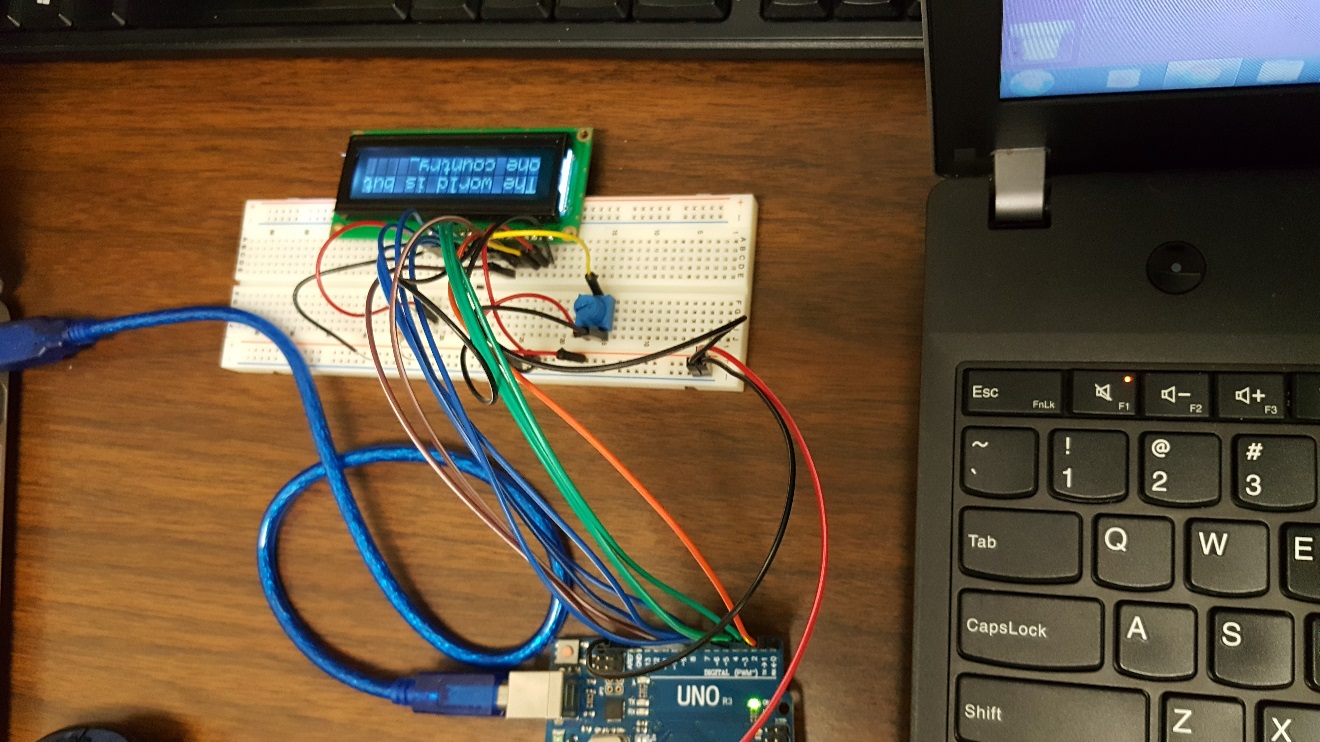
Obviously put better pic here

Group Project 2

LCD Display For Arduino

Casey Sanchez

Evan Coffey

Bryan Newsome

Amanda Hooge |

Table of Contents

[Introduction 2](#_Toc469181654)

[Development Tools 2](#_Toc469181655)

[Your experiment 2](#_Toc469181656)

[Conclusion 3](#_Toc469181657)

# Introduction

For our project, we decided to use the 16x2 LCD display and an Arduino microcontroller. To get it working all that was needed was around twelve jumper wires(what are those called?), a basic breadboard and the microcontroller. We decided to display a simple message on the LCD screen. This proved a very daunting task. After much research and meticulous reading of wiring diagrams, we were successful. (This part is if we get anything else working and word it better.) As soon as we were comfortable with the assembly language for the LCD, we were eager to add more components to out demonstration. Being in the holiday mood, we added blinking LED lights and a nice tune with the buzzer.

# Development Tools

When we began our experiment, we used the Arduino IDE 1.6.13. The software is open-source, written in Java, and works on Windows, Mac OS X, and Linux platforms. It can load code in various languages onto any Arduino microcontroller. It was very easy to install and get started right away. It even came pre-packaged with some helpful example code. Eventually, we stopped using the IDE and started loading code onto the Arduino via the command line on Amanda’s Mac machine.

\*\*\*The disassembler we used to get assembly language out of our C code is [\_\_\_\_\_\_\_\_\_\_\_\_\_]. Blah blah blah blah blah.

We communicated through email and collaborated on Github. We also used Google to find various pieces of documentation which helped us complete the assignment. No other development tools were needed.

# Your experiment

Our objective from the beginning was simply to get the 16x2 LCD to display some text. Given that there was no documentation for the LCD and that none of our group members felt confident writing the assembly language from scratch, we decided to think of other options.

We ended up trying a few, but before we could, we had to properly connect the LCD and our Arduino to a mutual breadboard. A google search or two later, and we had documentation which assisted us with this. We leaned in, put the wires in their place, and the screen lit up. “Hello world,” at last!

The first code we loaded onto our microcontroller was C code that we ran through the Arduino IDE. We knew that we’d eventually be turning in assembly language, but we wanted to make sure it worked before we got our hands dirty with the tough stuff. It worked like a charm, so we packed up for the day.

The above took place a few days after the group project was assigned. A week or so later, after some communication through email, chatting during lab time, and some internet research, we reconvened to try to get the LCD to work with assembly language code.

This is where things got hairy. As I mentioned before, we wanted to find assembly language code and modify it for our specific needs, but at this point, we had a hard time finding working code anywhere on the internet. We got very close with one attempt, finding some code that got the LCD to light up. Unfortunately, however, we couldn’t successfully modify it to display text. As much as we tried, for some reason, the only working assembly code we could produce was our disassembled C code.

Eventually, we decided that our best bet was simply to dig into the disassembled C code, make sense of it, trim the fat to make it more readable, and submit that. After breaking once more to complete our assigned sections of the lab report and look over the assembly language code, we came together one last time to piece the report together and agree on the assembly language code we wanted to turn in.

# Conclusion

We learned that transmitting data from a computer to a device is as simple as buying some wires, a breadboard, and an Arduino, and sitting down to write (or find!) some code. We only worked with one device, but it’s clear to see that one could let his or her imagination run wild with all the gizmos available on the market today.

Our assignment was to submit code written in assembly language, but that’s not at all a requirement of using these gadgets. Anyone with a basic understanding of programming and some experience with any popular high-level language could get started with microcontrollers.