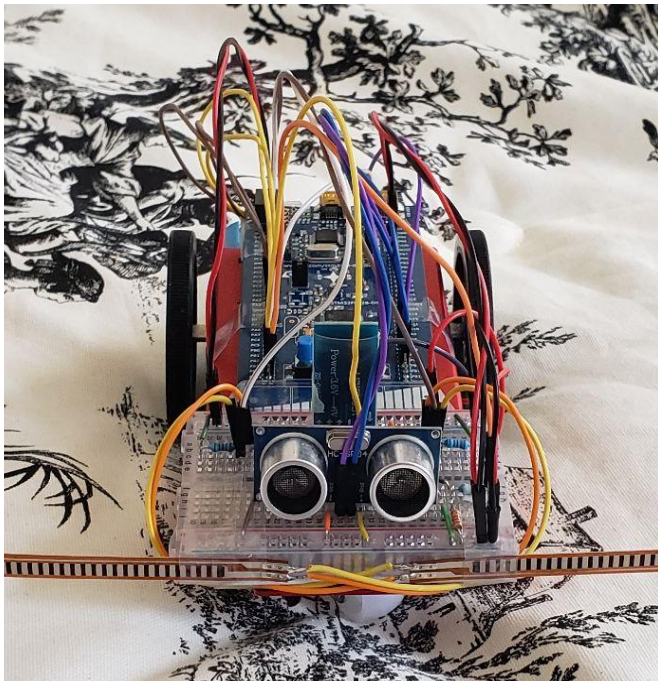
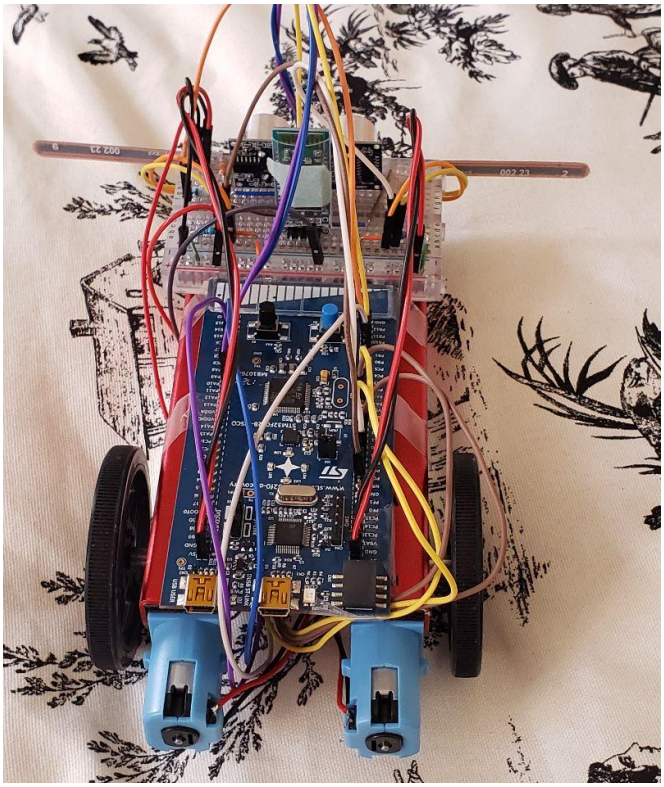
This is a README for a rover project created by Jared Miller, Miseal Nava, and Hyrum Bailey for ECE-5780. It was made in the Fall-2022 semester. We have used more than three lab techniques taught throughout the semester. We are using interrupts, pulse width modulation, timers, state machine, analog to digital converter, and motor driver control.

**Project Name:** LimeGreen
**Purpose:** The purpose of this project was to make a rover that reacted to its surroundings in such a way that it was able to navigate an obstacle course without a human looking directly at the course. It accomplishes this using a few sensors and bluetooth communication.
**Functionality:** The rover uses a combination of sensors to track whether or not it is safe to move forward. An ultrasonic sensor is on the front of the rover to ensure we don't get too close to things in front of us. There is a flex resistor on the left and right and these ensure that we don't hit anything on our left and right.
**Setup:** Using the code provided in the repository multiple different types of setups can be used. All parts that were used can be found in the BOM linked here: https://docs.google.com/document/d/1N5ZFklOcNdEwuQ63sDRkWUn_u8H6u7AHwvIRijccroU/edit?usp=sharing
A picture of how things were mounted on our rover can be seen below:

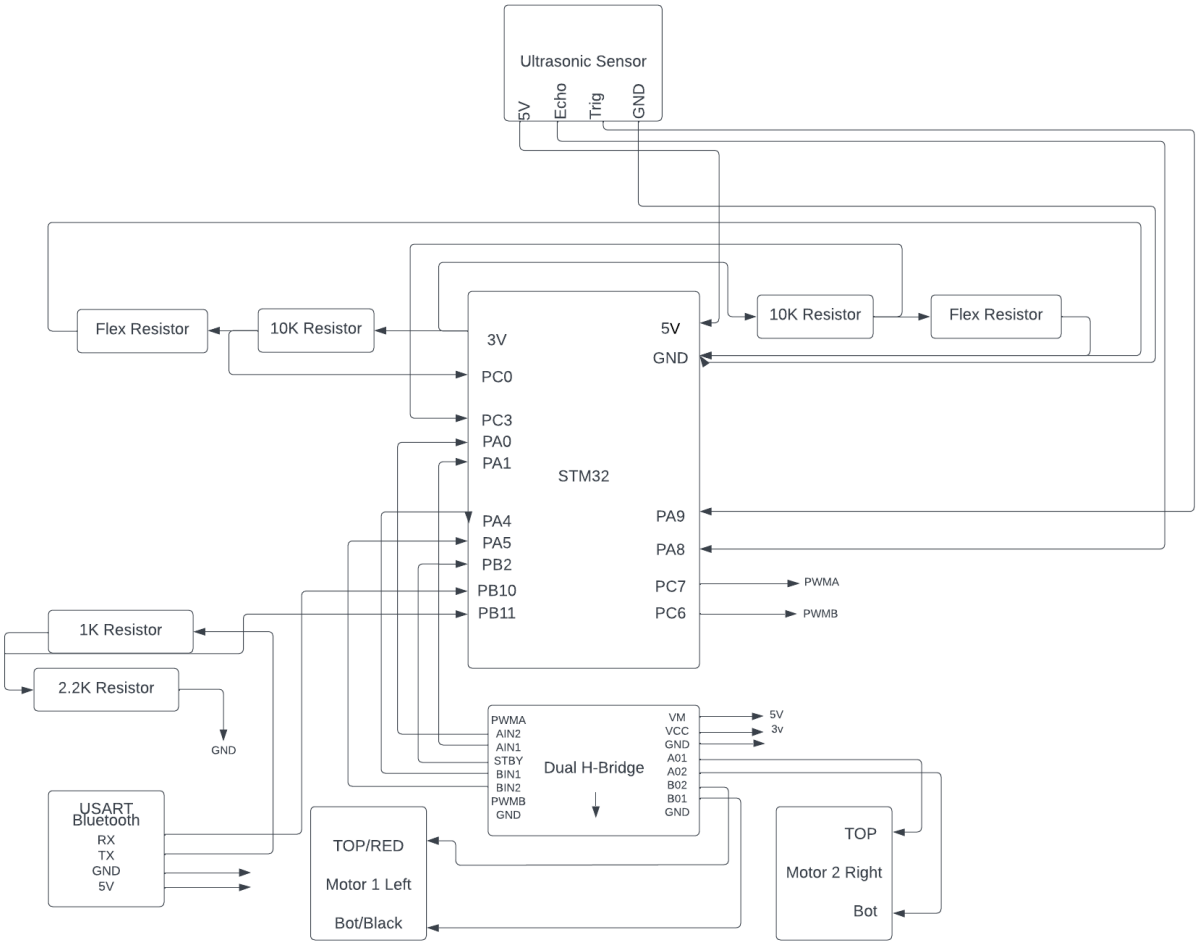




Once the code is flashed to the discovery board and the discovery board has been reset, all that is needed is a program that can connect to the bluetooth receiver. In our case we used putty. Ensure that you are using a baud rate of 9600 when connecting to the bluetooth receiver. The bluetooth receiver is named “LimeGreen” and has a pin of 0812, but this can be changed at

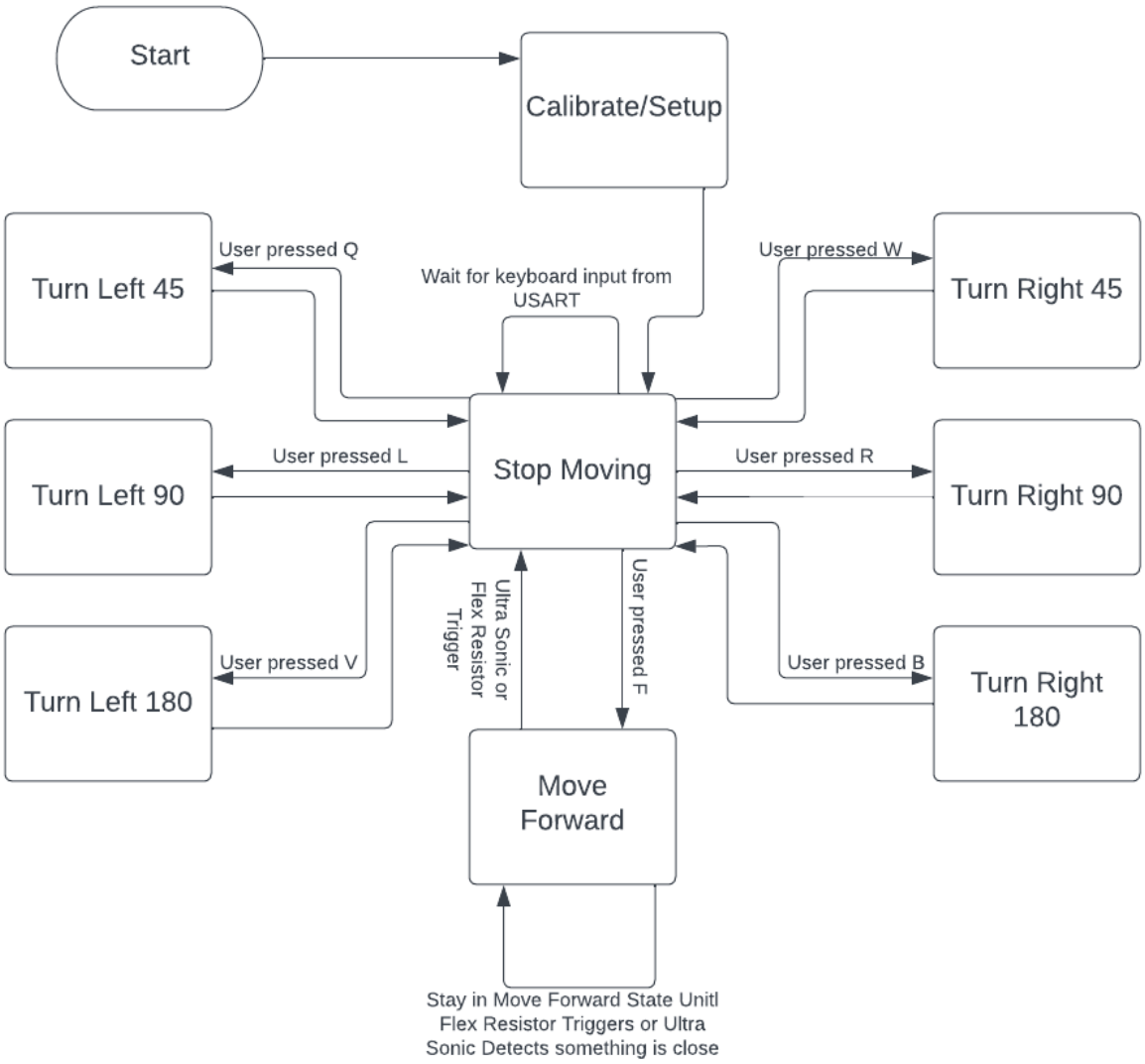
any time using the DSD software provided by the manufacturer to do so. Once you are connected through Putty if done correctly you should be receiving the distance the ultrasonic sensor is sensing. There are a few commands that work with the rover.
F: This makes the rover move forward until one of its sensors triggers. If a sensor triggers the rover will stop and await another command.
R: This makes the rover rotate right 90 degrees. After the rotation another command will have to be issued to make the rover move again.
L: This makes the rover rotate left 90 degrees. After the rotation another command will have to be issued to make the rover move again.
W: This makes the rover rotate right 45 degrees. After the rotation another command will have to be issued to make the rover move again.
Q: This makes the rover rotate left 45 degrees. After the rotation another command will have to be issued to make the rover move again.

**Wiring Diagram:**
Here is a wiring diagram of the rover, please follow it accordingly:



**Finite State Machine Flow Chart:**

The state machine for the rover is simple. As a preface, it is important to set up a bluetooth connection on a laptop and use Putty to be able to communicate with the board from a distance. When the rover is first started it will set up and calibrate the peripherals of the board(USART, ADC, timers, etc). Once the setup is complete it will immediately go into the Stop Moving state. In this state the motor speeds are set to zero and it waits for the user to press a key on the keyboard that will be carried through Putty and update a variable on the board. Depending on which key is pressed it will cause the rover to go into one of the seven states: Turn Right 45, Turn Left 45, Turn Right 90, Turn Left 90, Turn Right 180,  Turn Left 180, and Move Forward where the number represents an angle in degrees. For the turning state's, all they do is turn on one motor for x amount of time until the rover turns the desired amount and immediately go back to the Stop Moving state. However, if the user decides to go into the Move Forward state the only way to get out of that state is only through sensor feedback warning an object is close (Ultrasonic picks up something in front is close or on the flex resistor are bent by an object). The Move Forward state just sets the both motors speeds so that it goes straight as possible. See diagram on the next page.


Start
Calibrate/Setup
Turn Left 45
User pressed Q
Wait for keyboard input from USART
User pressed W
Turn Right 45
Turn Left 90
User pressed L
Stop Moving
User pressed R
Turn Right 90
Turn Left 180
User pressed V
Ultra Sonic or Flex Resistor Trigger
User pressed F
User pressed B
Turn Right 180
Move Forward
Stay in Move Forward State Unitl Flex Resistor Triggers or Ultra Sonic Detects something is close