



SANDTRONIC

CARPETA DE CAMPO.



Sandtronic_gr



eoffroad2023@gmail.com

Semana del 19 al 24 de abril: Evaluamos entre las opciones de proyecto realizar este arenero eléctrico ya que contábamos con la estructura construida, la cual nos hubiera llevado mucho tiempo si la tendríamos que haber construido en el colegio, ya que ese tiempo debería ser empleado para la gran cantidad de electronica que se precisa para este proyecto.

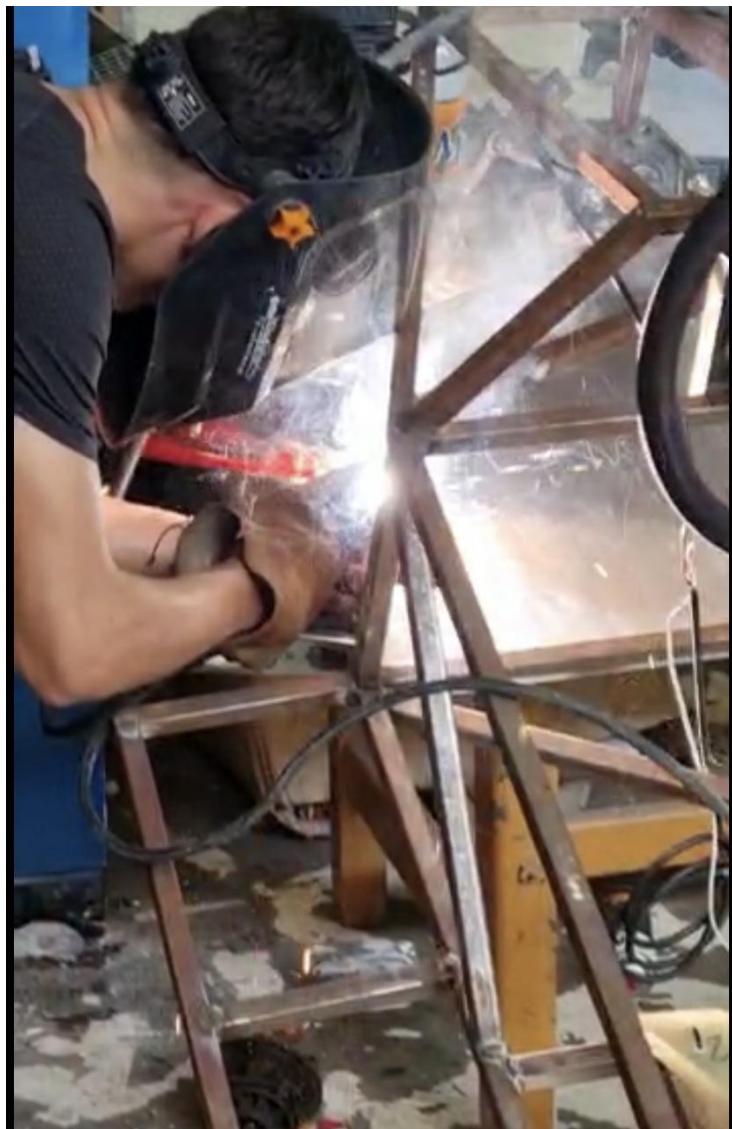


Imagen de la estructura a principios de 2023

Semana del 24 al 29 de mayo: Una vez de acuerdo con el grupo presentamos anteproyecto del arenero eléctrico el cual nombraríamos como Sandtronic

Semana del 14 al 19 de junio: Conseguimos un transporte para traer la estructura a la escuela, junto con los materiales y herramientas necesarias para trabajar en ello.

En aquel momento se arreglaron ciertos problemas estructurales, se diseñaron refuerzos para una mayor confiabilidad ya que luego de diseñar en 3D las baterías generaban desgastes en las uniones de soldadura.



Todas las soldaduras de la estructura fueron realizadas con Soldadora MIG.

- A la par que se realizaban los refuerzos de estructura el equipo estaba poniendo en condiciones el impulsor del arenero, el cual contaba con conexiones sulfatadas, carcasa en mal estado, rodamientos trabados, bobinas en mal estado por estar guardado en un lugar con mucha humedad.
- Se modelo en 3D una base para replicar las bobinas del motor las cuales fueron realizadas con alambre barnizado de cobre de 1mm



- También se desarmo el motor el cual tenía sus bulones en la mayoría clavados y dio mucho trabajo



- Restauramos la carcasa del motor torneandola.



Semana del 28 de junio al 3 de Julio: Al comienzo de esta semana se logro bobinar la bobina de campo del motor.



Compramos los MOSFET irfp4227 (20 unidades). Los cuales serian usados para realizar la primer placa de potencia.



Fabricamos la primer placa de potencia PCB



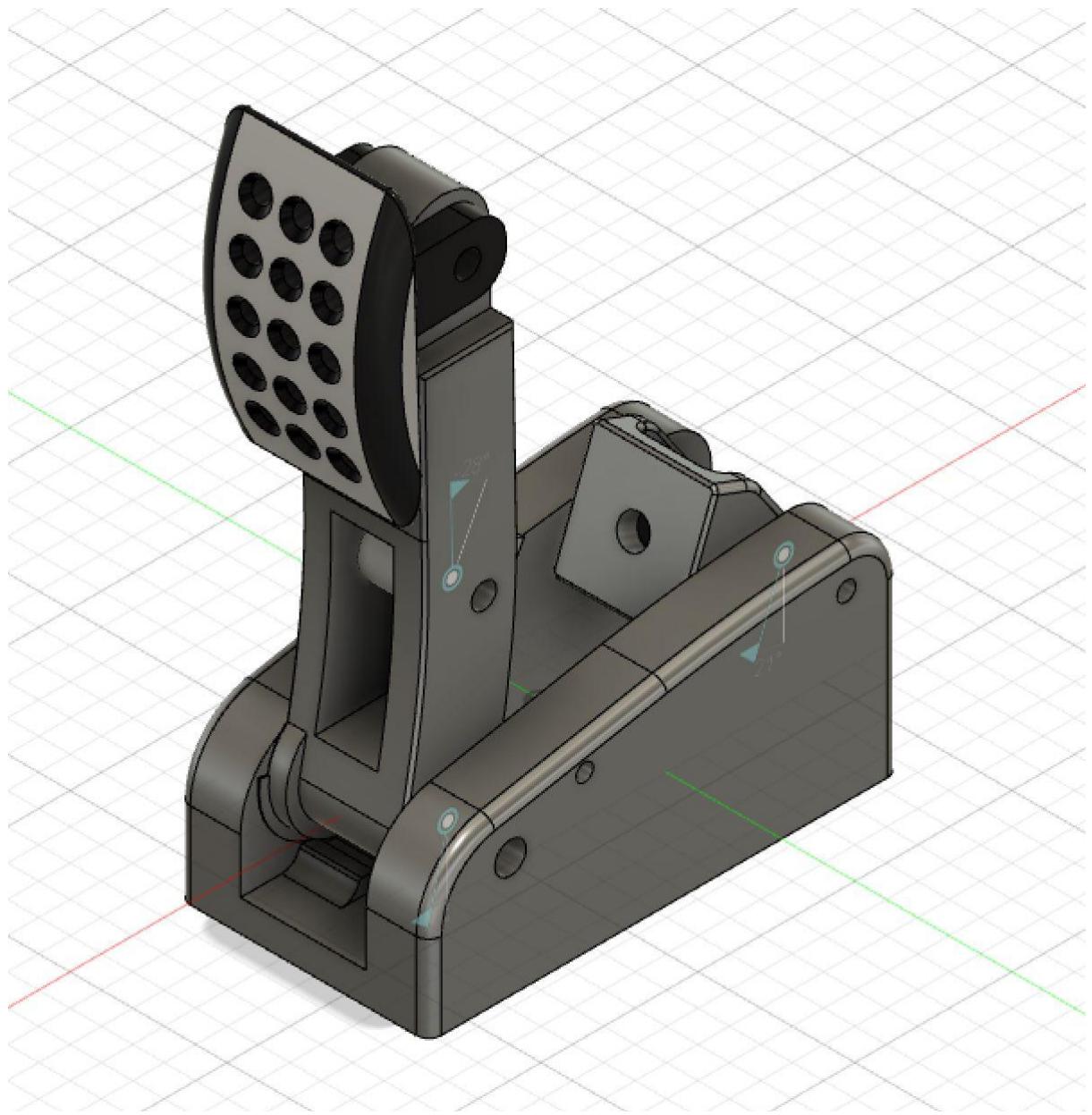
Aislamos y barnizamos las bobinas del motor, las cuales fueron rebobinadas por nosotros mismos.



Estañamos las pistas de la PCB para poder disminuir la resistencia y lograr la máxima conducción de la misma

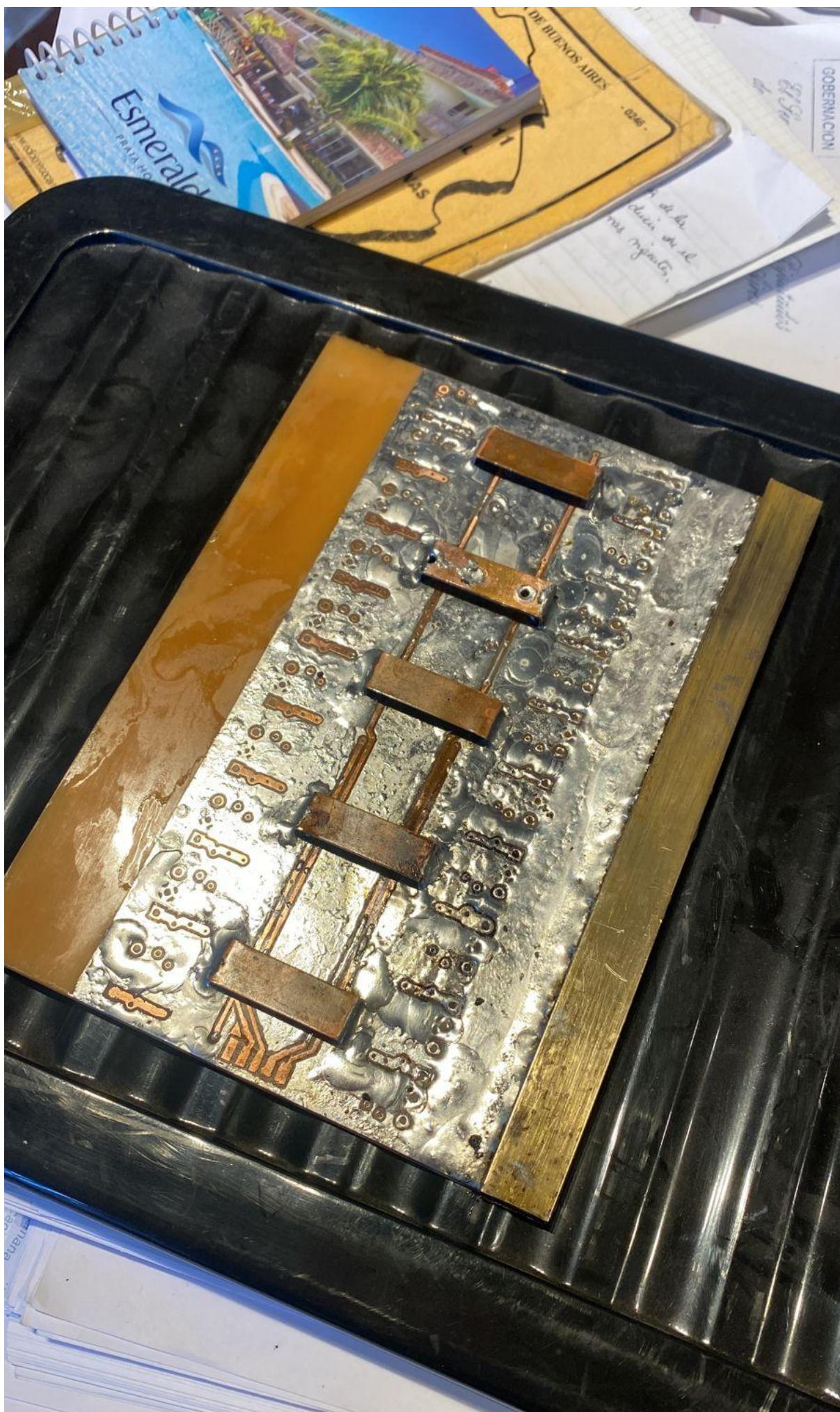


Semana del 5 al 10 de Julio: Diseñamos en 3D el pedal de acelerador para imprimirlo en la siguiente semana.



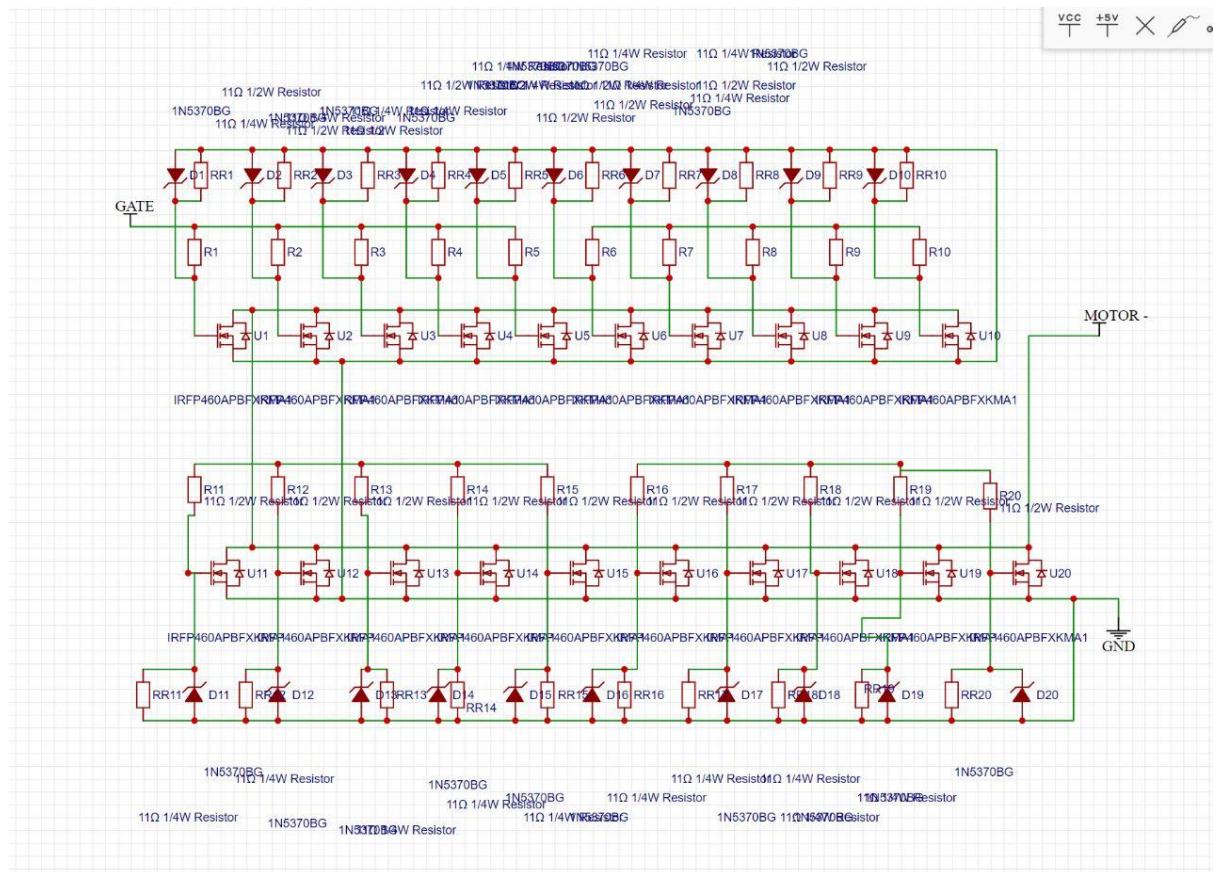
- Seguimos avanzando con la fabricación y estañando con la 1er placa de potencia y en el esquema de la placa de control



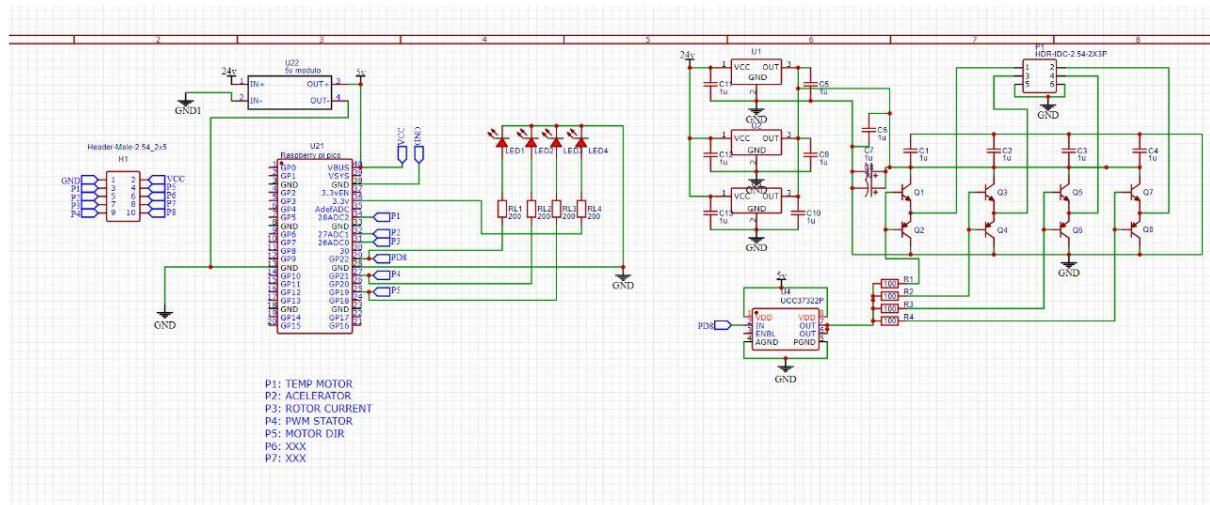


También realizamos los esquemas tanto de la placa de potencia como la de control

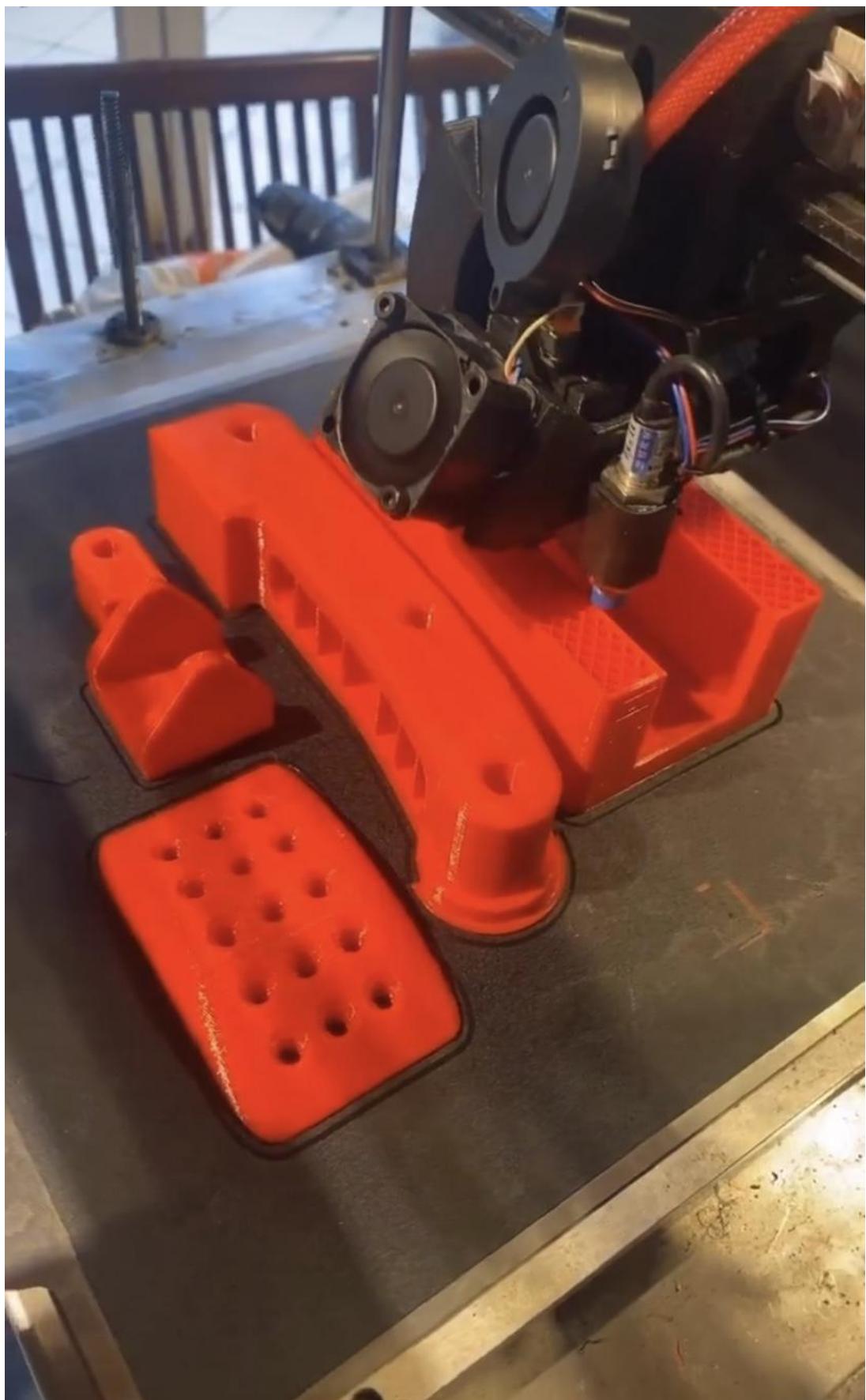
ESQUEMA PLACA DE POTENCIA, el programa que fue realizado el esquema de las placas fue en EASY EDA



PLACA DE CONTROL.



Semana del 12 al 17 de Julio:



Imprimimos el pedal del acelerador
Definimos cual seria el microcontrolador del proyecto.

RASPBERRY PI PICO, la cual integramos en la placa de control para realizar las primeras pruebas.



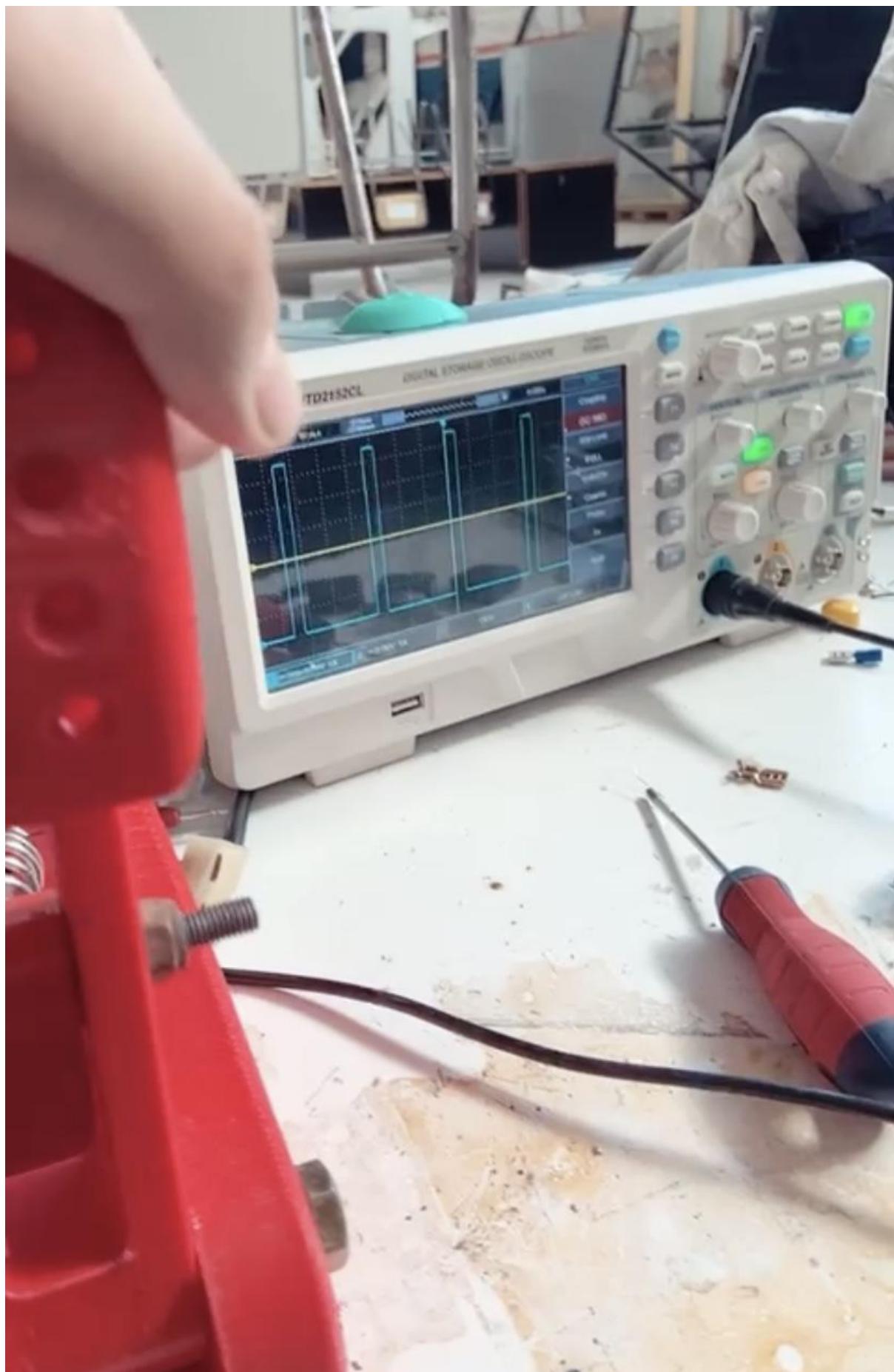
Programamos el primer código capaz de partir de un pwm que varíe el ancho de pulso, y también podamos registrar la velocidad dependiendo de cuantas veces pase un imán por un sensor hold y en cuanto tiempo se calcula y es mostrado analogicamente a través de un servo motor, el cual fue adaptado a un indicador de presión de aceite

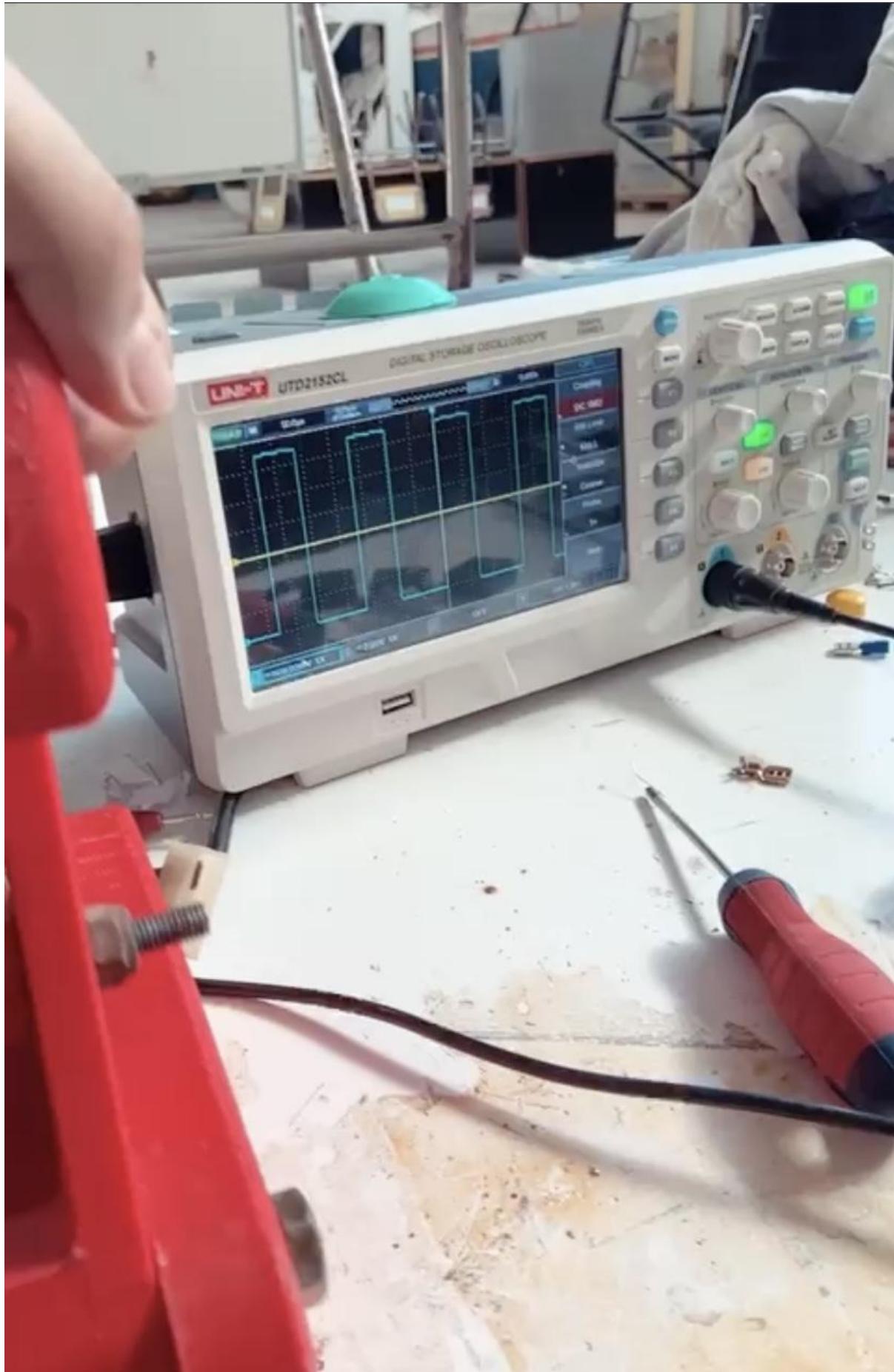
```

pps.py * 
1 import time
2 import machine
3 pwm = 0
4 pulse_count = 0
5 speed = 0.0
6 diametro= 0.55 //esta en metros diametro de rueda
7 circunferencia= 0.55 * 3.14 //circunferencia es igual al diametro por pi
8 relacion = 8.9 //relacion entre el moto r y rueda. cuantas veces gira el motor para que la rueda gire una vuelta
9 cant_imanes = 4
10
11 servo_duty_0 = 8000 //menor cantidad de pulsos en la ventana de tiempo equivalente a 5km/h
12 servo_duty_180 = 1333 //maxima cantidad de pulsos en la ventana de tiempo equivalente a vel max 80km/h
13 velocidad_max = 80
14 // Ancho de pulsoz
15 duty = servo_duty_0
16 // pendiente de la recta pulsos(v)
17 m = (servo_duty_180 - servo_duty_0) / velocidad_max
18 def calculate_speed():
19     global pulse_count, speed, duty
20     speed = pulse_count * 10 / (relacion * cant_imanes) * circunferencia * 3.6 // velocidad en km/h
21     duty = int(m * speed + servo_duty_0)
22     pulse_count = 0
23     print("Velocidad: {:.1f} km/h".format(speed))
24
25 def on_pulse(pin):
26     global pulse_count
27     pulse_count += 1
28
29 def velocimetro_init():
30     // GPIO para el servo de la aguja del velocimetro
31     global pwm
32     servo_pin = 20
33     pwm = machine.PWM(machine.Pin(servo_pin))
34     pwm.freq(50)
35     // GPIO de entrada para el sensor de efecto hall
36     sensor_pin = machine.Pin(16, machine.Pin.IN)
37     sensor_pin.irq(trigger=machine.Pin.IRQ_RISING, handler=on_pulse)
38     // Timer para ventana de tiempo velocimetro
39     t = machine.Timer()
40     t.init(period=100, mode=machine.Timer.PERIODIC, callback=calculate_speed)
41

```

Una vez que se logró que el código esté apto para variar el pwm, se realizaron pruebas ya con el sensor hold colocado en el pedal del acelerador, el cual dependiendo de su ángulo, es decir que tan pisado este varía el ancho de pulso.



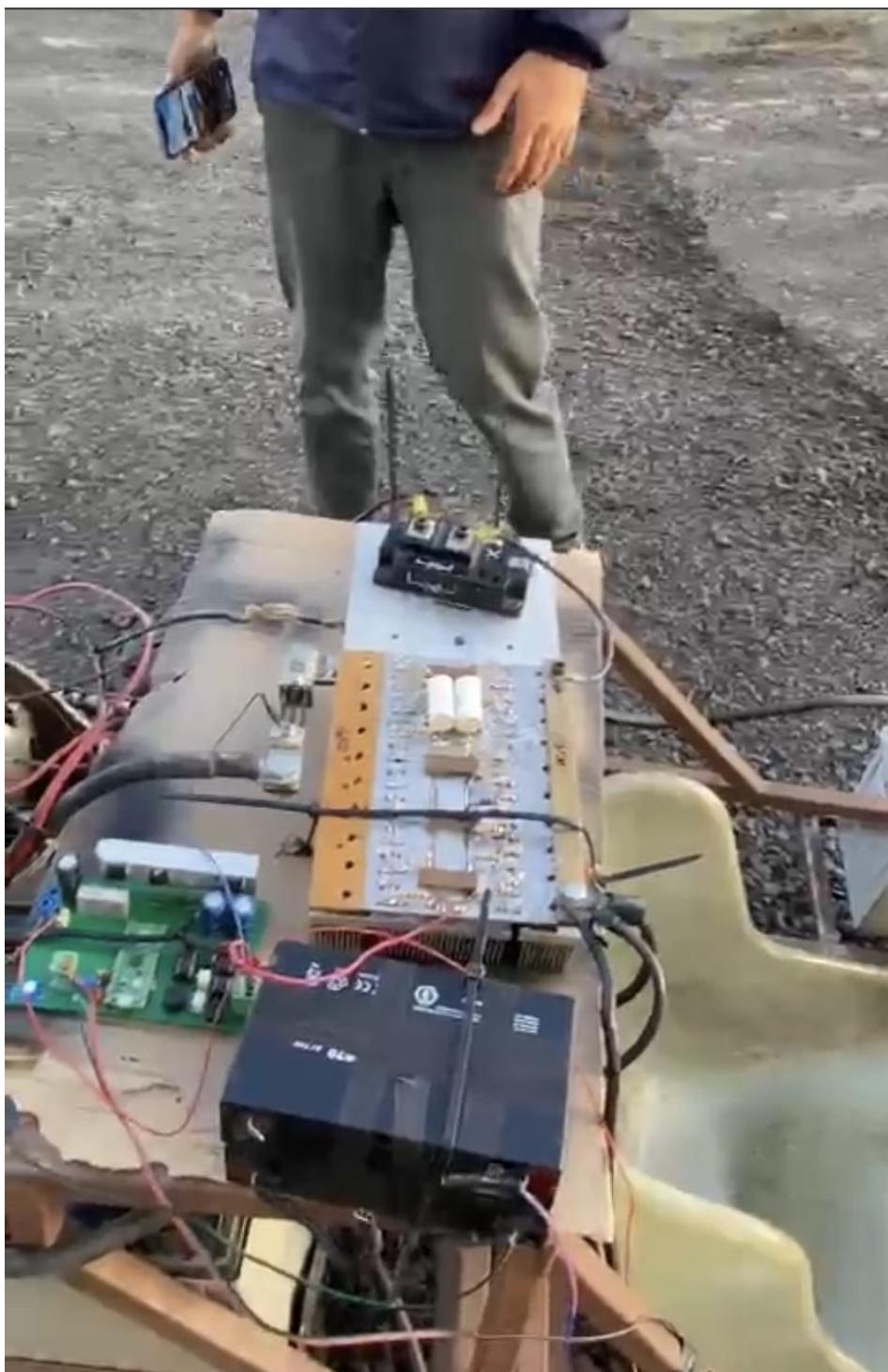


Semana del 19 al 24 de Julio: Realizamos varias pruebas con 1, 2 y 3 baterías de 12V 100amp.

Las pruebas fueron realizadas a lo largo de la semana ya que requerían tiempo y mucha observación en cada punto de conexión, comprobando voltaje amperaje, y consumo, que no ocurra ningún tipo de cortocircuito.



- Luego de esto se nos quemó la placa probando el arenero con carga.



-Momento en el que se quemó la primer placa de potencia, la causa de esto creemos que fueron picos de voltaje inesperados y oscilaciones que no deberían estar.

Semana del 9 al 14 de agosto: Luego del receso de invierno, volvimos y por lo que comenzamos fue por pintar el chasis del karting ya que nos dimos cuenta que después de un tiempo desarmado se oxida y podría llegar a perder fiabilidad la estructura debido a la humedad.



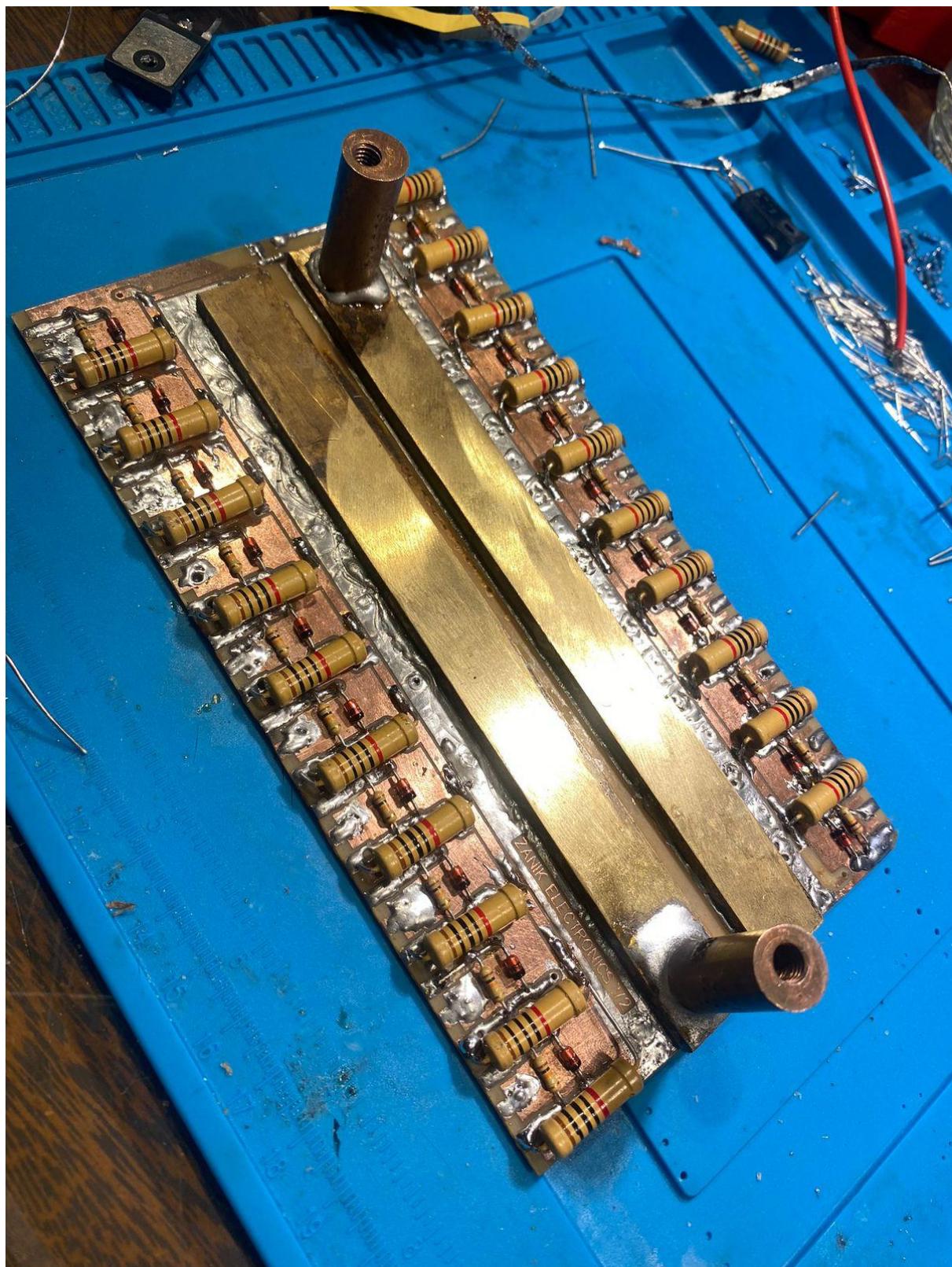
- También luego de la capa de Primer, colocamos abrazaderas para sujetar las baterías laterales, ya que nos dimos cuenta el dia que se quemó la placa de potencia, las baterías no estaban fijas a pesar de que su peso sea de 35kg cada una.

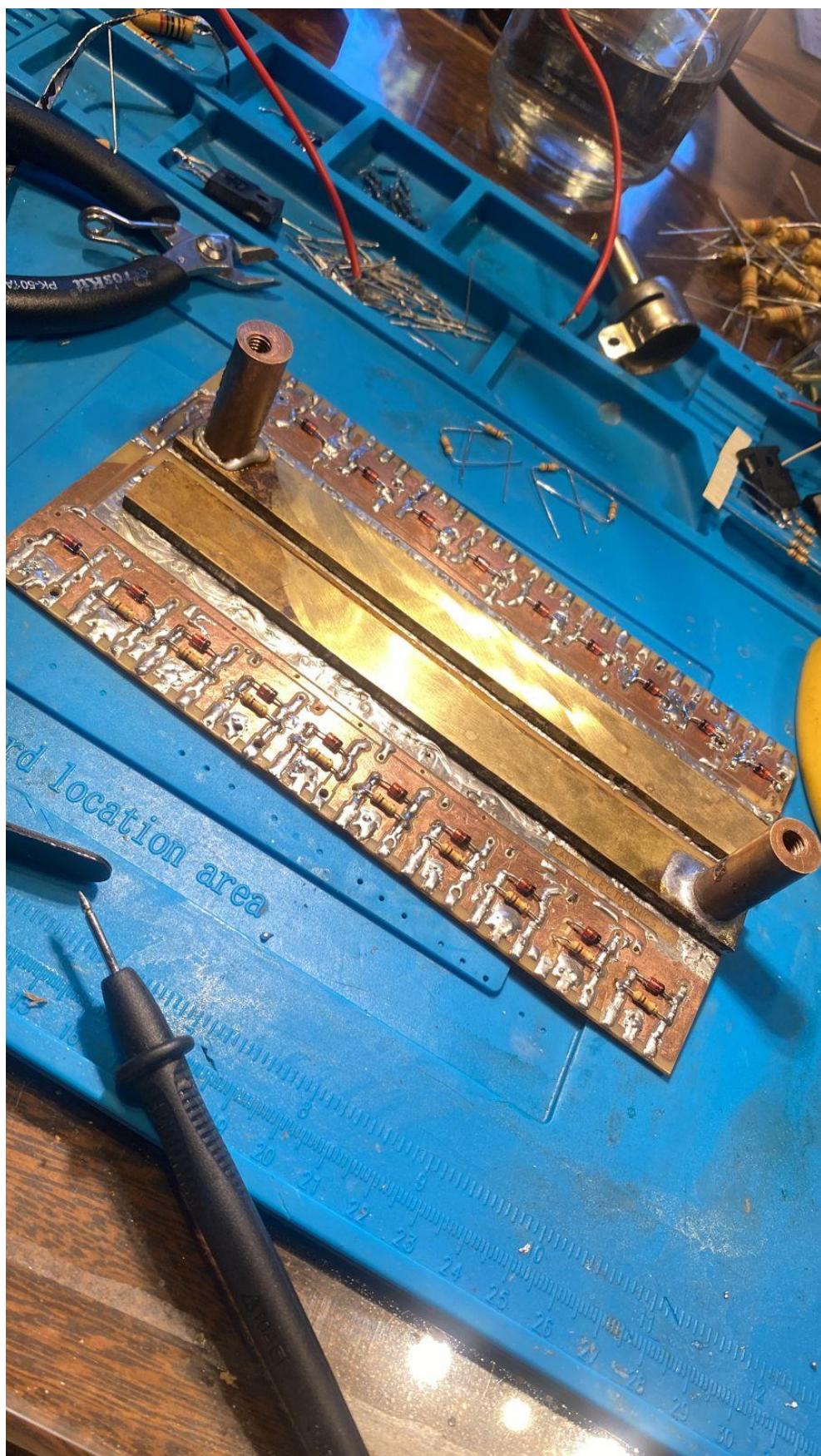


Semana del 16 al 21 de agosto: Esta semana lo que hicimos fue investigar que mosfet calidad/precio nos convenga para utilizar ya que necesitamos 20, por ende lo que hicimos fue realizar los pedidos a cooperadora, junto con el sistema de frenos para el tren trasero del sandtronic.

PRÁCTICAS PROFESIONALIZANTES - SOLICITUD DE MATERIALES			
NOMBRE DE PROYECTO: CA NOMBRE DEL PROYECTO: SANFERMÁN G.R. DIRECCIÓN: AVENIDA AVIC DATOS DEL PROVEEDOR: DIRECCIÓN: NOTO FRANCO TELÉFONO: 01 3551 4260 E-MAIL: DATOS DEL PROYECTO: DIRECCIÓN: NOTO FRANCO TELÉFONO: 01 3551 4260 E-MAIL: DATOS PERSONALES: NOMBRE: CARLOS C. BIANCO APELLIDO: RODRIGUEZ DIRECCIÓN: NOTO FRANCO TELÉFONO: 01 3551 4260 E-MAIL: DETALLE DE LA SOLICITUD: DESCRIPCIÓN: MATERIALES NECESSARIOS CANTIDAD: 1 PRECIO UNITARIO: \$34,999 PRECIO PARCIAL: \$34,999 PRECIO TOTAL: <u>\$34,999</u> FIRMA DEL RESPONSABLE: <u>Carlos C. Bianco</u>		NOMBRE DE PROYECTO: CA NOMBRE DEL PROYECTO: SANFERMÁN G.R. DIRECCIÓN: AVENIDA AVIC DATOS DEL PROVEEDOR: DIRECCIÓN: NOTO FRANCO TELÉFONO: 01 3551 4260 E-MAIL: DATOS DEL PROYECTO: DIRECCIÓN: NOTO FRANCO TELÉFONO: 01 3551 4260 E-MAIL: DATOS PERSONALES: NOMBRE: CARLOS C. BIANCO APELLIDO: RODRIGUEZ DIRECCIÓN: NOTO FRANCO TELÉFONO: 01 3551 4260 E-MAIL: DETALLE DE LA SOLICITUD: DESCRIPCIÓN: MATERIALES NECESSARIOS CANTIDAD: 1 PRECIO UNITARIO: \$11,999 PRECIO PARCIAL: \$11,999 PRECIO TOTAL: <u>\$11,999</u> FIRMA DEL RESPONSABLE: <u>Carlos C. Bianco</u>	

Semana del 23 al 28 de agosto: Una vez llegados los componentes necesarios para realizar la segunda placa de potencia, a raíz de esto decidimos también rediseñar ambas placas (control y potencia), para que sean más completas y tengamos más opciones para ir probando con el funcionamiento de sandtronic.







Semana del 6 al 11 de septiembre: Esta semana teníamos como objetivo diseñar, dos masas, una para cada rueda del eje trasero ya que las anteriores con las fuerzas que se ejercen al andar, se doblaban y barrían las roscas de los bulones.

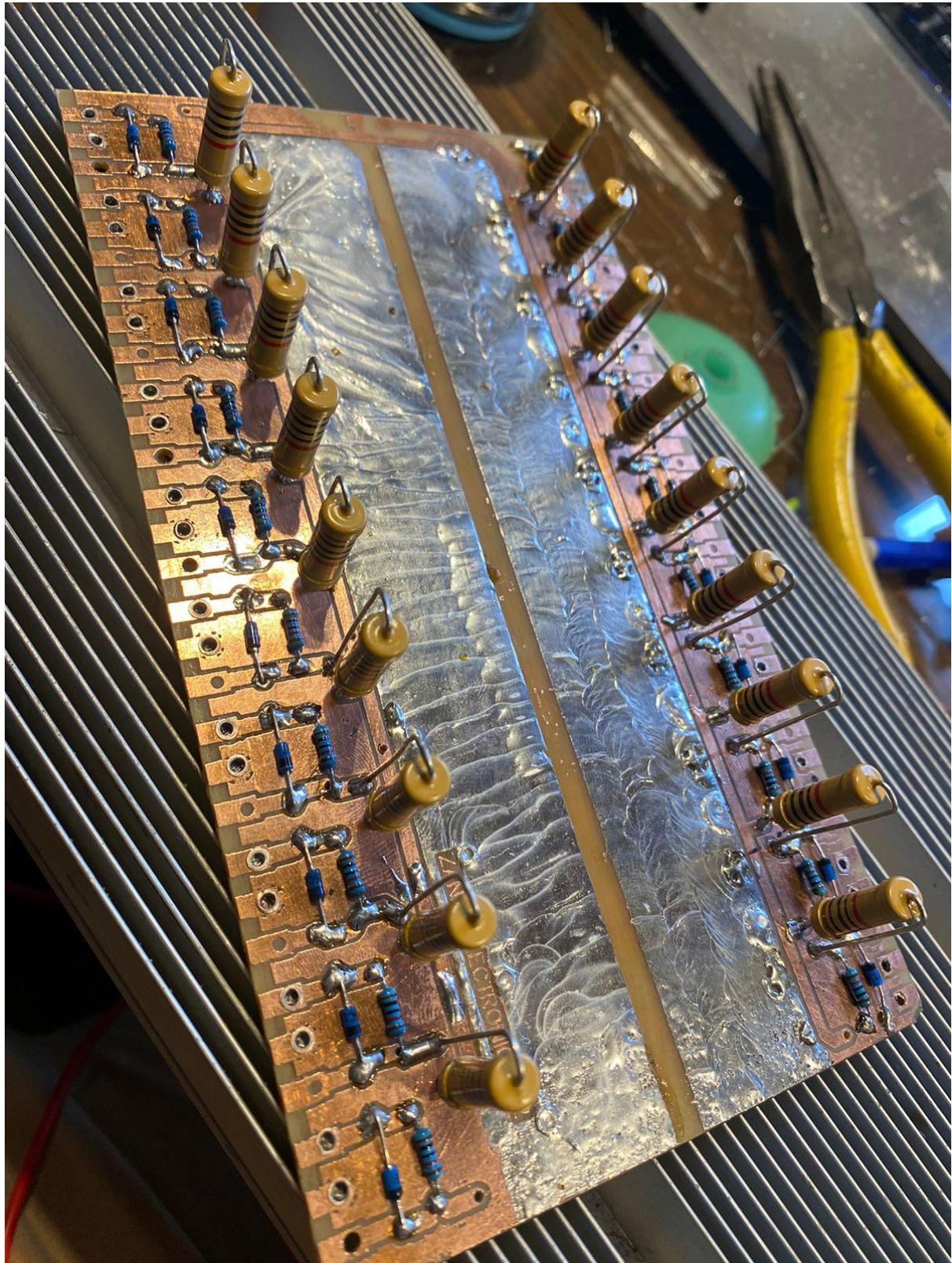


- Fueron soldadas con TIG y argón para un perfecto cordón de soldadura, y una muy buena confiabilidad y poder exigir al máximo el eje trasero ya que cuenta con mucho torque.



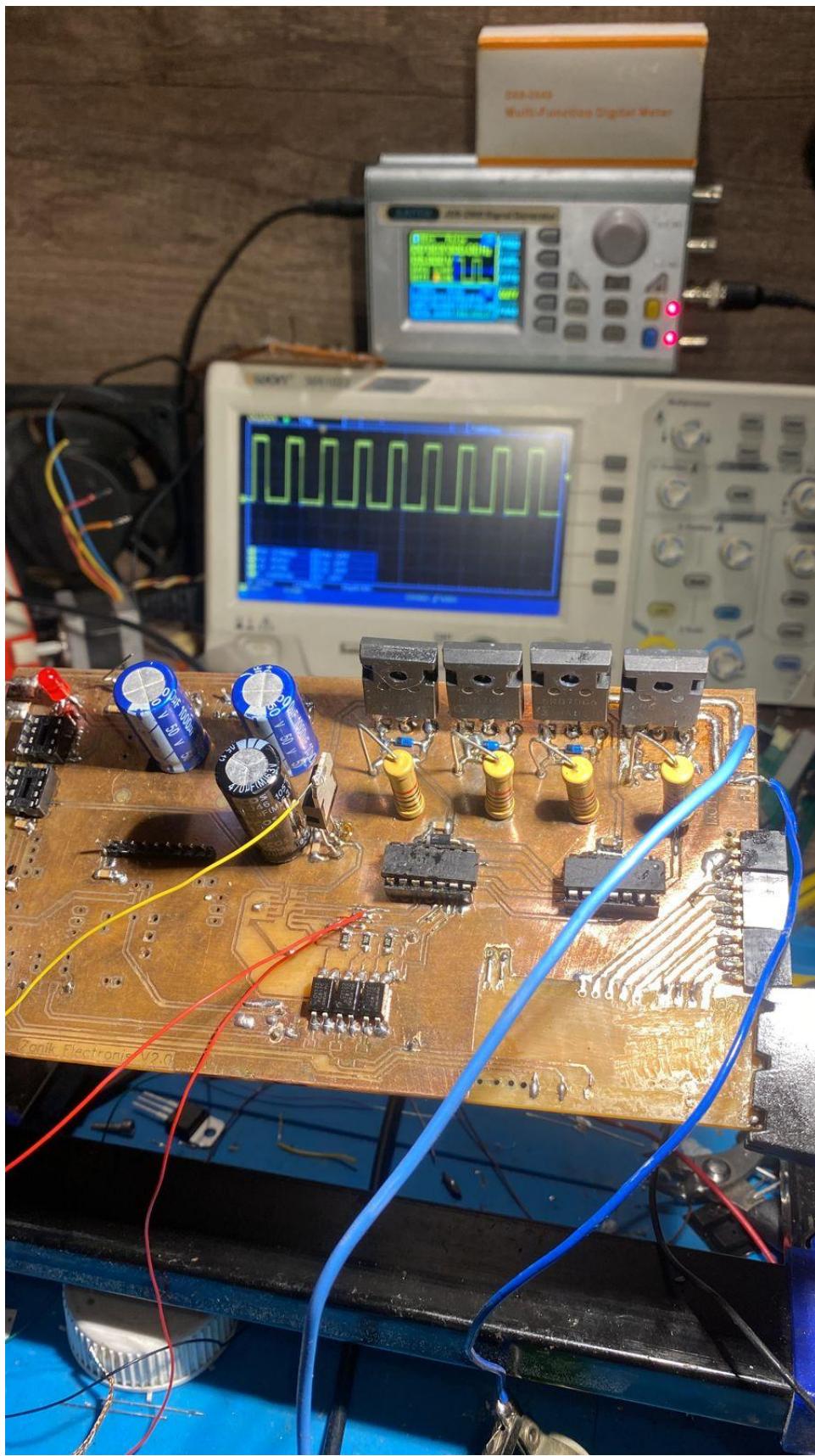
Semana del 13 al 18 de septiembre: Comenzamos a rehacer las placas tanto de control como de potencia, pero primero comenzamos por la de potencia.

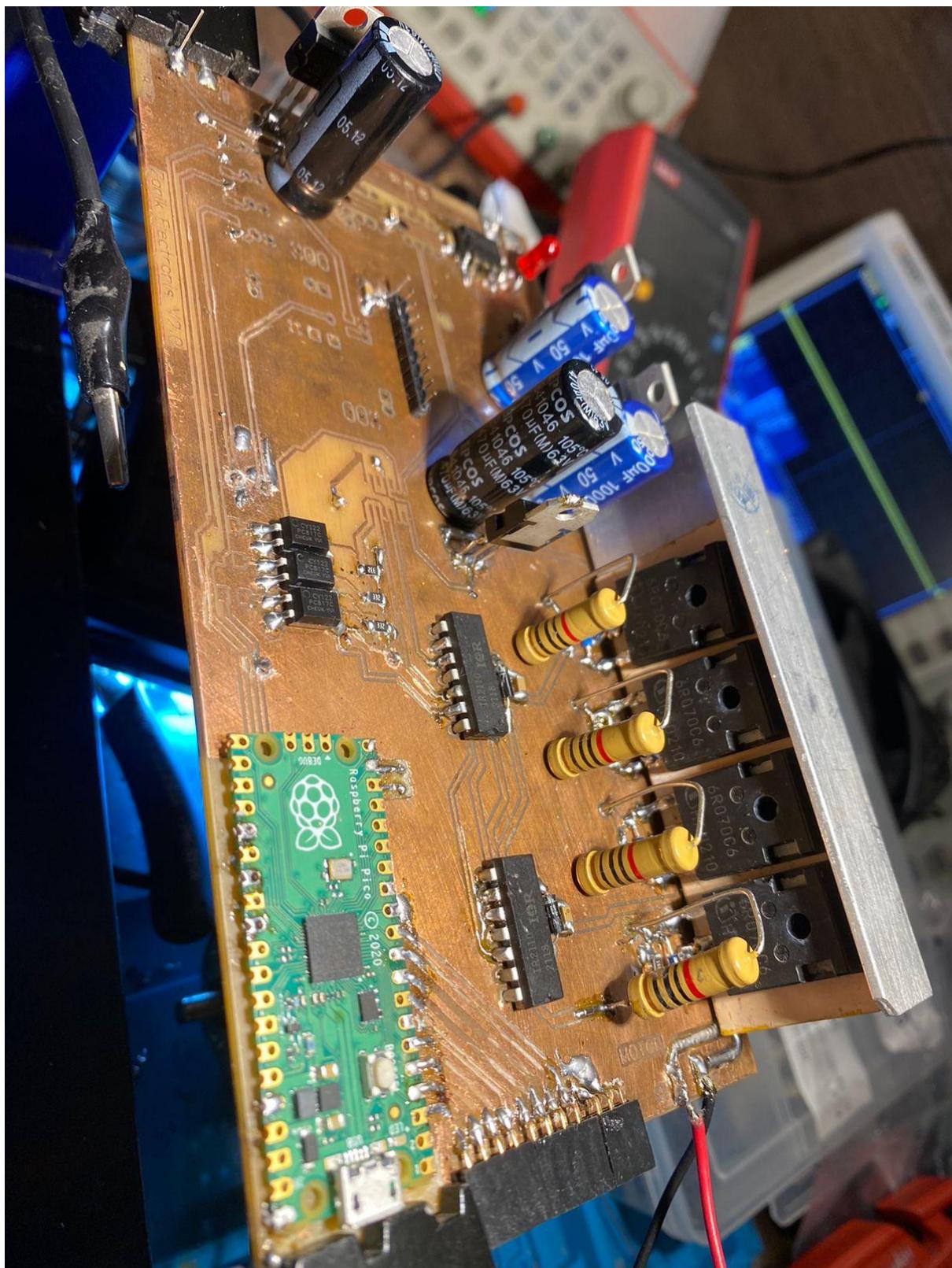


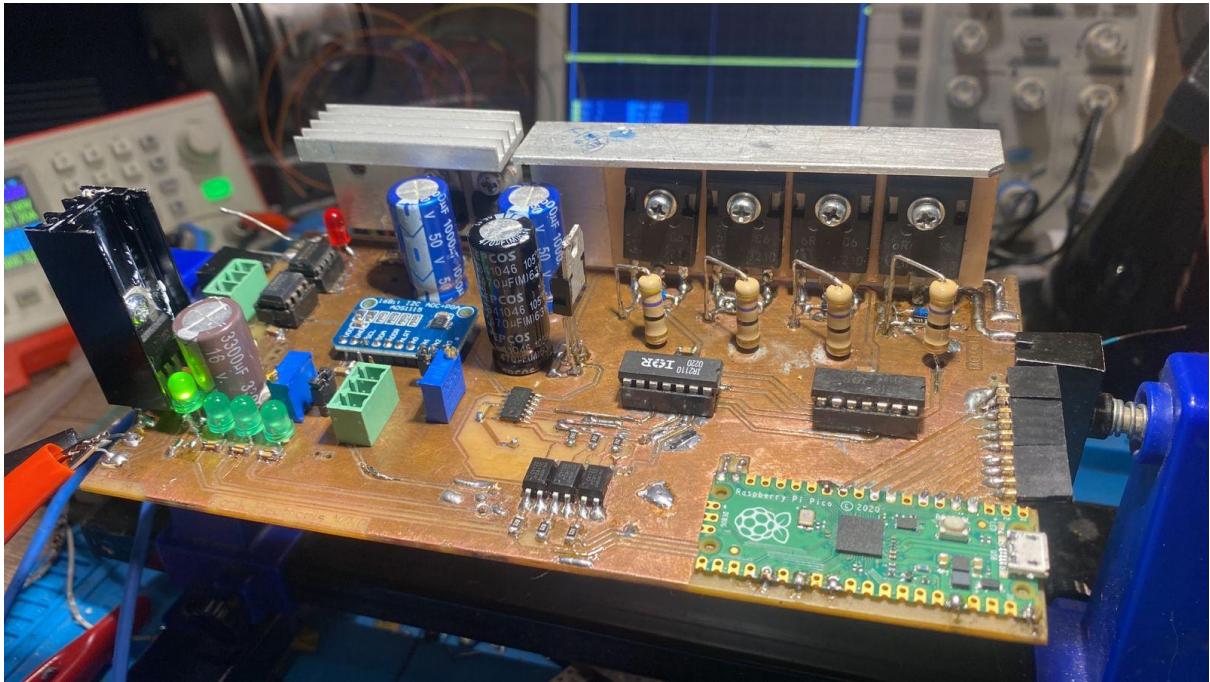


PLACA DE CONTROL SIN MOSFETS COLOCADOS.

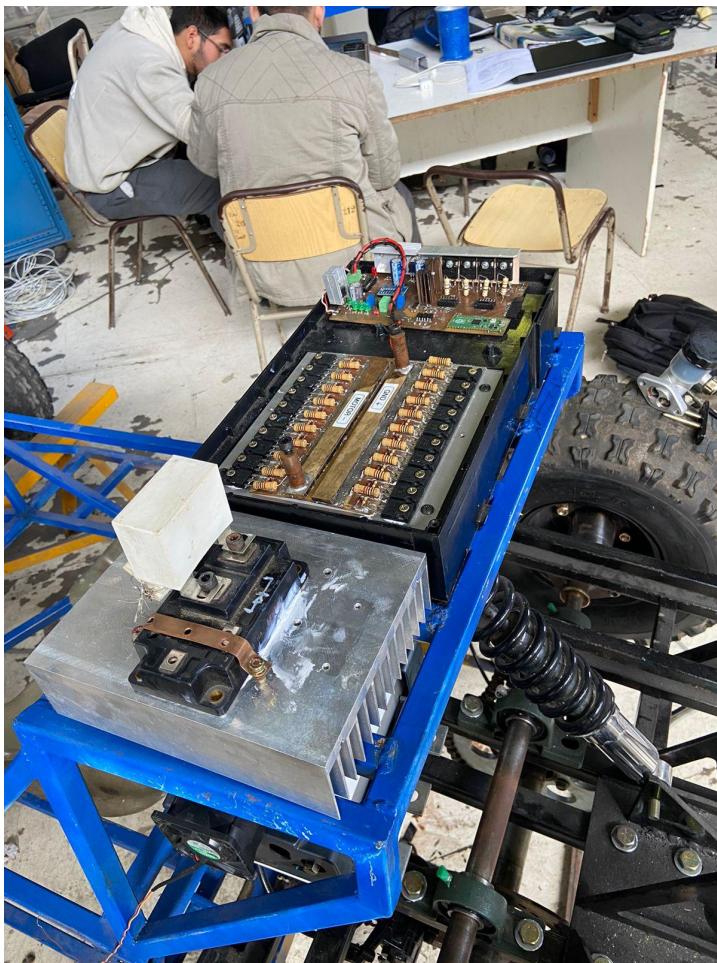
Semana del 27 de septiembre al 2 de octubre: PLACA DE CONTROL.







Semana del 4 al 9 de octubre: Esta semana, comenzamos con nuevas pruebas del funcionamiento.



Estas son las placas nuevas ya montadas acompañadas de diodos los cuales deberian eliminar/filtrar los picos de voltaje.



Luego de realizar algunas pruebas, estabamos teniendo picos de voltaje muy altos de 400v, a la par de esto lo que sucedió fue que el microcontrolador raspberry pi pico se colgo, lo cual provoco que siga enviando pulsos aunque sandtronic no esté acelerado. Logramos captar el momento justo de la explosion.

Semana del 11 al 16 de octubre: Volvimos a rehacer la placa de potencia, esta vez instalando el contactor, el cual es un botón de emergencia que en caso de necesitar se lo utiliza provocando que el circuito deje de estar cerrado y corte la señal.



Probando con el contactor y los nuevos componentes que conseguimos para la placas de potencia.

Semana del 18 al 23 de octubre: Colocamos 2 contactores para realizar el cambio de marcha, es decir que cada uno tiene una polaridad designada y provoca que el motor gira hacia un lado o el otro.



Siendo esta una de las ultimas semanas nos enfocamos en el código

Códigos que modificamos:

- ❑ PWM acelerador.py
- ❑ TEST MULTI THREADING.py
- ❑ VELOCIMETRO SANDTRONIC T...
- ❑ drive modes.py
- ❑ main.py

PWM

```
-
2
3     import machine
4     from machine import Pin, ADC, PWM
5     import utime
6     import time
7     def map(var, low_min, low_max, high_min, high_max):
8         low_range = low_max - low_min
9         per = var / low_range
10        high_range = high_max - high_min
11        result = high_range * per
12        return int(result)
13    adc = ADC(28)
14    pwm = PWM(Pin(7))
15    pwm.freq(2500)
16
17
18    while (1):
19        reading = adc.read_u16() - 19500
20        print(adc.read_u16())
21        res = map(reading,0,3.3,0,74000)
22        if adc.read_u16() < 20000:
23            res = 0
24        print("ADC: ", res)
25        utime.sleep(0.01)
26        pwm.duty_u16(res)
```

TEST MULTI THREADING

```

1 import _thread
2
3 def imprimir_valores_velocidad():
4     print('soy veloz')
5
6 def imprimir_valores_autonomia_corriente():
7     while True:
8         print('soy autonomo nashei')
9
10 _thread.start_new_thread(imprimir_valores_autonomia_corriente, ())
11 while True:
12     imprimir_valores_velocidad()
13

```

Velocimetro

```

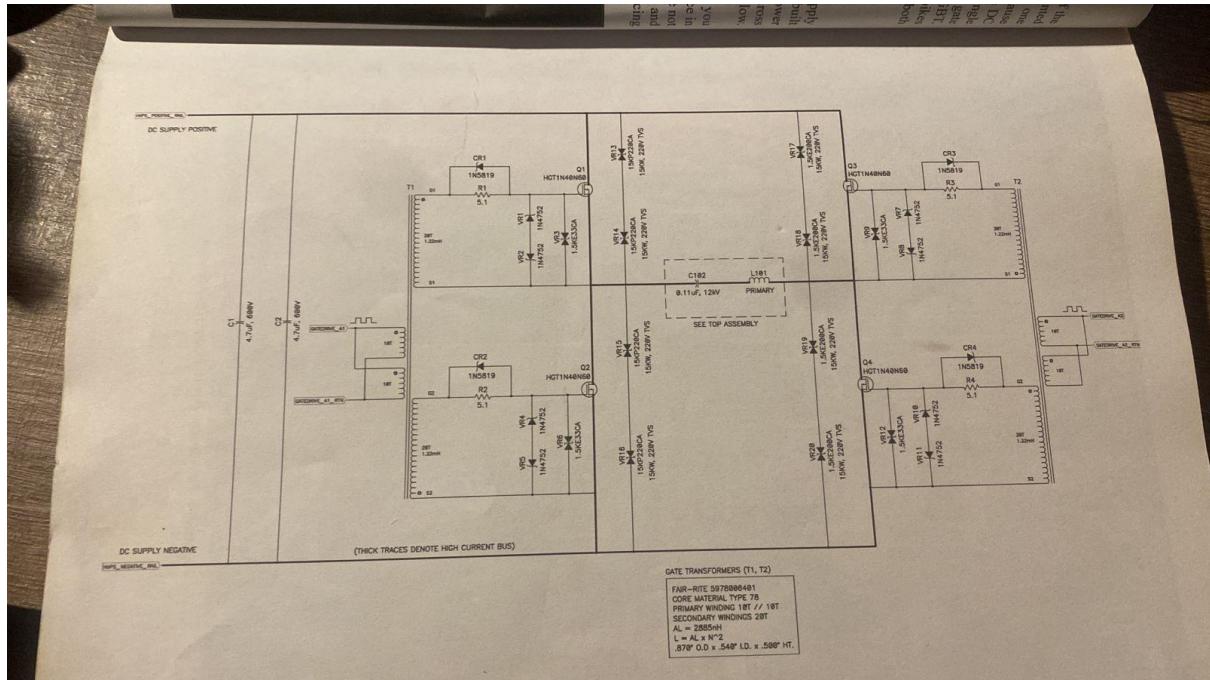
1
2 import time
3 import machine
4
5 servo_pin = 13
6 pwm2 = machine.PWM(machine.Pin(servo_pin))
7 pwm2.freq(50)
8
9 sensor_pin = machine.Pin(20, machine.Pin.IN)
10 pulse_count = 0
11 speed = 0.0
12 diametro= 0.55
13 circunferencia= 0.55 * 3.14
14 relacion = 8.9
15 cant_imanes = 4
16
17
18 servo_duty_0 = 8000
19 servo_duty_180 = 1333
20
21 duty = servo_duty_0
22
23 m = (1333-8000) / 80
24
25 def calculate_speed(t):
26     global pulse_count, speed, duty
27     speed = pulse_count * 10 / (relacion * cant_imanes) * circunferencia * 3.6
28     duty = int (m * speed + servo_duty_0)

```

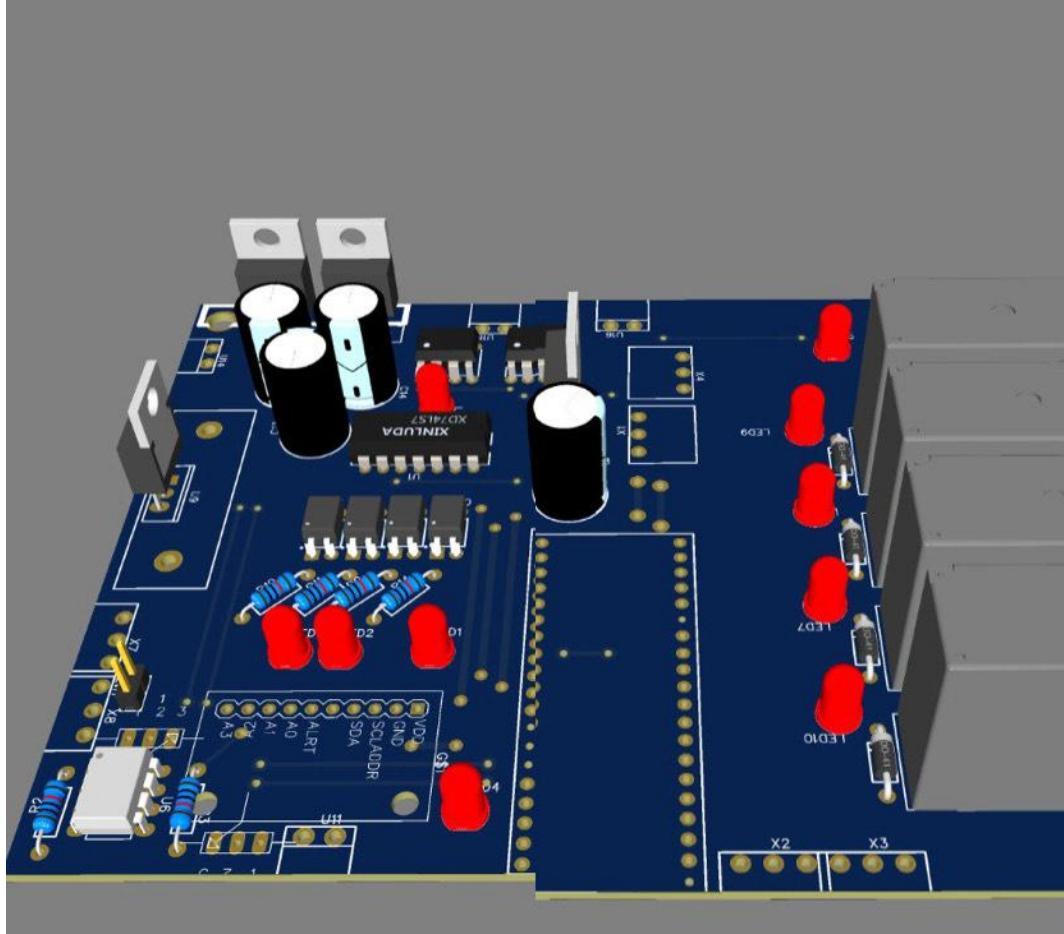
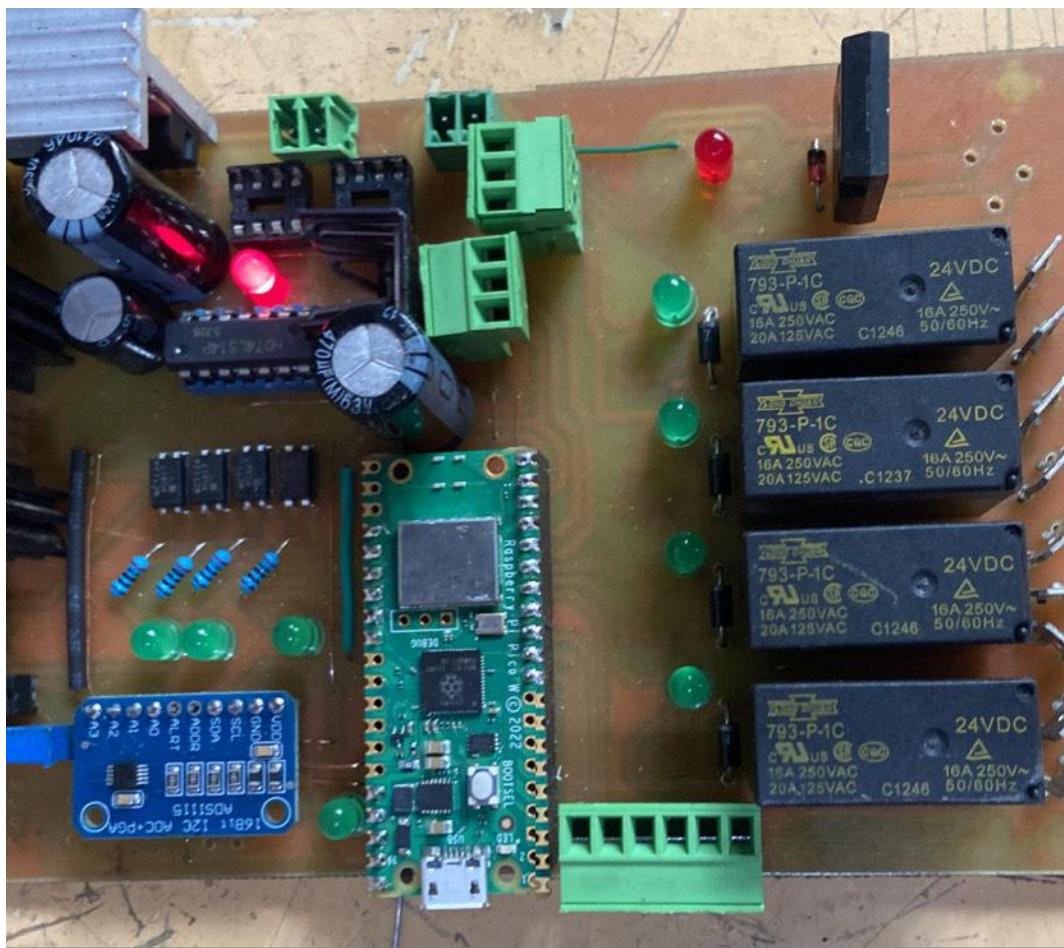
Modos de los drivers

```
1 import machine
2 import utime
3 import time
4 import ads1115
5 def map(var, low_min, low_max, high_min, high_max):
6     low_range = low_max - low_min
7     per = var / low_range
8     high_range = high_max - high_min
9     result = high_range * per
10    return int(result)
11
12 from machine import Pin,I2C, PWM
13 a1 = PWM(Pin(21))
14 a2 = PWM(Pin(19))
15 sw1 = PWM(Pin(3))
16 sw2 = PWM(Pin(4))
17 sw3 = PWM(Pin(5))
18
19 pwm = PWM(Pin(21))
20 pwm.freq(1500)
21
22 i2c = I2C(1, scl=Pin(15), sda=Pin(14), freq=400000)
23 adc = ads1115.ADS1115(i2c, 72, 0)
24
25
26 while (1):
```

Semana del 25 de octubre al 2 de noviembre: Esta semana investigamos en el libro de electronica **DRSSTC BUILDING THE MODERN DAY TESLA COIL McCauley**, llegando a la solucion que deberiamos armar el siguiente circuito para reducir los picos de voltaje.



Luego realizamos el diseño de una nueva placa de control.



SEMANA FINAL DEL 2 DE NOVIEMBRE AL 9 DE NOVIEMBRE:

Esta semana terminamos de fabricar las baterías de litio que alimentan el contactor, cambiamos los cables los cuales generaban inductancia, por barras de cobre que recuperamos, de un banco de capacitores, y fuimos agregando filtros probando en distintas partes del circuito.



BATERIAS DE LITIO



CAPACITOR QUE FILTRA EN LA SEÑAL DEL CONTACTOR



