EZ – SCOOTER

Tony Naylor, Carolina Geary, Jacob Jardines

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

Abstract --- The usage of electric scooters in cities has increased and become more important, as not only to reduce carbon emission but also to reduce the dependency on normal combustion engine vehicles. For this reason, we believe that electric scooters have their segment in the transport networks of the present-day and future. However, ongoing technology has addressed many problems with high cost and the lack of safety for the consumer. Therefore, this paper presents the design and development of a compact, portable, safe, efficient and more affordable electric scooter. The scooter will run on an electric hub motor powered by the batteries and our own design electric speed controller with a regenerative braking system, and an LCD display. For safety features a tilt sensor, headlight and taillight are integrated in the scooter. The proposed system with electrical and software features will help towards enhancing the performance of an electric scooter for easy mobility. The results obtained from the on-road test, software and implementation of the system has been discussed.

***Index Terms* --- Brushless DC Motor (BLDC), Electric Speed Controller, Atmega328p**

I. INTRODUCTION

Nowadays, millions of people depend on motor vehicles as their means of daily transportation. Unfortunately, owning a motor vehicle can be expensive in terms of purchasing the vehicle, routine maintenance, and repair costs. Additionally, the majority of motor vehicles burn fossil fuels and emit greenhouse gasses that contribute to the current climate trend of global warming. In recent years, many countries have begun shifting to being more ecologically minded. One of the ways they have started to implement these shifts is through policy changes and agreements, like the Paris Agreement which is "a legally binding international treaty on climate change" (United Nations), to hold countries accountable[1]. The United States, in the past years, has encouraged society to be less dependent on fossil fuels in order to achieve this goal.

Electric scooters have started to appear in big cities around the world as an evolution of the basic scooter which uses the driver's foot. These scooters are so compact designed that they could be stored in a small place anywhere wherever you go or in the house. However, it can be expensive for the consumer. We want to create a system that not only is beneficial to the environment, but to the consumer by being less expensive than the ones that already exist in the market. To find a scooter with similar capabilities to our design, costs could easily exceed a thousand dollars. Our electric scooter will run on electric power by activating a direct drive hub motor that will facilitate the scooter's acceleration. The scooter will have a motor on the rear wheel, a battery with a regenerative braking system as well as a wall charger, a LCD display, as and added safety feature several tilt sensors will be used to determine when the scooter is on its side and to prevent the activation of the motor, the speed controller will support regenerative braking and will switch the motor to this mode given user input, and lastly several light sensors will be used for automation of the scooter's lights.

II. SYSTEM COMPONENTS

The complete system is better presented in terms of system components; as individual physical elements whether purchased or designed. This section will be focused on the technical introduction of some of these components.

A. Block Diagram

To better understand the importance of each component a block diagram is very helpful.

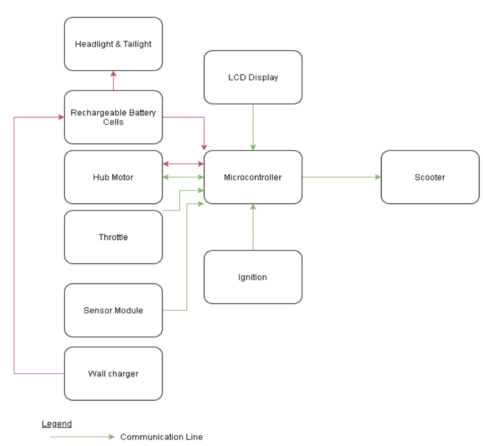


Fig 1. Complete system Block Diagram

As shown in Fig 1. The system is powered by batteries. The motor receives energy from the batteries by using the electric speed controller. The speed at which the scooter rotates is regulated by the joystick, once the rider presses the throttle or joystick, the electric speed controller engages the motor and starts to rotate. Then as safety features are the LCD display, tilt sensor, headlight and taillight they are all controlled by the microcontroller that is powered by the batteries.

B. RBSD Hub Motor

The usage of permanent magnets in electrical systems results in many advantages such as improved efficiency, less losses, fast dynamic performance and high power per unit volume. The brushless dc motors (BLDC) is a synchronous electric motor which is powered by DC electricity and it has a controlled commutation system. In the BLDC motor the electromagnets do not move instead the permanent magnets rotate around a fixed armature.

The BLDC motors offer various advantages over DC motors, like increased reliability, increased efficiency producing more torque per watt, and longer lifetime of the motor by eliminating erosion. There is already an analysis between DC and BLDC motors as seen in Table 1. and Table 2.

Motor	Advantages	Disadvantages
DC	Torque control Easy speed control Simple design	High maintenance Hard control Dusty brushes
BLDC	Low noise Long life High efficiency	Complex control High cost
Stepper	No feedback Long life High low-speed	Noisy Low efficiency

Table 1. Comparison between Motors

	MY1016 Brushed Motor	RBSD Hub Motor
Power	350 watt	350 watt
Voltage	24 volts	24 volts
Gearbox	yes	No
Efficiency	75%	85%
Weight	5.8 kg	3.5 kg
Regenerative	No	yes

Table 2. BLDC vs Brushed motor Parameters

For this project the goal is to use regenerative braking and the right type of motor or only motor that can be used for regenerative braking is a direct drive hub motor (non geared motors). The geared motors have a clutch and they are not directly driven when going forward, these motors coast so there is no resistance. The direct drive hub motor can regenerate electricity because we are basically driving it forward instead of letting the motor drive us. When we coast we are pushing it the opposite way or when we break and using regenerative braking the motor engages and sends electricity back out the opposite way to charge the battery. To engage regenerative braking, we can use e brake or electronic brake levers. It contains a micro switch so when we pull the lever the micro switch closes and lets the system know we just pulled the brake, another option is to use a button or a joystick.

C. Speed Controller

The speed controller (ESC) receives input from the throttle and precisely controls the flow of the current from the battery to the motor. A controller with higher-current, higher-voltage is capable of driving more powerful scooters. Controllers range from having sustained max outputs of tens of volts and little amps all the way up to 100 volts and 400 amps.

D. Throttle

The throttle is a device that is connected across a fixed input voltage of 5 volts and that outputs a variable voltage on the output pin. When selecting our throttle control the characteristics we were looking for were ease of use, a distinct implementation to set our scooter apart, dual use for regular motor operation, regenerative operation, and affordability.

The throttle controller that we selected to meet these requirements was a joystick, the HiLetgo Game Joystick Sensor. The design of the joystick with x, y axis analog output, and z digital output leads us to be able to use a 3-pin special line really plugged into the expansion board. With an input voltage range of 3.3

volts to 5 volts DC, a weight of 0.03 pounds, and with a high quality.

To integrate the joystick into our design it will be connected to the Atmega 328p by starting with the switch which is a digital pin, it will be connected to D2, then the y axis into A1, the x axis on A0, the 5 volts into the 5 volts and ground to ground. Likewise, the microcontroller will control the motor as a regenerative braking system given a voltage input below a certain threshold from the thumbstick. The above logic of the voltage thresholds and thumbstick direction is assuming that the maximum voltage delivered from the thumbstick is when it is in the most forward position and the most negative voltage output is when the thumbstick is in the most backward position.

E. Power Switch

When thinking of an implementation to turn on and off the electric scooter we looked into some of the methods they have on the market as the key switch and the button switch but none of the choices were good to fit our design. However we had the solution in our design already: the joystick. At this point, there is no planned function for the left and right directions for the thumbstick throttle control.

We chose to implement the joystick as our power switch. First to be able to use the scooter, it must take input from the joystick X and Y directions as a password. It is a four direction password and it has a delay of inputting each direction of two seconds, if the password is incorrectly inputted or took too long to input it resets after two seconds and waits for user input once again however once the password is inputted correctly it takes two seconds to turn the scooter on.

The implementation of this operation is to add in our software a safety feature, by adding a joystick combination. The joystick combination starts with a combination with right or left and this is because if you are already operating the motor and want to shut off the motor we do not want the driver to push the up or down because that is to go forward and the down position is to break. So for this reason the combination that we create and we test is right, right, left, up, left and then the motor will operate. We can test if the code works by unlocking the motor by looking at the waves on the Oscilloscope as shown in **Fig 2** We will know if the motor is unlocked even if the motor was not operating.

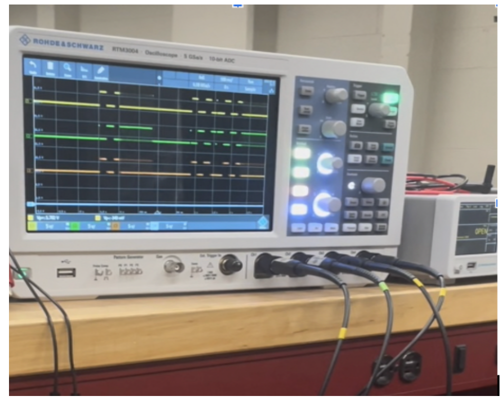


Fig 2: Joystick Test

F. Rechargeable Battery Cells

When selecting a battery the most important were low weight, high energy density,easy capability and cost effectiveness. The battery we selected needs to have enough energy storage to be a viable power source, and have an acceptable operating voltage range. As mentioned previously some parameters we take into account when choosing the battery include as follows:

Durability: It is important to consider how long the battery will last, each battery system has a unique system of adverse temperatures, rough roads, charging and discharging that we need to take into account. All these aspects mentioned that can affect the durability of the battery.

Performance: Is how much power it is capable of generating and how much running time it will offer in each charge. Some batteries will have the same performance, but it is possible that one performs better than the other one. The only way to know it is by testing each battery.

Voltage: Batteries with higher voltage allow the electric scooter to produce more power but by being with higher voltage they can be heavier because they contain more cells. We also need to match the battery voltage with the one in our motor. It is important to make sure our motor is able to support the voltage of the battery, since we are limited in our speed we do not need a very high voltage battery.

After taking into account all the parameters mentioned previously the battery that we selected to meet these requirements was the Ovonic 3S Lipo Battery 25C 2200mAh 11.1V Lipo Battery. This battery when connected in series with a second duplicate battery will be able to supply power to our motor within the correct voltage range.

III. HARDWARE DESIGN

This section is to focus on our methods for integrating and designing the electric speed controller of our system for regeneration and forward drive. With an understanding of the motor controlling theory this section deals with the theoretical concepts of the use of MOSFETs for the speed controller.

A. Electric Speed Controller

In order to implement our speed controller we first needed to understand what we needed to achieve. Our electronic speed controller needed to follow an analog joystick signal to change the speed of a switching network of field effect transistors. By switching the duty cycle of the transistors and/or the switching frequency, the motor speed can be changed. First, we looked at our brushless DC motor. It has a triple phase Motor with three input wires, because of the triple phase we cannot apply a normal differential voltage to it. The three wires create three coils inside the motor, the coils and the follow of current through them create a magnetic field and the motor spins due to attractive and repulsive forces between the permanent magnets and the coils. In order to make the motor rotate we needed to switch the polarity of the three-phase input wires and this quick switching of the polarity needed to be accomplished by control signals from our electronic speed controller.

We designed the electric speed controller to control the motor rotation using electronic switching of the three-phase input polarities and the design implementation that we went with was constructing a MOSFET H-bridge to sequentially alternate the internal polarity of the motor, and drive or resist the rotation of the motor. Using three pairs of power MOSFETs arranged in a bridge structure, each pair controls the switching of one phase of the motor. The MOSFETs are all controlled by using amplified signals from the microcontroller applied to the gate of each MOSFET. While the gate voltage is high, the MOSFETs act as a short and when the gate voltage is low they operate as an open. The MOSFETs operate in a rotating order where only two MOSFETs are active at a given time instance, creating a path from the battery high terminal, into one of the three-phase wires, out of another one of the three-phase wires, and to the battery low terminal. The PWM frequency is a trade off between the switching losses that occur at high frequencies, increased driving/braking force, and limiting average power to prevent damage to the motor.

Initially we believed that in order to interconnect the microcontroller and the triple phase bridge with our motor we needed to use drivers to step up the voltage from the microcontroller output pins to the MOSFET gate pins. We were under the impression that because MOSFETs act like a switch where we apply voltage at the gate and it will give us voltage at the source that we would need to match the voltage of our batteries to achieve substantial power. But in reality, our MOSFETs have a turn on voltage of 4 volts and our microcontroller can provide a maximum voltage of 5 volts. This along with the fact that effectively zero voltage will be dropped on the other side of the MOSFETs because it will be effectively tried to ground means that we will be able to drive a very high current through the motor without the use of amplifiers. We were able to repurpose the amplifiers to be used with the MOSFETs facilitating the headline and taillight automation. Because the lights were substantial loads voltage would be dropped on the opposite side of the MOSFETs and 5V is on the edge of the effective voltage range for the lights, adding the amplifier there would have a substantial effect on the functioning of the lights. Additionally, by allowing for a higher voltage we were able to reduce the current drawn by the lights.

Additionally, our electronic speed controller needed to provide regulated supply voltage for our drivers and microcontroller. And because the only batteries included in our design are the main scooter batteries, the constructed regulators had to be made to handle at least ~22.2V. To accomplish this we first added a 5 V regulator to our design to provide the required power for our microcontroller. Providing the required supply voltages to our drivers was slightly more complicated, thus was caused because the amplifiers we used as our drivers require both a negative and positive side supply voltage. Because of this we had to construct both a voltage regulator and voltage inverter to provide the required supply voltages.

For our project and for the motor we needed to drive, we decided to go with two HiLetgo HD44780 1602 batteries with 11.1 V connected in series, resulting in a considerable amount of voltage. As such, we needed to make sure that the MOSFETs could tolerate the maximum voltage that the battery is capable of outputting (~25V was observed in testing) and a comparable amperage rating to the motor (~10-15 A). The MOSFETs we ended up including in our project were the Bojack IRF540 MOSFETs. These MOSFETS have a drain to source voltage rating of 100 V, a gate to source voltage rating of 20 V, and current rating of 33 A. It was noted during our testing that these MOSFETs when used in our design have a tendency to reach very high temperatures. We observed that when these MOSFETs failed, they failed shorted, which would exacerbate the excess heat and had the capability to start electrical fires. Thus we included heat sinks into our design and added additional MOSFETs in parallel, to split the current passing through each MOSFET. These were low manpower and efficient methods to deal with dangerously high heat levels.

B. Lighting System

The lighting system is used to provide automated control over the head and tail lights. This is done by monitoring an externally mounted light sensor module and given a sufficiently low reading, sending a HIGH control signal to the input of an OP-AMP which setups up the voltage to 9V at the gate of a pair of MOSFET gates. These MOSFETs provide a path to ground when opened, from the battery high terminal, through the head and tail lights, to ground. We included two mosfets into the design to allow for the option of controlling the lights individually, as a new braking during daylight hours but due to time constraints we had to prioritize other features.

During our initial design of the lighting system included voltage doublers, which were connected to a regulated 5V. This would allow for greater voltage to be applied to the loads of the head and tail lights, reducing current drawn from the batteries. However, while testing our design we discovered that the voltage doublers LOW output voltage level was ~5V. This meant the MOSFET gates were always open and lights were always always drawing some amount of current, even though 5V wasn't sufficient to power them on. To eliminate this issue we went with the OP-AMP, where the LOW voltage level was 0V, which allowed the gates to properly close.

IV. SOFTWARE DETAIL

Before I can go into detail about the software portion of our EZ Scooter, I will go through the goals and features I am focusing on to complete it fully. The main goals with the project is for a fully functioning electric scooter to be able to move forward, backwards, and stop with regenerative brakes. The additional features include a LCD display that will show a functional speedometer, name of the scooter (EZ Scooter), and battery level. With all of these different types of features for the software system we will end up with a unique and easy to understand system for any user.

When it came to choosing the microcontroller for us that we needed to code in we were looking at mainly two options, one being the ATmega328P and ARMv8-A, which both had their advantages and disadvantages for their uses as you can see below in *Table 3*.

Microcontroller	**Advantages**	**Disadvantages**
ATmega328P	Cheaper Analog Inputs Flexible power	Fewer Pins Slow 16MHz C/C++ only
ARMv8-A	Access Linux Add-Ons Functionality	Costlier Digital only Less Power

Table 3: Comparison between Microcontrollers

We chose to go with the ATmega328P as we felt for our project it worked a lot better with being able to use analog inputs with our joystick that will have the ability to move the scooter forward, backwards, and stop. The ATmega328P also has a cheaper price to fit our budget and flexible power requirements for us to be able to test with. The ARMv8-A did not fit our needs and would be overall better as a learning experience, but did not provide the analog inputs that the ATmega328P has.

A. System Firmware

The software that controls our EZ Scooter is written in C++, uploaded to a software called Arduino IDE, where we can verify and test our code over and over until we get a desired result. It refrences a few libraries such as LiquidCrystal for the LCD, TinyGPS++for the GPS module, and SoftwareSerial to output our results onto the serial monitor. It is over 200 lines of code that is compiled with digital outputs and analog inputs that allow us to get our desired result. When coding for the ATmega328P we focus on two main functions: the void setup(), which runs the setuo code once and the void loop(), which focuses on the main code to run repeatedly. In this void loop() function, we call other functions that we made for specific features.

```
float batteryRead();
void milesperHour();
void lcdDisplay();
void joyStick();
void hubMotor();
```

Fig. 3 Function Calls

Above we can see *Fig.2* which represents some of the function calls we will perform within the void loop function(). (1) The batteryRead function focuses on reading the battery voltage per % and calculates the voltage through float numbers to get more of an accurate representation of the percent of the voltage. (2) The milesperHour function focuses on calculating the miles per hour of the scooter with the use of the GPS module constantly updating the LCD the values that are being calculated based on your location. (3) The lcdDisplay() function focuses on all the features within the LCD such as showing battery level, the name "EZ Scooter", mph, latitude/longitude, time, and even date. This prints all the results on the to LCD and shows the calculated values. (4) The joyStick function focuses on the analog inputs of our project where it will map the x and y values of the joystick itself and will recognize whether to move forward, backwards, or stop depending on the direction you point it in using analogRead functions. (5) The hubMotor focuses on the main issue on how it will detect highs and lows with different cases that it must check from the joyStick function. Using the digitalWrite function to write the highs and lows of each movement to a specific digital pin. All of these functions consist of constant ints and floats that will allow for proper calculations.

If all the functions pass through correctly then we will have a consistent system that will work in tangent with each other like no other. If one of the functions don't pass through for specifically dealing with the GPS Module then that most likely means the GPS module has lost the satellite signal, which is mainly caused by being inside or near tall buildings.

The functions I listed above is just a small glimpse of what is occurring in the system firmware, so to visualize it better and understand it I have a flowchart that represents the systems thought process:

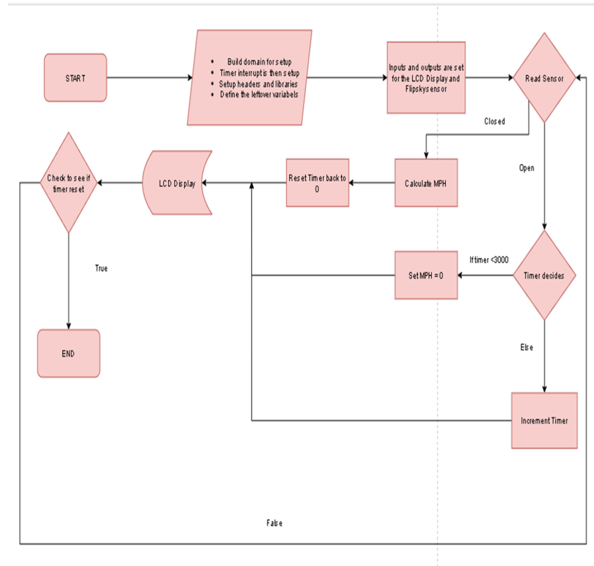


Fig. 4 Coded Flow Chart

B. Memory and Structure

The firmware would not be the same without an effective and reliable way of being able to store and retrieve the data that our code results in. The digital and analog inputs and outputs are stored within the ATmega328P that allow for multiple uses for each individual pin that is pointed to within the code such as declaring a variable to a digital pin (const int rs = 12, en = 11, d4 = 4, d5 = 5, d6 = 6, d7 = 7;) or declaring a variable to an analog pin (const int analogPin = A4;). All these declarations are positioned right in the beginning of the code right after the libraries are set, so that the variables can be called within any of the functions from *Fig 2*.

```
// include the library code:
#include <LiquidCrystal.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int RXPin = 0, TXPin = 1;
const uint32_t GPSBaud = 9600;
const int rs = 12, en = 11, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
const int analogPin = A4;
```

Fig.5 Libraries and Initializations

The RX pin focuses on receiving the data and the TX pin focuses transmitting data and is mainly used for the in tangent with the GPS module to be able to constantly update the ATmega328P information on the data that needs to be displayed on the LCD.

V. BOARD DESIGN

The whole system is implanted on a PCB board which has a low-cost, re-workable, and compact size that is organized rather than the use of many wires. Our PCB contains a lot of parts that we have to commit to and make sure we do not mess up with how the parts are put on. This method is efficient and the best way to approach our system with a plus of low electronics noise.

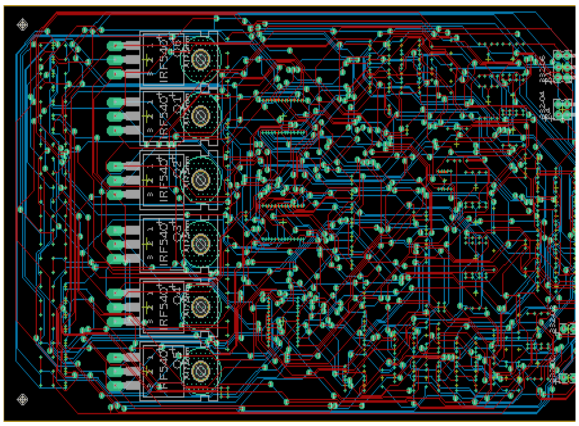


Fig 6 PCB

VI. CONCLUSION

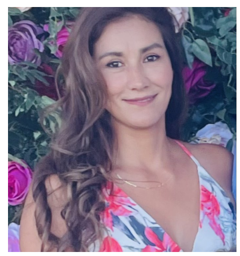
Working as a group project for these two semesters has given us a socially significantly enhanced learning experience in terms of the strength of the memories formed related to the engineering domains encompassed by our project. These experience will stay with us for many years to come and have given us a breadth of understanding that would be difficult to achieve in a classroom setting. Thanks to this Senior Design and the amazing learning opportunity it gave us, we reinforced the fundamental concepts that we will take with us as we begin our careers in the private sector. Some of these key concepts include device testing, design implementation at both the component and system level, and project management.

ACKNOWLEDGEMENT

The authors wish to acknowledge The assistance and support of Dr. Richie, and professors Hao Zheng and Mark Llewellyn; University of Central Florida and our fourth member that pass awy, Elizabeth Lovera.

THE ENGINEERS

Jacob Jardines is a 23-year-old graduating Computer Engineering student who hopes to eventually work as a software engineer for a large company such as Microsoft, Apple, Amazon, Google, Tesla, etc. Software design and research keeps him motivated and focused.



Carolina Geary a 35 year-old graduating Electrical Engineering student at University of Central Florida. Hopes to pursue a career in the specialized area of power systems, working for a company such as Tesla, Northrop Grumman.

Tony Naylor a 28 year-old graduating Electrical Engineering student at University of Central Florida. Hopes to pursue a career in the defense industry, working for a company such as Lockheed Martin or Raytheon.

REFERENCES

1. “The Paris Agreement.” *Unfccc.int*, United Nations, https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement
2. MacArthur, John, Jennifer Dill, and Mark Person. "Electric scooters in North America:

Results of an online survey." *Transportation Research Record* 2468.1 (2014): 123-130., https://journals.sagepub.com/doi/abs/10.3141/2468-14

3. Apostolou, Georgia, Angèle Reinders, and Karst Geurs. "An overview of existing experiences with solar-powered e-bikes." *Energies* 11.8 (2018): 2129., https://www.mdpi.com/1996-1073/11/8/2129
4. "Motorcycle, Motor Scooter, Moped and Motorized Scooter." *Florida Department of Highway Safety and Motor Vehicles*, 21 Mar. 2022, https://www.flhsmv.gov/driver-licenses-id-cards/motorcycle-rider-education-endorsements/motorcycle-motor-scooter-moped-and-motorized-scooter/.
5. *, N. (2021, November 30). *Choosing the right electric scooter throttle: Thumb vs trigger*. Varla Scooter. Retrieved April 26, 2022, from https://varlascooter.com/blogs/knowledge/choosing-the-right-electric-scooter-throttle-thumb-vs-trigger/
6. ago, yida 2 years, Yida, posts, yidaS. author's, & posts, S. author's. (2019, November 15). *Arduino uno alternatives - which microcontroller to buy?* Latest Open Tech From Seeed. Retrieved April 26, 2022, from https://www.seeedstudio.com/blog/2019/11/15/arduino-uno-alternatives-which-microcontroller-to-buy/
7. *Amazon.com: Hiletgo Game Joystick Sensor Game Controller ...* (n.d.). Retrieved April 26, 2022, from https://www.amazon.com/HiLetgo-Controller-JoyStick-Breakout-Arduino/dp/B00P7QBGD2
8. *Build a DIY electric scooter*. Nuts and Volts Magazine. (n.d.). Retrieved April 26, 2022, from https://www.nutsvolts.com/magazine/article/build-a-diy-electric-scooter