

ELEGOO UNO R3

ATmega328

Manuel Ponce | Name2 | Name3 | Name4

# Table of Contents

1. Table of Contents

2. Introduction

3. The Microcontroller Platform

4. The Test Device

5. Development Tools

6. Your Experiment

7. Conclusion

8. Contributions

9. Project Code

Introduction

In this group project, we were given the assignment of taking an Arduino UNO R3 board and mixing it with another piece of hardware in an environment where we were responsible to get the correct signals to be read from the device in a timely manner. Instead of using the Arduino IDE, we needed to come up with original code in assembly for only the specific device.

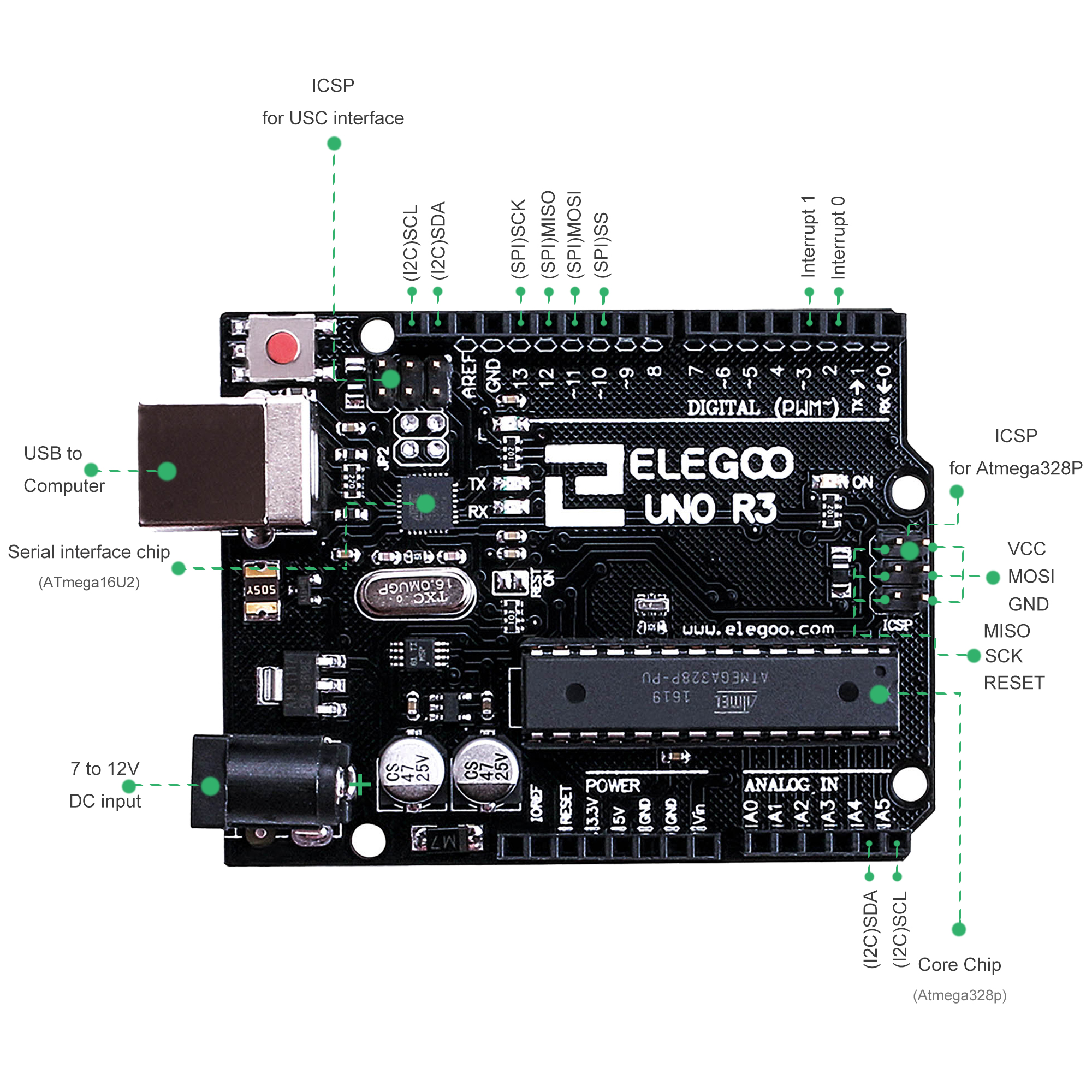
The specific device the group chose was the analog joystick with a switch. The joystick is very simple as it has the ability to move in the X and Y axis and when pressed straight down it can activate a switch.

The experiment that we aimed for was to read the input from the analog pins on the Arduino that were being provided with values from the joystick when it is moved in any direction. Then we take those values and input it into a delay function in order to change the speed a LED would blink.

The Microcontroller Platform

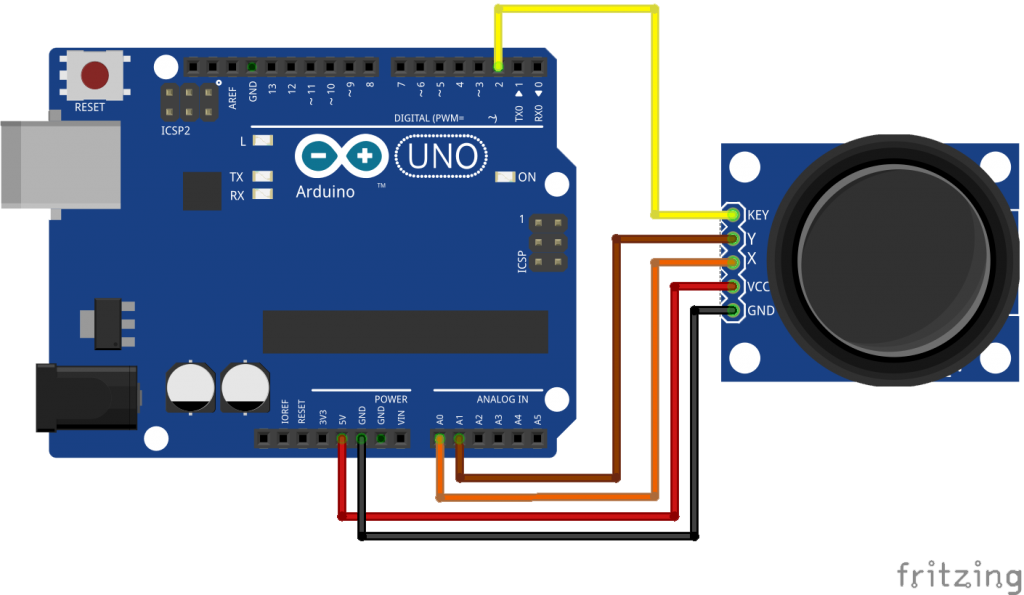
For this project, most of the experimenting was done on an ELEEGO UNO R3 replica of the original Arduino R3 board. This board was purchased on the official ELEEGO Industries website. This board can be purchased by itself or also as a kit with many pieces of hardware to create simple to difficult projects.

This board came with an ATMEL ATMEGA328 microcontroller and an ATMega16U2 serial interface chip. The board comes with two sets of pins made up of one side of 8 digital pins and the other side of 14 analog pins. The board also has a USB to PC port for data transfer and power, a 16 Mhz clock speed, and 32K flash memory.



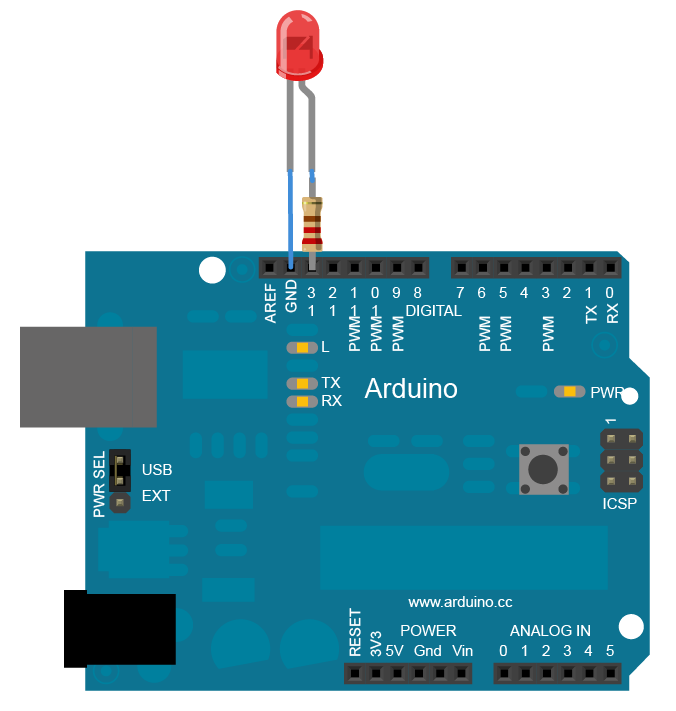
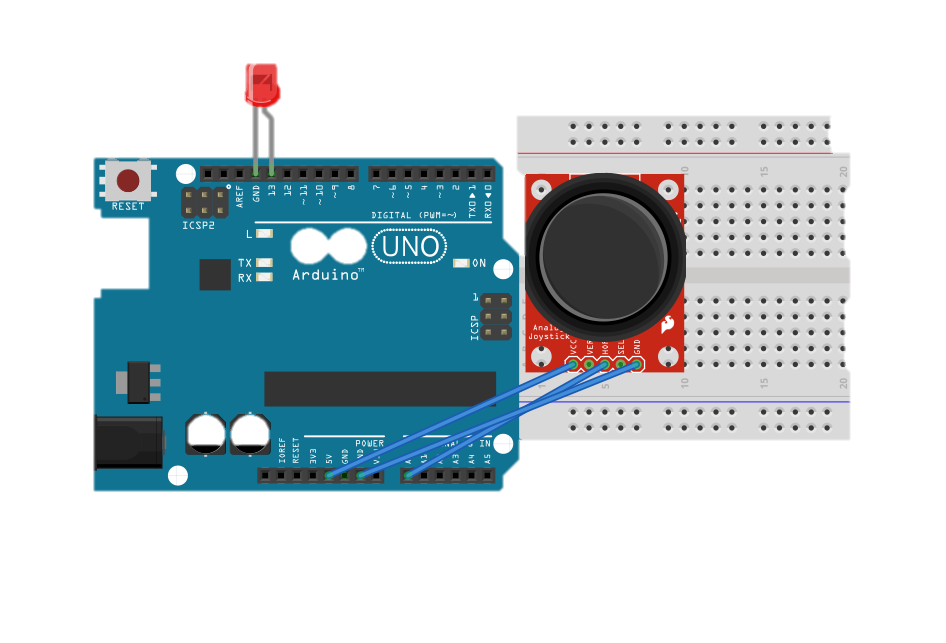
The Test Device

The test device we used was a joystick module that came included in the purchased kit of the ELEEGO UNO R3. Most joysticks are used for controlling things that need a more complex value rather than a button switch to be pressed. The most common use of small joysticks like the one used for this experiment are used in video game controllers to shoot the space aliens in the games! The joystick has only 5 pins and they are the ground pin, 5v power pin, x-axis pin, y-axis pin, and the switch pin for when pressed down. To get this device to work with the ELEEGO board was very simple. We used Jumper wires to connect all the correct pins to the ports on the board.

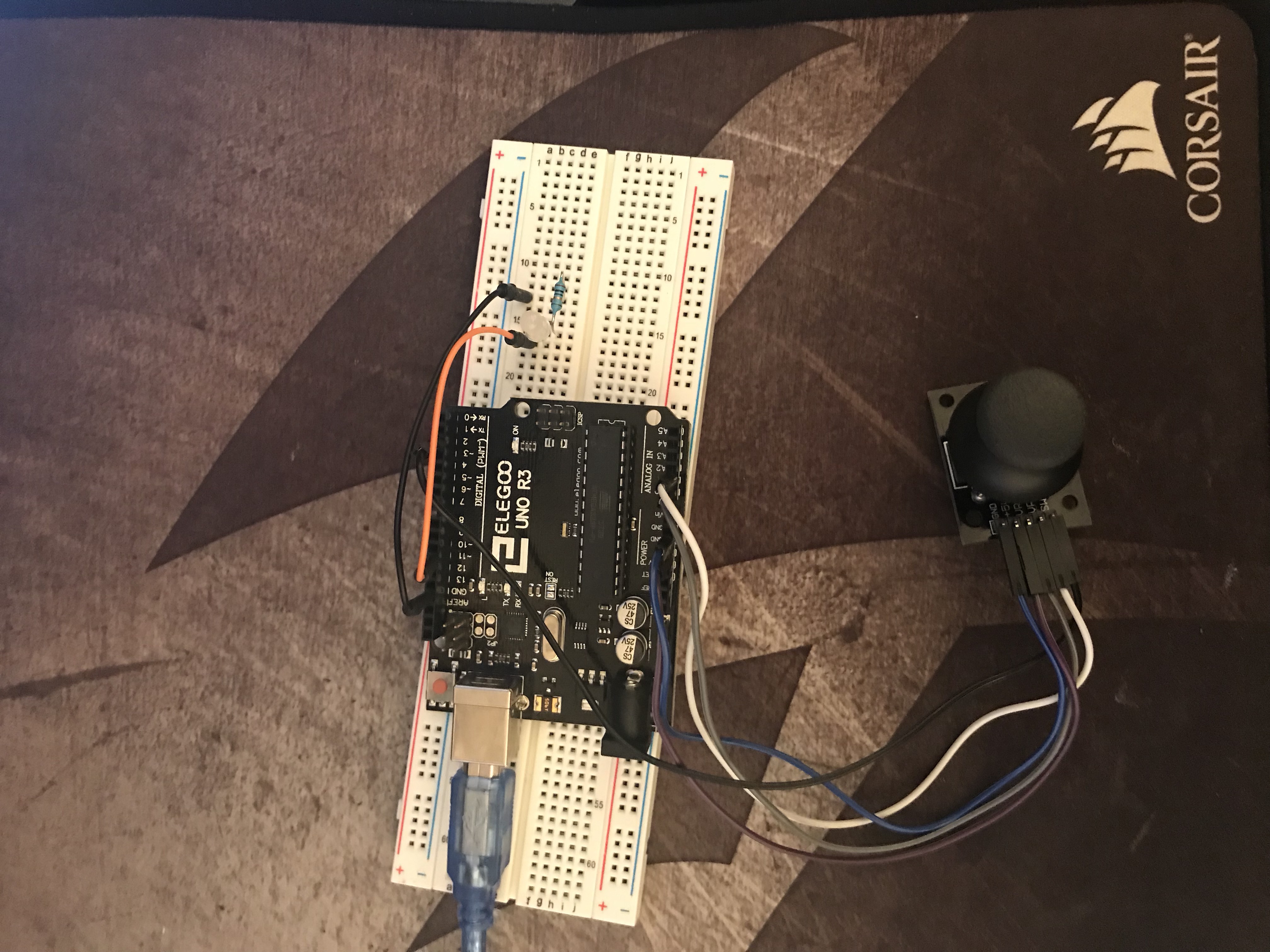


Development Tools

Basic hardware tools used to wire up the whole project was a total of 5 female-to-male Dupont wires, 1 LED, 1 120 Ohm resister, 2 jumper wires, a breadboard, and the USB to PC for transferring the code written onto the ELEEGO board and to power it as well. For software nothing much was needed that we already did not have. For starters, we began the project by starting in the Arduino IDE software to make sure the board was working and hooked up correctly. From there we moved onto transferring the Arduino Software code into C code. The final step was to be able to use the supplementary C code and assembly code we came up with, to work together to get the final product. To do this we used a Makefile that would use the help of AVR Dude to help load it onto the board itself.



Your Experiment

 With this experiment, the first ideas for the joystick was to read in the values and be able to use those values to move a character in an emulated environment of a game console running a game that uses a controller with a joystick controller normally. Choosing the joystick was easy because we saw it as a typical and useful device that could be neat to learn to manage and code for future projects. For the Experiment we were unable to get the joystick connected to the emulator of the game console. Instead we settled for the use of an LED to change the speed of the blink and the delay after the blink based on the position the joystick was at in real time. This project was very different compared to past programming projects, this involved a lot more of a plan to tackle the problems encountered. The biggest problem faced on this project was reading the value given from the joystick. Then passing them on to the register to load the delays and blinks.

Conclusion

Learning to deal with the hardware control on this project was very time consuming and a lot of extra work, but overall the information gathered and retained from doing this project is and will be very beneficial for future reference. Doing this project was testing the waters for future projects that will go right along with other devices that will work very similar. Another project that I could see myself trying is being able to connect the joystick to an LCD Screen to display a code of a small replica of a snake like game. This would be a little more complicated in getting them to work together but doable with a larger time frame.

Contributions

Manuel Ponce was responsible for writing the final report, the basic IDE Arduino code, transferring the IDE Arduino code into c, and supplying the assembly language code for the joystick.

Emilio Campos contributed to project concept.

Project Code

* Arduino IDE Code
* C code based on the IDE code
* Assembly code that is only to set up the joystick and retrieve the values.