SparkFun Kit Based Window Blind Flap Open/Closer
By Bryan Ritter
2023-12-09

1. Scope
This procedure will demonstrate how to use a SparkFun kit to simulate a window blind open rotation opener for a blind that uses two pulleys to twist open a set of blinds. This procedure assumes a basic understanding of Arduino and it's coding.

2. Reference Documents
SparkFun Kit
https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v41/all
Arduino IDE
https://www.arduino.cc/en/software

3. Equipment Used
Dental Floss
Tape
Computer compatible with required software
Some way to hold down the assembled SparkFun kit in a way that holds the tires above any ground, and keeps the kit from being pulled up.

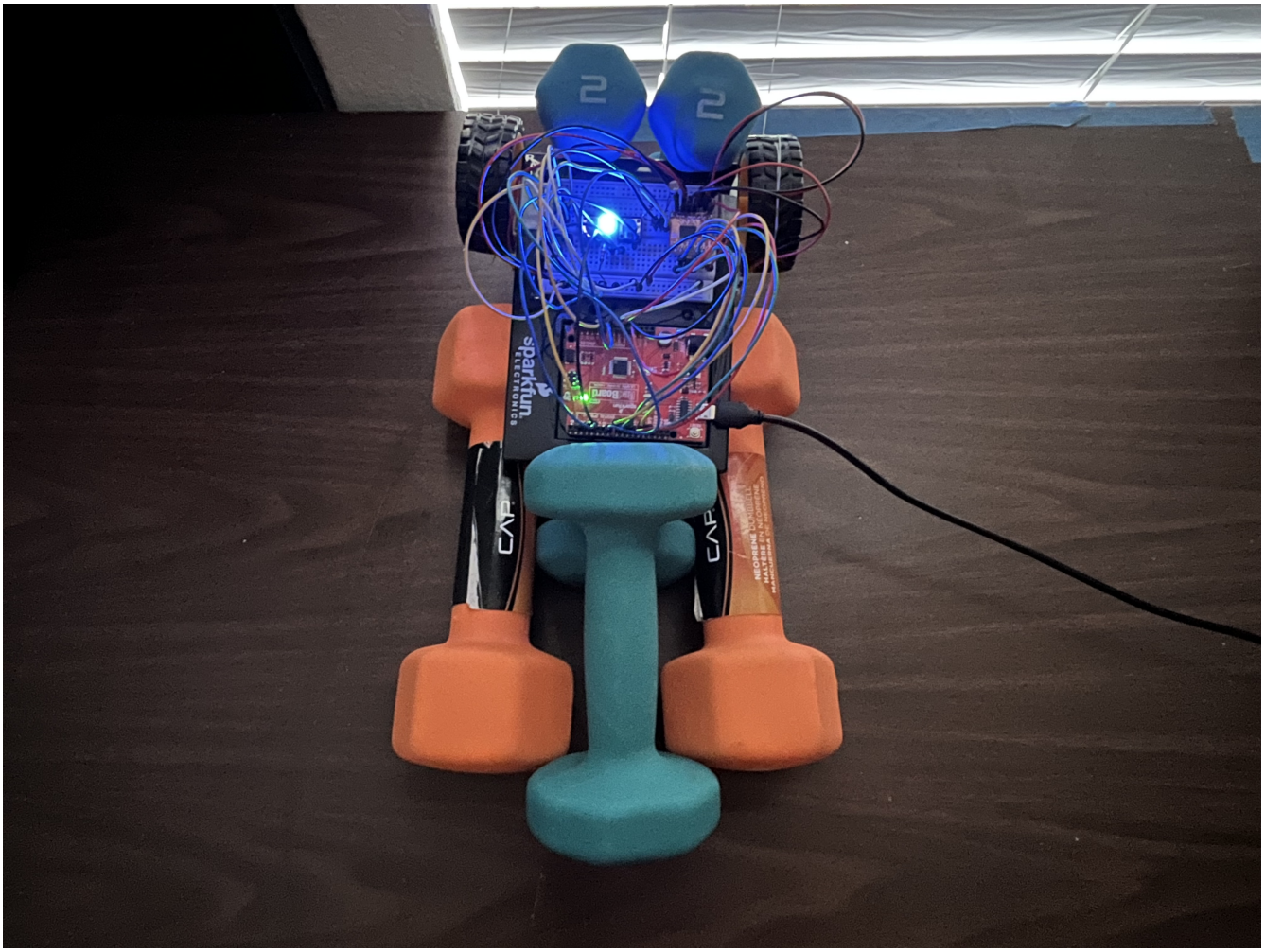

Figure 1: An image of the setup with the SparkFun kit held down with wheels in air

SparkFun Kit Items Used

- 1 BreadBoard
- 1 RedBoard
- ~30 Jumper Cables
- 1 Motor Driver
- 2 Gear Motors
- 1 Potentiometer
- 1 LED
- 1 330 Ohm Resistor
- 1 Photoresistor
- 2 Push Buttons
- 1 Switch
- 2 Wheels
- 1 USB A to USB Mico cable

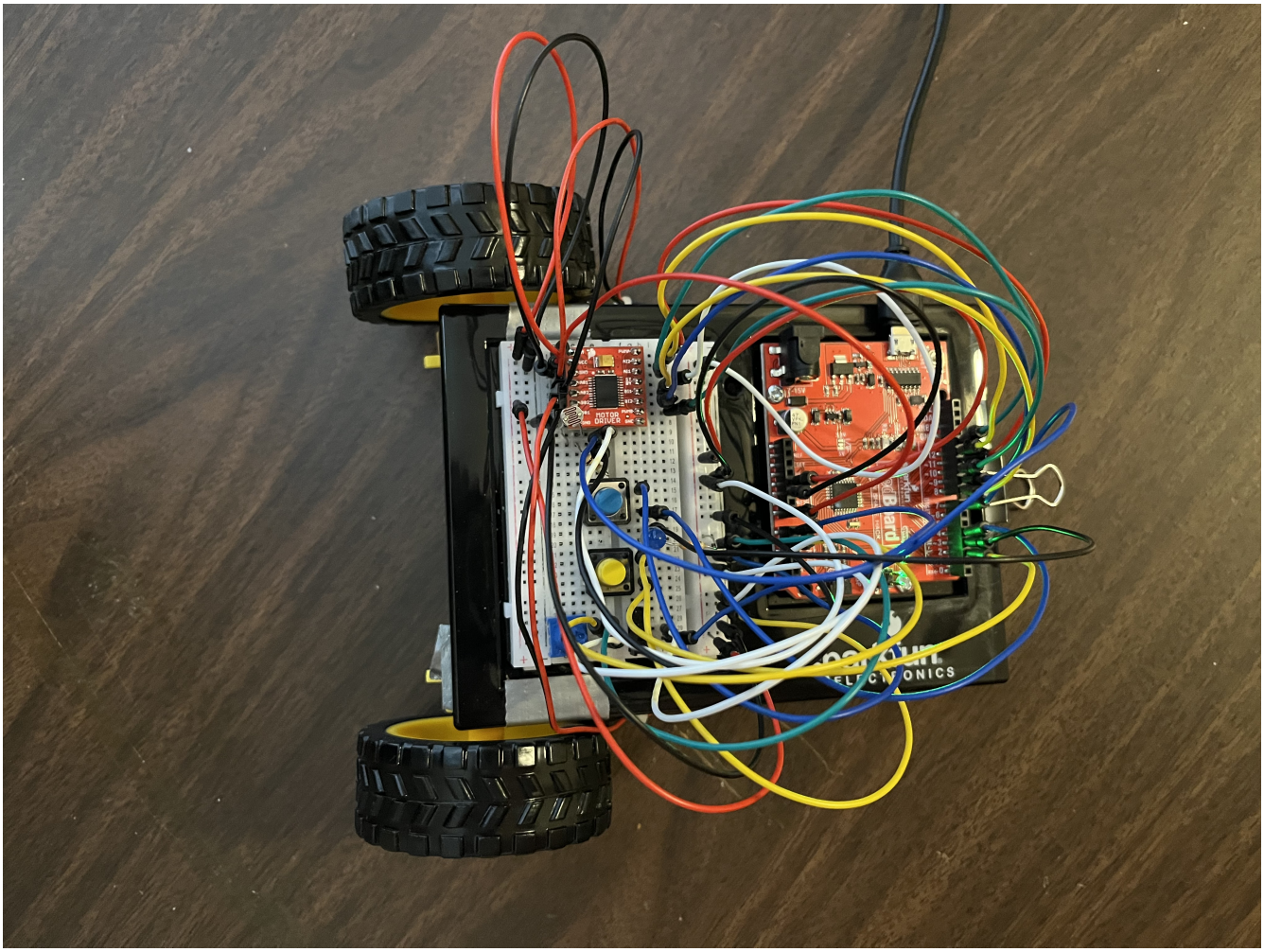

Figure 2: An overhead view of this SparkFun kit setup, showing the wiring and sensors used

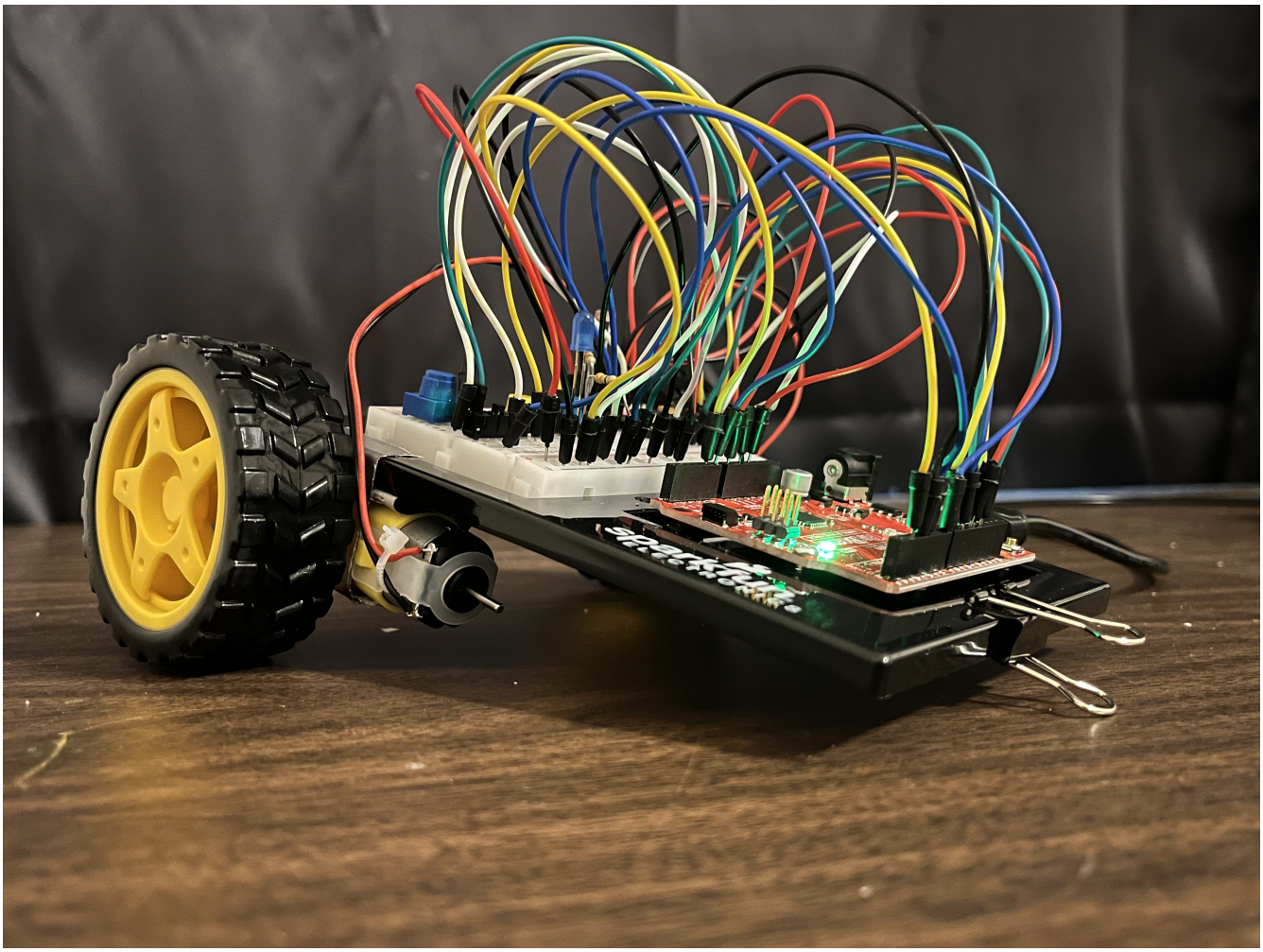

Figure 3: A side view of this SparkFun kit project

4. Procedure

a. Gather required parts
b. Assemble the equipment for the kit
c. Install software for your computer to run Arduino IDE
d. Create and upload code to Arduino
It may help to create it like this
I. Start with initializations
II. Create code that can rotate tires in opposite directions
III. Create code that can call the above code based on inputs like button presses, amount of light present in a room for a given amount of time, etc…
IV. Debug and upload code to Arduino
e. Use tape to stick dental floss between a tire from the end of a blind rotation string. Do this also between the other blind rotation string and the other tire.
f. adjust software initialization settings for your setup

5. Results / Conclusions

The SparkFun motor doesn't have enough power to pull the strings with enough force to move the blinds, but has enough capabilities to simulate doing so.

6. Discussion

Conceptualization stage:
The basic idea wasn't that hard to grasp, but the details to do things were. The primary goal of the project was to have a photo-resistor open/close the blinds. The first thing done for the was trying to decide what values of inputs should result in what values of outputs. e.g. If it's X brightness for Y length, rotate motor for Z amount of time to get A amount of distance moved by the blinds shut/open. Originally it was desired to have a light sensor out side of the room and inside the room, then compare the values, but there was only one light sensor, so that idea had to go away, and replaced with the idea to open blinds when it's been bright enough for a period of time, and close them when it's been dark for a period of time.

Getting something down to begin stage:
Looked at all the parts in the SparkFun kit, and tried to come up with gists of how things could potentially be used and put together. Tried to set things up in a way that adding extra stuff could be done, but wouldn't be required. Next was to gather existing code and wirings from the SparkFun projects and determine which projects had these similar enough to either reuse or use as reference for future actions. Main ones chosen to use for reference were the remote controlled robot(5B), and "Simon Says" game(2C).

Getting things organized stage
Next was how to I want this data processed? Use absolute or relative values? Which processing is desired inside loop() and/or in separate functions? Use global and/or pass values around? Does the engine rotation number use a time, distance, and/or rotation amount? Somethings didn't work like initially thought, and code had to be rewritten, and reorganized, and there some new things to learn, not covered in the SparkFun manual. One thing that helped was switching to paper and pencil to write down ideas, and organize thoughts.

Debugging stage:
When creating this code Serial.print("string"); type statements were used to view how variables were changing as the code was running. There even some periods of head scratching asking why isn't this variable changing, or how does it keep getting reset? It was said online that use of the "delay()" statement isn't that good of a coding practice, yet the SparkFun kit used it, and didn't show an alternate way of doing things to avoid use of it. At one point the engine stopped working, and had to find out if the projects code quit working it, or did some wire come out? (It was the later) There were many times variable names were changed to make more since with which they were being used. There were some bugs that one would think the compiler would have complained about, but didn't.

Extras added:
Once the core functionality was done, it was easier to add the optional items that were thought about earlier, even had some new usages with them that wasn't thought up of earlier.

Some neater add-on things that were done:
Using the potentiometer to control the blind angle that counts as open.
Buttons to manually initiate a blind closing/opening.
Using the switch to stop blind rotations, without having to manually adjust the blinds to a specific spot.

If the rotations don’t end up at the right spot(e.g. the switch got set to off), an error message is displayed to Serial.print().
An LED light brightness that changes based on how open the blinds are.

Some ideas that were scrapped, and/or potentially saved for later:
() Using the Ultrasonic Distance Sensor to tell if the blinds were open or not. (How spread out does sensor go? Would it count the blinds open as something being there too?)
() Using the Piezo Buzzer to make sounds when the blinds were being opened/closed. (Does one really want this? The only potentiometer was already being used to adjust what’s considered the open spot. Guessing without this the buzzer probably would sound off at full volume.)
() Using the temperature sensor to help decide if the window should be open/closed. (neat, but it could only measure the inside temperature, unless it is put outside.)
() Use more buttons for something. Like manually going up/down a bit. (More coding to do would be needed.)
() Using the LCD to display data. (Not enough wires to do this and other things, running out of space.)
() Using the battery pack. (Didn’t want to buy new batteries, especially since the motor doesn’t have enough power to open/close the blinds.)

Some issues with this as it is now:
The angle of blinds must be setup before the code is started.
Serial.print() output could be done improved.
Code comments could be organized more.
Something doesn’t seem to work unless Serial.print() debug lines are added. (The delay(1) line?) Why? This doesn’t make since. Is Arduino software itself messes up, and not the code? Probably shouldn’t use delay() lines anyway, even though that’s what the sample code in the SparkFun kit uses.