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.

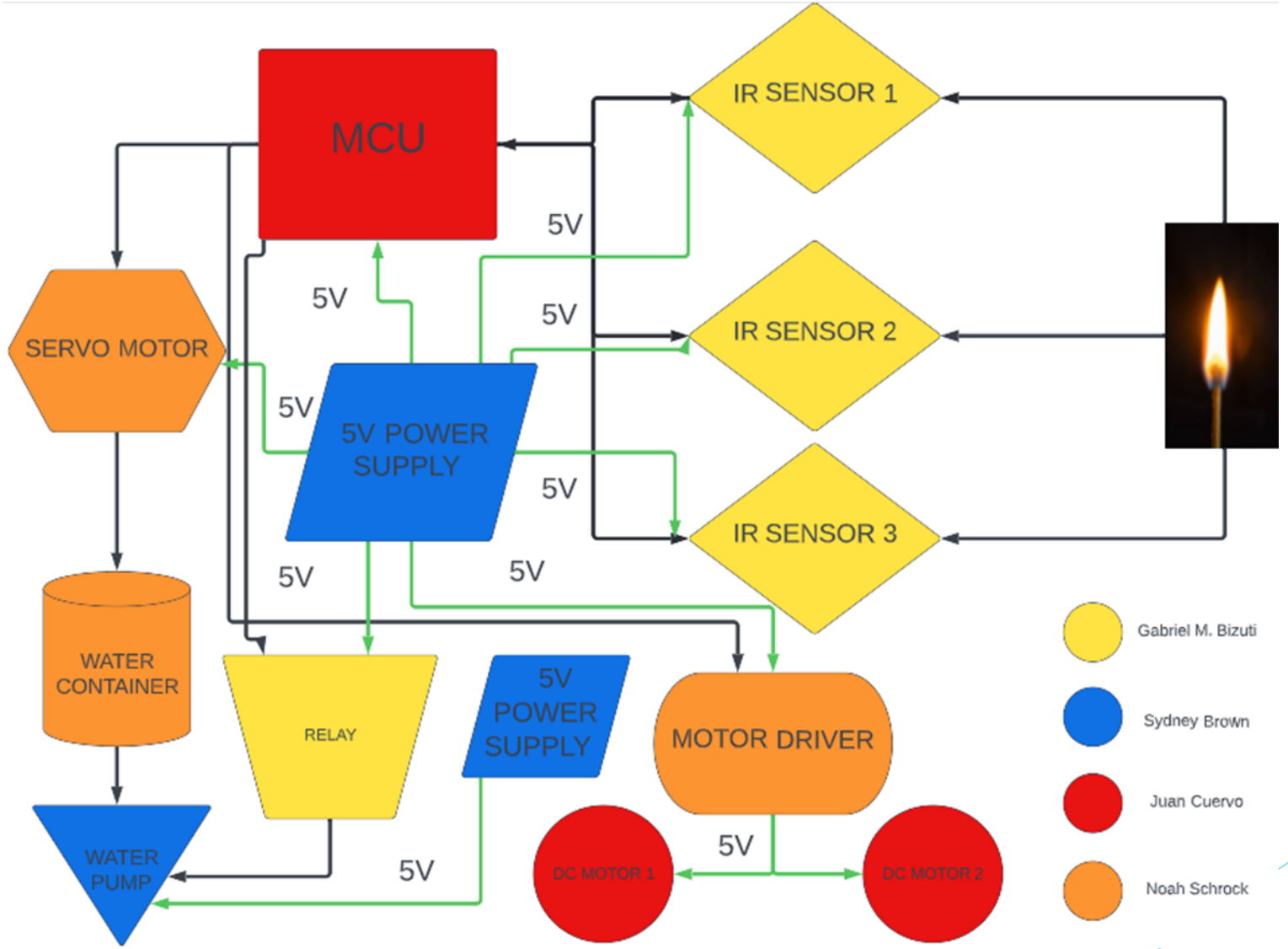
Goals and Objectives:

- 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:

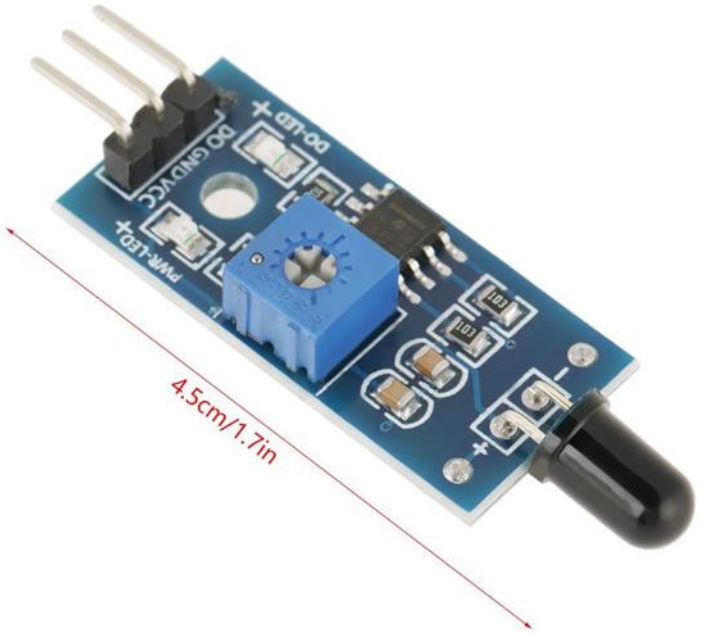
Specification	Description
Robot Size	Between 20 cm and 35 cm
Number of Sensors	3 IR sensors
Sensing Range	30 cm
Extinguish Time	3 Seconds
Movement Speed	Close to 2 Km/h
Response Time	Instantaneous once within range
Power Supply	5 V

System Block Diagram:



Infrared Fire Sensor:

- Function: Detect the presence of fire and its direction.
- Uses the infrared radiation emitted from fire to detect it.
- All flames of all sizes emit infrared light that can be detect by the sensors.
- Detects infrared light wave between 760 nm and 1100 nm; interference is reduced.
- 3 pin sensor that includes:
  - Digital Output pin.
  - Ground pin.
  - Vcc pin.

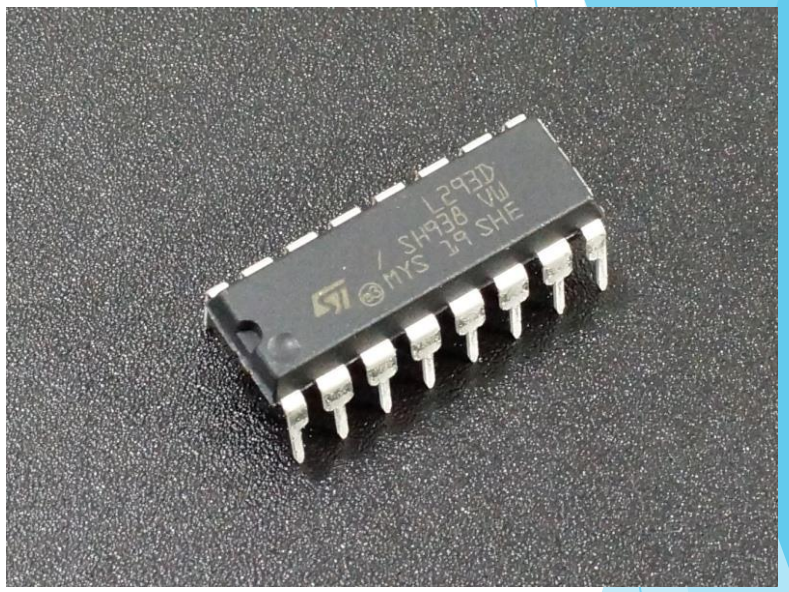


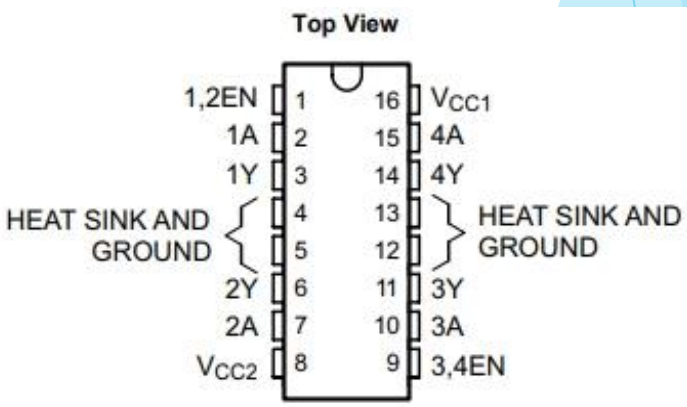
Infrared Fire Sensor Selection:

	Walfront IR Fire Sensor	HiLetgo IR Fire Sensor
Operating Voltage	3.3V to 5V	3.3 V to 5 V
Detection Angle	60 degrees	35 degrees
Wavelength	760nm to 1100nm	760 nm to 1100 nm
Sensing Range	35 cm	30 cm
Compatible with MCU	Yes	Yes
Price	$1.50	$0.879

Motor Driver IC:

- Function: The motor driver is responsible for controlling speed and direction of the DC motors.
- This driver is a general-purpose high voltage and high current component.
- Uses the H-Bridge circuit to control the rotation direction of the DC motors.
- The H-Bridge circuit uses 4 switches around the motor driver.
- When two switches close at the same time the polarity of the voltage applied to the motors change and therefore, the direction of the motors change as well.





Motor Driver Selection:

	TI L293D Motor Driver	ACEIRMC L293D Motor Drivers
Operating Voltage – Internal Logic (Vcc1)	5V	5 V
Operating Voltage - Motor (Vcc2)	4.5V to 36V	7V to 36V
Channel Output Current	Around 600 mA	Around 600 mA
Type	Dual H-Bridge	Dual H-Bridge
Compatible with MCU	Yes	Yes
Price	$0.95	$0.90

DC Motor and Wheel Set:

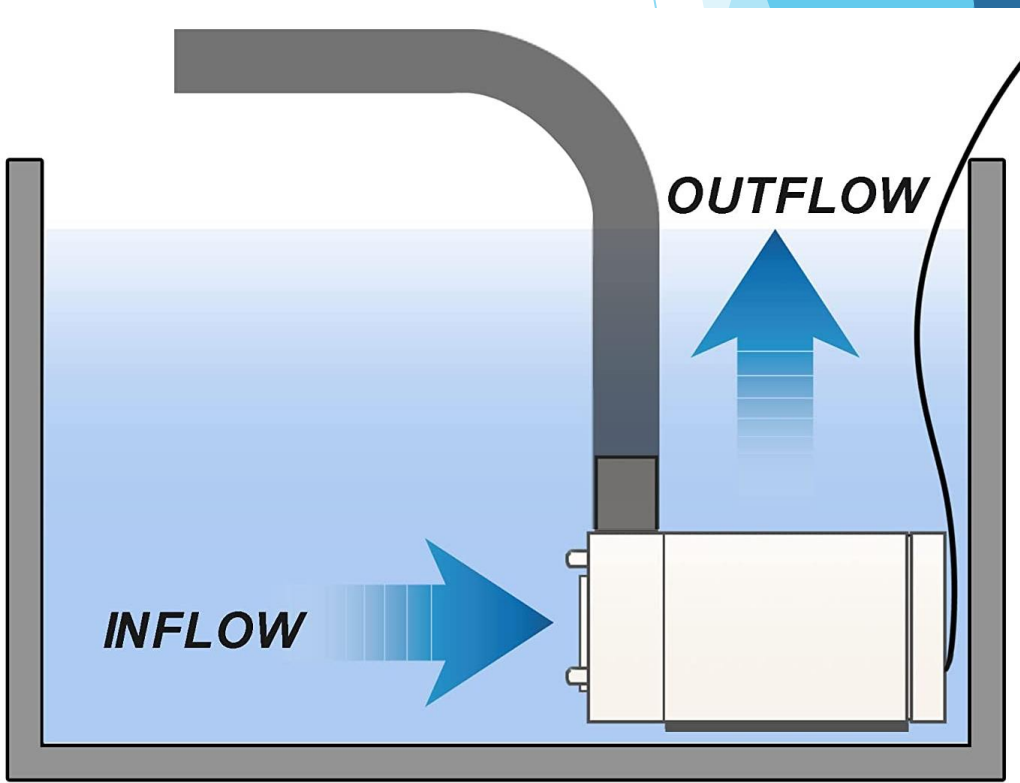
- Function: Move the autonomous robot close to the fire when detected.
- The motors are controlled by the motor driver and the microcontroller.
- Motor speed varies according to the amount of voltage supplied.
- Reversing the polarity provided to the motor can result in backwards movement.
- The motors can be easily attached to the side of the robot using screws and nuts for mounting.

DC Motor and Wheel Set Selection:

	ProtoSupplies DC Motor	S-snail DC Motor
Operating Voltage	3V to 12V	3V to 6 V
Typical Motor Current	140mA	150mA
Motor Speed at 5V	180 RPM	160 RPM
Compatible with MCU	Yes	Yes
Price	$6.50	$8.99

DC Mini Water Pump:

- Function: Pump water when close to the fire to put it out.
- Important for the water pump to work as expected because it is a crucial component to the design and the goal of our project.
- The flowrate of the pump depends on how much current it is being supplied.
- The water pump is going to be controlled by the MCU.

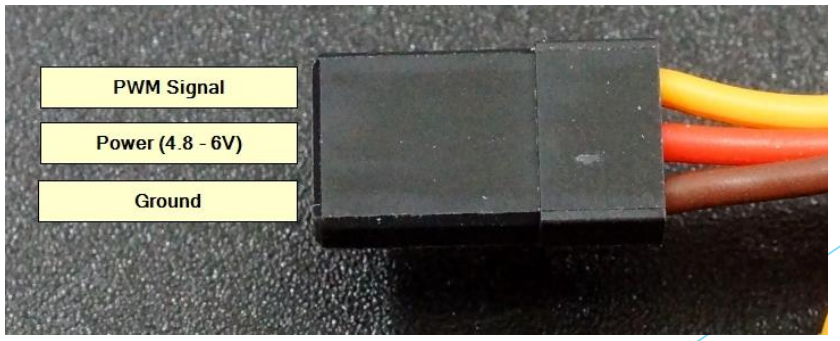


DC Water Pump Selection:

	ALAMSCN Micro Submersible Mini Water Pump	OCESTORE DC Micro Submersible Mini Water Pump
Operating Voltage	3V to 5V	3V to 6V
Current Consumption	150mA	150mA to 250mA
Water Flowrate	100 L/H	80 L/H
Lifting Height	40 cm to 110cm	40 cm to 110 cm
Price	$2.85	$3.40

Servo Motor Micro SG90:

- Function: Rotate the water container to cover a wider range with the pump water spray.
- The rotation speed of the servo is determined by the voltage it receives.
- The servo is driven by a PWM signal that is received from the MCU.
- Good option to use for spinning the water container.
- Typical operating voltage is 5V.
- The servo motor has 3 connections:
  - PWM Signal received from the MCU.
  - Power connection.
  - Ground connection.



Servo Motor Selection:

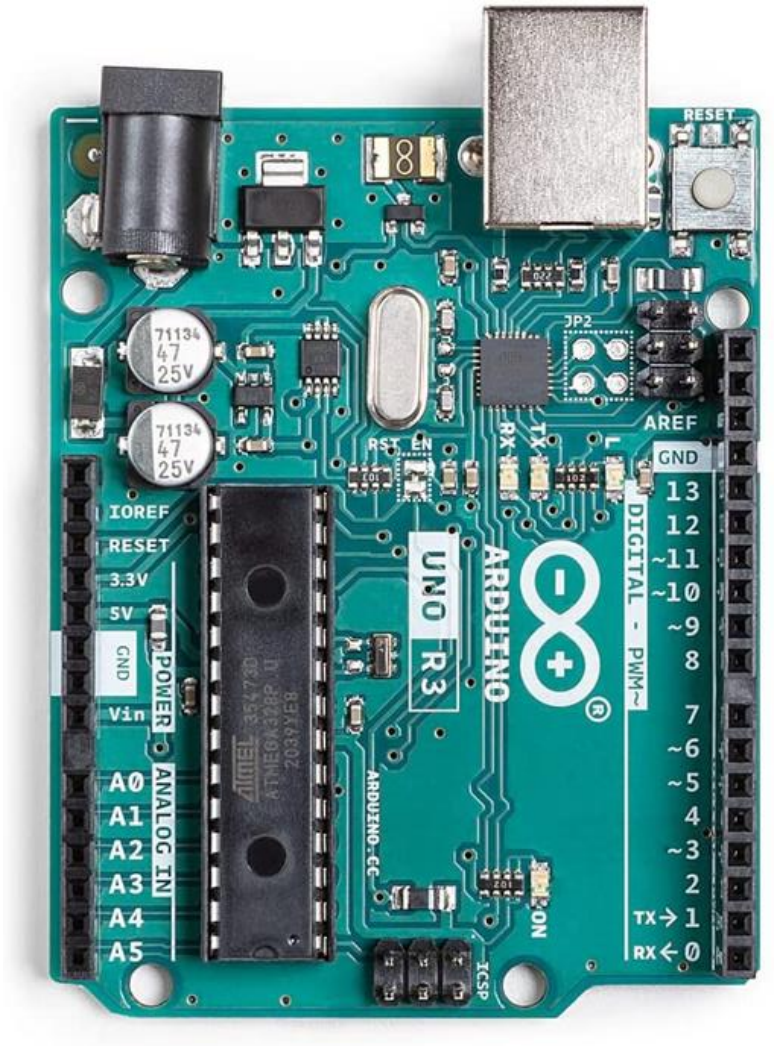
	ProtoSupplies Micro SG90 Servo Motor	Smraza Micro SG90 Servo Motor
Operating Voltage	4.8V to 6V	4.8V to 6V
Angular Rotation	360 degrees	180 degrees
Rotation Speed	120 RPM	100 RPM
Typical Current	100mA to 250mA	100 mA to 200mA
Compatible with MCU	Yes	Yes
Price	$3.79	$2.50

Power Supply:

- Anker PowerCore 10000 PD Redux
- Capacity of 10000 mAh
- Trickle-charging mode for low-power devices
- Ports: 1 USB type C, 1 USB type A
- USB C Output: 5V/3A, 9V/2A, 15V/1.2A
- Maximum output of 18 W
- Weight: 6.8 oz
- Recharging time: 3.5 hours

Arduino Uno:

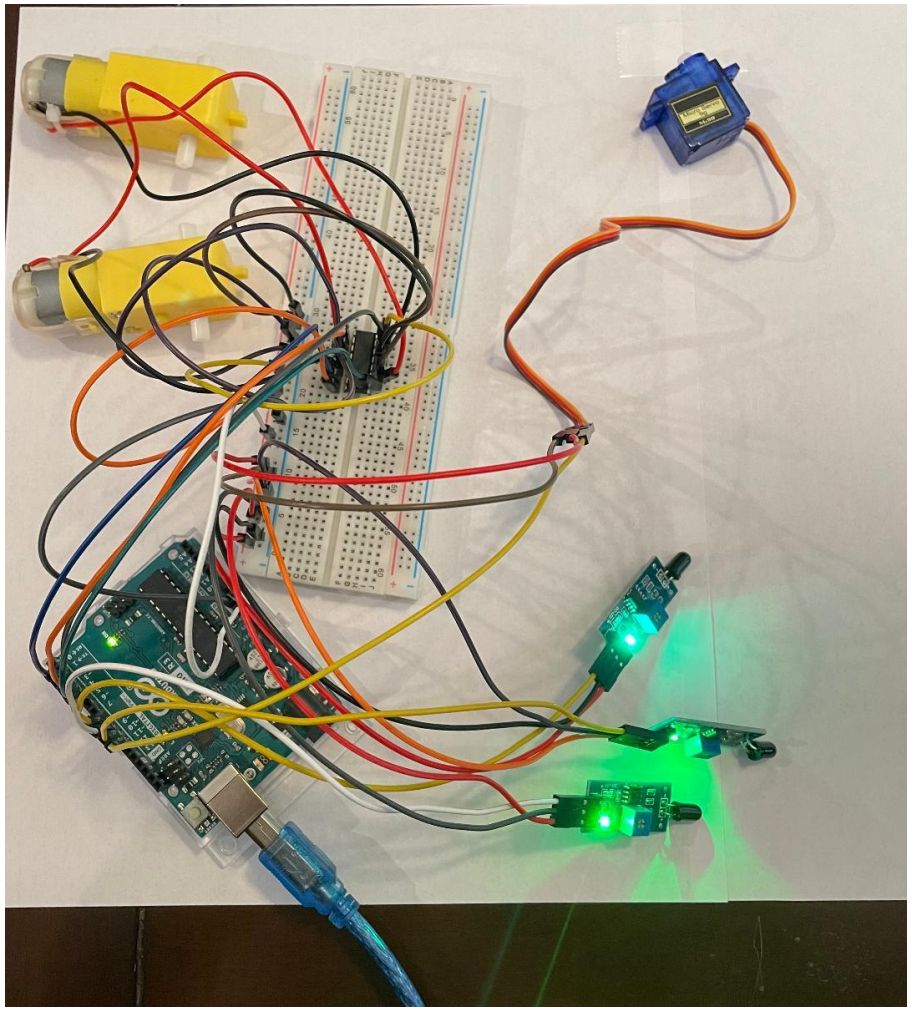
- Microcontroller board based off the ATmega328P
- 14 digital input/output pins (6 of which can be used as PWM outputs)
- 6 analog inputs
- 16 MHz ceramic resonator
- USB connection
- Power jack
- ICSP header and a reset button

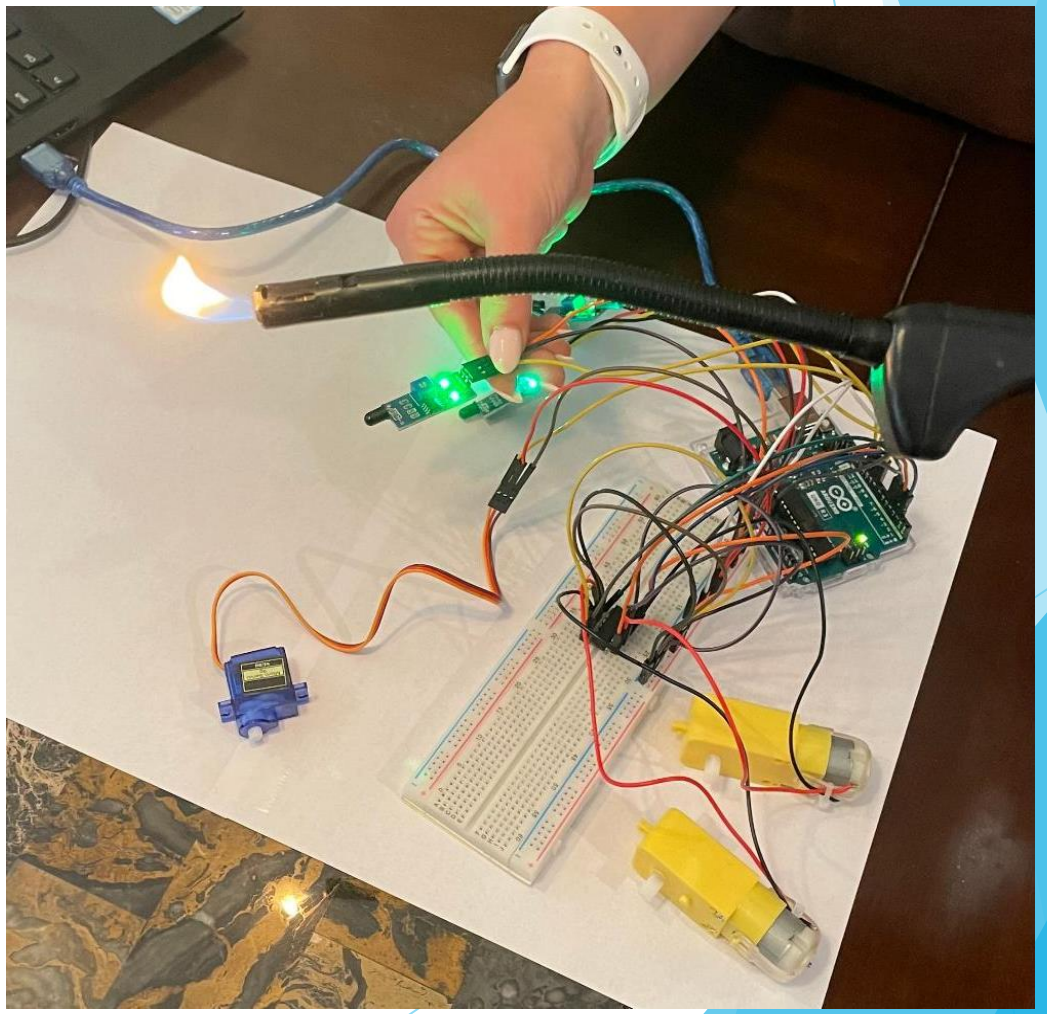


Microcontroller Selection:

	Arduino Uno – ATmega328 chip	Raspberry Pi 3 – RP 2040 Chip
Operating Voltage	5V	5 V
Maximum Total Current Drawn	20 mA	54 mA
Internal RAM	2 KB SRAM	1 GB DDR2
GPIO Pin Count	28-pins	40-pins
Power Consumption/Low Power	0.3 W Low Power MCU	3.7 W Low Power MCU
Price	$19.75	$35.00

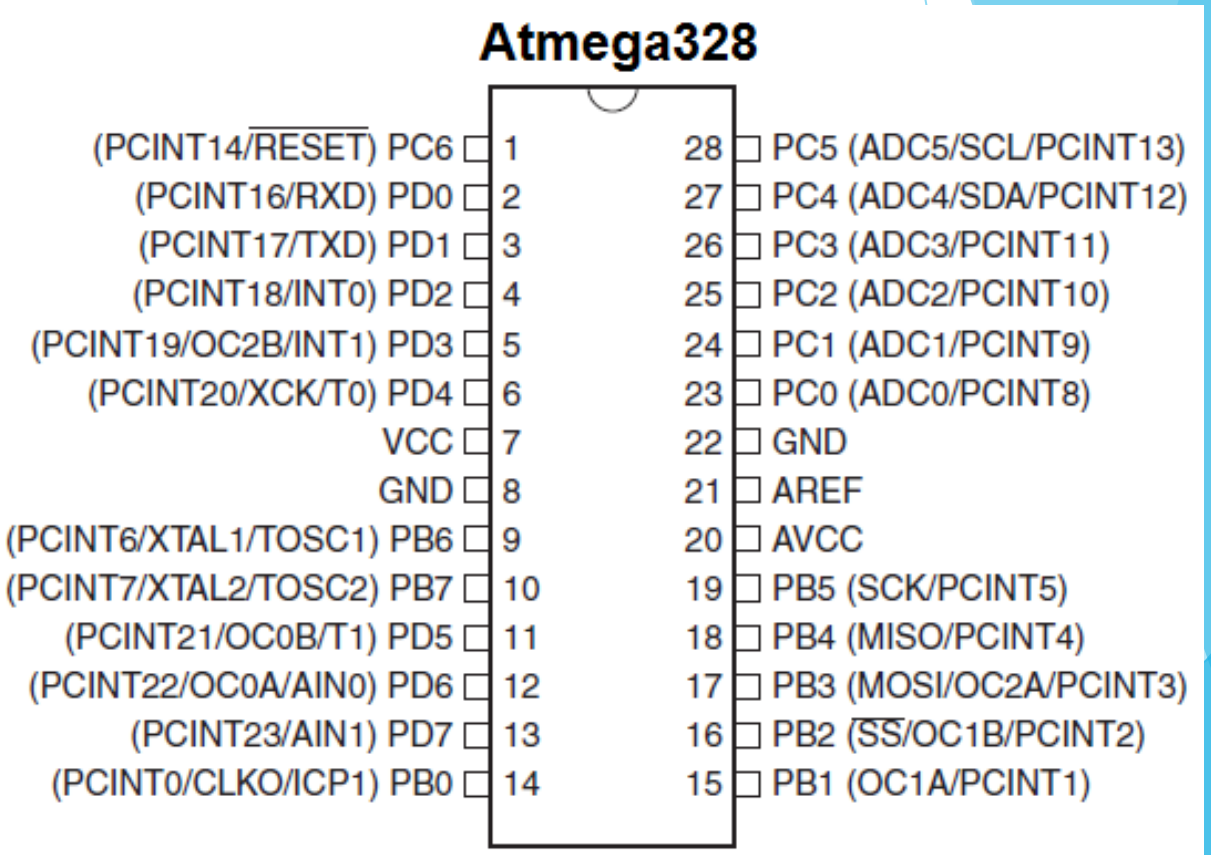
Initial Testing Using the Arduino Uno:





MCU:

- The ATmega328 is the microcontroller that powers the Arduino Uno
- 8-bit RISC processor core
- Clock frequency from 1 MHz to 20 MHz
- 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

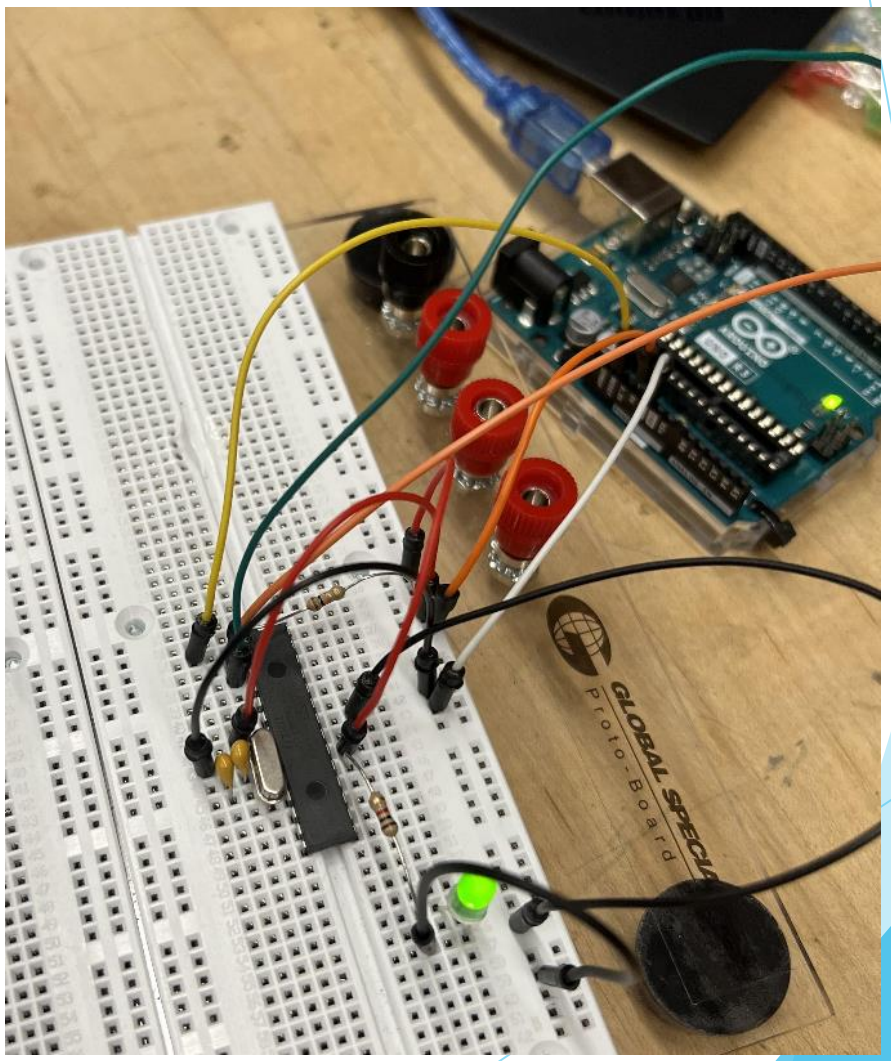


Arduino Bootloader:

- 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

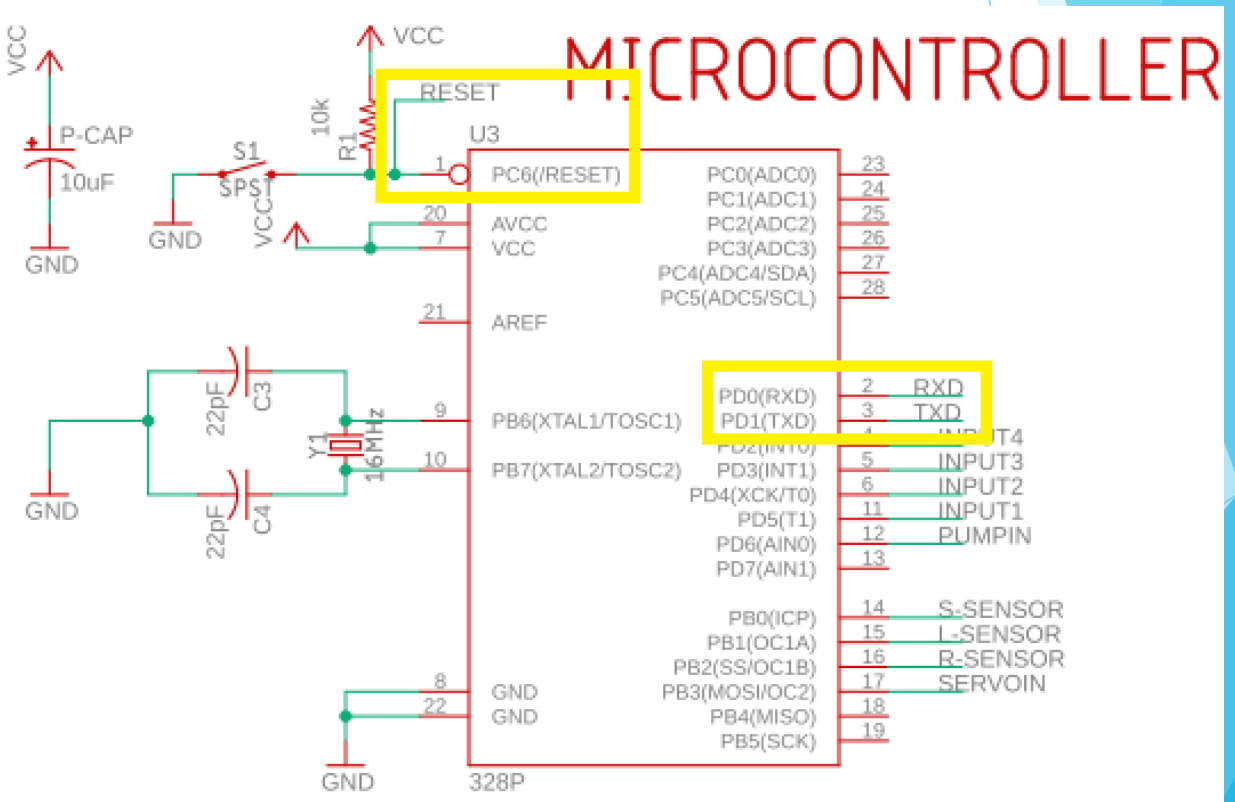
ATmega328 Connections:

- In order for the chip to function like it would on 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 22 pf capacitors
  - The ATmega328 also needs to have a bootloader burned onto it



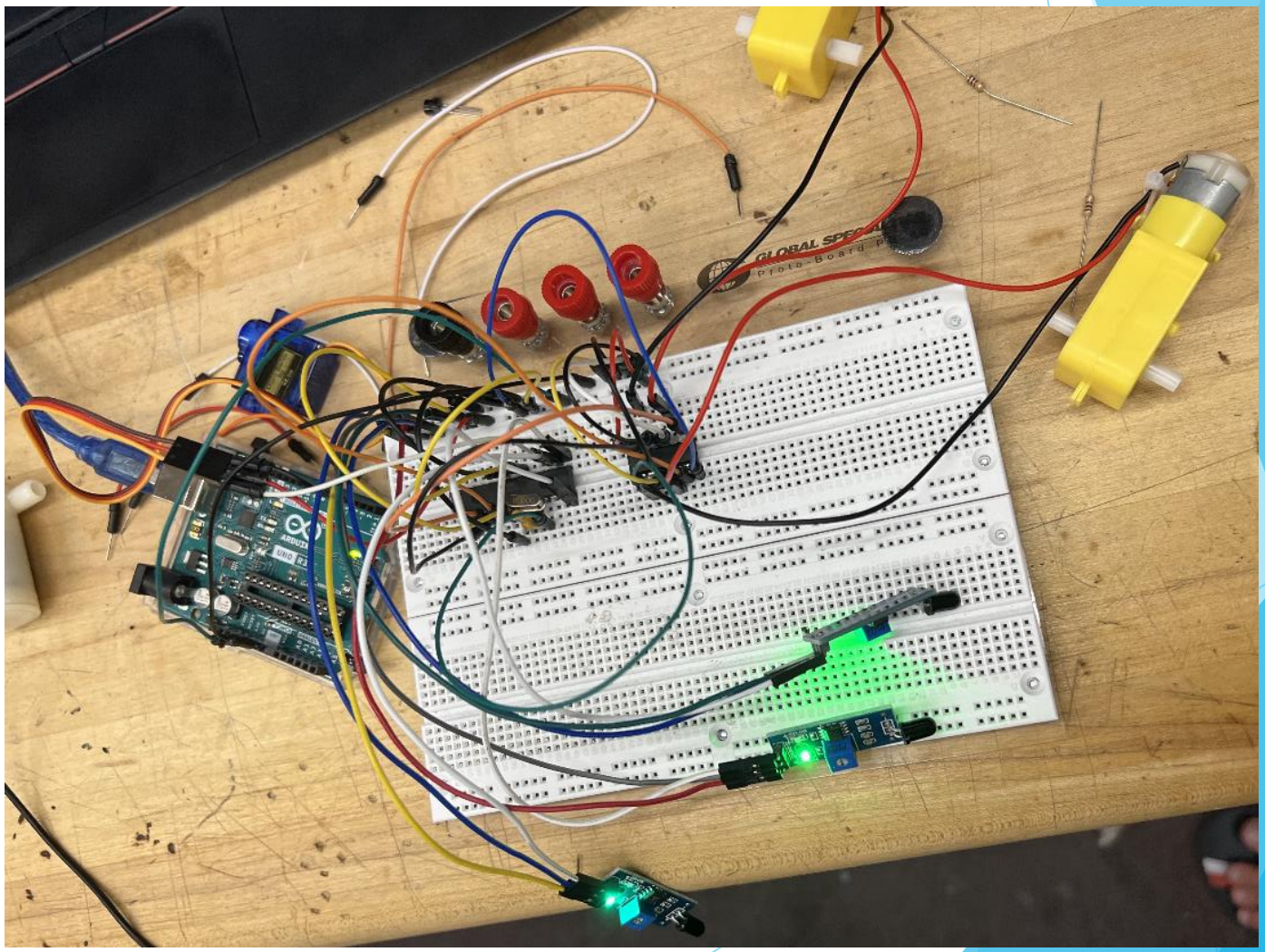
Loading Code Onto the MCU:

- 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

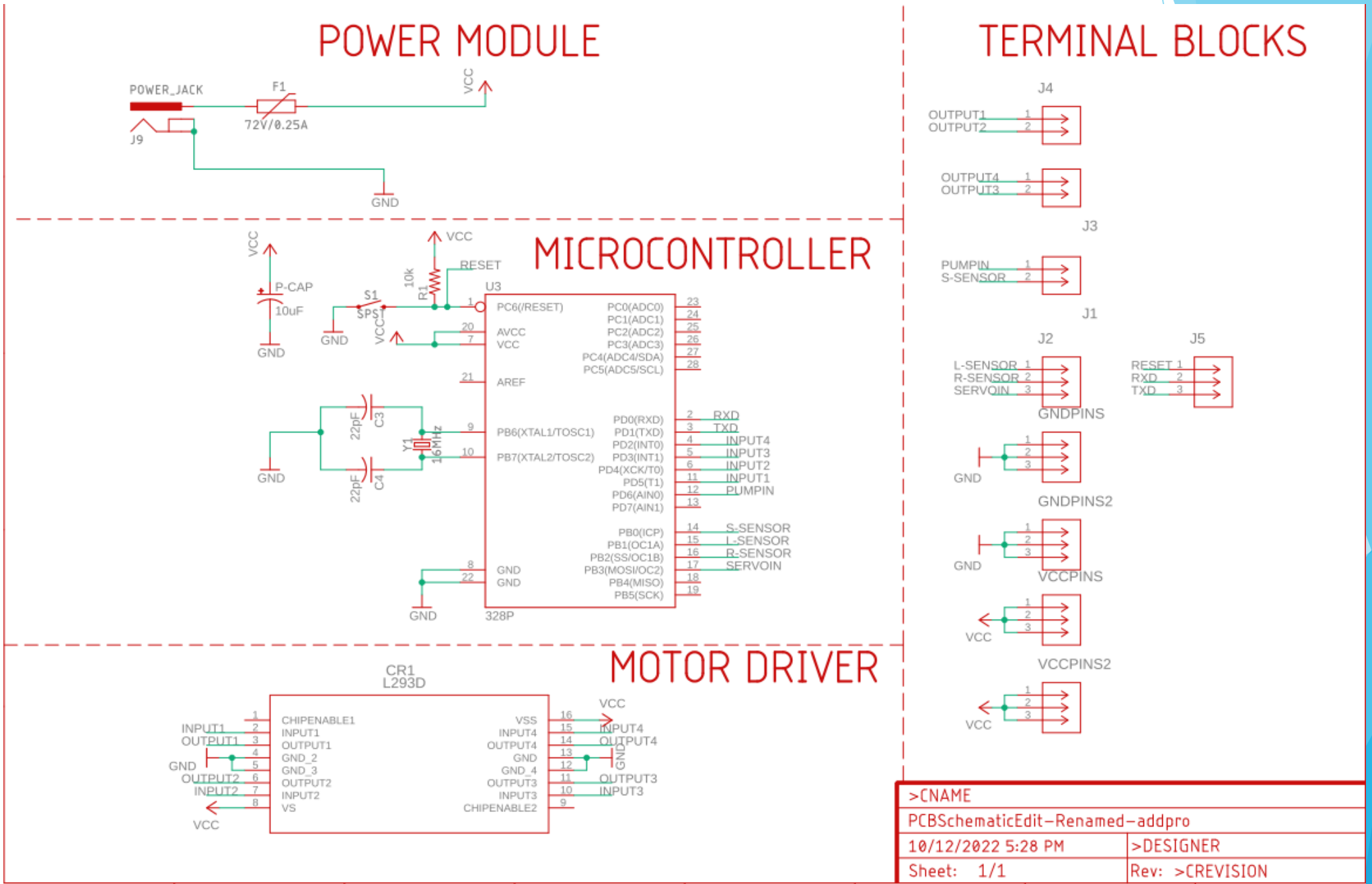


Breadboard Testing Using the MCU:

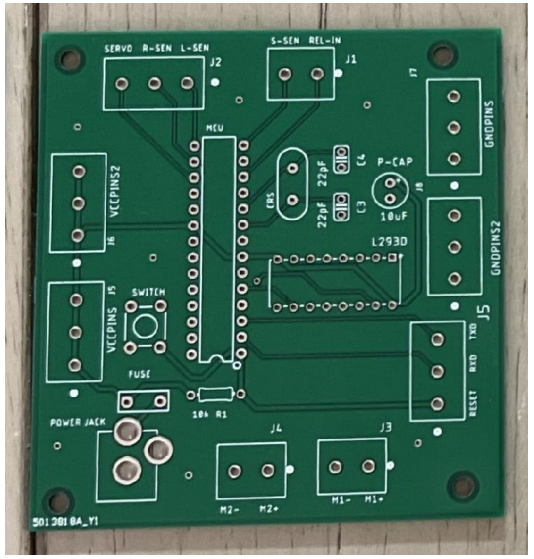
- Testing done with just the MCU of the Arduino
- Code was downloaded into the chip when it was connected to the Arduino and then removed and placed on the breadboard to do further testing
- All the components tested worked as expected

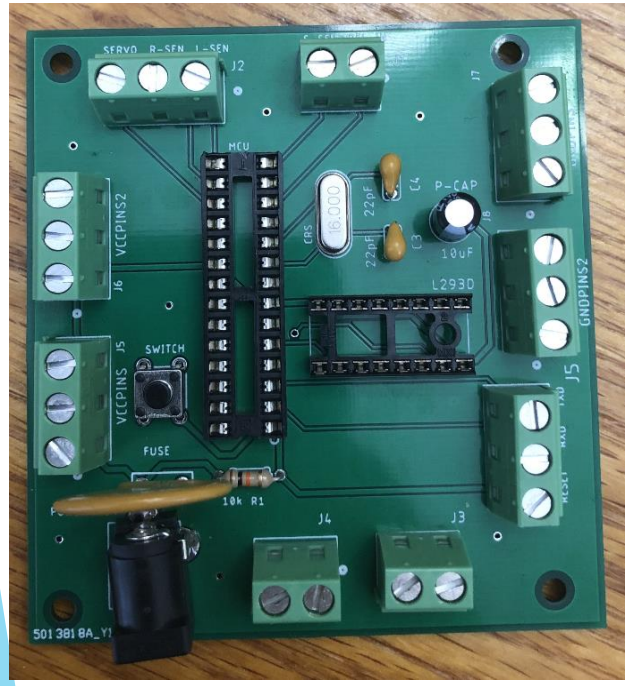


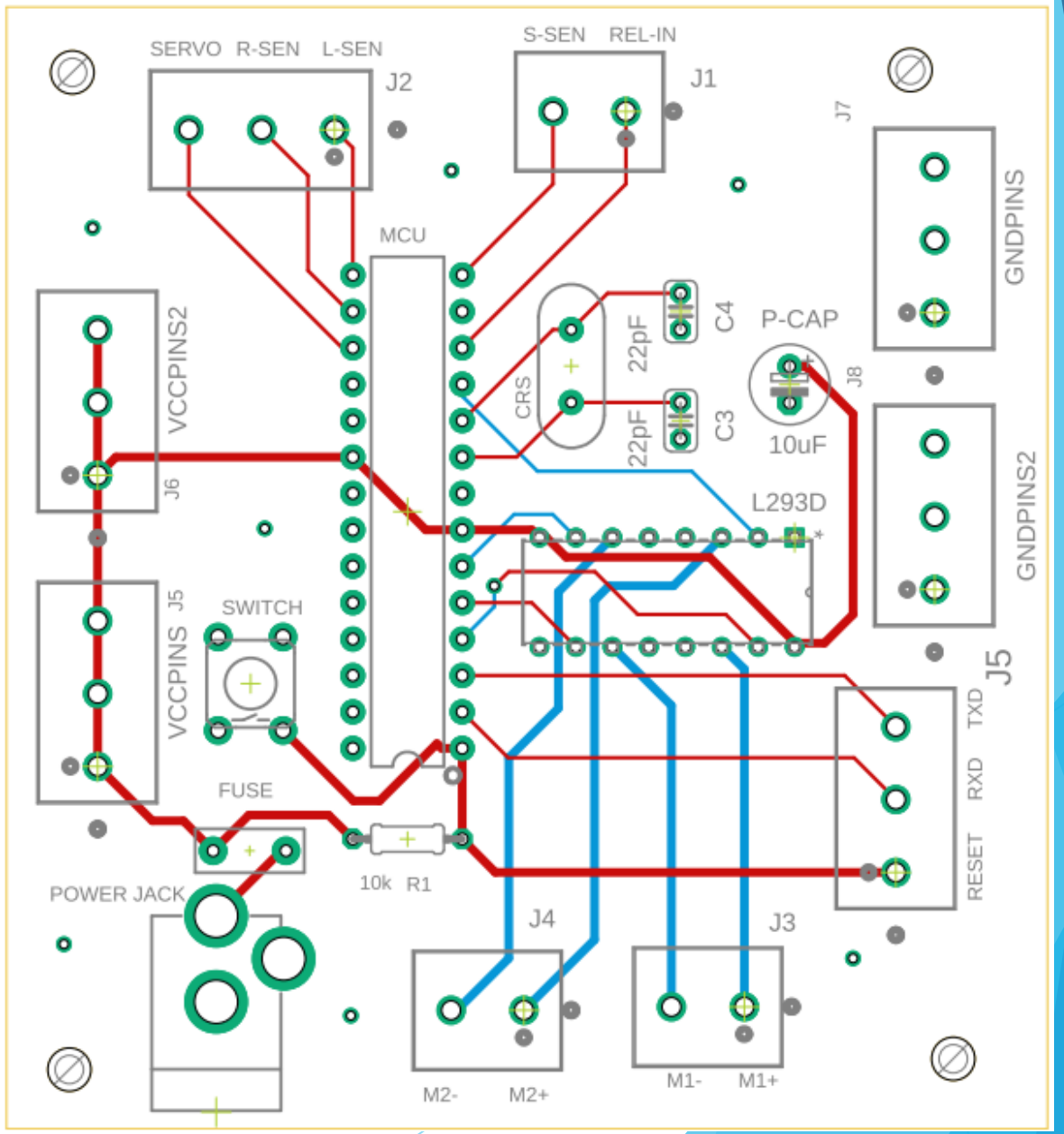
Overall Schematic

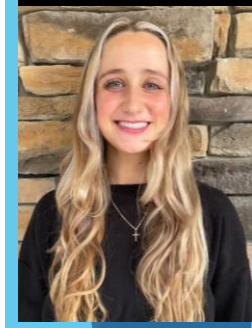


PCB:


SERVO R-SEN L-SEN
J2
S-SEN REL-IN
J1
MCU
P-CAP
22pF
10uF
L293D
GNDPINS
GNDPINS2
VCCPINS2
VCCPINS
SWITCH
FUSE
POWER JACK
10k R1
J4
J3
M2- M2+
M1- M1+
RESET RXD TXD


SERVO R-SEN L-SEN
J2
MCU
P-CAP
10uF
L293D
VCCPINS2
GNDPINS2
SWITCH
FUSE
10k R1
J4
J3

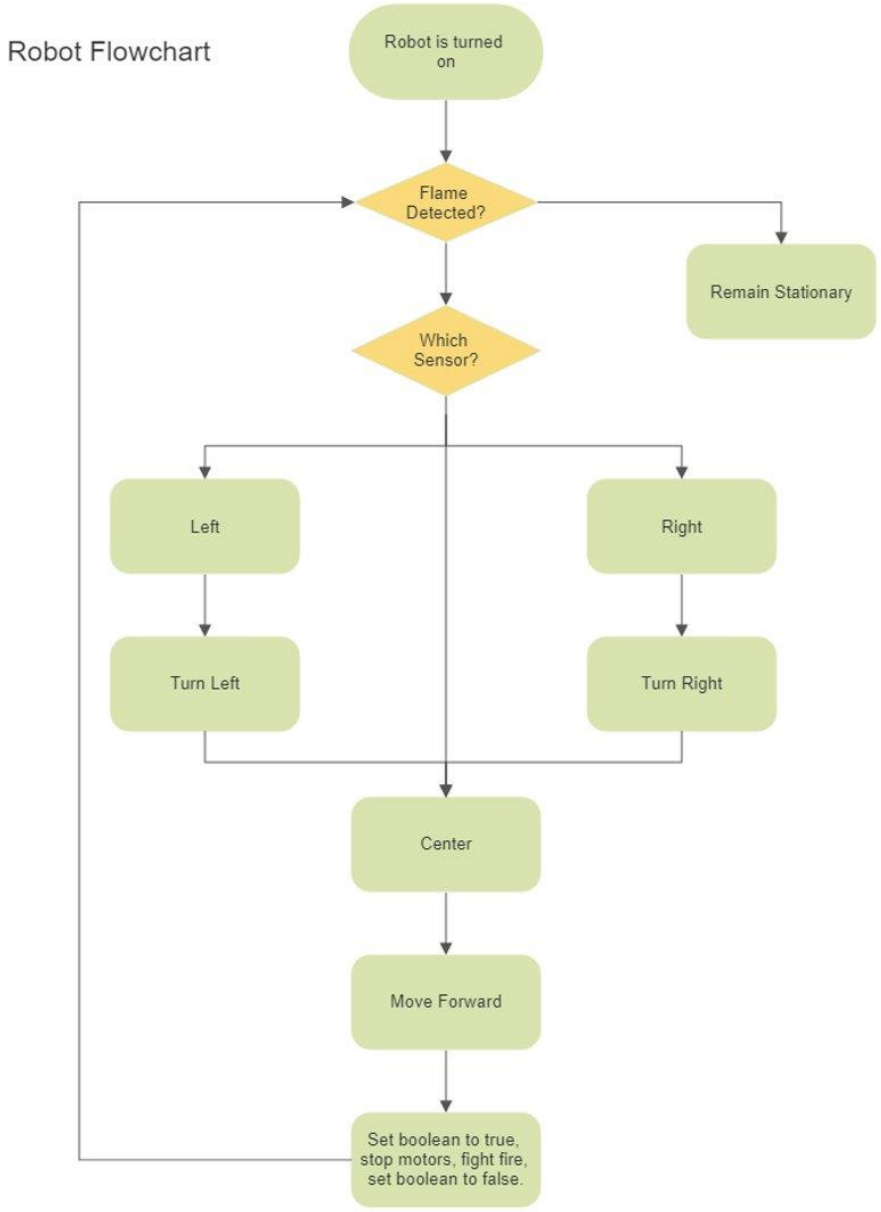

SERVO R-SEN L-SEN
J2
S-SEN REL-IN
J1
J7
MCU
GNDPINS
VCCPINS2
J6
CRS
22pF
C4
P-CAP
C3
J8
10uF
L293D
GNDPINS2
J5
SWITCH
VCCPINS
FUSE
RESET RXD TXD
10k R1
POWER JACK
J4
J3
M2- M2+
M1- M1+



Software:

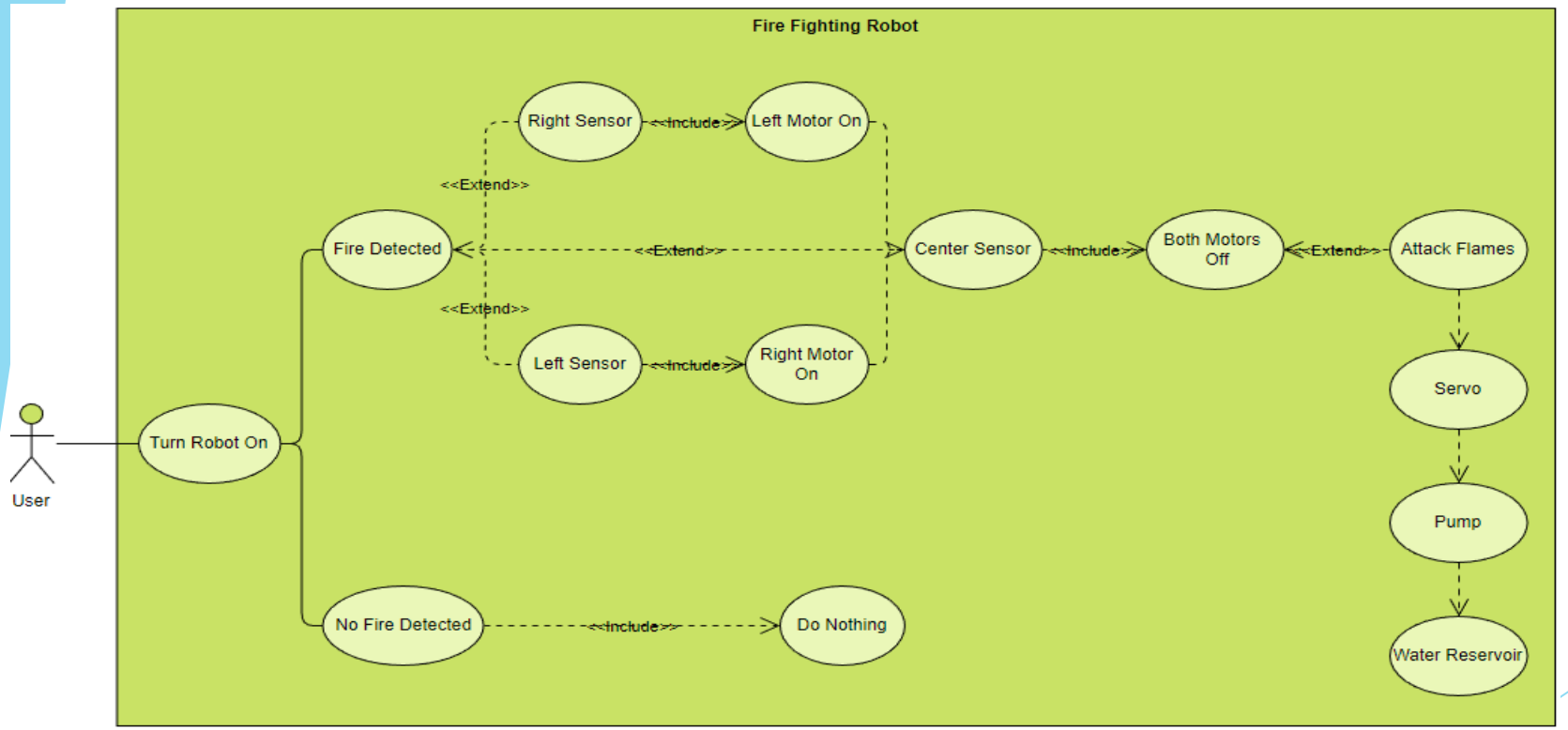
- The software is broken up into a few stages
  - Setup
  - Extinguish
  - Sensor input/Motor output
- **Setup**: This is the first part of the code where the parts are defined, and their pins are set so the micro controller knows which part is which. All parts of the robot are setup here: sensors, motors, 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 robot remains stationary unless one of the 3 IR sensors pick up a flame in range.
- Once a flame is detected the robot will respond according to which sensor saw the fire.
- After positioning itself in front of the flame it will trigger a series of actions to fight the fire.
- The pump will switch on, and the servo will sweep from right to left and then back to the right.
- Then it will reevaluate if there is still a flame present.
- If there is still fire in front of the robot it will sweep the flames with water again.
- If not, it will either turn to fight more flames or remain stationary depending on the conditions around the robot.

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 to dispense water and attack the flames.
- 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.

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.

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.

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%**

Budget & Financing:

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**

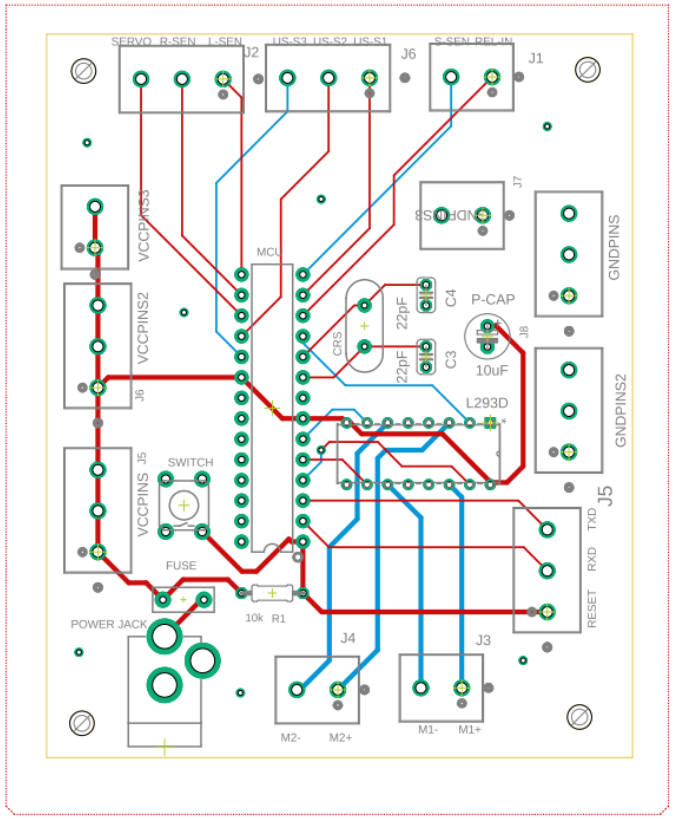
This is not the final breakdown of all the parts ordered as some did were not compatible, so they were substituted with parts that were. These are the parts that are currently being used but this chart is subject to change as the project evolves.

Administrative Content

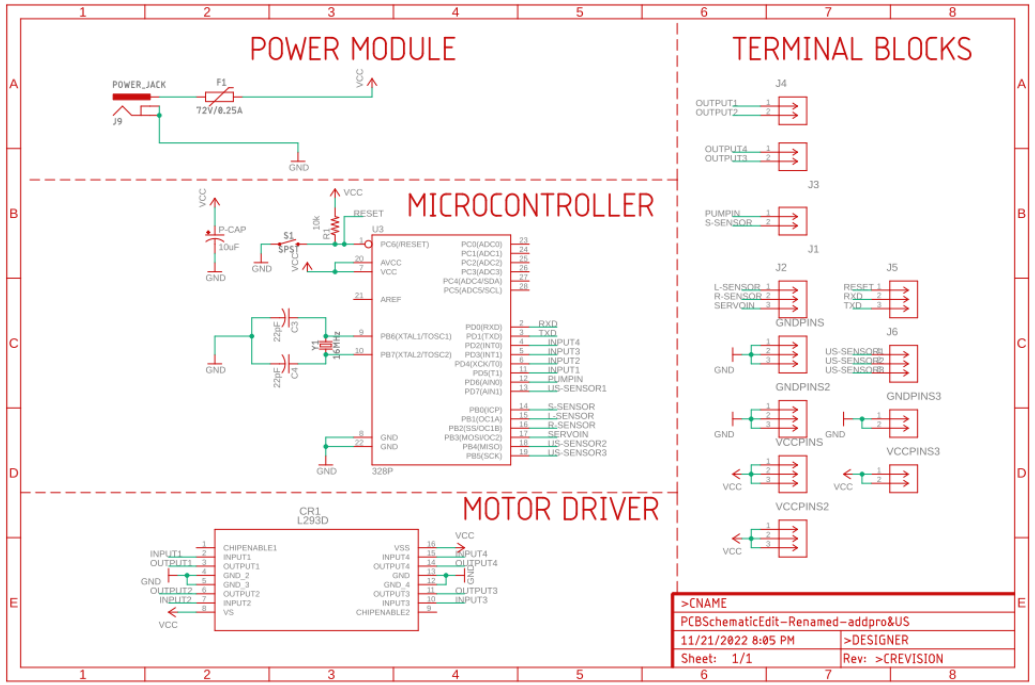
Stretch Goals:

- **New PCB:** A new PCB was designed, and code was written for it as well. However, due to manufacturing and shipping time we were not able to get it on time. This new PCB would've included more pins for ultrasonic sensors.
- **Ultrasonic Sensors:** These ultrasonic sensors would have allowed the robot to see fires from farther away using computer vision. Also, this would have allowed the robot to roam around a room searching for fires instead of staying stationary. The code and design for the implementation of these sensors is already written, but without the new PCB the implementation is not possible. Also, due to budget constraints we were not able to obtain these sensors.
- **Roaming Mode**: Right now, the robot 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 the robot goes into action. A potential upgrade would be to program the robot to actively roam around and search for fires with the use of the ultrasonic sensors.

Stretch Goals (New PCB):



New PCB Design



New PCB Schematic

Stretch Goals:

- **Additional IR Sensors**: 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, potentially giving it 360-degree detection.
- **Spray Nozzle:** Our current version has the robot spray water out the end of a tube from the water pump. A potential upgrade would be to add a spraying, or misting, nozzle to the end of the tube in order to have the robot cover a greater surface area when it sprays the water. This could be misting, spraying, or changing the output of the spray so that it sprays horizontally or vertically.
- **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.

Work Distribution Overview:

	Central Hardware	Peripheral Hardware	Embedded System	PCB Build	Software	CAD
Gabriel	X	x				
Sydney	x	X				
Juan			X	x		
Noah					X	x

Work Distribution Specifics:

- **Gabriel**
  - **Primary:** Central hardware components such as the chip, power sources, and motor driver, and how they interact with the motors, water pump, and servo.
  - **Secondary:** Peripheral hardware components such as the IR sensors, motors, pump, servo, and relay.
- **Sydney**
  - **Primary:** Peripheral hardware components such as the IR sensors, motors, pump, servo, and relay, and how they interact with the central hardware.
  - **Secondary:** Central hardware components such as the chip, power sources, and motor driver.
- **Juan:**
  - **Primary:** Embedded system, MCU, and the peripheral hardware's communication and compatibility with the software.
  - **Secondary:** PCB build and soldering of the PCB components onto the board.
- **Noah:**
  - **Primary:** Software design, writing, and modifying of the robot's code during testing.
  - **Secondary:** Computer Aided Design (CAD) of the robot's final chassis and outer shell covering.

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.

Progress Specifics:

Task/Milestone:	Date Completed:
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
Final tests	November 18-22, 2022
Final video demonstrations	November 23, 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.

Thank you