Project's Report –surfboard

May 2020

Project's Report –surfboard

Students:

Yann AORTE

Ran GUO

Yilu MENG

Tutors: George Antoine

Company: ARUE

THANKS

During the process of the completion of this project, we have received much guidance, help and support from many people. Here I'd like to extend my sincere gratitude.

Firstly, I am greatly indebted to Mr. George Antoine for his patiently explain through the entire process from preparation to accomplishment. Without his great patience and encouragement, the accomplishment of project wouldn't have been possible.

Secondly, many thanks go to Mr. Langella and other teachers who have taught me during the last two semesters. They have enriched my knowledge in the field of electronic. I am also deeply impressed by their strong senses of responsibility and professional dedication. Whenever I encounter any difficulties, they offer me valuable suggestions and constant encouragement.

Last but not the least I owe much to my teammates and classmates who are helpful and important in making the project a reality. I deeply appreciate their contributions.

TABLE OF CONTENT

Content

I Introduction

The aim of the project is to design and develop a water-jet system designed to equip and facilitate the practice of surfing with foil. This profiled wing located under the surface of the water is directly integrated on boards surfactants giving them sufficient lift to raise them above the level of the water.

Our prototype is intended to provide adequate thrust to enable the user to "foil" without outside help (sailing, jet ski ...) and whatever the water conditions (restless, calm, with or without waves ...).

This system therefore contains a user-controlled motorization system. Of numerous researches, simulations and modeling have been undertaken to determine the parts, equipment, connectors, or the manufacturing processes to be adopted for this type of product.

The entire mechanical and manufacturing part of this project has already been carried out by other groups of students during the tutored projects.

In addition, the motor, the VESC (card that control the remote control of the motor), the battery and the remote control have already been selected and ordered by another group of students.

However, Because the battery size is too large, it is not conducive to putting it in the surfboard. During the project of this semester, we re-optimized the selection and ordered the battery.

For the completion of this project, the tasks to be carried out were divided into three parts.

-A person will implement new buttons from the remote such as OFF system and an activation of LEDs under the board, to accomplish that the person will use an ST LINK V2 in order to flash a new firmware with new options implemented.

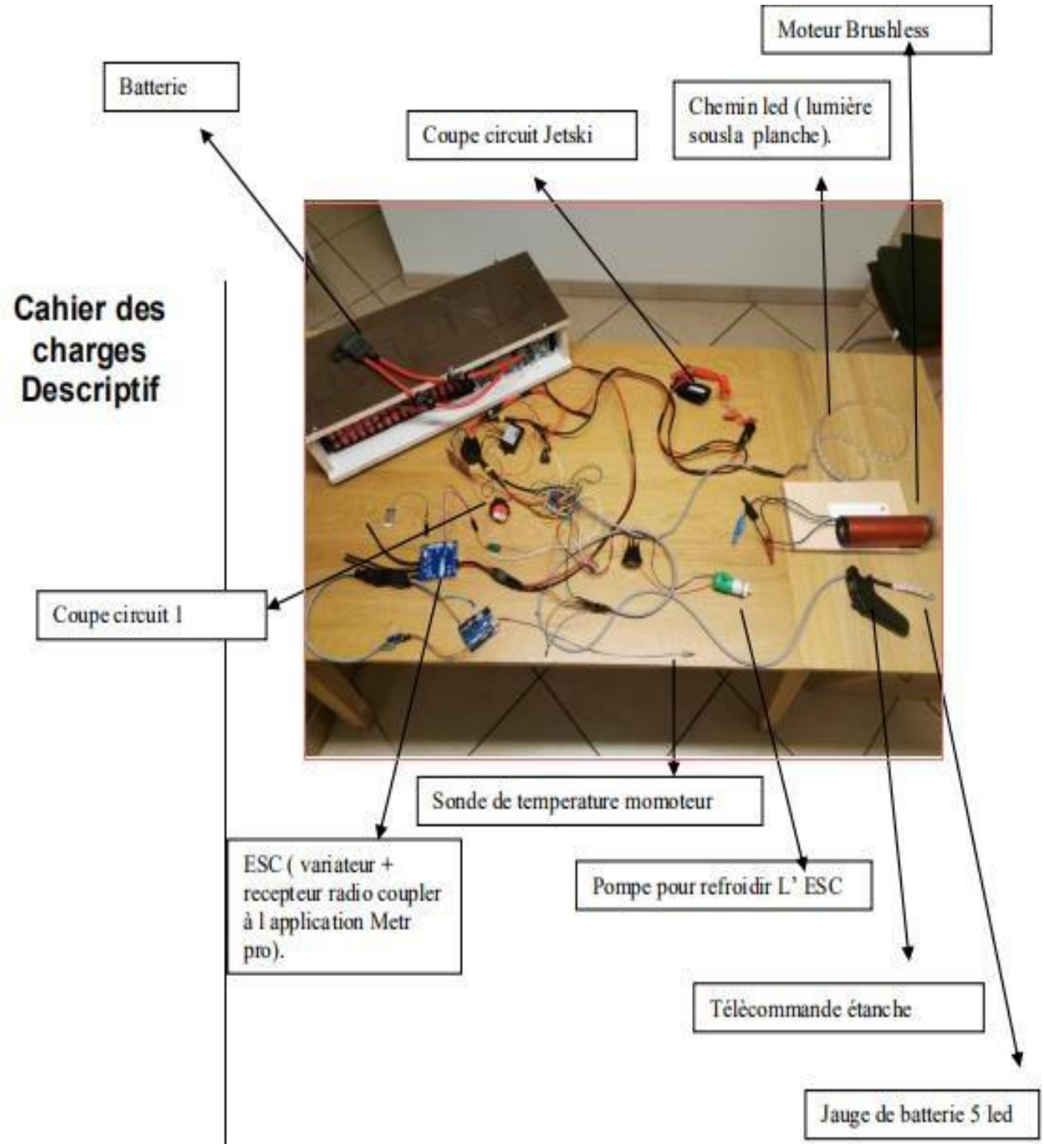
-A person that will be responsible of the part of Arduino. write programs of Arduino for controlling the LED band for beauty and write programs for controlling the LED band which present the battery power at the time.

-Another person will oversee the UART communication between the VESC Controller and the Arduino to get and use the information of the VESC such as the RPM, the battery charge, the temperature of the controller/motor...

We started by understanding the principle, then we checked if the circuit is working properly and draw the diagram of the circuit to understand better the existing circuit.

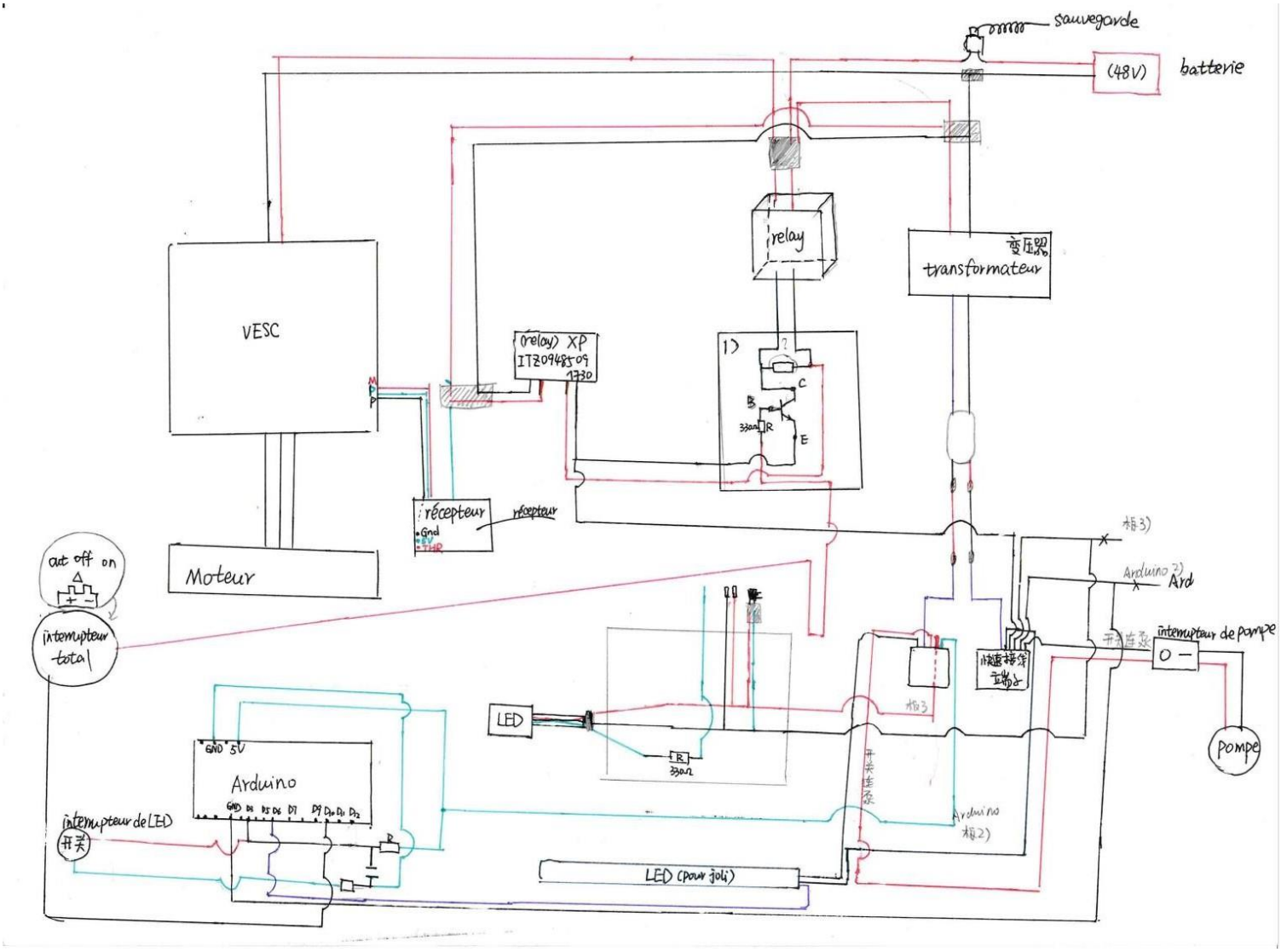
I. 1 Description from semester 1

I.1.1 Picture of internal circuit structure:



I.1.2 Circuit diagram

The battery supplies power to the whole system through a transformer.



There is a main switch on-off which commands the whole circuit and a small switch commands pump.

There is a communication between VESC and Arduino Mega.

VESC connects with the motor. And VESC connect with a receiver which receives the signal of the remote control.

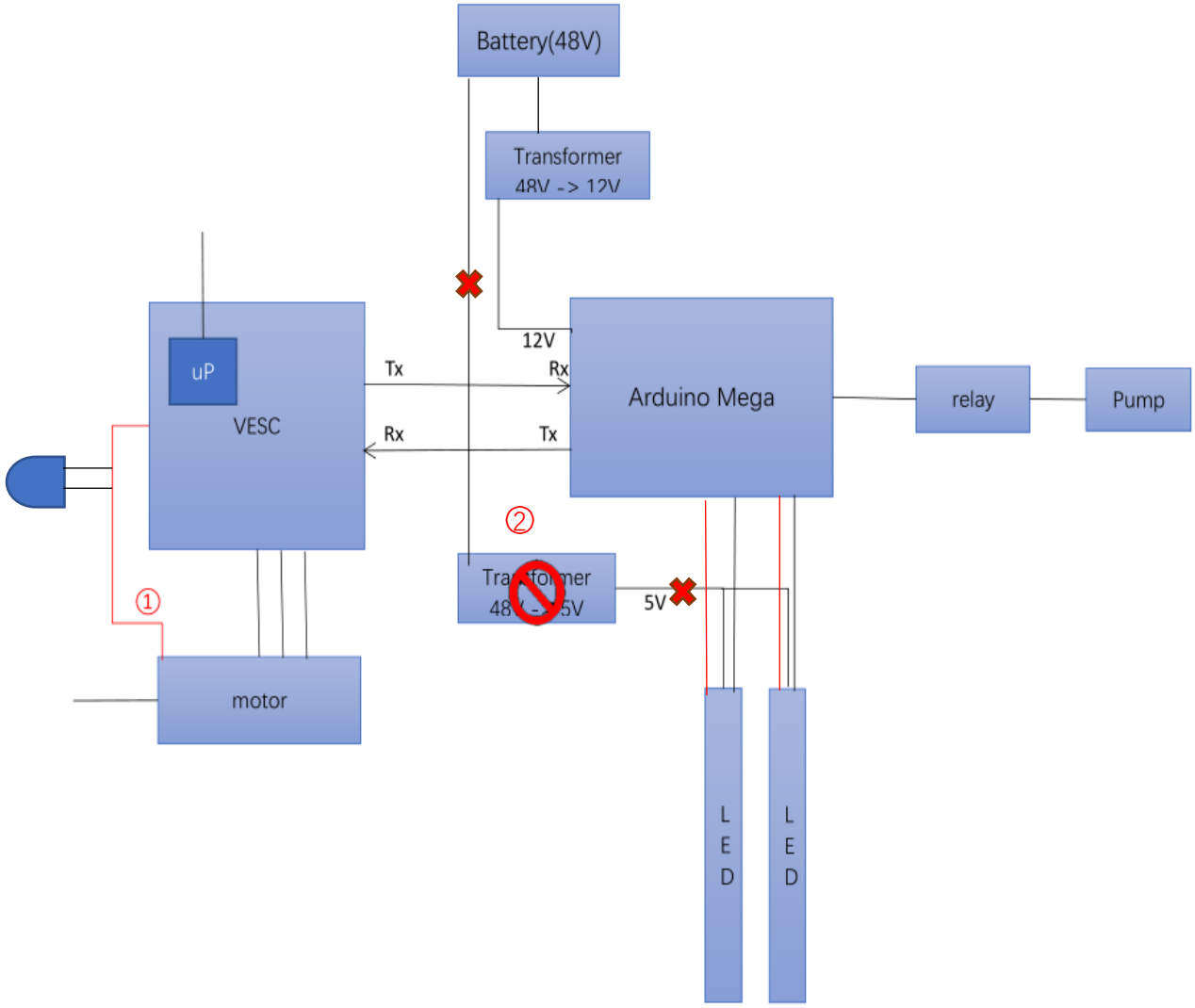
There are totally 3 circuit boards:

The first one is connected to the main switch on-off.

The second one is connected to the small LED, which lights will display battery level

The last one is connected to the small Arduino, which controls the LED (for beauty).

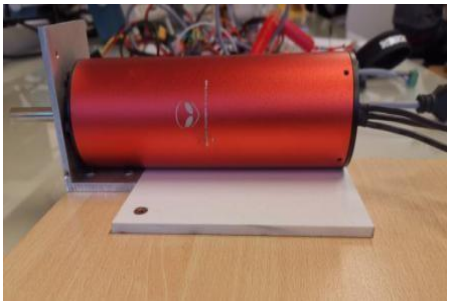
I.1.3 Figure: the structure of the foil



In Semester 2, due to the arrival of a new student in the project we changed our approach, we decided to start with only the main components to have something safe, refined and clearly understandable.

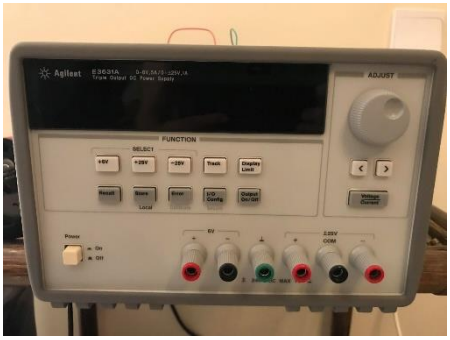
I.1.4 Components of the new system

- Motor 1 APS 56115 Inrunner brushless motor 100KV 5000W
  Imax=50A.
  Vmax=100.8V.

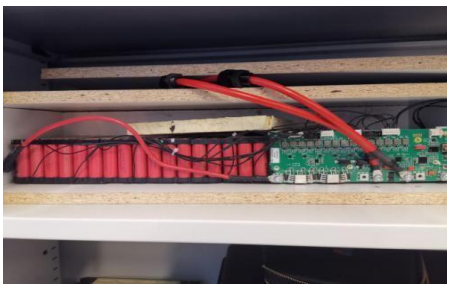


- Motor 2 (final motor) APS 56200 Inrunner brushless motor 100KV 8000W
  Imax = 80A.
  Vmax = 100.8V.

- Power supply for safety reason (from Polytech):



- Battery 1: 48V

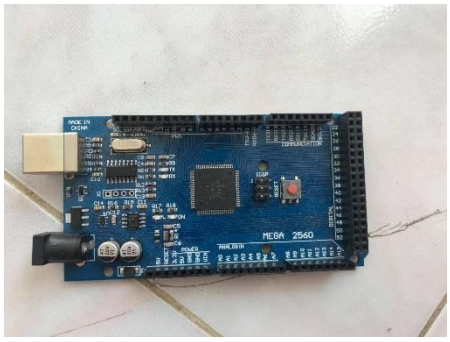


- Battery 2 (final battery): 48V 49ah

- VESC 6:

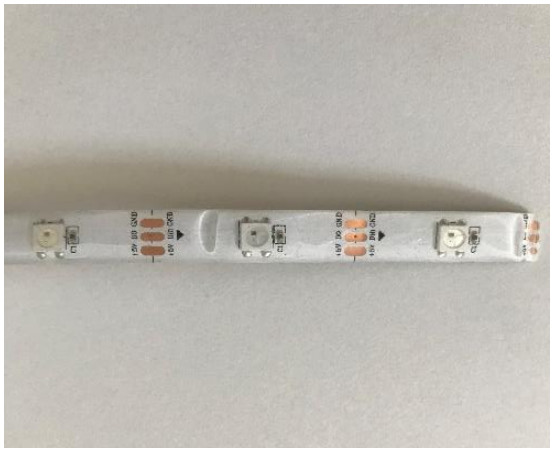
- VESC 75/300 (final controller) with remote receiver:

- Arduino MEGA:

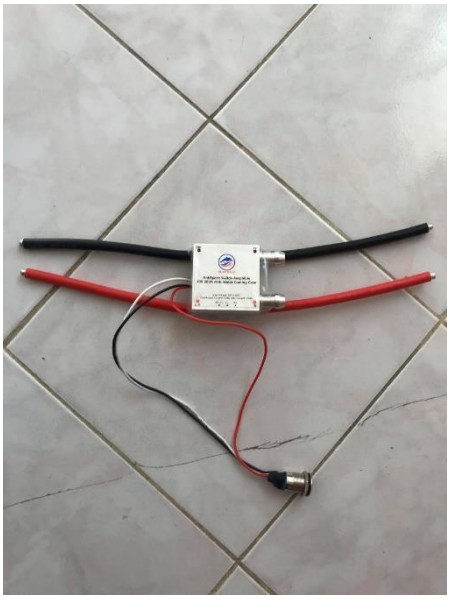


- Pump:

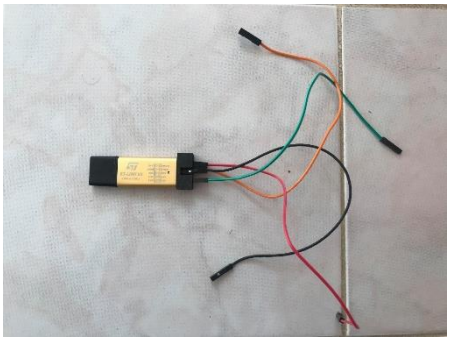
- LED:



- Switch On-Off AntiSpark Switch Amphibia 30S 200A:



- ST LINK v2 usb :



I.1.5 Principle of this system

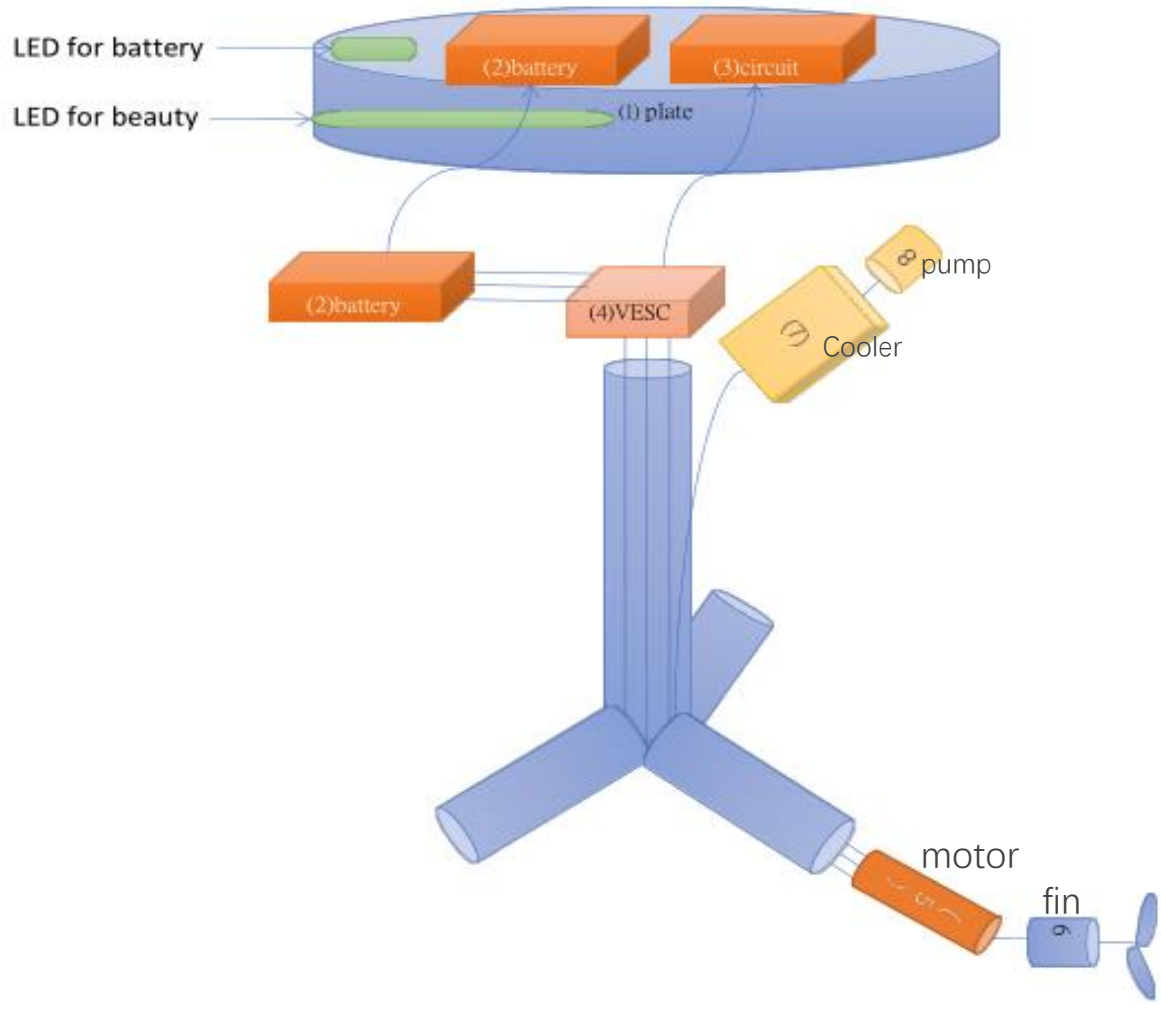
There are two main parts of the surfboard. The plate is on the water and the propeller is under the water.

The battery of power, the circuit and cooling system (including pump) are within the plate, and the light will display battery level. And the LED is at the bottom of the plate for embellishment, when the user is surfing at sea, the color of the LED strip flashes gradually Then, the motor which connects with the circuit and battery is inside the propeller.

The user stands on the plate and holds a telecommand in hand, when he presses the button of telecommand, the motor will rotate rapid, and it will become hot, but the motor is under the water, and the sea will cool it.

When the surfboard works for time , the temperature of VESC will rise, the pump will turn on and the water will enter the cooling system and take away some heat, but when it is too high , the power will also be cut off forcibly.

Schema of the surfboard

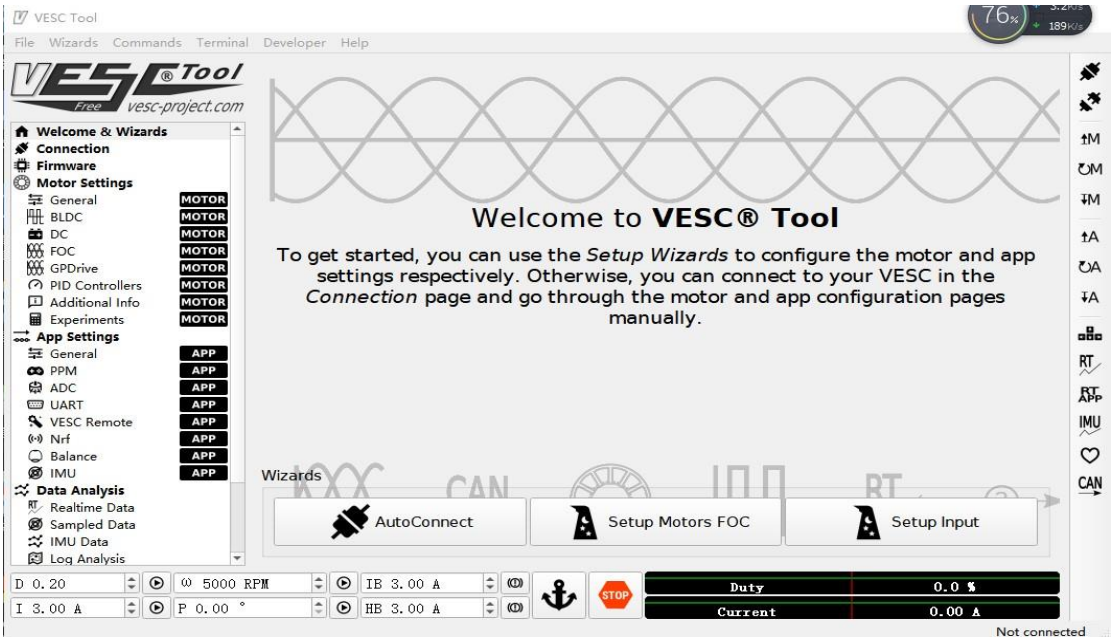


I.2 Tools

I.2.1 VESC tool

VESC Tool can be used with the VESC compatible Hardware and most VESC based Hardware of the past. It allows us to generate real time data and find out how to set up our drive with perfection. It is a strong and useful software for us. We wrote a tutorial in order to help our tutor or someone else in how to perfectly configurate your motor (refer to Annex).

In this project, we will use VESC tool to generate real time data. We will connect the VESC with our computer.



I.2.2 Telecommand

We used the telecommand maytec to control the speed of the motor. But it's not open source like the VESC, we can't change its function.

I.2.3 Arduino

We used the Arduino mega to write a program to control the LED.

I.3 Tasks

I.3.1 UART VESC

- UART communication between the VESC controller and Arduino for the purpose of getting all data that the VESC knows.

I.3.2 Control the LED

- the LED for battery capacity.

The light on the plate surface could display battery level. When the user is surfing at sea, it is easy to observe the remaining power of the battery.

- the LED for beauty.

The lamp at the bottom of the plate can change its own color, showing a cool effect at night.

I.3.3 Device optimization

In order to obtain higher performance and better user experience, we need to optimize some devices

- Optimize the battery.

Because the original battery size is too large to fit in the plate, we should find and order the battery in terms of battery size, voltage.

- Optimize the switch and button.

We want to add more cooling devices and make it easier to operate

- Optimize the motor and the VESC.

We hope for more power to bring users a better experience

II. UART VESC.

Description:

Thanks to the UART communication between VESC and Arduino we can get real time data and set some values from the VESC such as the input voltage, the current motor, the RPM, the temperature etc. These data will be useful for us for the LEDs or for implement new functions.

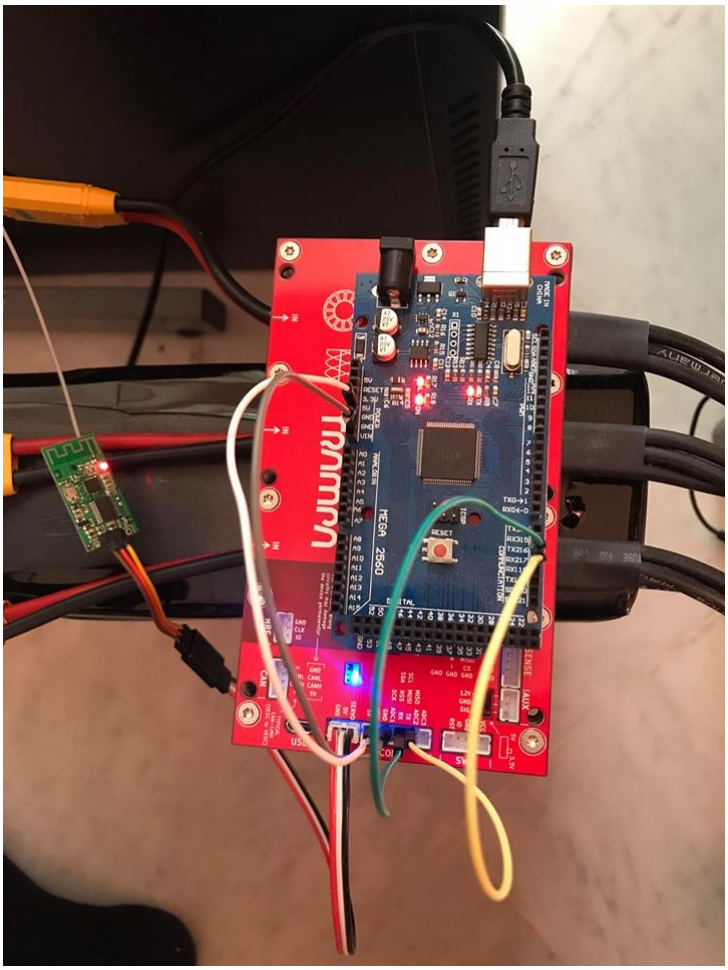
II.1 Hardware.

First, we need to connect correctly the VESC Controller and Arduino MEGA. The UART communication uses reception/transmission protocol. To achieve that we will use the COMM output of the VESC because it's the only pin where we can find Rx/Tx pins. For a good transfer of data, we need to cross each Rx/Tx from each device (we use port 2 of Rx/Tx from the Arduino):

Rx (COMM port VESC) <-> Tx2 (COMM port Arduino)

Tx (COMM port VESC) <-> Rx2 (COMM port Arduino)

It's not finished, we don't have to forget to connect each ground GND of each device. We obtain:



We just add a wire between PC and Arduino as means to display received data. We can add a wire between 5V (VESC) to 5V (Power port Arduino) in preparation for having the Arduino power supplied by the VESC. No need to add a converter between battery 48V and Arduino what refined the all circuit.

II.2 Software.

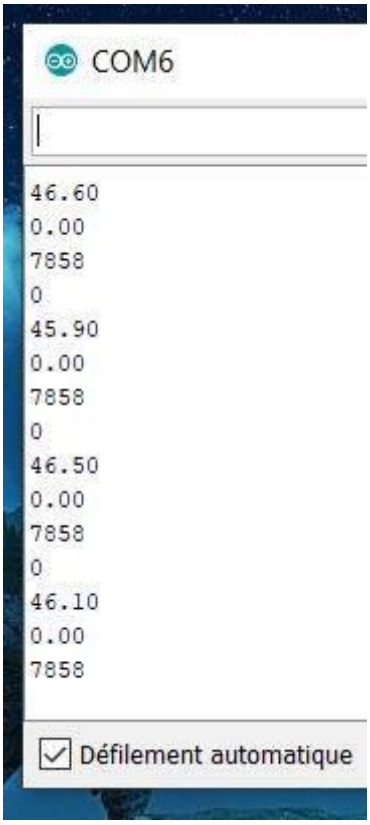
Now that everything is correctly connected, we need to implement the UART with the Arduino software for coding. To achieve this, we will use VescUart library written by Benjamin VEDDER. This is an Arduino library for interfacing with a VESC over UART and it's updated for the newest VESC firmware (FW3.40) and cleaned up a bit. That's why we must update our firmware using VESC Tool. To use the library, you will have initiate the

VescUart class and set the Serial port for UART communication:

VESC_UART

```
#include <buffer.h>
#include <crc.h>
#include <datatypes.h>
#include <VescUart.h>

VescUart UART;

void setup() {
  Serial.begin(9600);
  Serial2.begin(19200);
  while (!Serial2) {;}

  UART.setSerialPort(&Serial2);
}

void loop() {

   if ( UART.getVescValues() ) {

    Serial.println(UART.data.rpm);
    Serial.println(UART.data.inpVoltage);
    Serial.println(UART.data.ampHours);
    Serial.println(UART.data.tachometer);

  }
  else
  {
    Serial.println("Failed to get data!");
  }

  delay(500);
}
```

Furthermore, we must write the correct baud rate of the UART communication using VESC Tool to 19200 bps to achieve a perfect comprehension between each device. That is why in the Arduino code we put the serial port 2 Rx/Tx as well to 19200 bps. 9600 bps is used for the serial monitor from the PC intending to display data. You can now safely use the get functions and change the values of the class:



What is powerful with UART communication is that we have directly access to a lot value (Temperature, Voltage, Current, Duty Cycle, RPM, Tachometer…) but that is not limited to these values because we can calculate other useful data from the directly received data such as the power, the speed, the distance:

```
power = current*voltage;
c_speed = (VescMeasuredValues.rpm/38)*3.14159265359*0.000083*60;
c_dist = (VescMeasuredValues.tachometer/38)*3.14159265359*0.000083;
```

Another aspect is that we can set data now to control the VESC from the Arduino, we could add a function OFF motor by just doing UART.setCurrent(0).

III. Realization Arduino.

Introduction:

This part is mainly the connection between the Arduino and the light strip. In our project, there are two light strips, one is for good-looking and the other is for displaying the current battery level.

Because these two parts are carried out at home, due to lack of some equipment, some simulations were carried out.

III.1 controlling the LED for beauty.

Description:

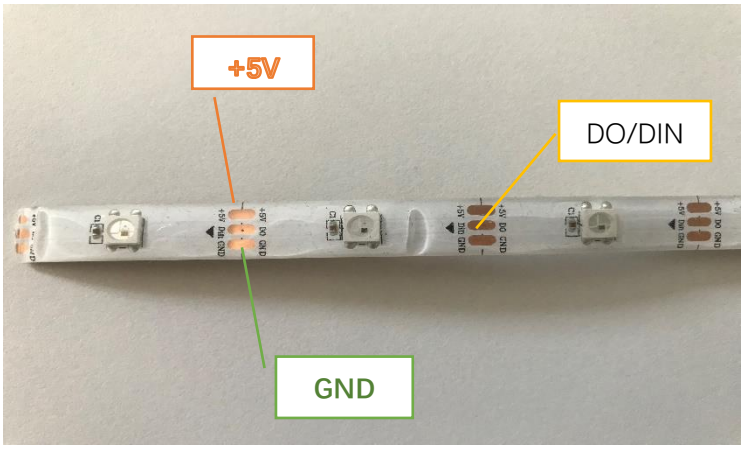
There is a total of 36 leds in this string of lights, which are located under the side of the surfboard and above the water.

We canceled the button to control the light strip and directly connected the light strip to the Arduino. To look good, we let the light belt continuously cycle through several colors, and each small light bulb changes colors in turn, adding some time delay to them.

Realization:

There is a light-emitting diode (LED) on the LED module, which has two states: on or off.

The module has three pins, GND is connected to the ground, + 5V is connected to the 5V output on the Arduino, and DIN/DO is connected to the IO pin on the Arduino.



We use the library *<FastLED.h>* to control the led band.

In our program, we set the LED_PIN to 6 port, the number of leds is 36, which means that connecting the DIN / DO interface to the 6 port of Arduino as the port to receive the signal.

```
#include<FastLED.h>
#define LED_PIN    6
#define NUM_LEDS   36

```

Then we use CRGB to control LEDs' colors, which is Representation of an RGB pixel (Red, Green, Blue).

We need to set up a storage block, which will be used to store and process led data using code bellows, thereby creating an array that we can manipulate to set / clear led data:

```
CRGB leds[NUM_LEDS];
```

We can find CRGB related reference on this webpage:
http://fastled.io/docs/3.1/struct_c_r_g_b.html

And then, in our setup function, we setup our leds.

This tells the library that there is a string of NEOPIXEL on pin LED_PIN which is 6, and there are NUM_LEDS (that is, 36), and those leds will use the led array leds we declared before.

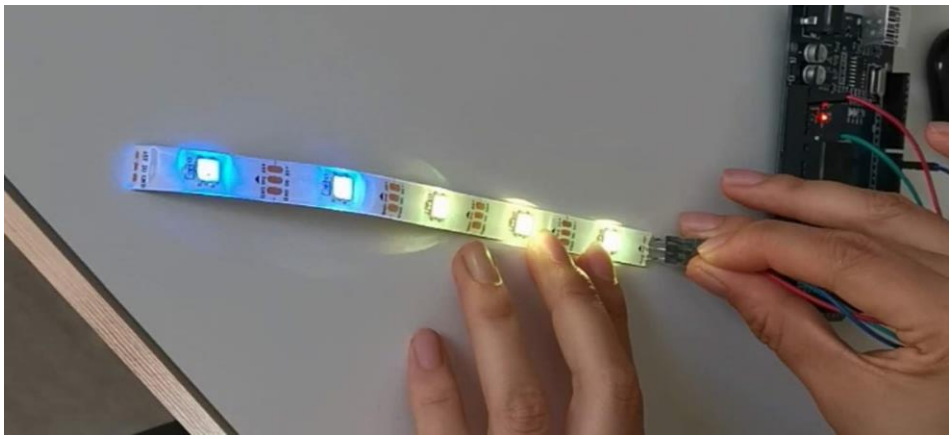
```
void setup() {
  FastLED.addLeds<NEOPIXEL, LED_PIN>(leds, NUM_LEDS);
}
```

We can find *addLeds* related reference on this webpage:
http://fastled.io/docs/3.1/class_c_fast_l_e_d.html#a79df28eb68fc2062b995f9000aed274c

Finally, in the loop, let the light strips light up one by one or change the color, first use CRGB to give the color to the small light, and use the show () function to make it light, and finally use delay () to generate a certain time delay, so that we can see Little lights change one by one.

```
13 void loop() {
14 for(int i=0;i<NUM_LEDS;i++){
15   leds[i]=CRGB(255,0,0);
16   FastLED.show();
17   delay(200);
```

I tested the code at home with a LEDs band with 5 LEDs, I got the result as below:



You can find more details which we realized at https://youtu.be/DJeYX31gvYo.

III.2 controlling the LED for battery capacity.

Description:

In use, we want to know the power of the battery, so that we know whether the battery power can support the work and charge the battery in time.

So, we intend to use five small lights to display the power. When the five small lights are green, it means that the battery is fully charged. When the battery voltage drops, the green light turns off and the red light turns on. When all five green lights turn off, only the red light is flashing which presents the battery is not enough.

we want to measure 46,2V-58.8V, but Arduino can only measure 0-5V and display between 0-1023. So we transform 46.2-58.8V to 3.93-5V, we use a converter to give the tension to Arduino and led 5V. And we connect the battery (which is converted to 5V) to pin A4.

Realization:

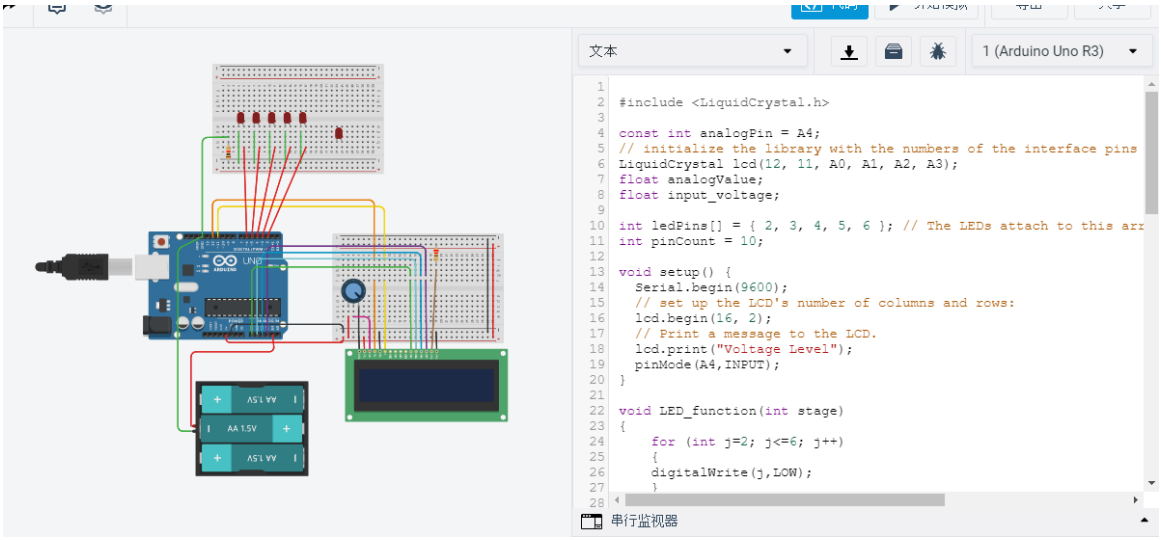
III.2.1 Simulation avec thinker card

First, at the beginning of this task, because there are no related devices, we performed Arduino simulation on the Tinkercad.com website and replaced the light strip with 5 small LEDs.

Here is the lien of the Tinkercad.com:

https://www.tinkercad.com/

And our realization:



We use the library <LiquidCrystal.h> to configure the LCD which use the pin 12, 11, A0, A1, A2, A3.

We setup Pin A4 as input which receive the voltage of battery after convention (0-5V). We add a function which named LED_function. When we receive the voltage signal, we call this function. The variable stage means the number of the LEDs which will turn on. And 2-6 is the pin number which output the signal to LEDs.

The LED_function is as below:

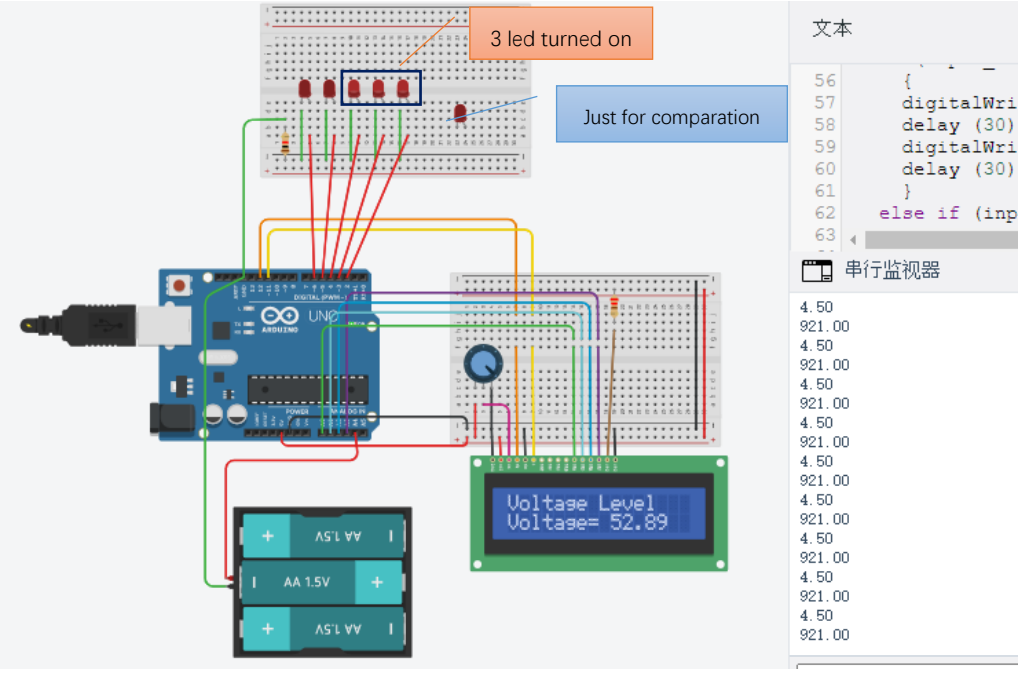
```
22  void LED_function(int stage)
23  {
24      for (int j=2; j<=6; j++)
25      {
26      digitalWrite(j,LOW);
27      }
28      for (int i=1, l=2; i<=stage; i++,l++)
29      {
30      digitalWrite(l,HIGH);     //delay(30);
31      }
32  }
```

We set all the led pins to low potential, and then re-turn on the LEDs after receiving the input signal.

In this case, when the voltage is very low (<3.9V), 5 LEDs turned off.

The voltage becomes higher, more LEDs turn on.

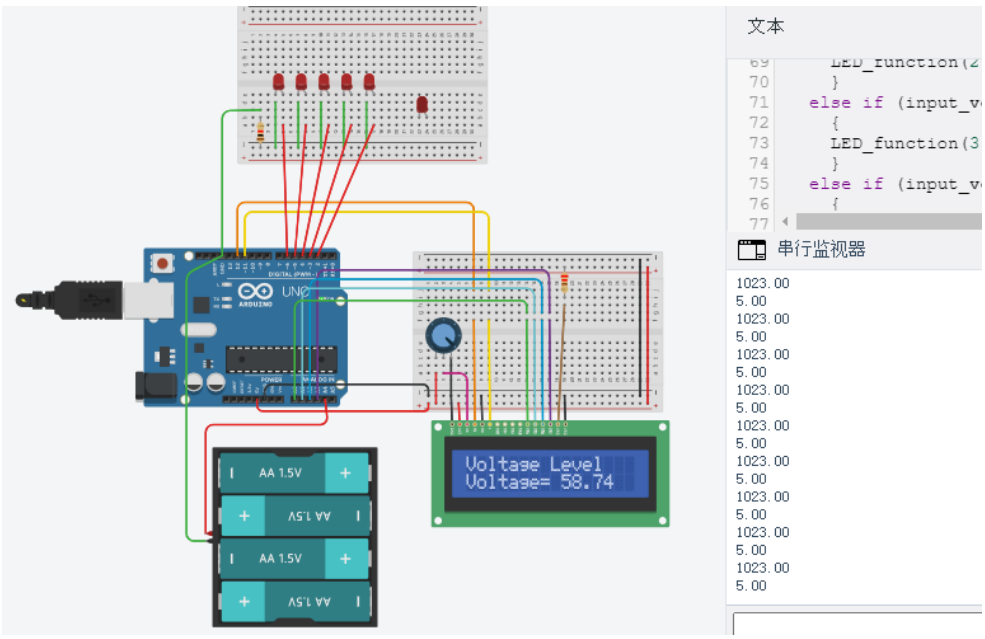
This is result of this simulation:



As the figure above, we use 3 batteries, so we receive the input is 4,5V which is equivalent to 52.89V. So, there are 3 LEDs which turned on.

As the figure below, we add 1 battery, we can see all 5 LEDs turned on. And the voltage now equivalent to 58.74V.

Finally, we test it with 2 batteries, all 5 LEDS turned off.



You can find more details which we realized at https://youtu.be/yeCEDsFHaAk .

This is the lien of our design at Tinkercad.com:
https://www.tinkercad.com/things/kAF53lgOk8F-epic-gaaris/editel?sharecode=Hgkelv1kpaFCHA8Qj2qKFaMCuk4Yh_DpYF1-SEitZHM

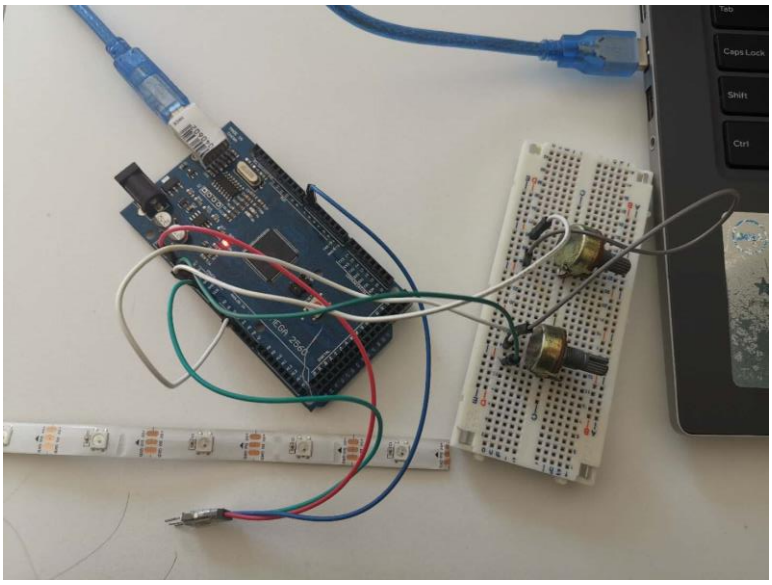
III.2.2 Test with Arduino and LEDs band.

We changed some codes before and add the codes for controlling LEDs band with using the library <FastLED.h>. This is my LED_function:

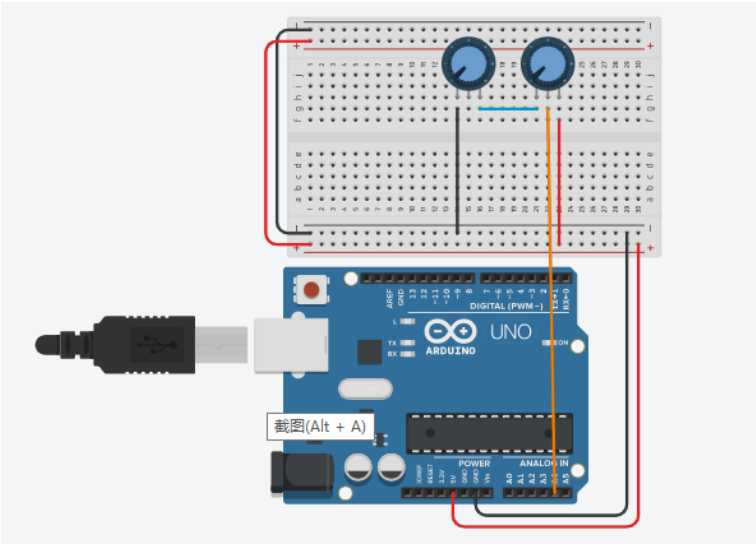
```
void LED_function(int stage)
{
  for(int i=0;i<stage;i++){
  leds[i]=CRGB(0,255,0);
  FastLED.show();
 }
 if(stage!=NUM_LEDS-1){
 for(int i=stage;i<NUM_LEDS;i++){
  leds[i]=CRGB(255,0,0);
  FastLED.show();
  }
 }
}
```

In this part, at first, we don't have the battery and we can't find simulation online with library <FastLED.h>, so we don't test it.

Later we thought of a way to solve it, with the circuit below:



The details of the circuit:



We use this circuit above to get the voltage input at pin A4.

We use 2 potentiometers to change the voltage input. We use 2 potentiometers because we do not have resistance and the other device. (These 2 potentiometers are borrowed from one of our friends, one is 5kΩ, the other is 10KΩ)

We adjust the potentiometer left as 5KΩ and we change the resistance of right potentiometer to change the voltage input.

We have tested with using just one potentiometer, but it does not work, the reason

may be the resistance is inconvenient to adjust with one potentiometer.

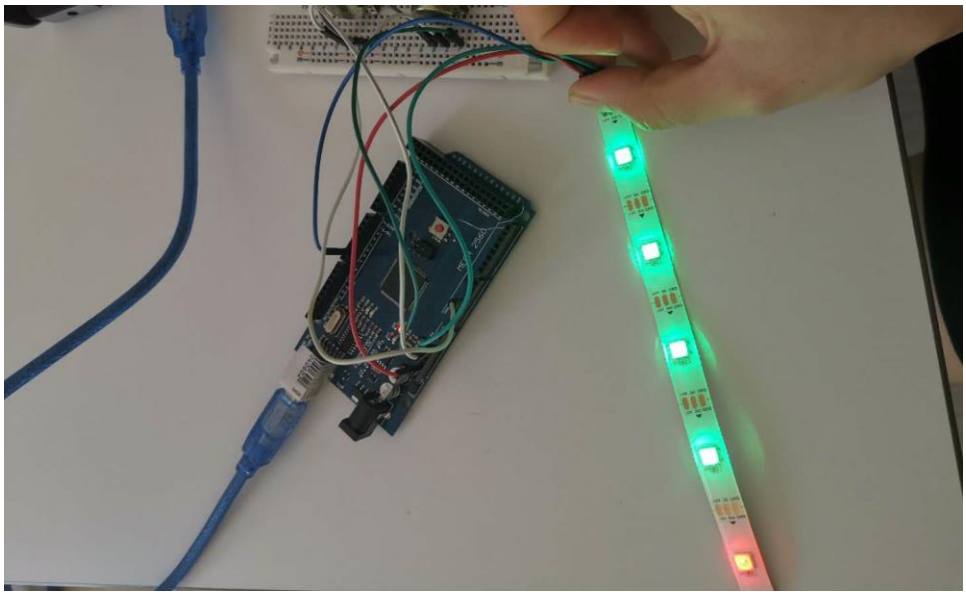
Then the two potentiometers are connected in series to divide the voltage, to achieve A4 to obtain the voltage value we want.

Using this voltage generator, we test it with serial monitor to show the voltage input which received by pin A4.

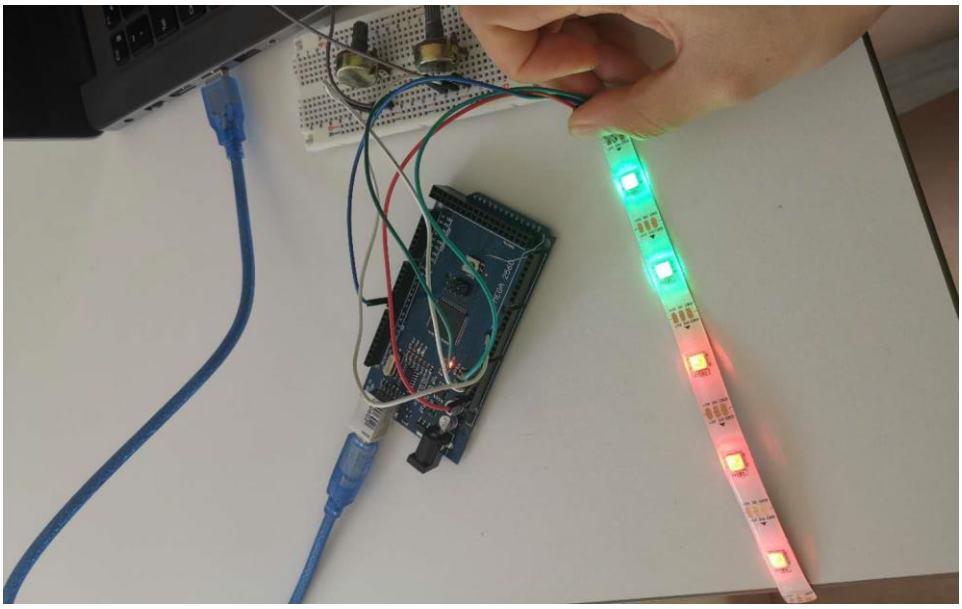
These are the results we get when we adjust the potentiometer:

1023.00	886.00
5.00	4.33
1023.00	871.00
5.00	4.25
964.00	825.00
4.71	4.03
956.00	824.00
4.67	4.02
938.00	755.00
4.58	3.69
930.00	754.00
4.54	3.68

We use these voltages to test our codes which control the LEDs band, we can get the result we want.



In this case, four green LEDs turned on and one red LED turned on.



In this case, two green LEDs turned on and three red LEDs turned on.

In this way, we can show the voltage of battery with LEDs bend, after a voltage converter which transform the voltage to 0-5V.

You can find more details which we realized at https://youtu.be/HEfV2464kw8

IV. Device optimization.

Introduction:

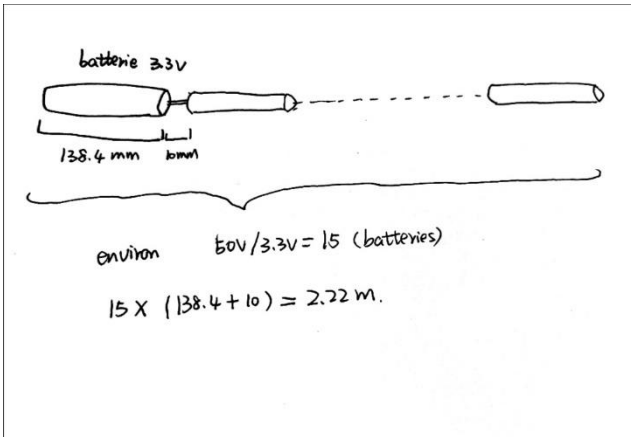
During the project, we have made some optimizations on the circuit devices, such as batteries, buttons, motor, VESC and so on. To make it easier to put the circuit in the surfboard and increase the tightness.

IV.1 Optimize the battery

Because the original battery size is too large to fit in the plate, we should find and order the battery in terms of battery size, voltage. That's why with our tutor we went in a battery constructor named b-volt located in Antibes (www.b-volt.com) to command a new battery with perfect size, we received and tested it mid-May.

At the beginning, we planned to order batteries in China. We look for the information at http://sz-battery.com/lithium-ion-battery/37v-74v-111v/268.html

We wanted to obtain a total voltage of 50V by connecting 3.3V batteries in series, but we roughly calculated that its size is too large, the counting process is as below:



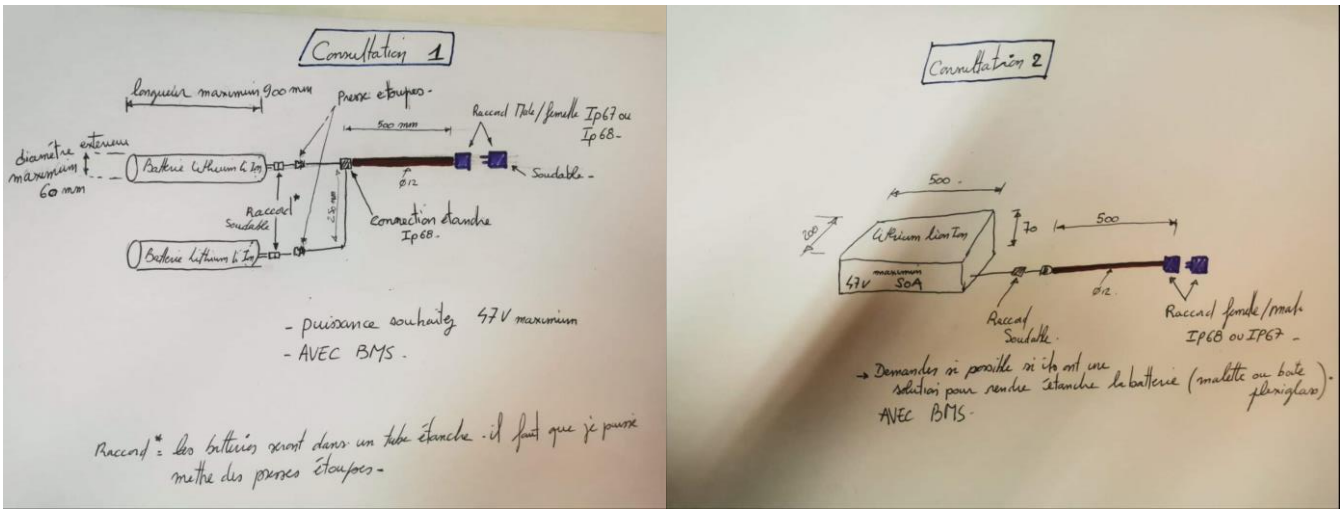
The size of each battery 3.3V is 138.4mm, the connections between batteries are ensured by a screw connection, so we provide a short cable screwed between the batteries, we suppose 1cm. Then in total, per battery we have: 138.4+10=148mm.

And we want the total voltage is 50V, so we need about 50/3.3 # 15 batteries.

Therefor the total size is about 15*148 # 2.22m!!!

It is too large to fit in the plate. so, we have the following two options:

The first option uses cylindrical battery, and the second one uses quadrate battery:

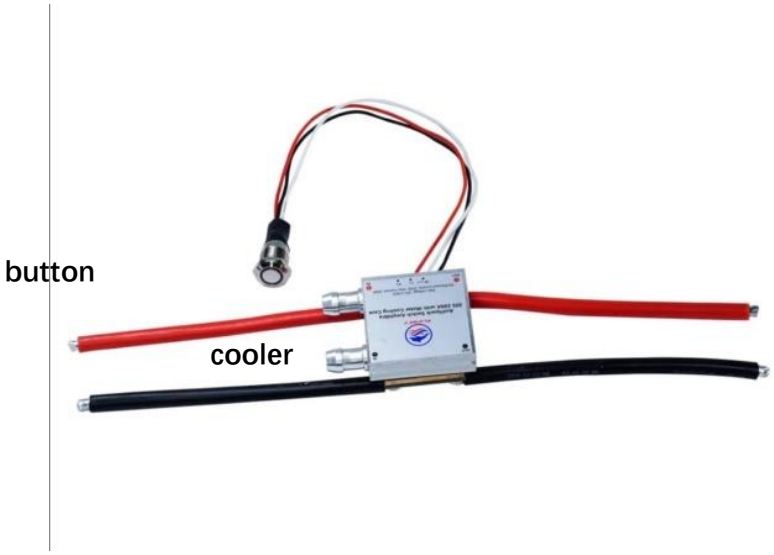


All these 2 solutions with BMS (Battery management system) which can control battery better. And we want it has cable gland and waterproof connection IP68 for waterproof sealing, to better protect the internal circuit

We are looking for battery production companies that can meet our requirements based on these two solutions, but encountered some difficulties, many companies are unable to produce such batteries. Finally, with our tutor we went in a battery constructor named b-volt located in Antibes (www.b-volt.com) to command a new battery with perfect size.

IV.2 Optimize the switch and button

Due to the long-term operation, the temperature of the device will rise, so we want to add more cooling devices. We upgraded the switch to add a cooler based on the original switch, using sea water for cooling. So, we choose Amphibia 30S 200A https://a.aliexpress.com/_BPoeNp



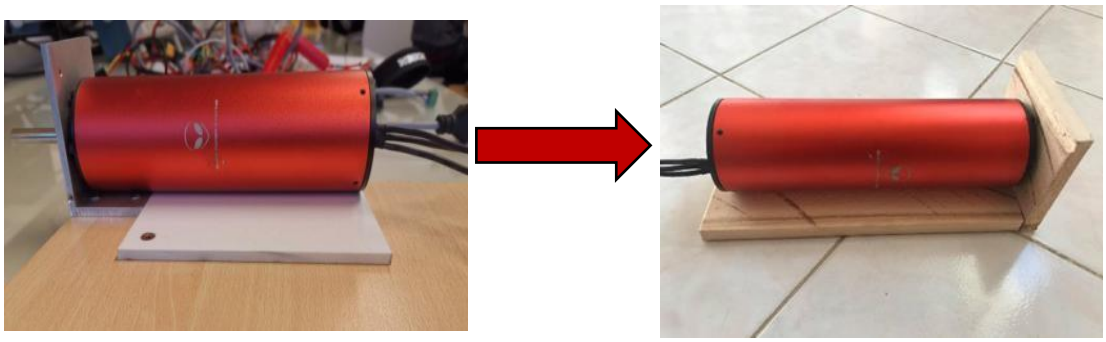
But the button of the new switch is too small, we need to optimize again.

First of all, there are two types of buttons, momentary and latchine ,Momentary is auto-reset switch, that is to say, if we press it, there will be power, but when we release it, there won't be power ; Latchine is a self-locking switch, that is to say, when we press it the first time, it works, we press it the second time, there won't be current. According to demand, we choose latchine.

Finally, we selected LANBOO IP68 12VDC 24DC.

IV.3 Optimize the motor and the VESC

We optimize the motor with more power to bring users a better experience. And the shape of the new motor is slenderer, making it easier to place in the surfboard.



Motor 1 APS 56115 Inrunner brushless	Motor 2 APS 56200 Inrunner brushless
Work on 100KV 5000W	Work on 100KV 8000W
Imax=50A.	Imax=80A.
Vmax=100.8V.	Vmax=100.8V.
https://alienpowersystem.com/shop/brushless-motors/56mm/aps-56115-inrunner-brushless-motor-100kv-5000w/	https://alienpowersystem.com/shop/brushless-motors/56mm/aps-56200-inrunner-brushless-motor-100kv-8000w/

Our project requires higher output power, so we choose the VESC75/300.

The VESC 75/300, most of its features would be handy to have in the VESC 6 format. But it's re-designed the entire PCB, powerful features available in a smaller & cheaper 60V rated package.

https://trampaboards.com/vesc-6-mkiv-in-cnc-t6-silicone-sealed-aluminium-box-with-genuine-xt90-connectors--vedder-electronic-speed-controller-trampa-special-p-27536.html

V. Flashing Firmware.

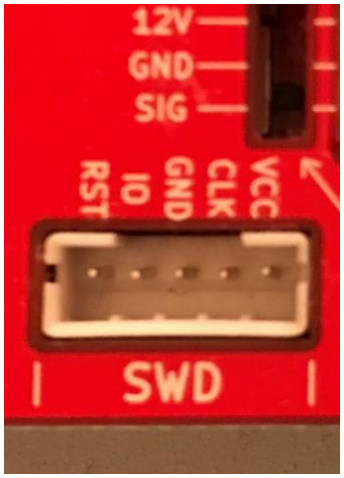
Purpose:

Our tutor Antoine asked us to implement two new functionalities controlled from the remote which are:
-a button for a complete OFF system
-a button to switch ON/OFF the LEDs under the board
To achieve that we need to change the properties of the firmware. First, we must be able to flash the firmware using an USB ST LINK v2 and secondly find the part of the remote implementation and change the sub properties of the code.

V.1 Flashing part

Fortunately, the firmware code is open source, our motor is a BLDC motor thus we use the BLDC firmware. We will use UBUNTU.

We need to connect the USB link to all the pins of the SWD port VESC:



Now that everything is correctly connected, we have some prerequisites to download for our UBUNTU such as the gcc toolchain and add the st link v2 programmer. When everything is setup, we need to open a terminal and do these commands:

1)Firstly we have to build and flash the BOOTLOADER
Clone the GitHub of the bootloader:
-git clone https://github.com/vedderb/bldc-bootloader bootloader
Build it:
-make
Flash it:
-st-flash write build/BLDC_4_Bootloader.bin 0x08000000

2)Now repeat the septs above for the VESC_FIRMWARE
Clone:
-git clone https://github.com/vedderb/bldc.git vesc_firmware
Build it:
-make
Flash it:
-st-flash write BLDC_4_ChibiOS.bin 0x08000000

Now to verify it, we need to open VESC Tool (windows) and check if the firmware is correctly done.

V.2 Code Searching part

Due to a lack of time we didn't find the right part in order to implement the new functionalities but recently our tutor had some new information, he called us and said

that it is not possible to change it from the VESC code and that is a part of the Maytec remote firmware (not open source). He told us too that the Maytec remote version 1 has not that possibility to add/change the button's functions, but the version 2 can. To wait this and have an eFoil almost finished, our tutor asked us to put a switch ON/OFF for beauty LEDs, the switch ON/OFF system is already controlled by the switch between the battery and VESC.

Conclusion:

In this project, we completed most of our goals. We tested whether the circuit was operating normally and realized the communication connection between VESC and Arduino.

In terms of device optimization, we replaced VESC so that it can support a battery voltage of up to 58.8V. We replaced the motor for higher efficiency and so on.

In terms of battery management, we have realized the observation of the battery power through the light strip in order to charge the battery in time.

We significantly refined the all circuit, the next incoming part to complete are the two new functionalities to implement with the Maytec version 2 remote control and weld everything like the Antispark switch ON/OFF between battery and VESC, weld to an Arduino shield the wires for LEDs and UART connexion to have something stable and solid.

Thanks again to Mr. George Antoine and Mr. Langella for their help in this project.

Annexes :

1. Code: LED for beauty:

```
#pragma once
#include<FastLED.h>
#define LED_PIN    6
#define NUM_LEDS  36

CRGB leds[NUM_LEDS];

void setup() {
  //FastLED.addLeds<WS2812,LED_PIN,GRB>(leds,NUM_LEDS);
  FastLED.addLeds<NEOPIXEL, LED_PIN>(leds, NUM_LEDS);

}

void loop() {
for(int i=0;i<NUM_LEDS;i++){
  leds[i]=CRGB(255,0,0);
  FastLED.show();
  delay(200);
}
for(int i=NUM_LEDS-1;i>=0;i--){
  leds[i]=CRGB(0,255,0);
  FastLED.show();
  delay(200);
```

```
for(int i=NUM_LEDS-1;i>=0;i--){
  leds[i]=CRGB(100,30,175);
  FastLED.show();
  delay(200);
}
for(int i=0;i<NUM_LEDS;i++){
  leds[i]=CRGB(0,255,255);
  FastLED.show();
  delay(200);
}
for(int i=NUM_LEDS-1;i>=0;i--){
  leds[i]=CRGB(255,0,255);
  FastLED.show();
  delay(200);
}
for(int i=0;i<NUM_LEDS;i++){
  leds[i]=CRGB(255,255,0);
  FastLED.show();
  delay(200);
}
for(int i=NUM_LEDS-1;i>=0;i--){
  leds[i]=CRGB(0,0,255);
  FastLED.show();
```

2. Code: controlling the LED for battery capacity (simulation with Tinkercad).

```
#include <LiquidCrystal.h>

const int analogPin = A4;
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, A0, A1, A2, A3);
float analogValue;
float input_voltage;

int ledPins[] = { 2, 3, 4, 5, 6 }; // The LEDs attach to this array of pin nu
int pinCount = 10;

void setup() {
  Serial.begin(9600);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Voltage Level");
  pinMode(A4,INPUT);
}

void LED_function(int stage)
{
    for (int j=2; j<=6; j++)
    {
    digitalWrite(j,LOW);
    }
    for (int i=1, l=2; i<=stage; i++,l++)
    {
    digitalWrite(l,HIGH);    //delay(30);
    }
}


void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);

  //  Conversion formula for voltage
  analogValue = analogRead (A4);
  Serial.println(analogValue);
  delay (1000);
  input_voltage = (analogValue * 5.0) / 1024.0;
  lcd.setCursor(0, 1);
  lcd.print("Voltage= ");
  lcd.print(input_voltage*11.76);
  Serial.println(input_voltage);
 // Serial.print(input_voltage);
  delay(100);

//we want to mesure 46,2v-58.8v, but arduino can only mesure 0-5v and displa
//so we transform 46.2-58.8v to 3.93-5v, we use a converter to give the tensi
//so we connect the battery (which is converted to 5 v ) to A4.
  if (input_voltage < 3.93 )
```

```
    {
    digitalWrite(2, HIGH);
    delay (30);
    digitalWrite(2, LOW);
    delay (30);
    }
  else if (input_voltage <4.144 && input_voltage >=3.93 )
    {
    LED_function(1);
    }
    //
  else if (input_voltage < 4.358 && input_voltage >= 4.144)
    {
    LED_function(2);
    }
  else if (input_voltage < 4.572 && input_voltage >= 4.358)
    {
    LED_function(3);
    }
  else if (input_voltage < 4.786 && input_voltage >= 4.572)
    {
    LED_function(4);
    }
  else if (input_voltage < 5 && input_voltage >= 4.786)
    {
    LED_function(5);
```

3. Code: controlling the LED for battery capacity (test with LEDs band)

batterie

```
#include <LiquidCrystal.h>
#include<FastLED.h>
#define LED_PIN    6
#define NUM_LEDS  5

CRGB leds[NUM_LEDS];

const int rs = 12, en = 13, d0 = A0, d1 = A1, d2 = A2, d3 = A3;
const int analogPin = A4;
LiquidCrystal lcd(rs, en, d0, d1, d2, d3);
float analogValue;
float input_voltage;

int pinCount = 10;           // the number of pins

void setup()
{
   Serial.begin(9600);      //  opens serial port, sets data rate to 9600 bps
   lcd.begin(16, 2);        //// set up the LCD's number of columns and rows:
   pinMode(A0,OUTPUT);
   pinMode(A1,OUTPUT);
   pinMode(A2,OUTPUT);
   pinMode(A3,OUTPUT);
   pinMode(A4,INPUT);
   lcd.print("Voltage Level");
   FastLED.addLeds<NEOPIXEL, LED_PIN>(leds, NUM_LEDS);
}

void LED_function(int stage)
{
  for(int i=0;i<stage;i++){
  leds[i]=CRGB(0,255,0);
  FastLED.show();
 }
 if(stage!=NUM_LEDS-1){
 for(int i=stage;i<NUM_LEDS;i++){
  leds[i]=CRGB(255,0,0);
  FastLED.show();
```

```
  }
 }
}


void loop()
{
  analogValue = analogRead (A4);
  Serial.println(analogValue);
  delay (1000);
  input_voltage = (analogValue * 5.0) / 1024.0;
  lcd.setCursor(0, 1);
  lcd.print("Voltage= ");
  lcd.print(input_voltage*11.76);
  Serial.println(input_voltage);
  delay(20);

  if (input_voltage < 3.93 )
    {
    for(int i=0;i<NUM_LEDS;i++){
    leds[i]=CRGB(255,0,0);
    FastLED.show();
    delay(200);
    FastLED.clear();
    }
 }
  //else if (input_voltage <4.144 && input_voltage >=3.93 )
  else if (input_voltage <4.144 && input_voltage >=3.93 )
    {
    LED_function(1);
    }
    //
  else if (input_voltage < 4.358 && input_voltage >= 4.144)
    {
    LED_function(2);
    }
  else if (input_voltage < 4.572 && input_voltage >= 4.358)
    {
    LED_function(3);
    }
  else if (input_voltage < 4.786 && input_voltage >= 4.572)
    {
    LED_function(4);
    }
  else if (input_voltage < 5 && input_voltage >= 4.786)
    {
    LED_function(5);
    }
}
```

4. Config Motor Tutorial for Antoine

TUTO DE CONFIGURATION DU VESC AINSI QUE SA TELECOMMANDE & AUTRES depuis VESC TOOL
Dans le menu du tout en haut cliquer sur Wizards -> Setup others Motor
NEXT et Motor Type : BLDC
NEXT
Motor Current Max : 80A (depend du Motor)
Motor Current Max Brake : -80A
Battery Current Max : 25A (cela peut changer regarder spec battery)
Battery Current Max Regen : -12A (variable aussi)
NEXT
Mettre nombre de Cells Battery : 14 et APPLY
NEXT
SENSORLESS (changer si on a mis un SENSOR)
NEXT
RUN DETECTION (bouton play, attention!!!! bien tenir le motor si il n'est pas déjà fixe)
APPLY
NEXT
FINISH

Maintenant nous allons configurer la télécommande
SETUP INPUT
PPM INPUT
NEXT
Alors ici il faudra :
Verifier que le min soit bien la même valeur lorsque la gachette est à vide
Appuyer à sur la gachette au maximum, et entrée en valeur max la valeur indiqué (par exemple ici 2,00000 ms)
Le centre devrait se faire automatiquement
APPLY
NEXT
Control TYPE : PID Speed Control No Reverse
NEXT
FINISH
Autre : Pour la communication UART avec l'arduino, il faut aller dans :
UART (Menu sur la gauche)
Mettre en Baudrate 19200
Sur la menu de droite, cliquer sur la flèche vers le bas avec un A (write app configuration) pour enregistrer la valeur dans le VESC

5. BOM (Bill of material)

1	name	quantity	price	cost
2	arduino mega	3	7.68	23.04
3	arduino mega proto screw	3	11.46	34.38
4	arduino relay 4CH	3	3.3	9.9
5	touch screen	3	12.95	38.85
6	connector VESC to arduino	1		
7	DC DC 60/12V	1	8.98	8.98
8	DC DC 60/5V recup	1		
9	temperature sensor for VESC	1		
10	motor	1		
11	button	1		
12	battery	1		
13	VESC	1		
14	switch	1		

There are all pointers:

Arduino mega:
https://www.ebay.com/itm/Mega-2560-Atmel-ATmega-2560-R3-Microcontroller-Board-Compatible-CH340G-Arduino/283606404588?hash=item4208424dec:g:8WAAAOSw~gRVk4m6

Arduino mega proto screw :
https://www.ebay.com/itm/MEGA-2560-R31-Prototype-Screw-Terminal-Block-Shield-Board-Kit-For-Arduino-Tool/392476753071?epid=2297423288&_trkparms=ispr%3D1&hash=item5b616ffcaf:g:kB4AAOSwnihdoZKA&enc=AQAEAAACQBPxNw%2BVj6nta7CKEs3N0qWkpAvWCkr3ZCThREv9%2Fkv0C4RudCfCEZ%2BgI01iQBuZado3GSH4EdLEnghHXB1Z1hmQ5VJGdgSMPbSAcqyJwsNJ6VBaqeHy%2BGD0FQ7iQdesdr%2B66K2dwvsZYYNLd%2F%2BvjZO9eiXhlFujKXs1i3RwMX8vAi3PRHCrPz03mVBz3XNyr6YDAqxtD0Jq5Ykm1uH4LrkqBRgJIJJ3TueP%2Bv%2F9I7vmI8knQ02Bl%2FI86zTINJ6YM68T%2Fmg6zNluFx8cnKVE4ZBdIsTq2jo7wNJJ3SbEE2wt0G%2B1cGbkTZjCfw0IQyuLzs73yMHUmRU63c%2B%2FIwwBcDN5fNGUfK62vIX3bmrM5I6fow1nuX5o3YXd0%2FDoGyR5ClpaZHbCA352cxz%2B5TpMczVv2BGW%2B1z4vJ4qMeMm%2BAzeQsOHME%2FbdejOWW%2B5%2BBgAIRiCyTFPE2zolByjA7%2BrKLUMD58Z7JwUQJRzHK1KJr84wNGRydea1esq%2FSxR9VIsBonxQ58VRmwSCFkIkpWdTZ01aSnLAmniLUEOJI7g6SA10%2BfwRkjJEsy68v6JyrNLLf2M6WK37vuUGBG%2B8UcuZYz1IT4GJs4DOC9Y%2BZXgHec7mo8SeJ5WwxjVS8CagFYPAKLT0pza

ZbLnuN%2FnIUapLNGZlvQu2LdK%2BaCoJR9jqbOsPLI5HPnYWMTkZZTRe4fvu7PJjsuSZ5y0AJKHEMga8KR3SgtshSYQV5IBtYv1PZmSooaxsew062SknRuoA2ihfA%3D%3D&checksum=39247675307192fc91c324104340890f3c4b56829b26&enc=AQAEAAACQBPxNw%2BVj6nta7CKEs3N0qWkpAvWCkr3ZCThREv9%2Fkv0C4RudCfCEZ%2BgI01iQBuZado3GSH4EdLEnghHXB1Z1hmQ5VJGdgSMPbSAcqyJwsNJ6VBaqeHy%2BGD0FQ7iQdesdr%2B66K2dwvsZYYNLd%2F%2BvjZO9eiXhIFujKXs1i3RwMX8vAi3PRHCrPz03mVBz3XNyr6YDAqxtD0Jq5Ykm1uH4LrkqBRgJIJJ3TueP%2Bv%2F9I7vmI8knQ02BI%2FI86zTINJ6YM68T%2Fmg6zNIuFx8cnKVE4ZBdlsTq2jo7wNJJ3SbEE2wt0G%2B1cGbkTZjCfw0IQyuLzs73yMHUmRU63c%2B%2FIwwBcDN5fNGUfK62vIX3bmrM5I6fow1nuX5o3YXd0%2FDoGyR5CIpaZHbCA352cxz%2B5TpMczVv2BGW%2B1z4vJ4qMeMm%2BAzeQsOHME%2FbdejOWW%2B5%2BBgAIRiCyTFPE2zoIByjA7%2BrKLUMD58Z7JwUQJRzHK1KJr84wNGRydea1esq%2FSxR9VIsBonxQ58VRmwSCFkIkpWdTZ01aSnLAmniLUEOJI7g6SA10%2BfwRkjJEsy68v6JyrNLLf2M6WK37vuUGBG%2B8UcuZYz1IT4GJs4DOC9Y%2BZXgHec7mo8SeJ5WwxjVS8CagFYPAKLT0pzaZbLnuN%2FnIUapLNGZlvQu2LdK%2BaCoJR9jqbOsPLI5HPnYWMTkZZTRe4fvu7PJjsuSZ5y0AJKHEMga8KR3SgtshSYQV5IBtYv1PZmSooaxsew062SknRuoA2ihfA%3D%3D&checksum=39247675307192fc91c324104340890f3c4b56829b26

DC DC 60/12V:
https://www.ebay.com/itm/Waterproof-DC-DC-Buck-Step-Down-Volt-Converter-20V-60V-24V-36V-48V-to-12V-3A-Car/123499957610?hash=item1cc12bcd6a:m:m2Xak4DKz7SW6ESaqKr78Dw

Motor:
https://alienpowersystem.com/shop/brushless-motors/56mm/aps-56200-inrunner-brushless-motor-100kv-8000w/

Switch:
https://a.aliexpress.com/_BPoeNp

VESC:
https://trampaboards.com/vesc-6-mkiv-in-cnc-t6-silicone-sealed-aluminium-box-with-genuine-xt90-connectors--vedder-electronic-speed-controller-trampa-special-p-27536.html

Bibliography

1. The document: All projects 2019-2020.pdf
2. http://fastled.io/docs/3.1
3. https://www.arduino.cc/en/Reference/LiquidCrystal
4. http://sz-battery.com/lithium-ion-battery/37v-74v-111v/268.html
5. www.b-volt.com
6. http://vedder.se/2015/10/communicating-with-the-vesc-using-uart/#more-935