Automatic Firefighting Robot

Group C:
Gabriel Martinho Bizuti - Electrical Engineering
Sydney Brown - Electrical Engineering
Juan Cuervo - Computer Engineering
Noah Schrock - Computer Engineering

University of Central Florida
Department of Electrical and Computer Engineering

Project Description and Motivation:

- A robot that moves autonomously to extinguish fire/flames.
- A simple design that is easy to operate.
- When a flame is detected, the robot moves towards it and puts it out.
- IR sensors will be used to detect fire/flames.
- MCU processes signals from the sensors.
- Once the fire/flame is in range the water pump is activated.
- Create a device that can keep humans safe when interacting with fire.
- Prevent the spread of fire to the rest of the vicinity.
- Initiate discussions on this topic to look for possible improvements and advancements.

Goals and objectives:

- Design a robot that moves on its own and accurately puts out fire/flames.
- Create a design that is simple and effective.
- Avoid complex set up and user interface.
- Have a product that can help users feel safer due to its presence.
- Robot can easily move and position itself where it needs to be.
- Once in range, the robot can quickly put out the fire/flame.

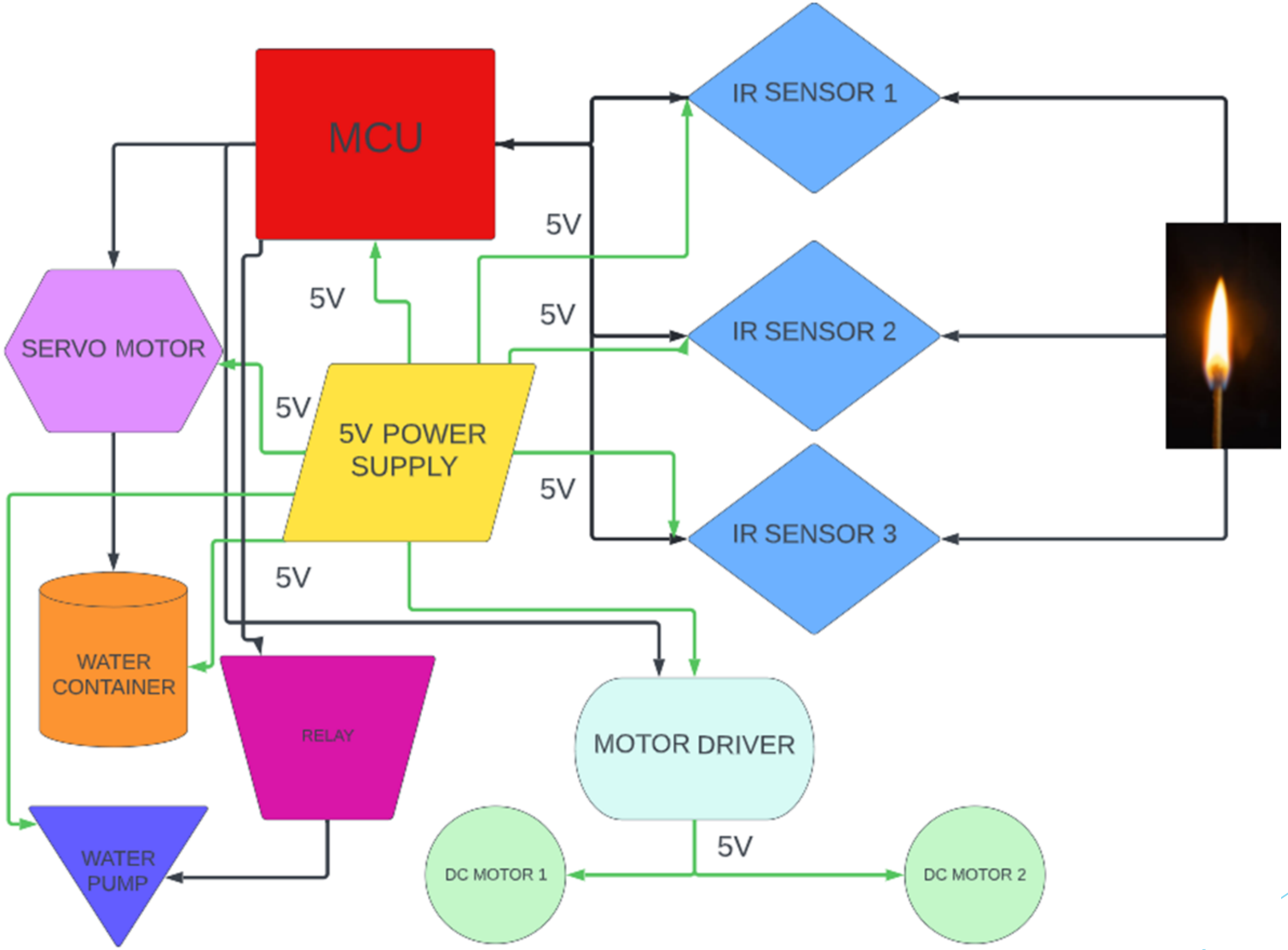
Specifications:

Specification	Description
Robot Size	Between 25 cm and 35 cm
Number of Sensors	3 IR sensors
Sensing Range	30 cm
Extinguish Time	3 Seconds
Movement Speed	Close to 2 Km/h
Response Time	Instantaneous once within range
Power Supply	5 V

Project Description and Motivation:

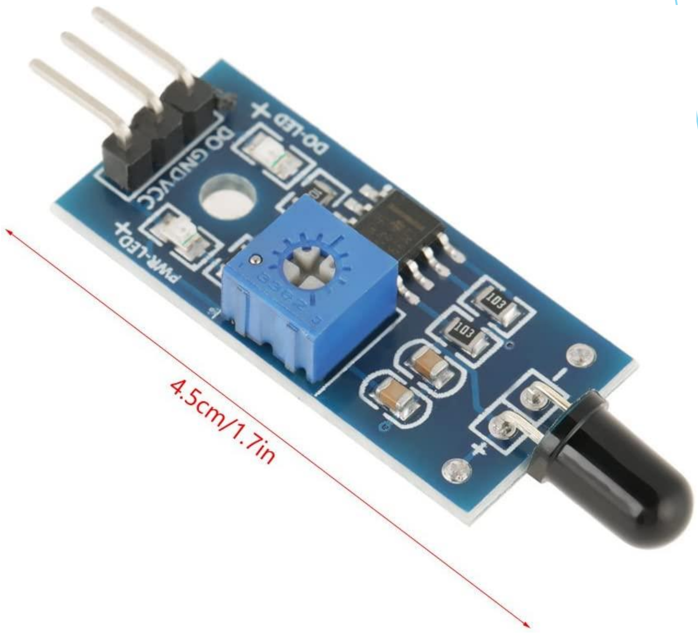
- A robot that moves autonomously to extinguish fire/flames.
- A simple design that is easy to operate.
- When a flame is detected, the robot moves towards it and puts it out.
- IR sensors will be used to detect fire/flames.
- MCU processes signals from the sensors.
- Once the fire/flame is in range the water pump is activated.
- Create a device that can keep humans safe when interacting with fire.
- Prevent the spread of fire to the rest of the vicinity.
- Initiate discussions on this topic to look for possible improvements and advancements.

System Block Diagram:



Walfront Infrared Fire Sensor:

- Function: Detect the presence of fire and its direction.
- Operating voltage for the sensor is 5V.
- 3 pin sensor that includes:
  - Digital Output pin.
  - Ground pin.
  - Vcc pin.
- Detection Range of 30 cm.
- Detection angle of 45°
- Detects infrared light wave between 760 nm and 1100 nm.
- Cost of $1.37 per sensor.
- Bought from Amazon.com



UCF

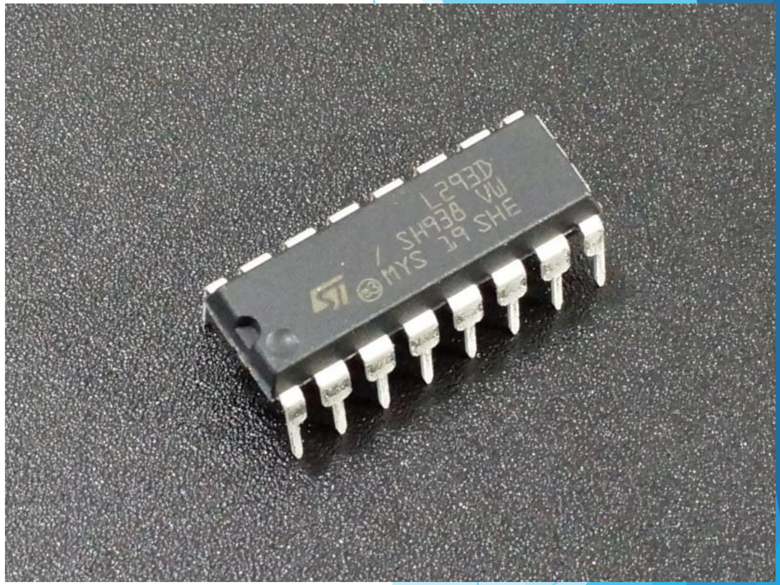
DC Motor and Wheel Set:

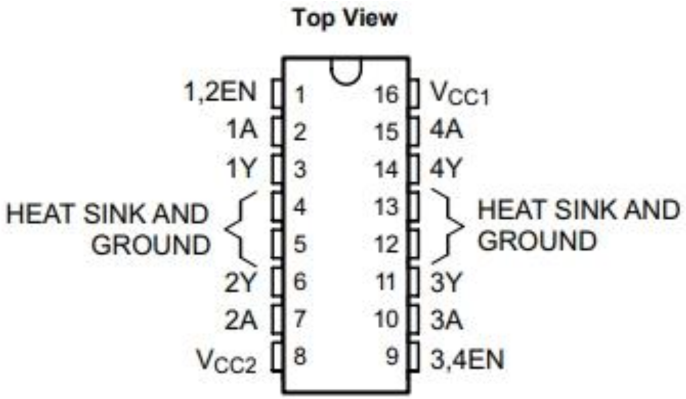
- Function: Move the autonomous robot close to the fire when detected.
- Operating voltage of the motor is between 3V and 12V DC.
- The DC motor has 2 terminals that are connected to the motor driver.
- Typical motor current is 140 mA.
- Motor Speed:
  - 180 RPM for 5V(No load) supply.
  - 120 RPM for 3V(No load) supply.
- Motor dimensions(L x W x H) are 70 x 22 x 18 mm.
- Tire dimensions(H x W) are 66 x 25 mm.
- Bought from protosupplies.com

UCF

L293D Motor Driver IC:

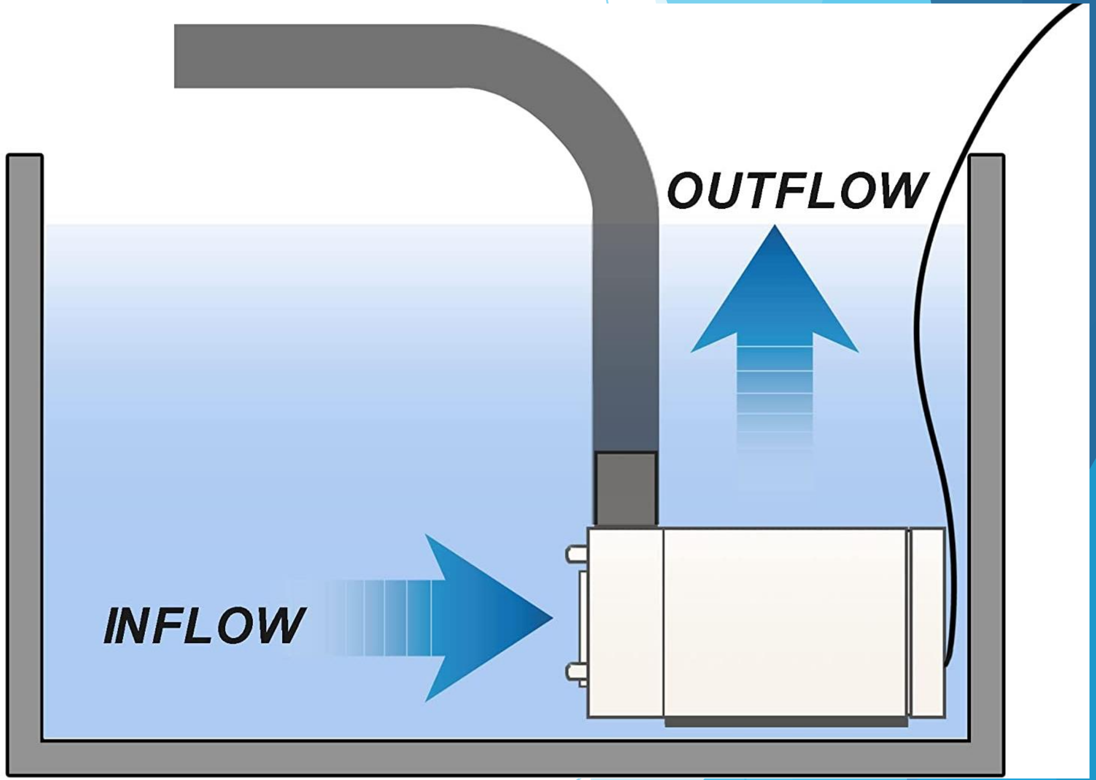
- Function: The motor driver is responsible for controlling speed and direction of the DC motors.
- Operating voltage for Vcc1 (Internal Logic) is 5V.
- Operating voltage for Vcc2 (Motor) is between 4.5V and 36V.
- Uses the H-Bridge circuit to control the rotation direction of the DC motors.
- Bought from protosupplies.com





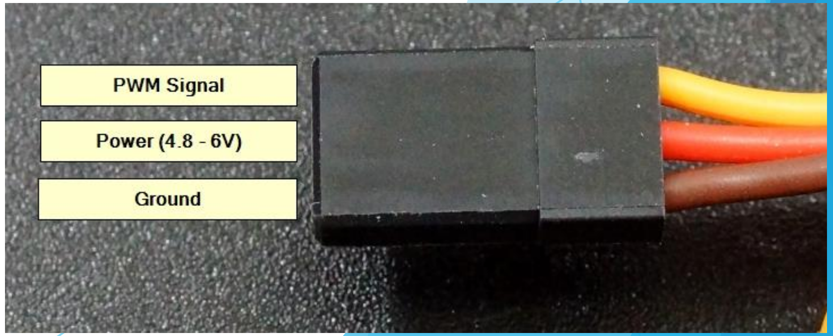
DC Mini Water Pump:

- Function: Pump water when close to the fire to put it out.
- Operating voltage for the pump is between 3V and 5V.
- Current consumption is 120 mA.
- Water flow rate of 100 L/H.
- The water pump is going to be controlled by the MCU.
- Bought from Amazon.com



Servo Motor Micro SG90:

- Function: Rotate the water container to align the water pump with the detected flame.
- Operating voltage for the servo motor is between 4.8V and 6V DC.
- Typical operating voltage is 5V.
- The servo motor has 3 connections:
  - PWM Signal received from the MCU.
  - Power connection.
  - Ground connection.
- The servo motor has 360° of rotation.
- At operating voltage of 5V, the full rotation speed is of 120 RPM.
- Bought from protosupplies.com

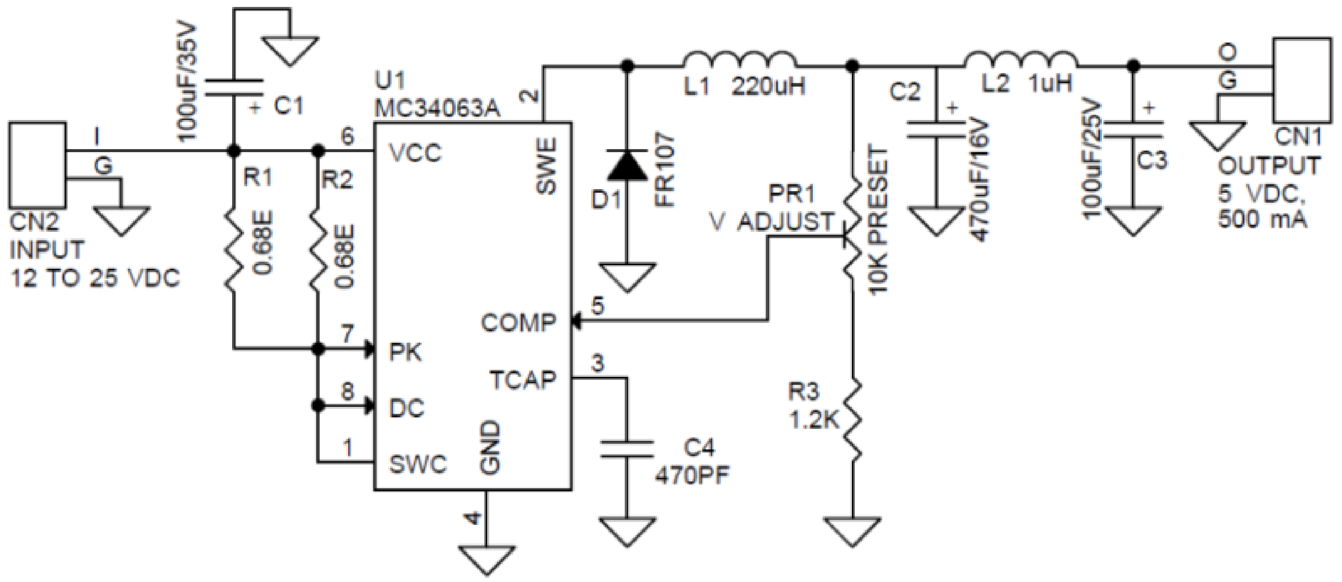


Power Supply:

- Lithium Polymer (LiPo) Battery
- Voltage Range for our project – 7.4 - 12 V
- High energy density – can deliver higher currents for critical applications
- Ultra-thin
- Lightweight
- High security
- Low cost
- Small internal impedance
- Low maintenance
- Specialty cells
- Applications – Li-ion batteries have replaced other rechargeable batteries in electronic devices

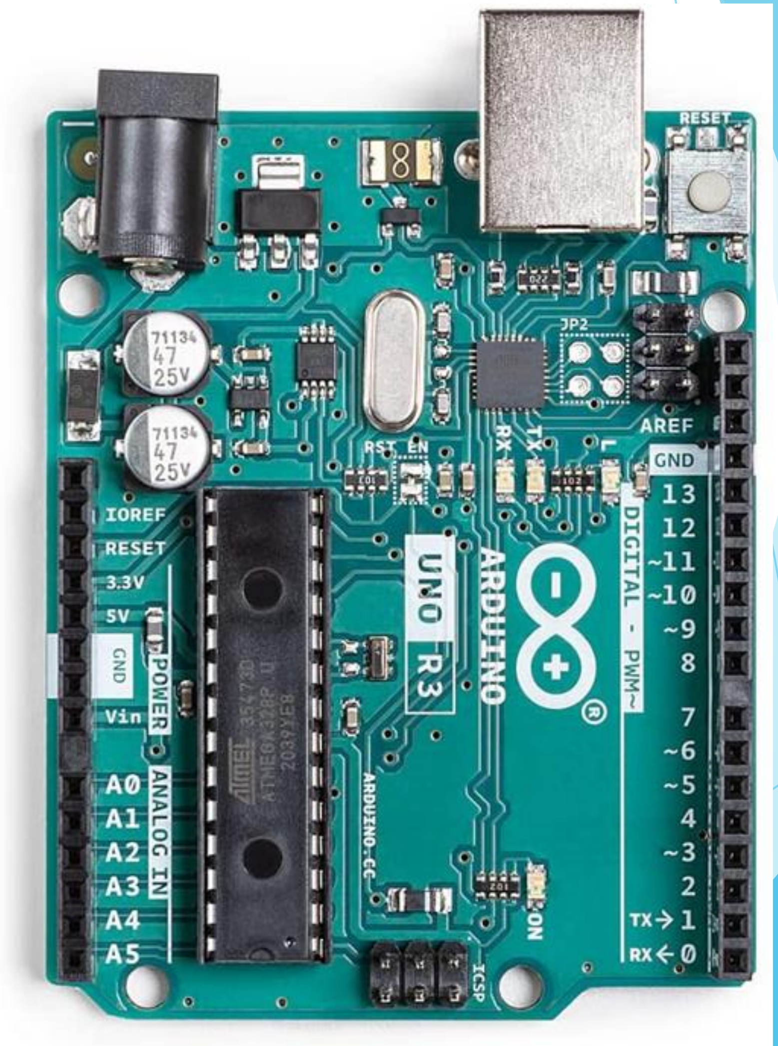
Switching Regulator:

- Switching regulators are able to step-up (boost) or step-down (buck) voltages
- A buck converter is a DC to DC converter that converts high voltage to a low voltage
- High efficiency across a wide range of input and output voltages
- Better transient performance
- Low heat generation
- Less expensive

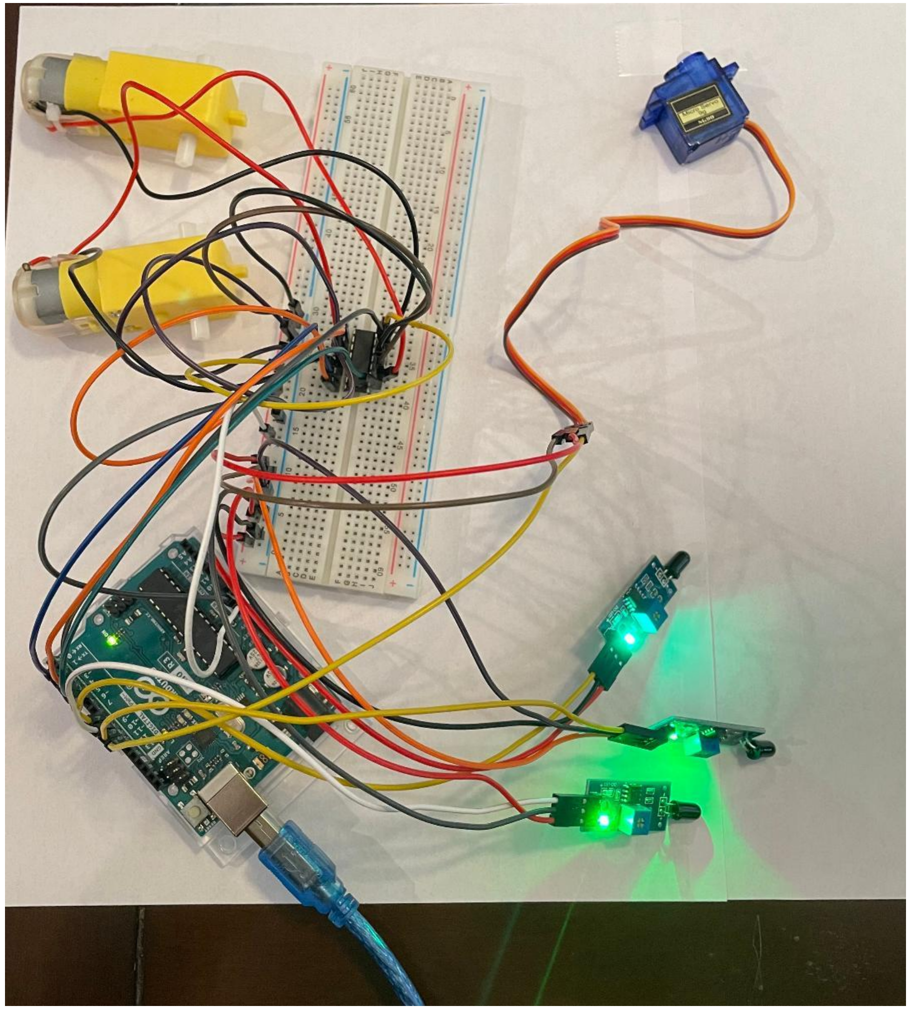


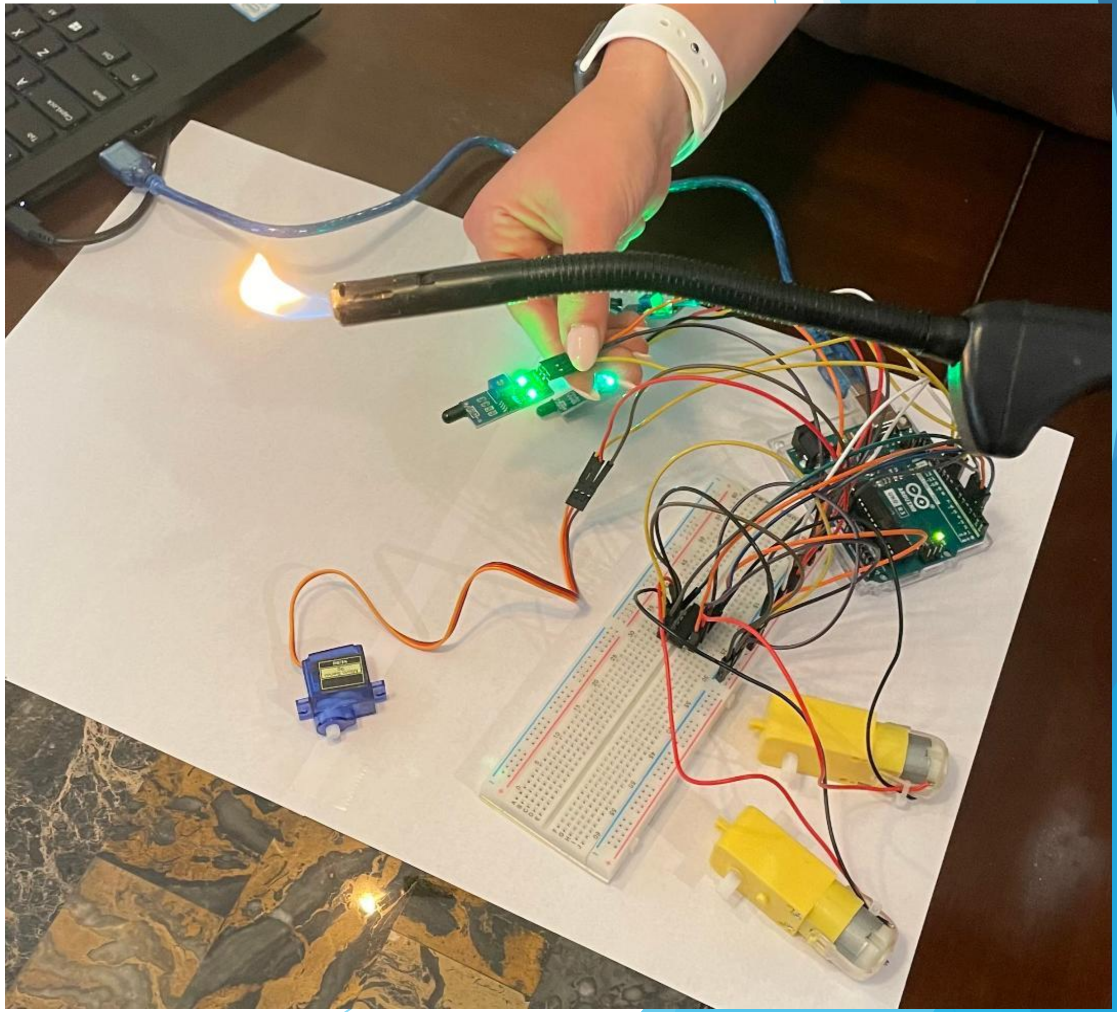
Arduino Uno:

- Microcontroller board based off the ATmega328P
- 14 digital input/output pins (6 of which can be used as PWM outputs)
- 6 analog inputs
- 16 MHz ceramic resonator
- USB connection
- Power jack
- ICSP header and a reset button
- Bought from Amazon.com



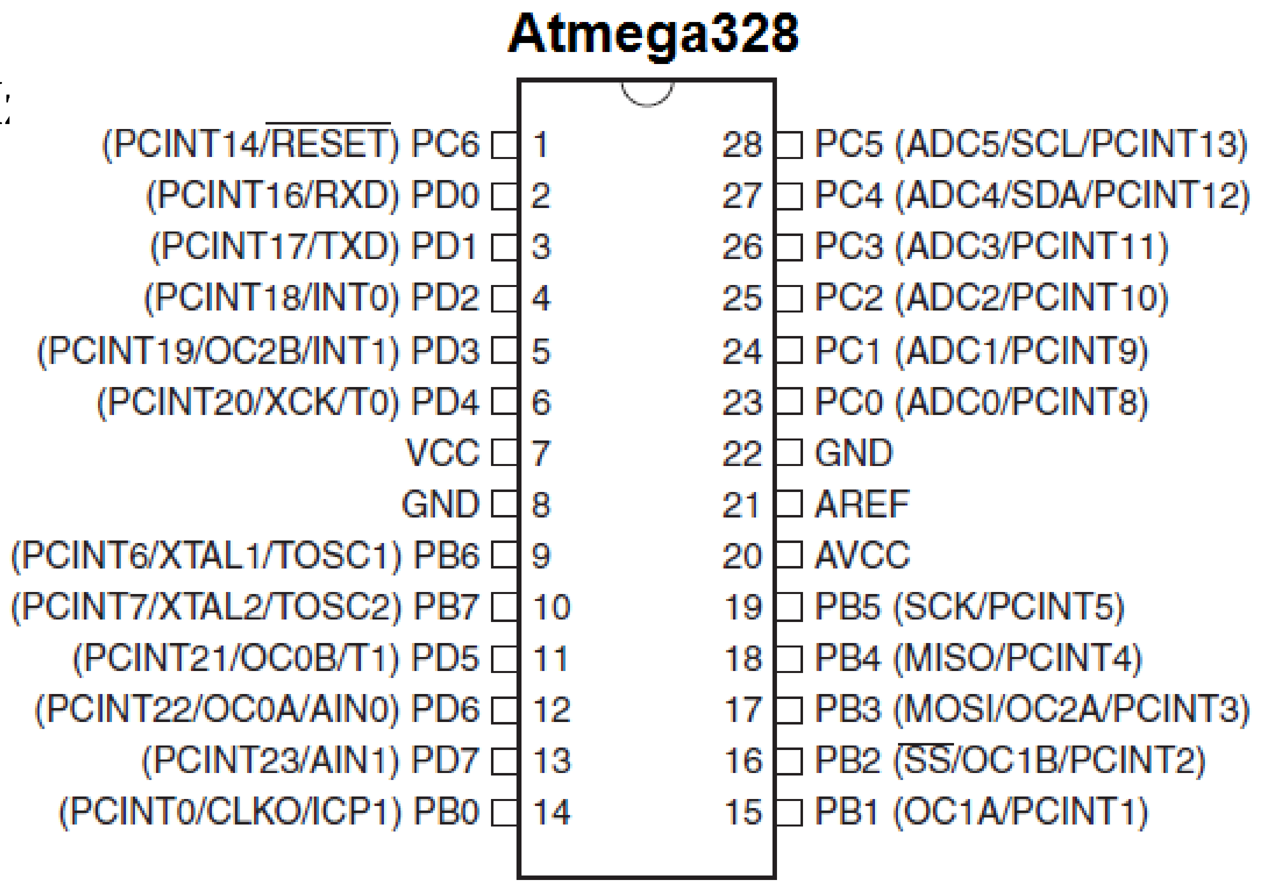
Breadboard Testing Using the Arduino Uno:





MCU:

- The ATmega328 is the microcontroller that powers the Arduino Uno
- 8-bit RISC processor core
- Clock frequency from 1 MHz to 20 MH
- 32 KB flash memory
- 2 KB SRAM
- 1 KB of EEPROM
- 23 GPIO lines
- 32 general purpose registers
- I2C, SPI, and Serial interfaces
- 28-pin DIP
- Bought from Amazon.com

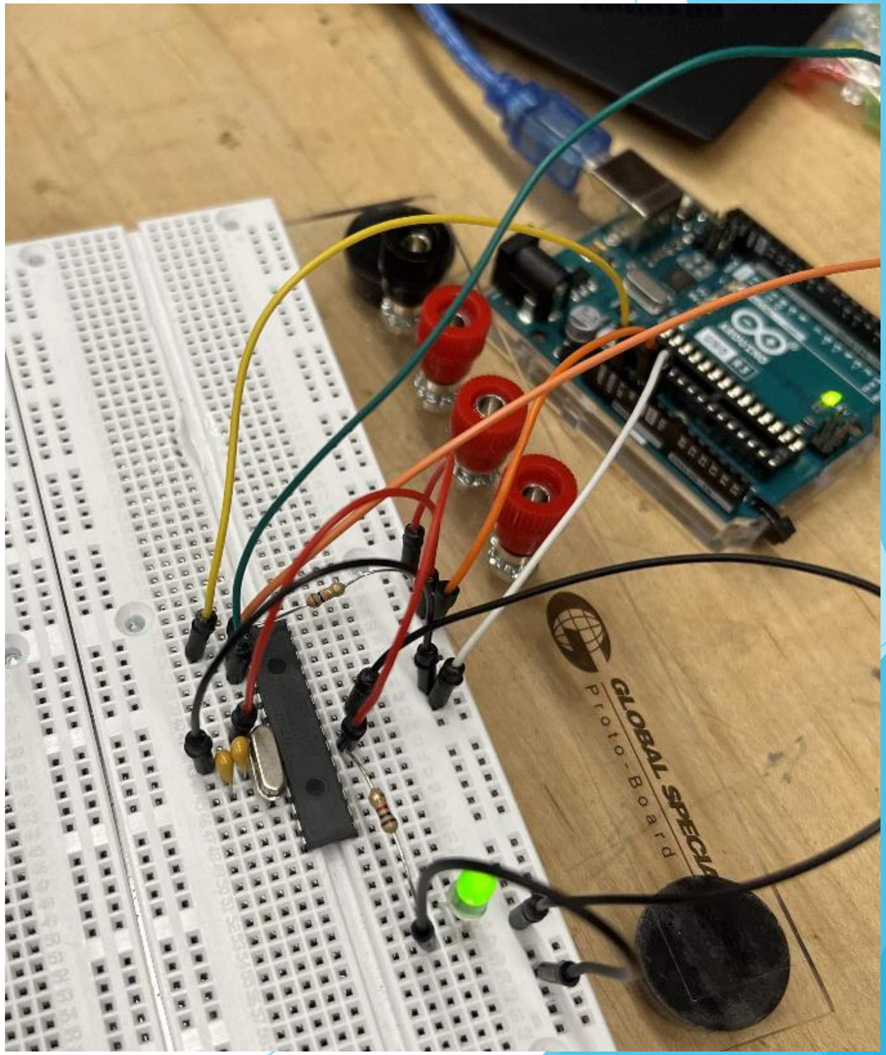


Arduino Bootloader:

- A bootloader is code that is burned onto the ATmega328 chips EEPROM
- The code is loaded when the processor is powered up or reset
- It sets the clock frequency, internal registers, etc.
- The bootloader allows the ATmega328 to accept programs from the Arduino IDE on its serial RX and TX pins
- ATmega328P-PU already has the bootloader installed

ATmega328 Connections:

- In order for the chip to function like an Arduino, a few steps need to be taken:
  - ATmega328 has to be wired with a 16 MHz crystal, a 10 K resistor, and two 20 pf capacitors
  - The ATmega328 also needs to have a bootloader burned onto it

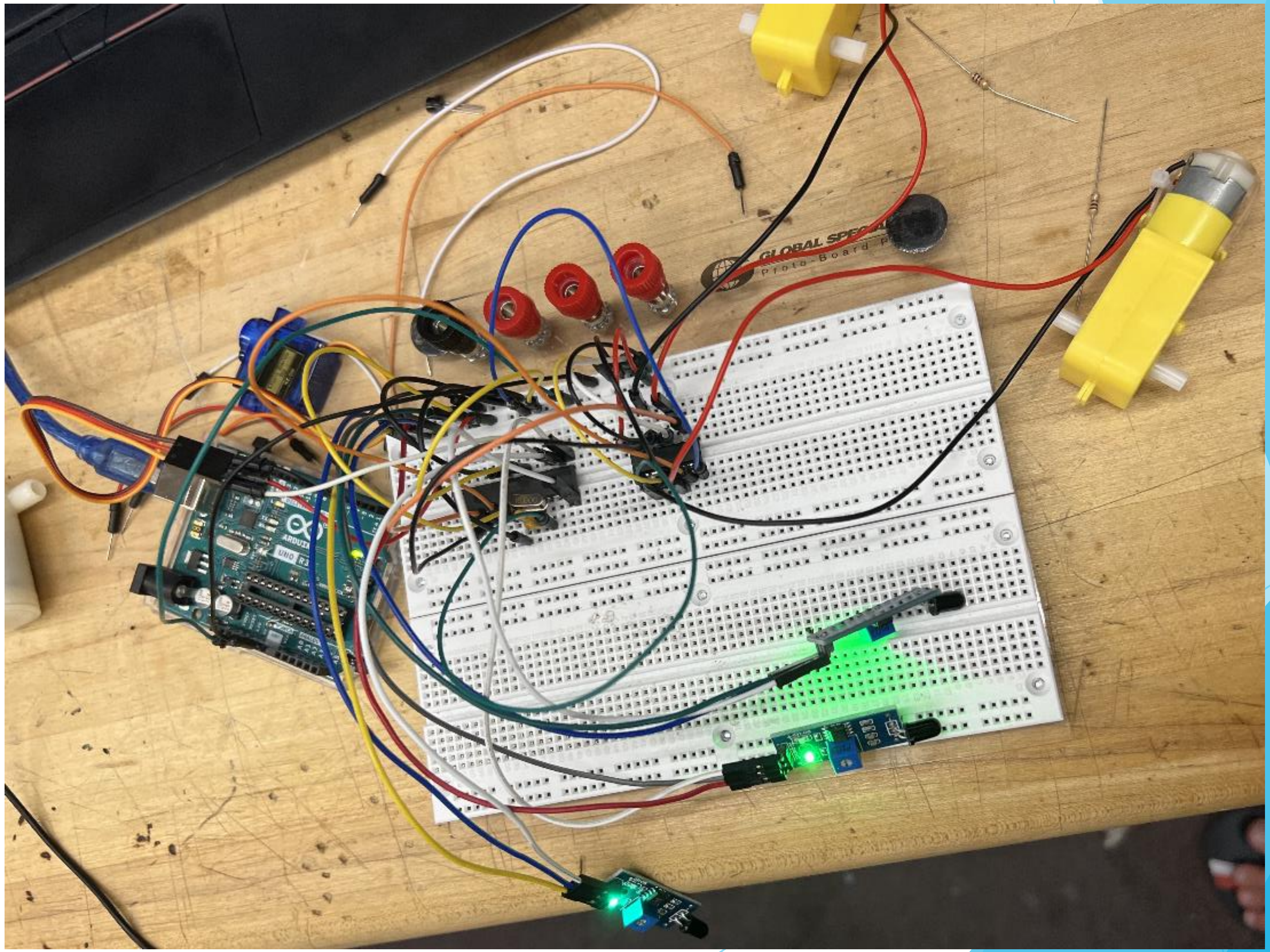


Loading Code Onto the MCU:

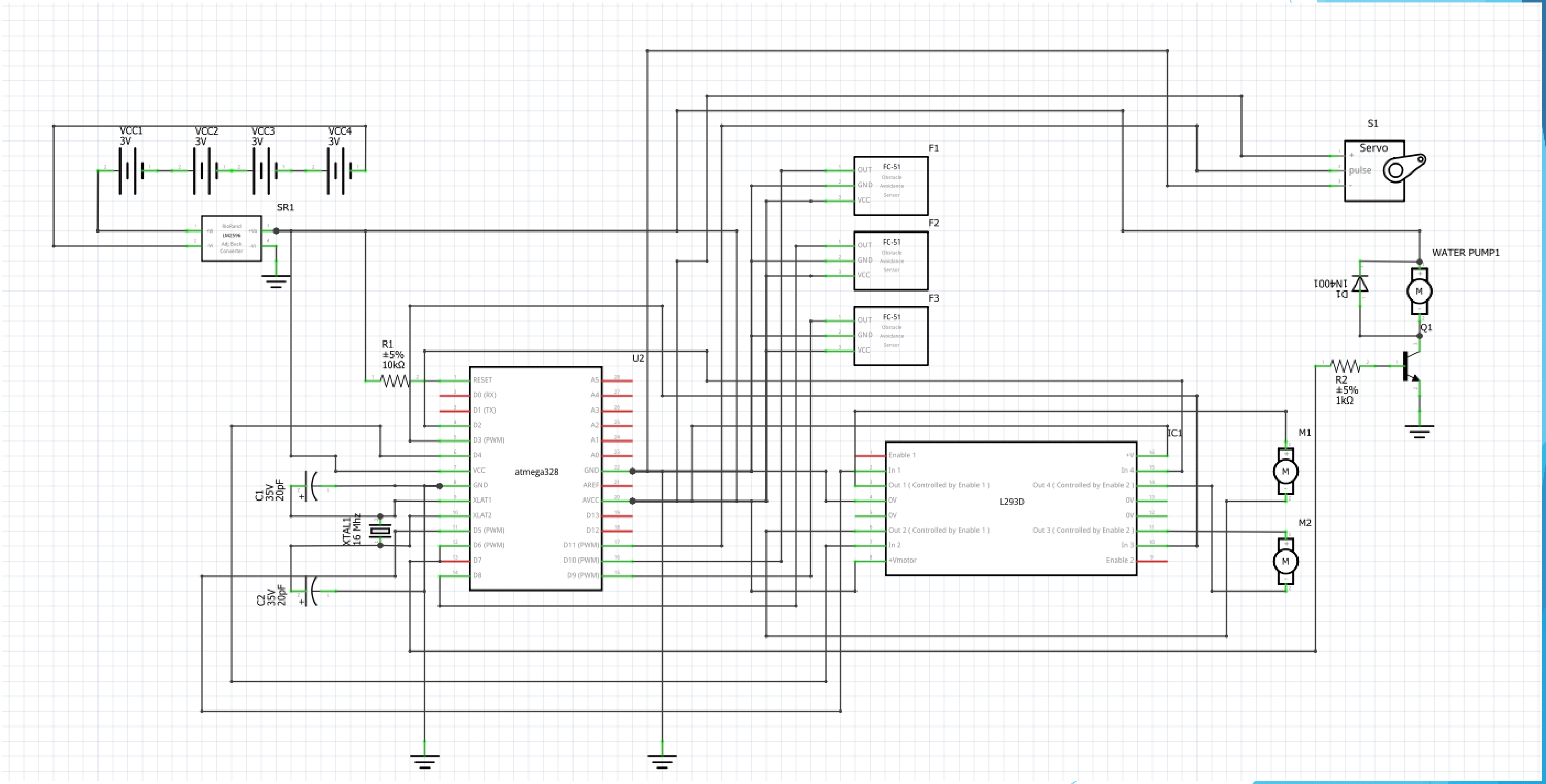
- There are 3 different ways to load a sketch into the MCU of the Arduino
- Method 1 – Use ATmega328 from Arduino
- Method 2 – Use the Arduino as a Serial Connection
- Method 3 – Load the program by using an FTDI Adapter
- The first two methods require the original Arduino Uno that has the mounted IC socket with the chip
- Our group used the first two methods to load our sketch into the MCU

Breadboard Testing Using the MCU:

- Testing done with just the MCU of the Arduino
- Code was downloaded into the chip when it was connected to the Arduino and then removed and placed on the breadboard to do further testing
- All the components tested worked as expected

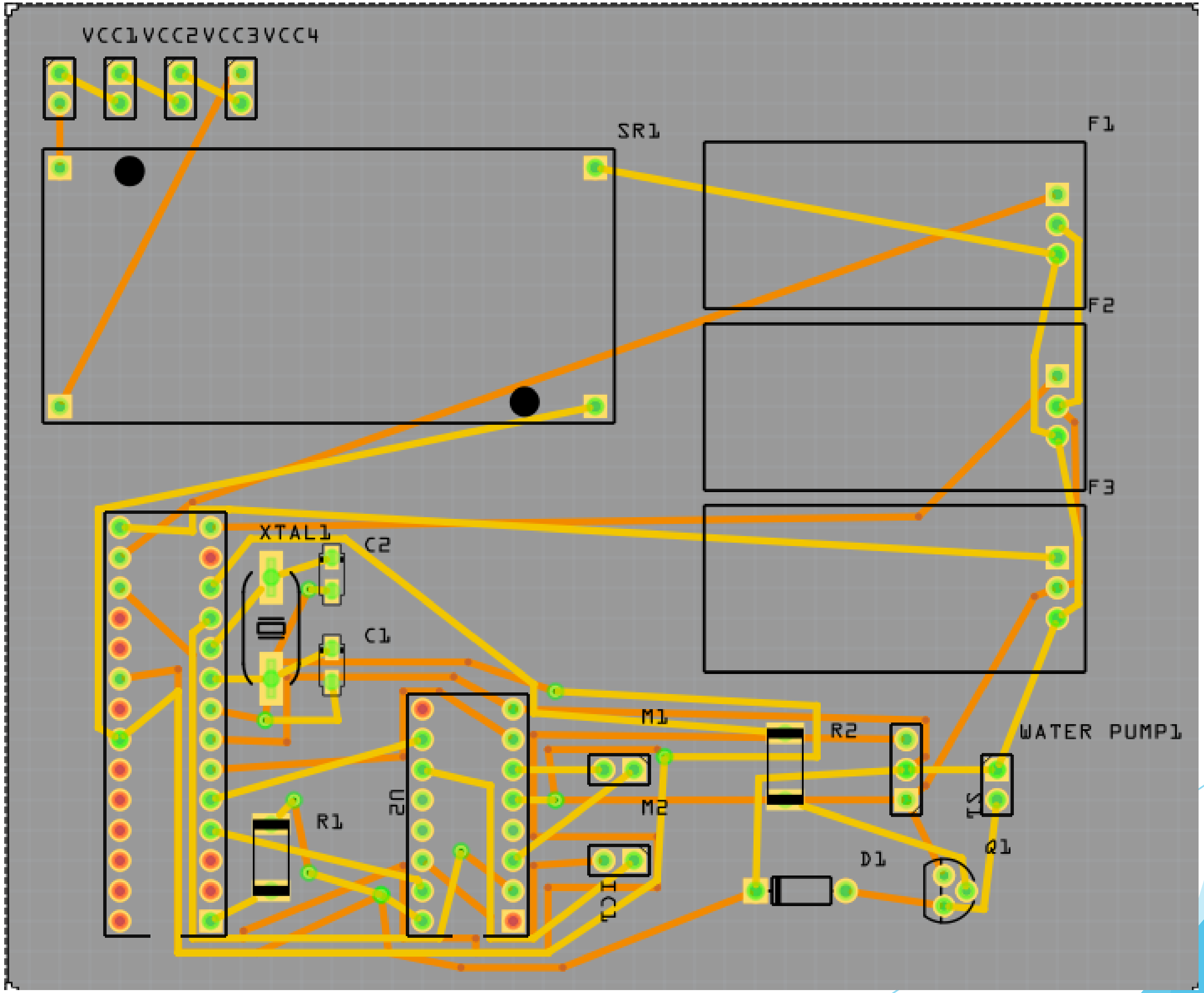


Schematic:



VCC1
3V
VCC2
3V
VCC3
3V
VCC4
3V
SR1
S1
Servo
pulse
F1
F2
F3
FC-51
OUT
GND
VCC
WATER PUMP1
Q1
R1
±5%
10kΩ
U2
atmega328
RESET
D0 (RX)
D1 (TX)
D2
D3 (PWM)
D4
VCC
GND
XLAT1
XLAT2
D5 (PWM)
D6 (PWM)
D7
D8
A5
A4
A3
A2
A1
A0
GND
AREF
AVCC
D13
D12
D11 (PWM)
D10 (PWM)
D9 (PWM)
R2
±5%
1kΩ
IC1
L293D
Enable 1
In 1
Out 1 (Controlled by Enable 1)
0V
0V
Out 2 (Controlled by Enable 1)
In 2
+Vmotor
+V
In 4
Out 4 (Controlled by Enable 2)
0V
0V
Out 3 (Controlled by Enable 2)
In 3
Enable 2
M1
M2
C1
35V
20pF
C2
35V
20pF
XTAL1
16 Mhz

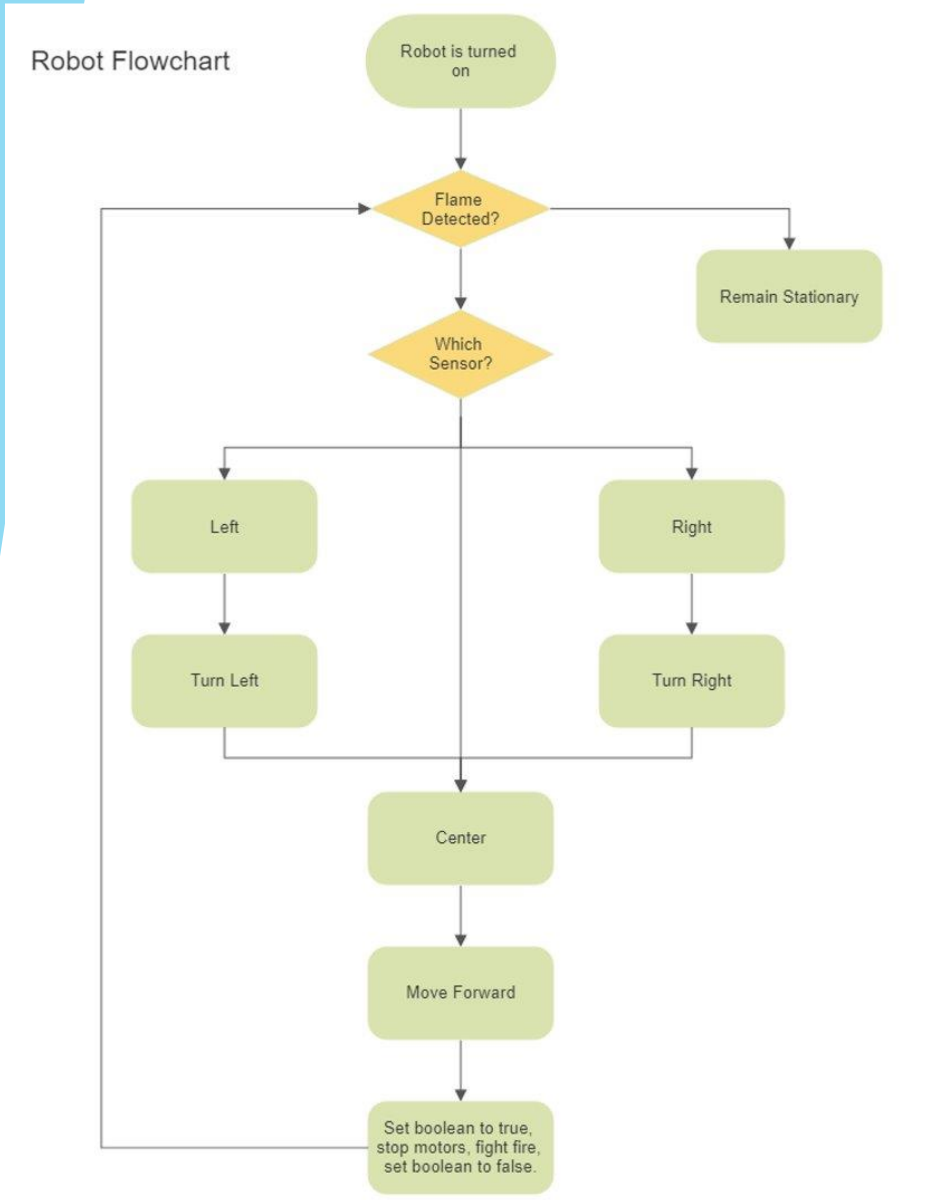
PCB:


VCC1 VCC2 VCC3 VCC4
SR1
F1
F2
F3
XTAL1
C2
C1
M1
R2
WATER PUMP1
M2
R1
U2
IC1
D1
Q1
S1

Software:

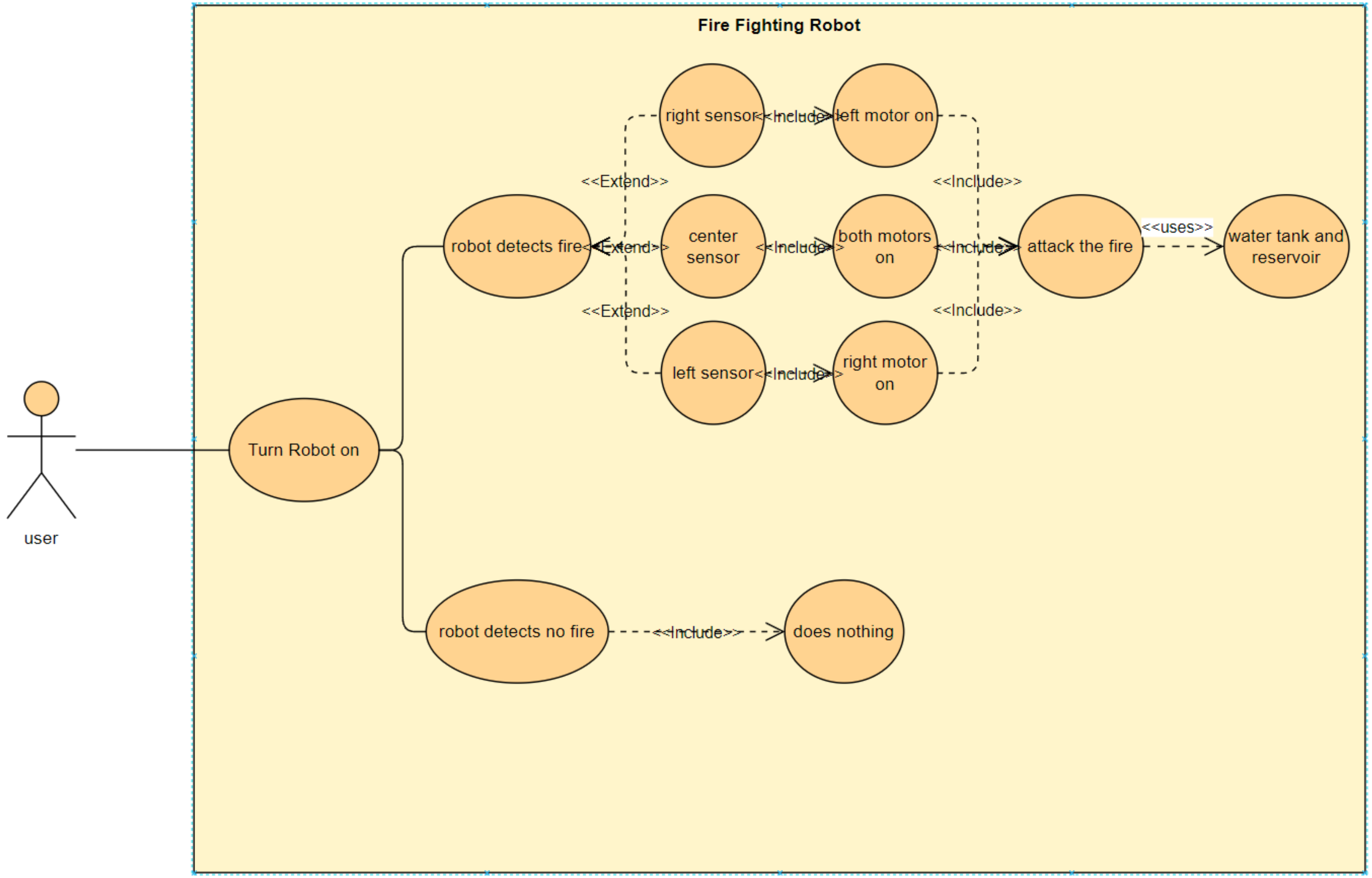
- The software is broken up into a few stages
  - Setup
  - Extinguish
  - Sensor input/Motor output
- **Setup**: This is the first part of the code where the parts are defined, and their pins are set so the micro controller knows which part is which. All parts of the robot are setup here: sensors, motors, servo, and pump
- **Extinguish**: The robot will stop movement, turn the pump on, and begin servo movement causing a sweep of sprayed water in front of the robot. Then the robot goes back to its initial state waiting for IR sensor input.
- **Sensor input/Motor output**: The robot will determine where the fire is depending on which sensors are turned on, it will then position itself near the fire using the motor controllers, and then begin the extinguish function.

Robot Movement Flow Chart:



- The Goal was to keep the software as simple as possible while still performing all functions
- This allows us to make easy upgrades after the working prototype is implemented
- There are two things the software evaluates:
  - Is there a flame?
  - If so, where is the flame
- The robot will then use one of 3 loops to respond until the fire is extinguished.

Use Case Diagram:



- The only user interaction is turning the robot on
- The robot will monitor its surroundings and respond accordingly
- After positioning itself the bot will use the pump water reservoir to dispense water and attack the fire.
- Following its initial attack, it will re-evaluate and respond accordingly.

Design Constraints:

- **Economic**: Due to the current supply shortages we have had to adjust from our original plans for our project to incorporate parts that are more easily acquired. We have also had to pay close attention to prices as our project is funded out of pocket, and with the rising costs and limited part availability we have had some challenges getting what we need.
- **Environmental**: Environmental constraints have been quite an issue as some of our team members don't live close making it hard to meet at specific locations. This also can be a constraint when the lab is busy or full making testing and working on the robot difficult. There are also limited areas where we can test the fire-fighting capabilities of our robot.
- **Health & Safety**: Safety measures have died down significantly, but Covid-19 is still an ongoing threat to us and the people around us. We are also testing a robot that extinguishes fire which in and of itself can be dangerous to work with, especially considering the limited range of the robot putting us in closer working proximity to the flames.

UCF

Design Constraints cont.:

- **Time**: This is the most crucial and difficult constraint to work around. Our team members have to balance school, work, travel, and other responsibilities. This has made team meetings incredibly difficult. Thankfully, using Discord we have been able to keep in contact with each other even when we can't meet.
- **Sustainability**: Unfortunately, due to other constraints we haven't been able to source parts that are specifically sustainable. Our project can also have particularly detrimental sustainability affects given the use of fire in testing and battery packs used in operation.
- **Manufacturability**: This constraint is a balancing act for our project because we have to balance an on-board water reservoir with electronics, and the robot getting in close proximity to flames. The availability of parts has been the biggest influence on this constraint.
- **Social, Political & Ethical**: There are not many social or ethical constraints to more efficiently and safer way of putting out fires. However, it does put the number of jobs for firefighters at risk given that with enough budget and research it could encroach on their job tasks.

Related Standards:

- There are a few organizations that cover standards relating to our project
  - American National Standards Institute (ANSI)
  - International Organization for Standards (ISO)
  - National Fire Protection Association (NFPA)
  - Association Connecting Electronics Industries (IPC)
- The Robotic Industrial Association (RIA) covers the standards for all Industrial Mobile Robots (IMR)
- Aside from those organizations there are standards organizations that cover individual parts of our robot as well
  - PCB: Institute for Interconnecting and Packaging Electronic Circuits (formally known as Institute of Printed Circuits)
  - Battery: International Electrotechnical Commission
  - USB: the USB Implementers Forum
  - C Language: ISO/IEC 9899:2018

General Project Successes:

- **The research phase** was smooth and for the most part our team handled the workload well and divided up the writing well.
- **The software** was easy to manage and worked well with the Arduino Uno that we used for testing the board.
- **The hardware team members** were all able to meet regularly to test the robot through the development stage until we had a working robot. There were some difficulties through this phase, but being able to regularly collaborate in person made the process much smoother.
- **Communication** was a strong suit of our team. Using Discord as our communication app we were able to effectively communicate and keep each other up to date on what we were working on as well as what issues we were facing so we could collaborate and find a resolution quickly.

UCF

Project Difficulties:

- **Part availability** was difficult as many of the parts we wanted were unavailable due to supply issues, so we had to choose other parts. This difficulty was eased though because we had all summer to order parts rather than taking consecutive semesters.
- **Teammate availability** was another issue that we have been running into. This was not an issue in the first semester of senior design since as it was mostly research, and it could be done without in person meetings. However, this semester with the building and development of the bot we have needed more in person interaction, but some of the teammate's schedules haven't lined up, making in person meetings more difficult.
- **Part compatibility** was an issue with some of the parts not working together. Some didn't meet certain voltage requirements and made getting the robot functioning more difficult than it could have been.

UCF

Budget & Financing:

Unfortunately, our project did not have a sponsor so the funding for the Automatic Firefighting Robot has been entirely out of pocket thus far. One team member will buy the item we need and the rest of us will reimburse them for ¼ of the item via Cash App or Venmo.

Budget Breakdown

Team member	Portion
Gabriel Martinho Bizuti	25%
Sydney Brown	25%
Juan Cuervo	25%
Noah Schrock	25%
Total	**100%**

UCF

Budget & Financing:

UCF

Cost Breakdown

Part Description	Quantity	Price
Arduino Uno	1	$22.79
Fire/Flame Sensor	3	$17.07
Servo Motor	1	$10.99
L293D Motor Driver Module	1	$9.50
Mini DC Submersible Pump	1	$11.39
Small Breadboard	1	$0
Robot Chassis (with 2 motors and 2 wheels)	1 Chassis 2 motors 2 wheels	~$20
Small Can/Water Holder	1	$0
Connecting Wires	Several	$0
Ceramic Capacitors	230	$12.99
Quartz Crystal Oscillators	5	$6.99
UNO R3 bootloader	1	$29.99
Total		**$121.71**

This is not the final breakdown of all the parts ordered as some did were not compatible, so they were substituted with parts that were. These are the parts that are currently being used but this chart is subject to change as the project evolves.

Administrative Content

Immediate Plans:

- The immediate plans are mostly concentrated around the hardware, mostly focusing on the central components. The most immediate plans are focused on resolving four main points that are causing slight setbacks.

1. **Submersible Water Pump:** The water pump is currently not receiving enough current from the circuitry and so it does not work with the current set up. This is a main concern since the whole point of the robot is to put out fires.

2. **Regulator:** Currently the group is trying to determine which regulator would be best for the robot. The decision as of now is between a switching regulator or a linear regulator.

3. **Power:** So far, all the testing has been done with 5V, but the group is currently deliberating as to what would be the best level of voltage to power the robot for it to last wirelessly.

4. **PCB:** The group has run into some difficulties in the designing of the PCB. This is a top priority that the group is presently focused on addressing so that no further setbacks take place in the building of the robot.

Longterm Upgrades:

- **Additional sensor FOV**: Adding more Infrared sensors around the robot will allow it to see more of the area around itself allowing it to respond to fires that are just out of view of the current configuration or even behind itself with the right sensor orientation.
- **Upgraded Chassis**: Currently the electronics of the bot are exposed which presents not only a cosmetic issue, but a functional one as well. There is a water tank on board the robot use in fighting the fires and without a shell on the top of the bot to cover the electronics it presents a potential hazard especially as the robot moves about.
- **Spray Nozzle:** Our current version has the robot push water out the end of a tube from the water pump which is currently being worked on. Once the pump is added to the bot and working properly, we would like to test some spray nozzles that change the flow and pressure of the water. Determining which fights fires the best using the small robot form factor.

Longterm Upgrades:

- **Mobility**: Right now, our bot is designed to stay stationary until it sees fire, while this would be useful with spreading fires, it gives the fire a chance to grow before it goes into action. We plan to add a mode where the robot will actively roam around and search for fires. This requires significant code changes and the addition of ultrasonic sensors so it can sense objects around itself.
- **Onboard battery**: Balancing the power draw and weight of all the components of the robot including an onboard battery is one of the tasks we are aiming for before we change the behavior of the bot. Having the bot wander around looking for a fire isn't feasible while the bot is tethered to a power source from a cable.
- **Power/Mode switches**: Once we have the battery pack and code written for the robot's different modes, we would like to allow users to choose the behavior of the bot without having to reprogram it by adding some switches or buttons that allow the user to choose a mode for the robot.

UCF

Work Distribution Overview:

	Gabriel	Sydney	Juan	Noah
Peripheral Hardware	X			
Central Hardware		X		
Embedded System			X	
Software Design				X

Work Distribution Specifics:

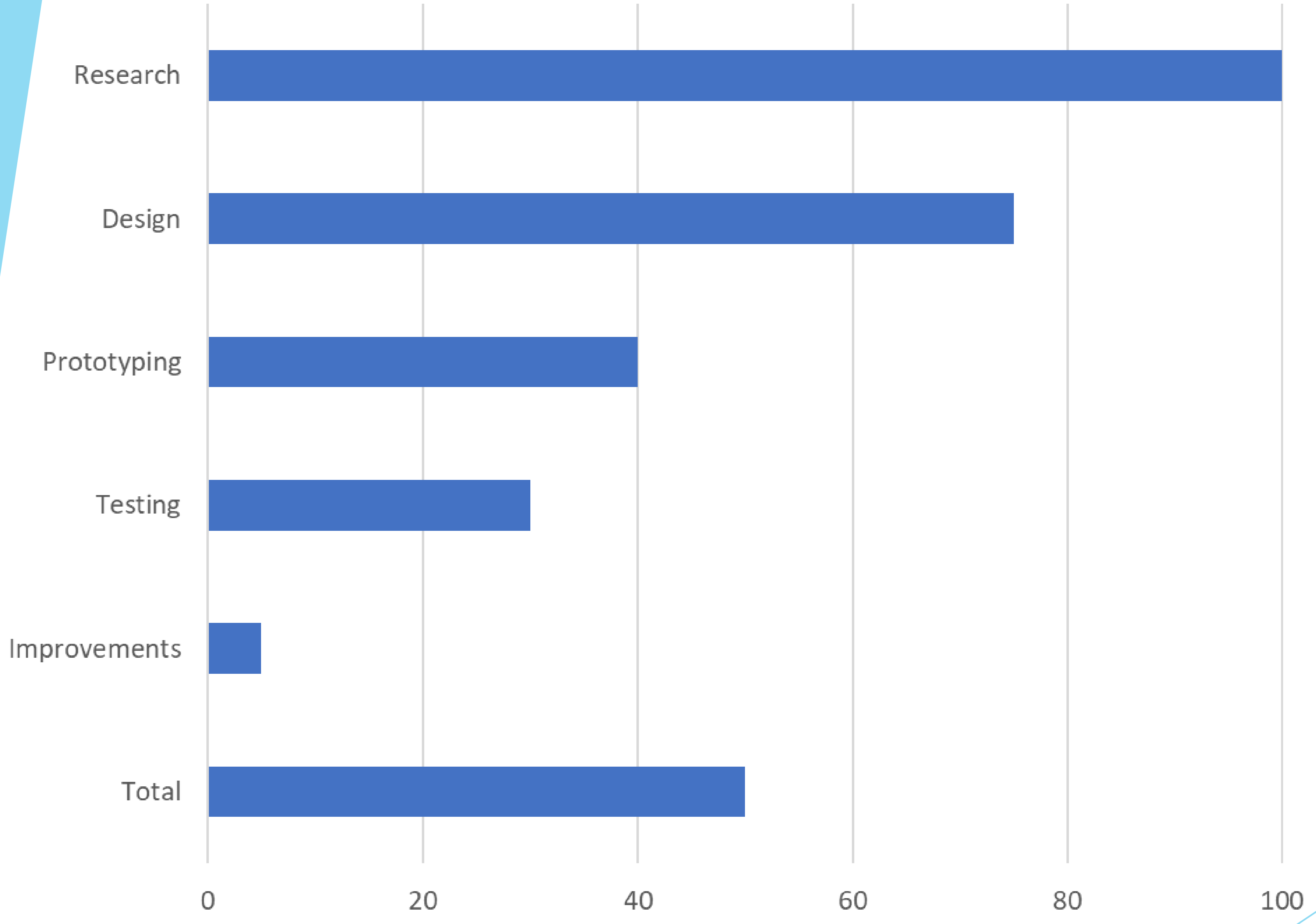
- **Gabriel:** Gabriel has specifically worked on those components of the robot that encompass the peripheral hardware. Gabriel's focus thus far has been on how the fire IR sensors, the motors, and the submersible water pump work individually as well as in relation to the central hardware (MCU, power, and motor driver).

- **Sydney:** Sydney's role has been primarily focused on the central hardware component of the robot, which includes how the MCU operates individually, how the motor driver communicates with the motors and water pump, and how the robot is powered. Also, Sydney has focused on designing the first PCB.

- **Juan:** Juan has worked primarily on the MCU's communication with the other hardware in the robot. This has consisted in making sure that the hardware and software designs are compatible and that the circuit connections ensure that all the peripherals respond appropriately to the software.

- **Noah:** Noah's role has mainly consisted of the software component of the robot. This role includes writing and modifying the logic in the code to ensure that the robot responds accurately to obtain the final goal of the project, given the powers and constraints of the peripheral and central hardware. He also 3D modeled and printed the chassis/shell of the robot.

Progress Overview:



Progress Specifics:

- **Research**: This phase is fully complete. No further research is needed.
- **Design**: The design for most of the robot has been completed except for a fully completed PCB, the chassis design, and the water pump.
- **Prototyping**: Some of the prototyping has been started, and so far, the group has a robot with fully functioning wheels, flame sensors, and servo motor in response to the code. Also, the basic software for the robot is fully completed.
- **Testing**: Testing for the robot has been approached in a series of steps.
  - **Individuals**: All individual parts have been fully tested and are operational.
  - **Mobility**: The motors have been fully tested and are operational.
  - **Vision**: The sensors have been fully tested and are operational.
  - **Fire Suppression**: The water pump has been tested but is still not operational in conjunction with the rest of the robot.
  - **Full Robot**: The full robot has not been tested yet, as it has not yet been built.
- **Improvements**: Some improvements on the original design have been considered and added into the final design, but none have been physically implemented yet.

Progress Specifics:

Task/Milestone:	**Date Completed:**
First Meeting	August 29, 2022
Basic software finished	August 30, 2022
All individual parts tested	August 30, 2022
Vision (sensors) functional	September 8, 2022
Servo motor functional	September 12, 2022
Motors functional	September 18, 2022
First PCB design	September 20, 2022

Note: When the word "functional" is used in the table, it designates a milestone in which the part properly and accurately responds to the software in a way which satisfies the end goal of the robot.

Tentative Progress:

Task/Milestone:	Target Date:
Water pump functional	September 30, 2022
First finalized prototype	October 7, 2022
Order PCB	October 7, 2022
Prototype full test	October 11, 2022
Test PCB	October 21, 2022
First full robot assembly	November 1, 2022
Testing and modifying	November 8-18, 2022
Finalize upgrades	November 18, 2022
Final tests	November 21, 2022

Note: When the word "functional" is used in the table, it designates a milestone in which the part properly and accurately responds to the software in a way which satisfies the end goal of the robot.

Questions?