



Robot cu detector de metale

Profesor coordinator : Trăsnea Bogdan

Nume : Sandu Octavian

Programul de studii : Automatica

Grupa : 4LF411

An : 2022-2023



Cuprins

1. Introducere.....	3
2. Arhitectură.....	5
3. Conectarea componentelor	9
4. Programarea.....	10
5. Ansamblarea finala	12
6. Probleme.....	14

1. Introducere

Ideea acestui robot a pornit de la robotii folosiţi pentru detectarea minelor, fiind un concept destul de interesant pentru mine.



Am vrut sa construiesc o varianta mai simpla care ar putea fi printata 3d. Astfel, am găsit 2 modele de roboti, eu dorind sa pic undeva intre acestea din punct de vedere al complexităţii.



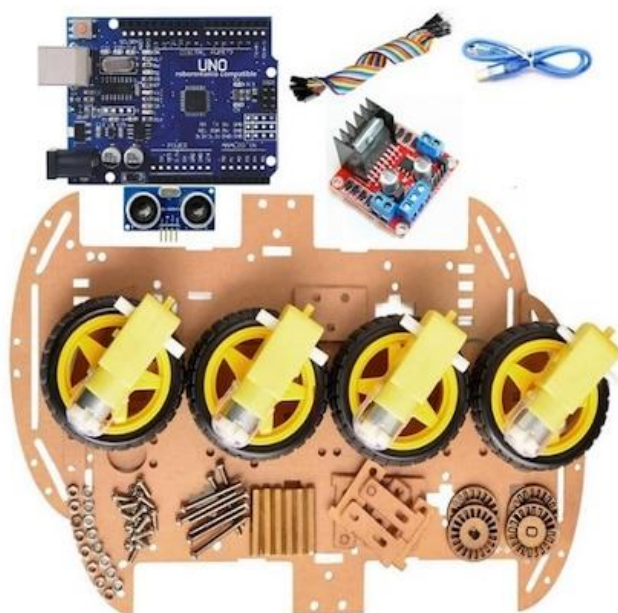
Mi-am dorit ca acesta sa fie complet autonom, cele doua modele prezentate anterior fiind controlate prin telecomanda.

Ideea inițială a acestui proiect a fost sa controlez totul cu un Raspberry Pi pentru ca știam deja cum sa folosesc sistemul de operare Linux, însă am descoperit destul de repede ca nu era posibil. In final, am ajuns

la configuraţia actuala, aceasta fiind un Raspberry Pi 3A+ si un Arduino Uno cu un CNC Hat V3 conectate prin Serial.

2. Arhitectură

Inițial am comandat un Kit Robot 4WD ca si baza pe care sa montez brațul robotic. Acesta a venit cu șasiul, 4 motoare DC de 12V, o punte H L298n si un Arduino Uno.



Pe lângă aceste am mai utilizat următoarele:

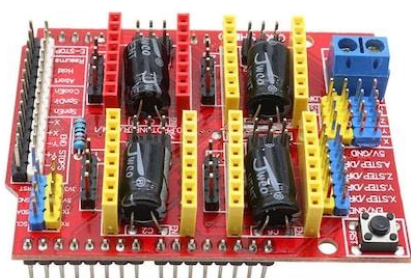
- Un kit circuit detector de metale



- 2 Drive pentru motoare pas cu pas (Stepper)



- O placa de expansiune CNC V3



- Un motor servo MG996



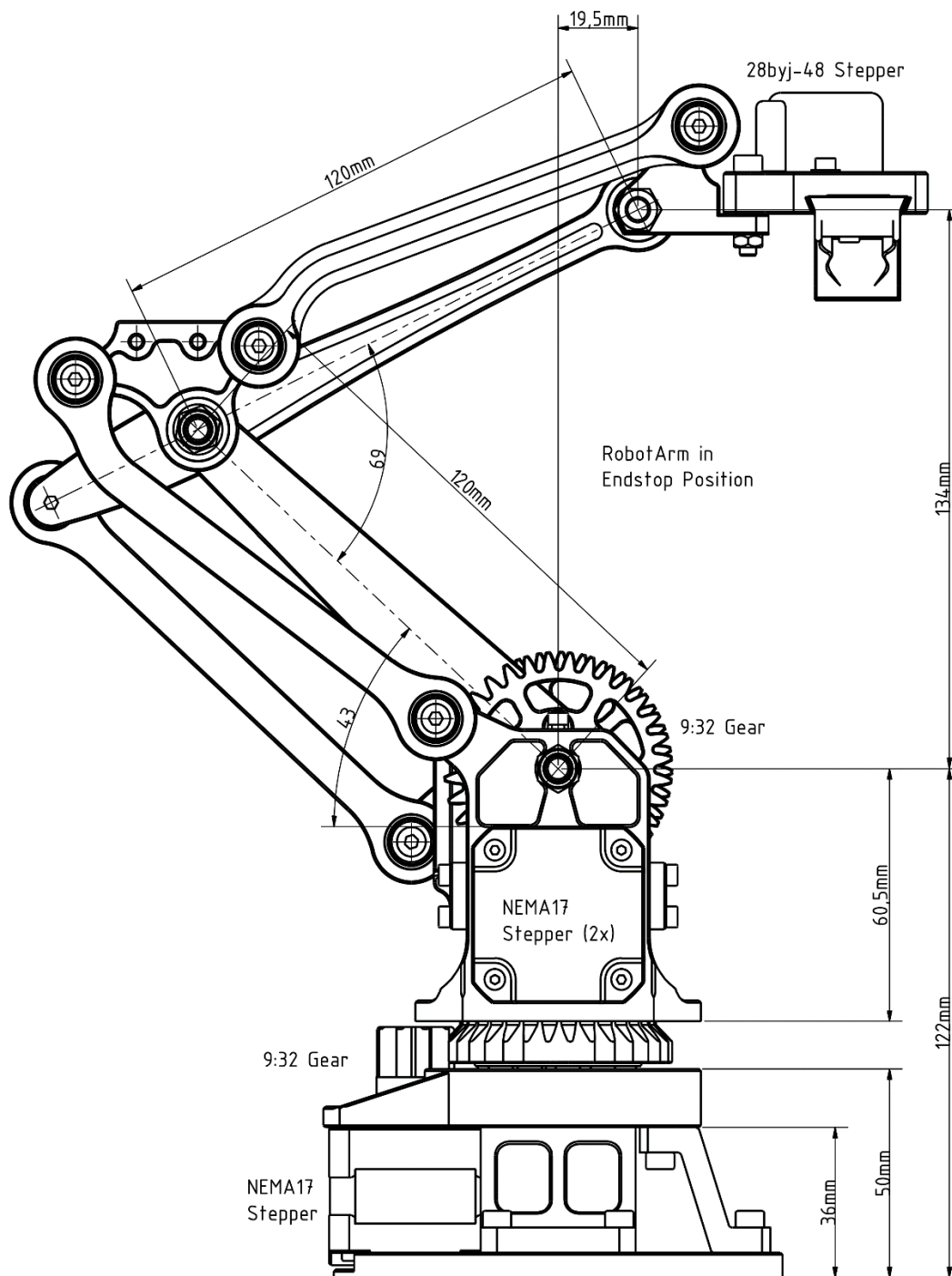
- 2 motoare pas cu pas (stepper) NEMA OKY7414



- Cabluri, baterii de 9V si conectori pentru baterii

Pe lângă acestea, am avut nevoie de un braţ pe care sa montez detectorul de metale

printat 3D un braţ (<https://www.thingiverse.com/thing:1718984>) , conceput original in jurul unei placi RAMPS pentru un Arduino Mega (https://www.reprap.org/wiki/RAMPS_1.4).



Bratul a fost printat folosind imprimanta Ender 3 Pro si utilizând PLA+.

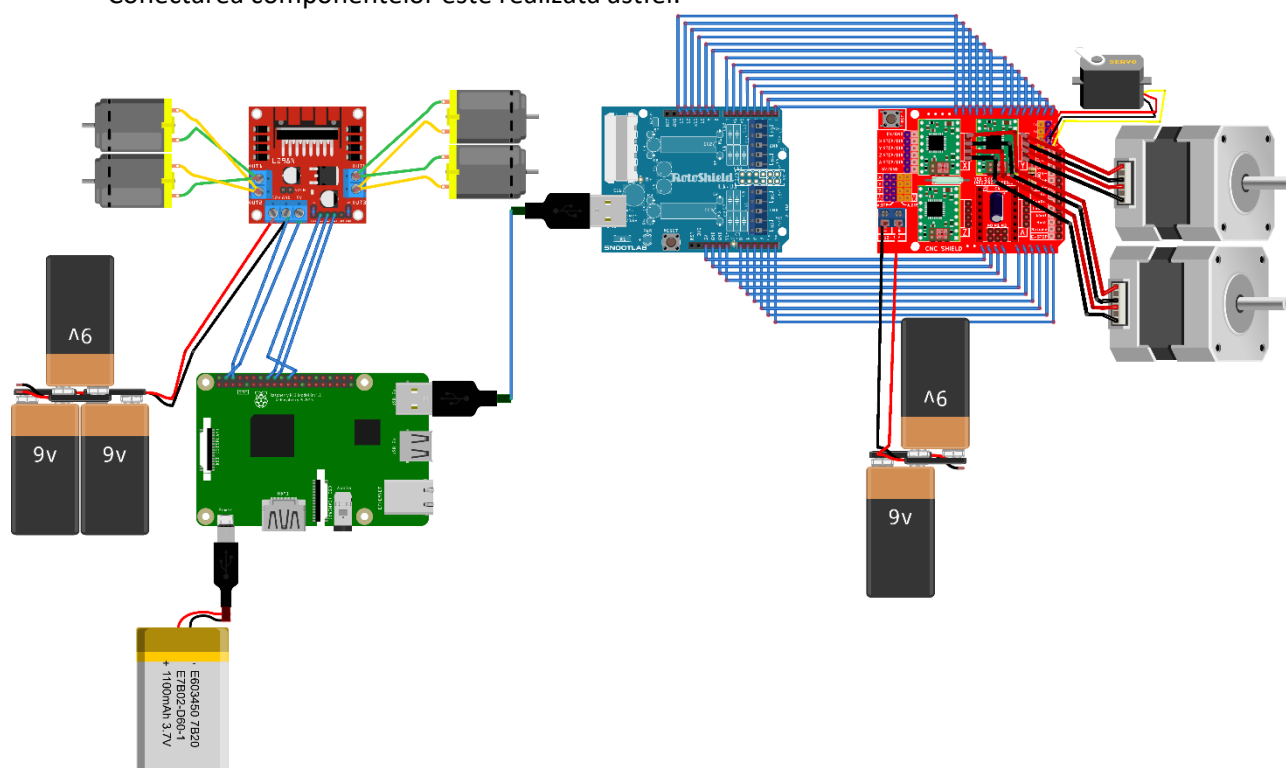
Acest model foloseşte 3 motoare stepper si un servomotor pentru a acţiona cleştele, însă pentru mişcarea pe axa Z am decis sa folosesc un servo pentru a păstra greutatea cat mai mica.

Pentru acesta am avut nevoie de:

- 10 rulmenţi 686 ZZ SRBF (686-2z)
- 1 rulment 51105 SRBF (51105-srbf)
- 6 rulmenţi 624 ZZ SRBF (624-zz-srbf)
- Şuruburi M3 si M4 împreună cu piuliţele adecvate

3. Conectarea componentelor

Conectarea componentelor este realizata astfel:



fritzing

4. Programarea

Arhitectura acestui proiect este bazată pe GRBL, un software pentru interpretarea codului G-Code. G-Code este cel mai utilizat limbaj pentru platformele de printare 3D şi CNC, care ajută mişcarea componentelor în spaţiul 3D utilizând axele XYZ.

În mod oficial, GRBL acceptă doar motoare stepper, însă există o versiune modificată a acestuia ce poate controla şi motoare servo, numită [grbl-servo](#).

Comenzile în G-Code sunt transmise din Raspberry Pi prin serial către Arduino pe baza detectorului de metale, acesta transmitând un semnal High când detectează un obiect metalic pe pin-ul 36 (GPIO 16).

Pe lângă acesta, programul pentru a controla motoarele este standard, scris în python:

```
import RPi.GPIO as GPIO
from time import sleep

in1 = 24
in2 = 23
en = 25
temp1=1

GPIO.setmode(GPIO.BCM)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(en,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
p=GPIO.PWM(en,1000)
p.start(25)
print("\n")
print("The default speed & direction of motor is LOW & Forward.....")
print("r-run s-stop f-forward b-backward l-low m-medium h-high e-exit")
print("\n")

while(1):

    x=input()
```

```
if x=='r':
    print("run")
    if(temp1==1):
        GPIO.output(in1,GPIO.HIGH)
        GPIO.output(in2,GPIO.LOW)
        print("forward")
        x='z'
    else:
        GPIO.output(in1,GPIO.LOW)
        GPIO.output(in2,GPIO.HIGH)
        print("backward")
        x='z'

elif x=='s':
    print("stop")
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.LOW)
    x='z'

elif x=='f':
    print("forward")
    GPIO.output(in1,GPIO.HIGH)
    GPIO.output(in2,GPIO.LOW)
    temp1=1
    x='z'

elif x=='b':
    print("backward")
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH)
    temp1=1
    x='z'

elif x=='l':
    print("low")
    p.ChangeDutyCycle(25)
    x='z'

elif x=='m':
    print("medium")
```

```
p.ChangeDutyCycle(50)
x='z'

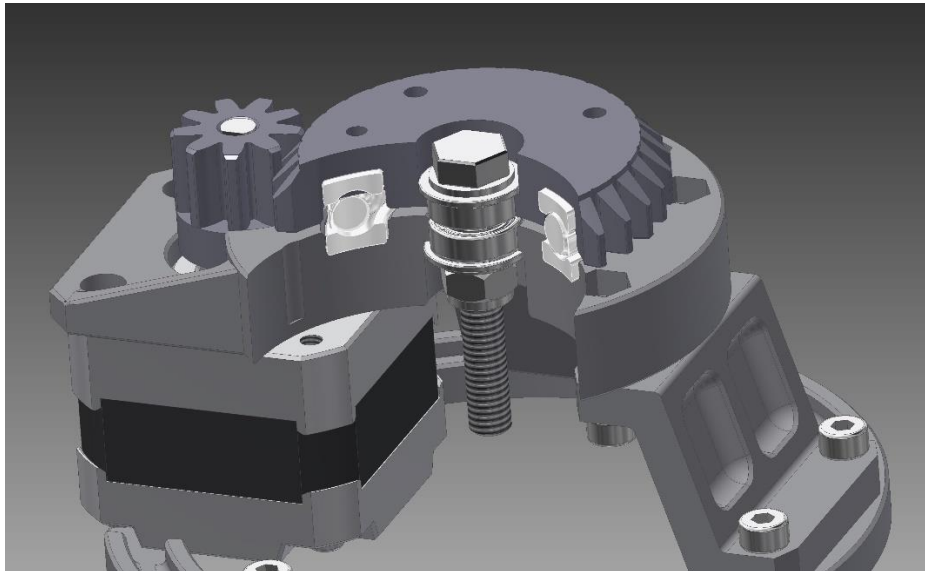
elif x=='h':
    print("high")
    p.ChangeDutyCycle(75)
    x='z'

elif x=='e':
    GPIO.cleanup()
    break

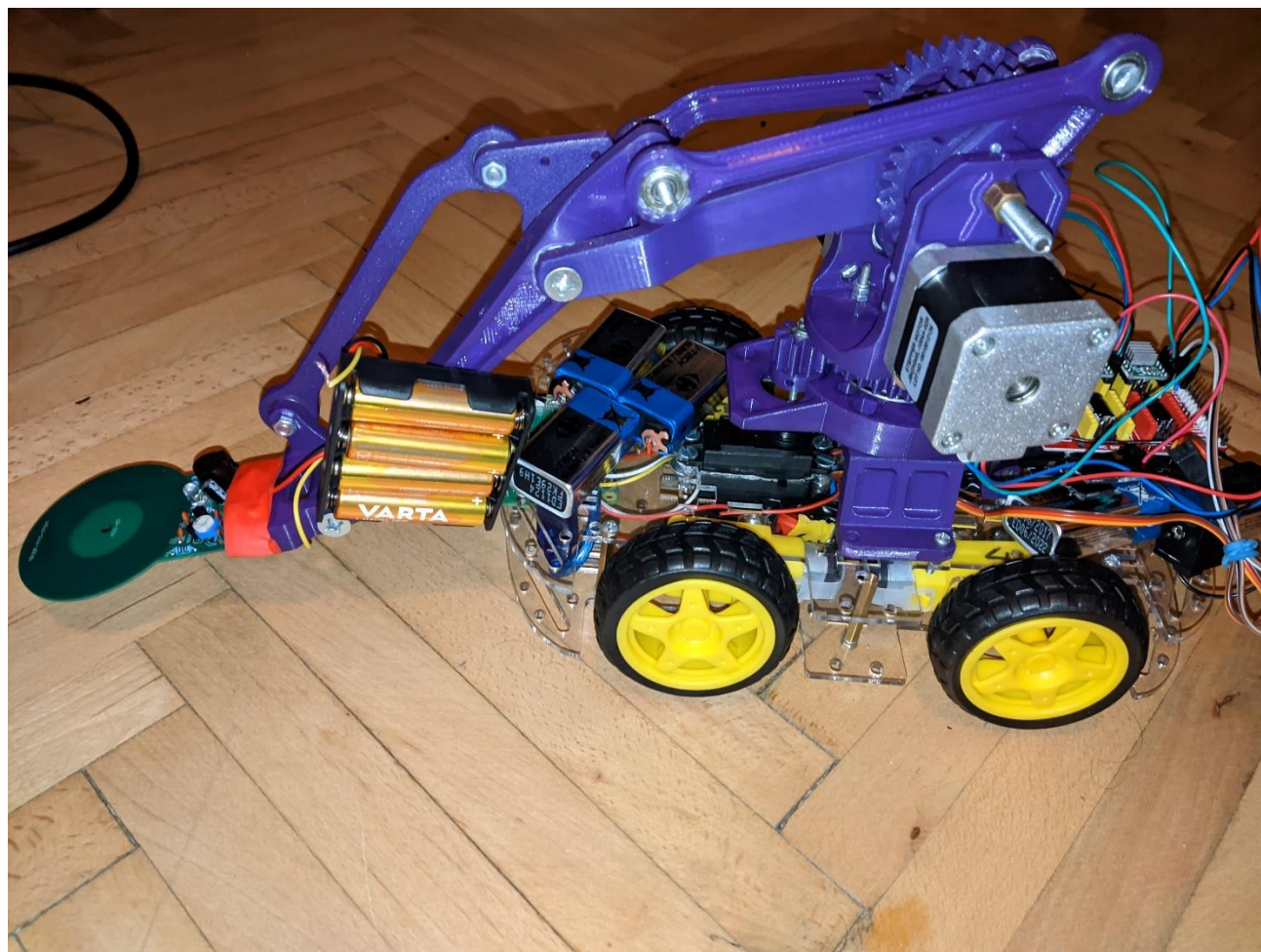
else:
    print("<<<  wrong data  >>>")
    print("please enter the defined data to continue.....")
```

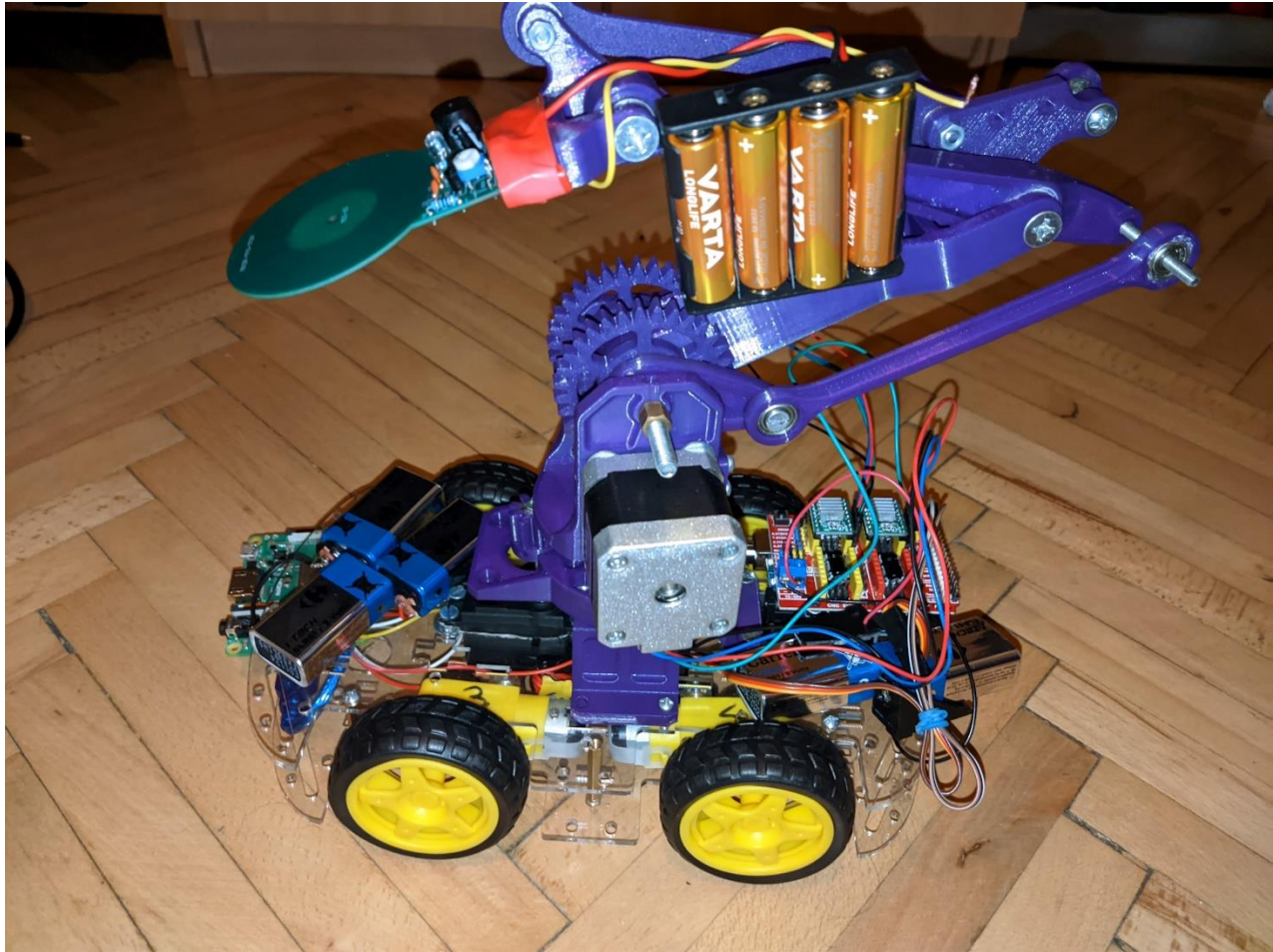
5. Ansamblarea finala

Cea mai complexa parte a robotului a fost miscarea pe axa Z. Bratul este situat pe 3 rulmenti prinsi intre un surub si o piulita.



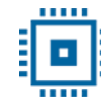
In urmatoarele imagini este prezentat robotul final





6. Probleme

- Puntea H nu accepta mai mult de 12V, ceea ce este necesar pentru alimentarea tuturor celor 4 motoare
- Puntea H nu distribuia tensiunea echivalent, astfel decat 2 motoare primeau curent.
- Am incercat multiple programe pentru a controla puntea H, insa nici unul nu functiona corect
- Puntea H a cerut mai mult de 5V din Raspberry Pi, astfel arzandu-l
- Motoarele DC au conectori foarte fragili, s-au rupt de 3 ori pe fiecare motor
- Cablurile din kit sunt foarte fragile
- Arduino-ul nu ruleaza instructiunile din GRBL in mod corect, nu proceseaza mai mult de 4 instructiuni fara a trimite erori, instructiunile pe care le proceseaza sunt inconsistente (aceeasi instructiune produce rezultate diferite)
- Potentiometrul driverelor se invarte in gol (in mod normal au un minim si maxim)



- Am ars un driver deoarece l-am pus invers
- Contactele puntei H si al placii de expansiune pentru sursa de curent nu prind cablurile
- Rulmentii si suruburile au fost foarte greu de gasit, fiind nevoie sa fie comandate de pe internet (in unele cazuri imposibil de gasit)
- Dimensiunile unor modele 3D sunt gresite (gaurile pentru rulmenti sunt prea mari sau prea mici)