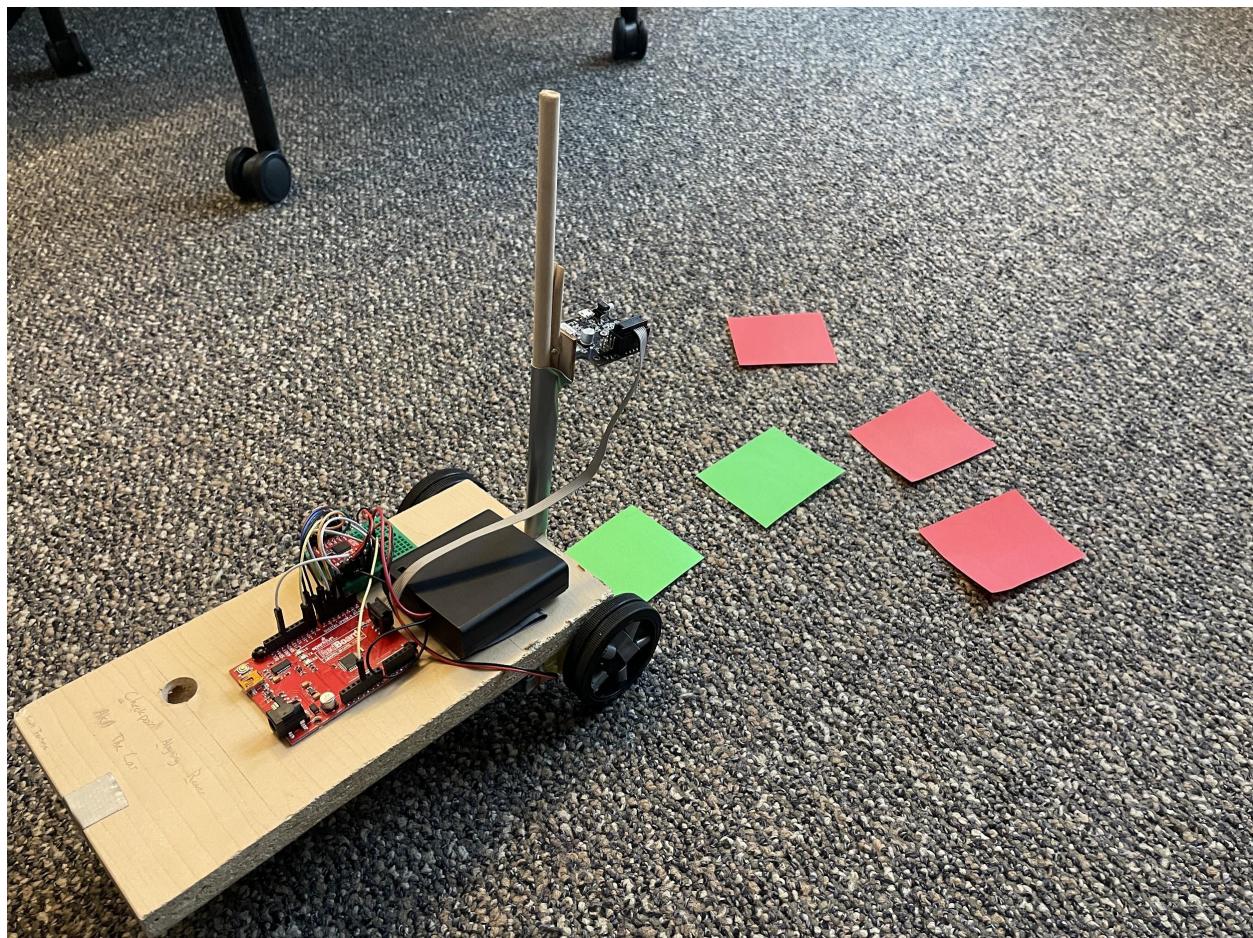


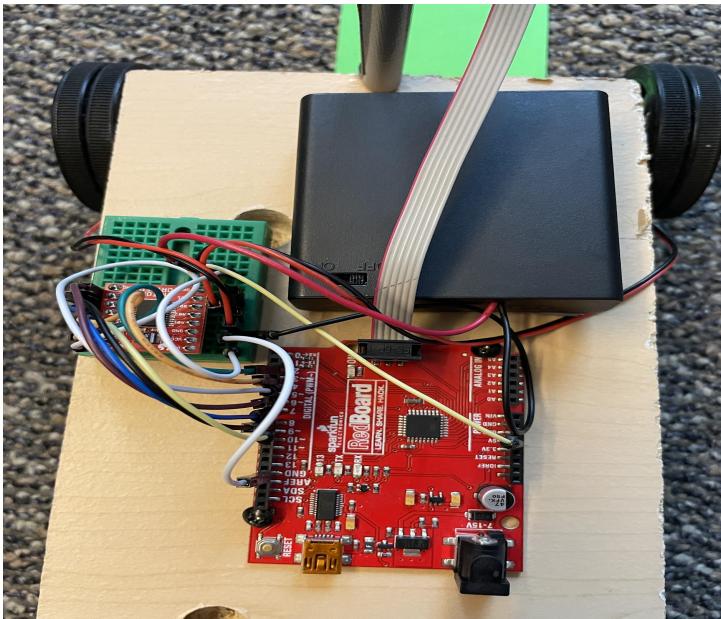
# Color Detecting and Course Adjusting Rover

for 391AH Spring 2022



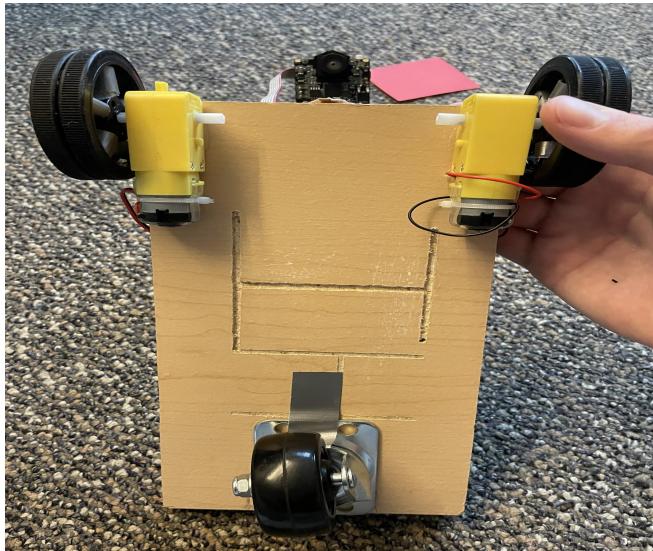
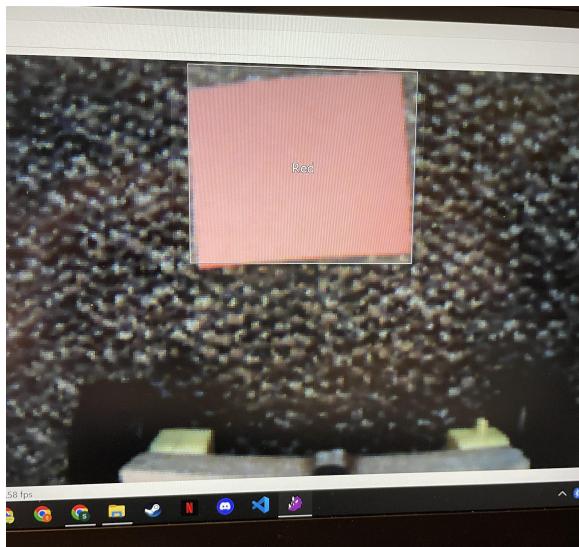
Scott Fortune

## Photos:



The arduino is connected to the motor driver using a breadboard. The connections are the same as the resource given for the TB6612FNG hookup, only adapted to work on the arduino uno. The motors are powered using four AA batteries in a battery pack with a convenient on-off switch. Connected to the top part of the arduino is the cable for the pixycam, which is mounted on a wooden pole to get a wider view (seen on the title page).

On the underside of the rover, the motors are attached via hot glue. The caster wheel is attached with hot glue as well, and tape is used to keep it in place. Without the tape, the wheel would turn and the rover became too unpredictable.



This is a picture of the pixycam software, pixyMon, used to teach the camera and troubleshoot. The pixycam is first taught an object and its color by placing the object in front of it. Then, when it sees the object again, it creates a “block” with properties such as x-coordinate, y-coordinate, time seen, etc.

### **How it works:**

When the rover is powered on, its default setting for when no cards are detected is to move forward. As it moves forward, a while-loop constantly iterates through the array of seen blocks and checks their signatures:

- If a block has a signature of 1, then it is red. Code for the red state causes the rover to stop until the red card is removed.
- If a block has a signature of 4, then it is green. The code also compares the x-coordinate value to a threshold. If less than the threshold, the left motor rotates backward to turn left. If higher, the right motor rotates backwards to turn right.
  - The rover checks the x-coordinate condition and rotates as such until the green card is lined up in front of it. It then drives past it as normal.

### **Video**

<https://drive.google.com/file/d/14RR92mWs-IUhhy6bkzCTAo58YtvEt3Ml/view?usp=sharing>

- Even with the caster wheel slightly out of place, the rover is still able to course adjust.

### **Difficulties:**

- The pixycam proved itself to be the most challenging part of the project to troubleshoot. Even in the finished product, it would incorrectly detect objects that weren't there and fill up the blocks array, slowing down the program and causing errors. I partially solved this by setting the sensitivity of the other signatures to 0, effectively making the pixycam only respond to red and green colors. However, light became an issue. Depending on the light level of the room, the arduino wouldn't recognize the red and green cards and simply drive past them. This almost happened during my presentation, but I arrived early and adjusted the sensitivities beforehand.
- Selecting colors for the pixycam also was difficult. Any object that was too shiny would reflect too much light and cause the pixycam to miss it. I also couldn't choose yellow, since the yellow motors were in frame and were detected as blocks of their own. Blue was too similar to the carpet, so it was too hard for the pixycam to distinguish. Red and green were the best choices.
- The hot glue wasn't as good of a binding agent as I initially thought, especially on the smooth plastic of the motor. There were several times when the rover hit something and the motor fell off.
- The thin wheels had difficulty moving on the carpet. I tried to overcome this by taping two wheels together per motor, but in the end, the table proved to be the best surface.

### **Possible applications:**

The Color Detecting Course Adjusting Rover could serve as a basis for a mine-field path plotting rover. The mines would act as green cards, since careful precision is necessary to drive over the mine without a tire setting it off. The goal or an especially dangerous mine would act as red cards, since that is when one would want the vehicle to simply stop and wait for instructions.

## Project Logs:

Feb 24th

- Picked up motor, motor driver, battery pack.
- Watched soldering in action, now I know the safety procedures.
- Took a look around MakerSpace and started brainstorming some ideas.

Feb 28th

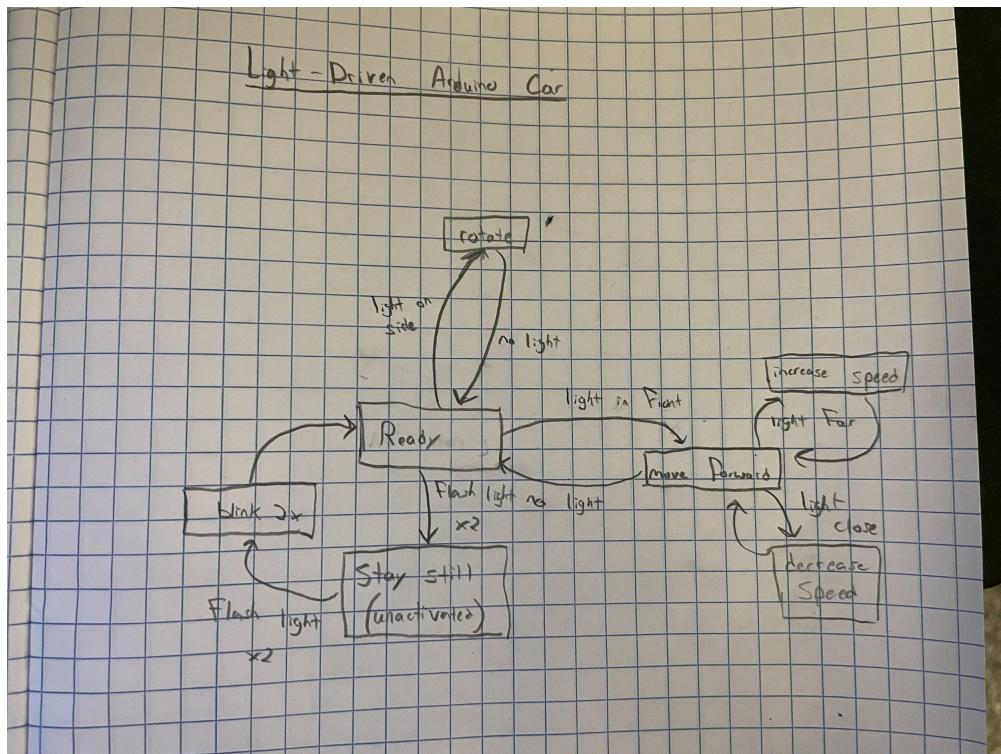
- Tested motor: spins relatively fast. Spins faster with a better connection, which is hard to achieve.

Feb 29th - March 2nd

- Looked at some of the example code on the TB6612FNG website to understand the library. [learn.sparkfun.com/tutorials/tb6612fng-hookup-guide/all](https://learn.sparkfun.com/tutorials/tb6612fng-hookup-guide/all)
- Researched the pixycam, may want to integrate into the project. Its color detection and tracking abilities seem very useful.
- <https://pixycam.com/pixy-cmucam5/>
- Attempted to connect everything to the motor driver. Ran out of pin cables. I will try again. (later realized you are supposed to connect the motor driver directly to the breadboard).

March 3rd - 9th

- Took apart the RC car frame. Missing a gear and axle piece -> might be too complicated to try and replace.
- Tested the motors and steering mechanism on the RC car. Worst case scenario, I use another frame that models the mechanism.
- Began designing code for light component (later scrapped).



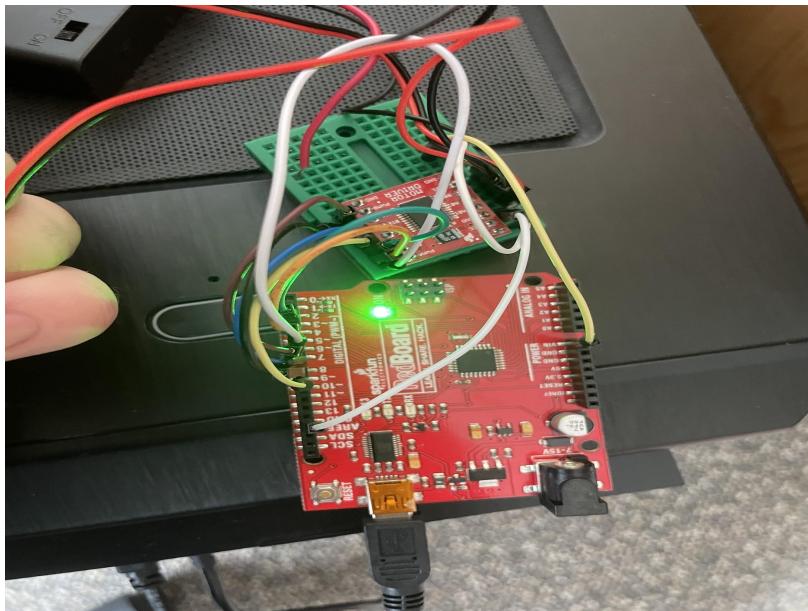
- Sent in parts list that I will need after spring break.

March 10th - 23rd

- Obtained 4x motors, pixy cam, clamps, and wood base for car frame
- Played around with pixy cam and did further reading on its capabilities. Might decide to have car follow something more complex than light, like a drawn line, in the future

March 24th - 30th

- Connected two of the new motors to the arduino and breadboard. Now the example code from the motor driver library runs and the wheels spin like they are supposed to. (hurray!)
- Studied the example code and library functions for the driver. Should be easy to modify it to work the way I want it to.



March 31st - April 6th

- Downloaded necessary libraries and drivers for PixyCam v2 to work with arduino.
- Tested object recognition with PixyMon.
- Studied and ran the example code.
- The usb connection on the pixycam is not very good with the provided cable. I will have to troubleshoot.

April 7th - 13th

- Found a better cable for Pixy.
- I have the arduino and the pixy talking to each other: the pixy is able to detect an object and its position, and the arduino can output that information to a console.
- all that's left is setting up states for the car to be in when it sees an object.

April 14th - 27th

- I have testable code for the car and pixycam.
- The arduino and circuitry is connected to the car frame.
- Went to the make space to hot-glue the motors and caster wheel onto the frame. Also have the pixycam connected to a spool now.

April 28th - Last day

- Finished code and attempted to troubleshoot all the problems.
  - Had to tape two tires together to compensate for driving on the carpet.

- Had to change pixycam settings so the rover would work in a wider range of light conditions.
- The rover is able to detect objects relatively well and change its course based on whether the card is red or green. Ready for presentation.