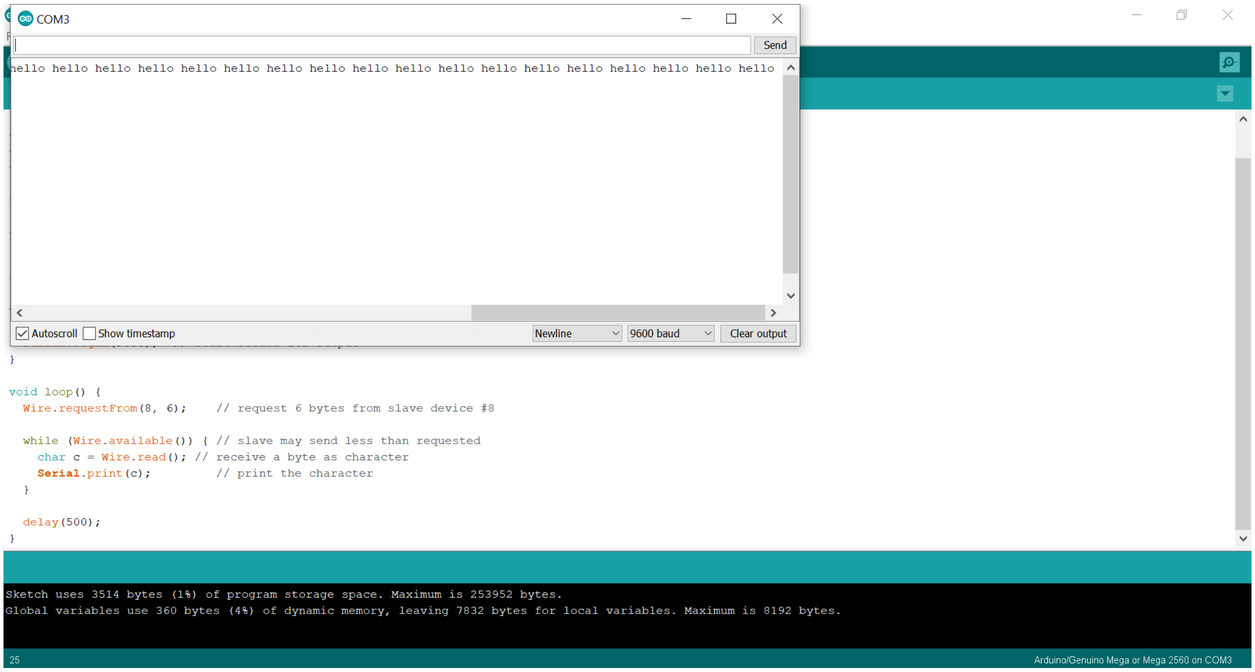


Figure 4: Serial Monitor

The third major test was the integration test which tested each aspect of the PID controller, the heating unit, set temperature, and LCD displaying all of the important parameters. This test is shown below through an image of the LCD screen which displays the set temperature “Set: “ and the real temperature “Real temp: “ The heating unit as seen in the picture never climbs to the desired temperature but did climb to 21 degrees Celsius. The display depicting this information can be found in appendix A.

Analysis of Results

Test one performed extremely well and had the accuracy that was expected from research and communication with the client. The important requirement that was met here is the ability to read the correct real temperature through the k-type thermocouple.

Test two performed well throughout the entire test except for one aspect. The system did not have the ability to communicate between Arduinos and display it on the LCD screen like the team had thought. The Arduino could transmit to the LCD screen or the other Arduino but not both. This was a small hindrance in our test that could have been changed by utilizing a Zigbee Arduino module for wireless communication. Unfortunately, this problem was not found in our research and the team paid for it toward the end of the project. The important requirement here was not met which was communication between Arduinos and displaying it to the LCD screen.

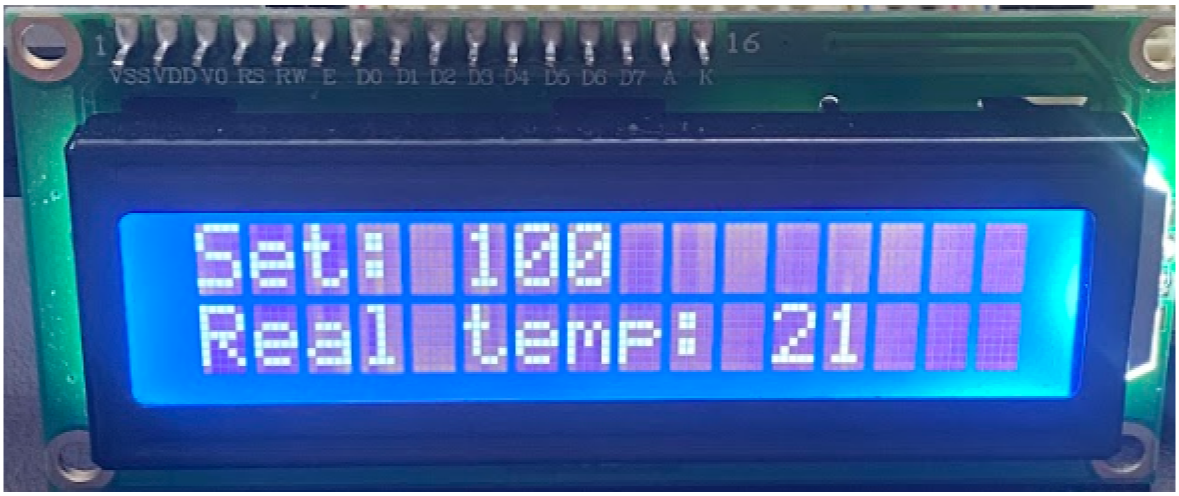
Test three underperformed during the testing phase. The PID temperature controller did not control the temperature to the accuracy that the team expected or wanted. This is mainly due to the fact that we had trouble controlling the power supply converter to apply the correct current and voltage to the SSR and heat the oven heater to the setpoint. The most important requirements were not met as this was the heater control test and we did not heat the oven to a temperature that was within 3 degrees of the setpoint.

Lessons Learned

In the testing of our project, we encountered a few roadblocks that were described in the Major Tests and Analysis of Results sections of this document. Due to issues we faced while testing, we had to change and get rid of some of our requirements to complete the project. Also, during testing we had to modify different aspects of our design. We initially were working with an I2C LCD screen but had to change to a regular LCD screen when we faced issues with the first one. After we began testing, we needed more space on our Arduino Board so had to change back to the I2C LCD screen to save space on the board for other parts of the project. Although we modified our design, we still did not have space on our board to operate the LCD screen and use Arduino to Arduino communication as they required the same pins. As this was done at the end of our testing, we did not find a quick and easy solution. In the future we could have experimented with different forms of Arduino communication that were not wired.

We did a number of regression tests with our design, as the project was dependent on the code working properly with our hardware. This is how we were able to fix bugs that came up as we were completing our other testing. For example, this was done when we encountered issues with our LCD screen when we were testing our Arduino to Arduino communication. The LCD screen worked before we set up the communication but would not as we were testing the functionality of the communication. As we were testing, the importance of testing procedures became apparent. There are aspects of the project that we would not have known to check on until working with the final product if we had not tested them, the main example being our communication system working with the LCD screen. Testing was important to discover the issue and, with more time, find a different approach.

Appendix A



This display shows the set and real temperatures for our system. Where the set temperature is given by the user, and the real temperature is measured using the k-type thermocouple.