**Proiect de atestat profesional la Informatică**

Robot SCARA

**Profesor coordonator:**

Florea Delilah

**Olăreanu Alexandru**

**Sibiu, 2021**

# Rezumat:

Această lucrare descrie designul și construcția atât elementelor de hardware dar mai ales software-ul uni Robot SCARA. Tot robotul este proiectat și constituit de la zero. Acesta poate fi folosit ca o bună platformă de dezvoltare a viitorilor roboți industriali la costuri mult mai mici.

Cuprins

[Rezumat: 2](#_Toc71910895)

[Introducere 2](#_Toc71910896)

[Hardware 3](#_Toc71910897)

[SCARA-Commander – Codul sursă 3](#_Toc71910898)

[Descrierea programului 4](#_Toc71910899)

[Schema componentelor software 5](#_Toc71910900)

[Stratul de abstracție al hardware-ului 5](#_Toc71910901)

[Interpretor Gcode 6](#_Toc71910902)

[Planificatorul de mișcări 6](#_Toc71910903)

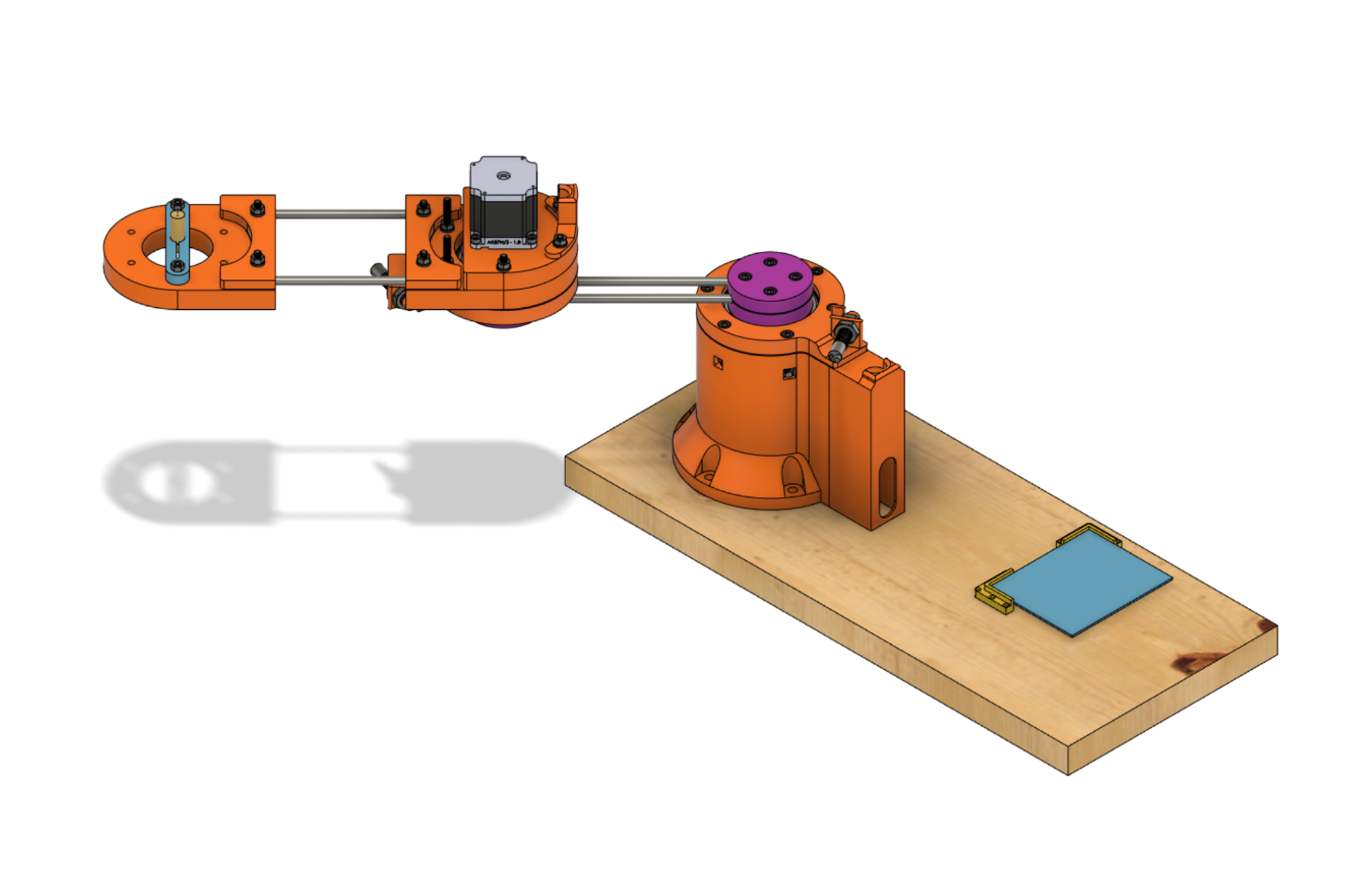
[Bibliografie 6](#_Toc71910904)

[Anexa – Scheme tehnice 8](#_Toc71910905)

# Introducere

[SCARA](https://en.wikipedia.org/wiki/SCARA) este un acronim pentru Selective Compliance Articulated Robotic Arm. Aceste brațe robotice sunt de obicei formate prin îmbinarea a mai multor segmente unul după altul, cu alte axe perpendiculare pe ele. Exista multe tipuri de geometrii ale roboților SCARA, cum ar fi [Morgan-SCARA](https://en.wikipedia.org/wiki/RepRap_Morgan). In industrie se aceștia sunt produși de companii populare cum ar fi ABB, Kuka sau Fanuc.

Acest proiect constituie dezvoltarea de la stadiul de design la construcție și testare a unui robot SCARA cu două axe. O diodă laser poate fi adăugată in vârful celui de al doilea segment pentru a putea observa mai bine mișcările brațului. Robotul primește comenzi în lmbajul standardizat Gcode de la un computer și execută comenzi.



Figură 1Robotul SCARA în CAD

# Hardware

Hardware-ul este baza fiecărui sistem. Acest robot are două articulații cu două segmente unul după altul cu lungimi de 225 respectiv 275 mm pentru o acoperire de 0,5 m. Tot ansamblul este montat pe o placă de lemn, pentru a ușura transportul. Majoritatea robotului este tipărit 3D cu o imprimantă Prusa I3 Mk3 și în mare parte filament Prusament. Acest fapt reduce semnificativ costurile de producție.

Piese precum profilele de 8mm ce constituie structura segmentelor, șuruburi, piulițe, cabluri sunt cumpărate de la companii locale. Motoarele pas cu pas sunt de la StepperOnline iar senzori de la un distribuitor local. Proiectarea a fost făcută în Autodesk Fusion360.

Electronica este făcută la comandă și comandată din China, proiectată în easyEDA și comandată la JLCPCB cu componente adiționale de la Mouser Electronics. Acest proiect rulează pe un microcontroler STM32F411CE împreuna cu drivere de motoare pas cu pas Trinamic TMC2660 și ele pe o plăcuță personalizată. Vezi anexe pentru scheme tehnice și [acest link](https://github.com/Olareanu/SCARA-Hardware) pentru datasheet-uri.

# SCARA-Commander – Codul sursă

Codul sursă se poate evalua aici: [SCARA-Commander](https://github.com/Olareanu/SCARA-Commander)

## Descrierea programului

Programul are ca scop controlarea în timp real al brațului. Acesta trebuie, în mare, să implementeze următoarele funcții:

* Comunicare Serial prin USB
* Interpretare Gcode
* Planificarea și execuția mișcărilor
* Funții de siguranță

Codul este scris în C++ în mediul CLion IDE cu ajutorul platformei PlatformIO ce facilitează programarea microcontrolerului. Git este folosit pentru controlul versiunii.

Proiectul are 3 mari componente:

* Stratul de abstracție al hardware-ului (HAL)
* Interpretorul de Gcode
* Planificatorul de Mișcări

## Schema componentelor software

Computer

Senzori

Interpretor Gcode

HAL

Planificator

Drivere Motoare pas cu pas

## Stratul de abstracție al hardware-ului

Numit pe engleză „Hardware Abstraction Layer” – HAL această parte a codului creează o interfață între funcțiile low level ale sistemului și logica high level a acestuia.

Microcontrolerele nu oferă un sistem de operare așa că programul în sine trebuie să conțină definiții pentru toate componentele, să ofere comunicare prin diferite protocoale, cum ar fi SPI sau Serial prin USB, să prezinte funcții de siguranță, să seteze regiștrii etc. Aceste funcții sunt oferite de un HAL.

Aceste funcții se concretizează în apelarea a multor librării ce implementează diferite funcții. Platforma Arduino, folosită aici și compatibilă cu microprocesorul oferă Arduino.h, ce include standarde de comunicare SPI (într microcontroler și driverele de motoare pas cu pas), Serial prin USB (comunicare cu computerul) și acționare de pini prin comenzi simple ca „pinMode()” în loc de setare a regiștrilor interni.

Librăria TMC2660.h originează din software-ul popular open-sorce Marlin și este folosită sub o formă puternic modificată pentru a comunica prin SPI cu driverele de motoare pas cu pas. Aceasta oferă abilitatea de a seta regiștrii driverelor ușor și oferă wrapere pentru a seta viteza de rotație a motoarelor și alți parametri ai acestora.

Stratul de abstracție al hardware-ului mai oferă și citirea senzorilor

## Interpretor Gcode

Interpretorul de Gcode primește o linie de la Computer cu ajutorul HAL și o interpretează pentru a îi trimite informațiile din ea planificatorului de mișcări. O comandă timipă de Gcode arată așa:

G0 X20 Y53

„G0” este comanda în sine. Aceasta reprezintă o mișcare rapidă la anumite coordonate. „X20” și „Y53” sunt argumente care constituie coordonatele carteziene la care trebuie să se miște robotul, de obicei exprimate în mm.

Comenzi populare mai sunt „G1”(mișcare în linie dreaptă) și „G28”(mișcare spre poziția acasă). Argumente des folosite sunt „F” urmat de un număr ce reprezintă viteza de mișcare (F de la termenul englezesc feedrate).

La roboți ca cei SCARA se mai pot folosi și coordonate polare cu argumente „A” și „B” ce reprezintă poziția în grade a fiecărei axe, ceea ce este implementat în acest proiect.

## Planificatorul de mișcări

Scopul acestei componente este să planifice o mișcare validă și să o execute. În primul rând trebuie sa se asigure că poziția primită de la computer este validă și apoi sp calculeze o viteză și accelerație potrivită pentru mișcare, apoi să execute mișcarea respectivă prin HAL.

# Bibliografie

(fără an). *Forward Kinematics.* Preluat de pe https://en.wikipedia.org/wiki/Forward\_kinematics

(fără an). *Gcode.* Preluat de pe https://en.wikipedia.org/wiki/G-code

(fără an). *GRBL.* Preluat de pe https://github.com/grbl/grbl

(fără an). *Inverse Kinematics.* Preluat de pe https://en.wikipedia.org/wiki/Inverse\_kinematics

(fără an). *Marlin.* Preluat de pe https://marlinfw.org/

(fără an). *Morgan SCARA.* Preluat de pe https://en.wikipedia.org/wiki/RepRap\_Morgan

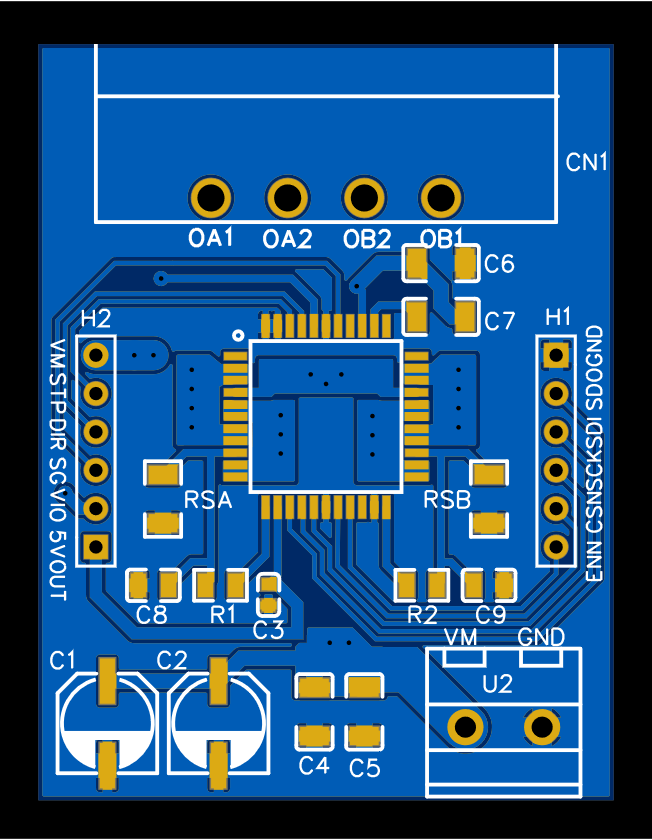
(fără an). *PlatformIO.* Preluat de pe https://platformio.org/

(fără an). *RepRap.* Preluat de pe https://github.com/grbl/grbl

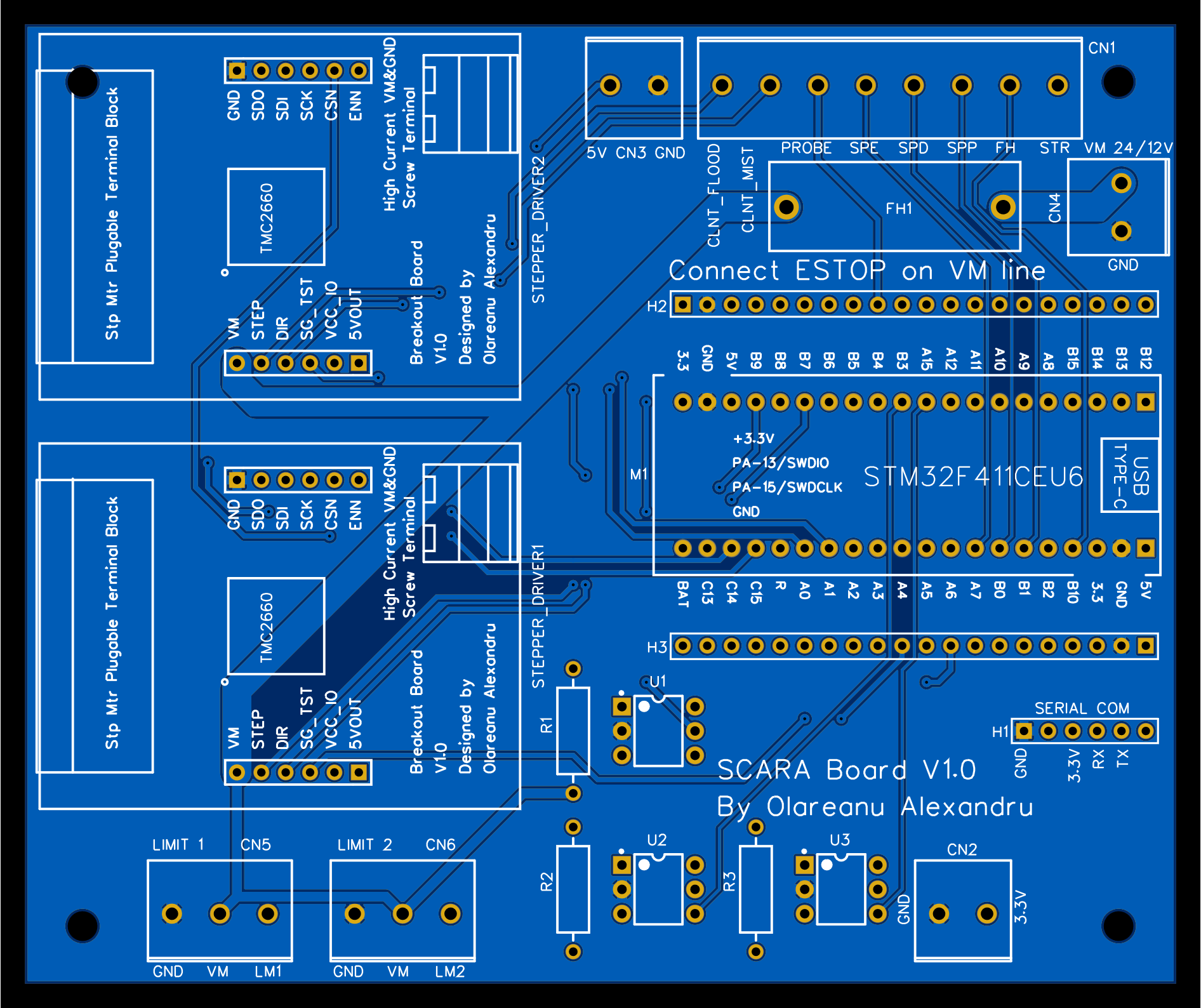
(fără an). *SCARA Robots.* Preluat de pe https://en.wikipedia.org/wiki/SCARA

(fără an). *Stepper Motors.* Preluat de pe https://en.wikipedia.org/wiki/Stepper\_motor

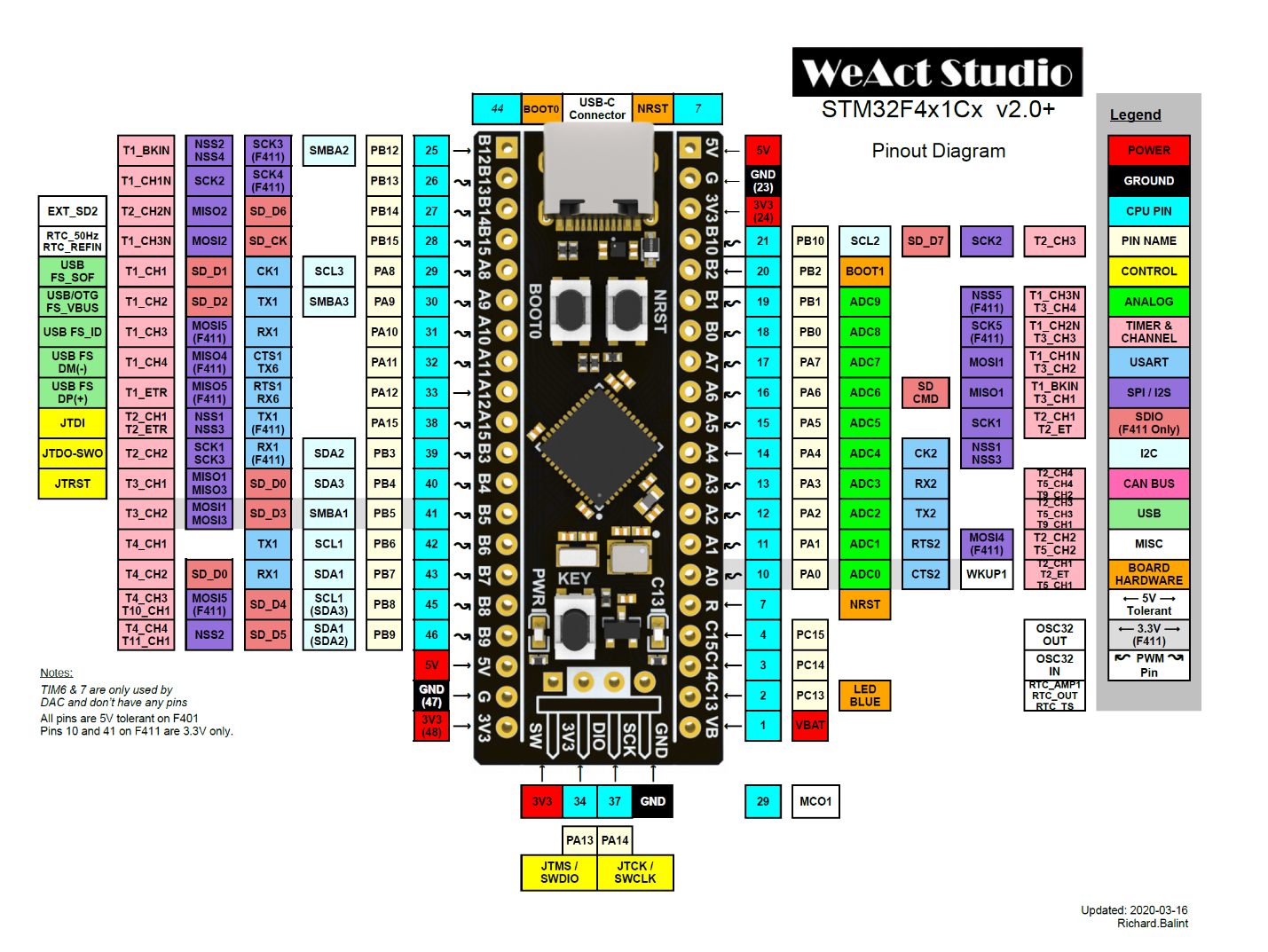
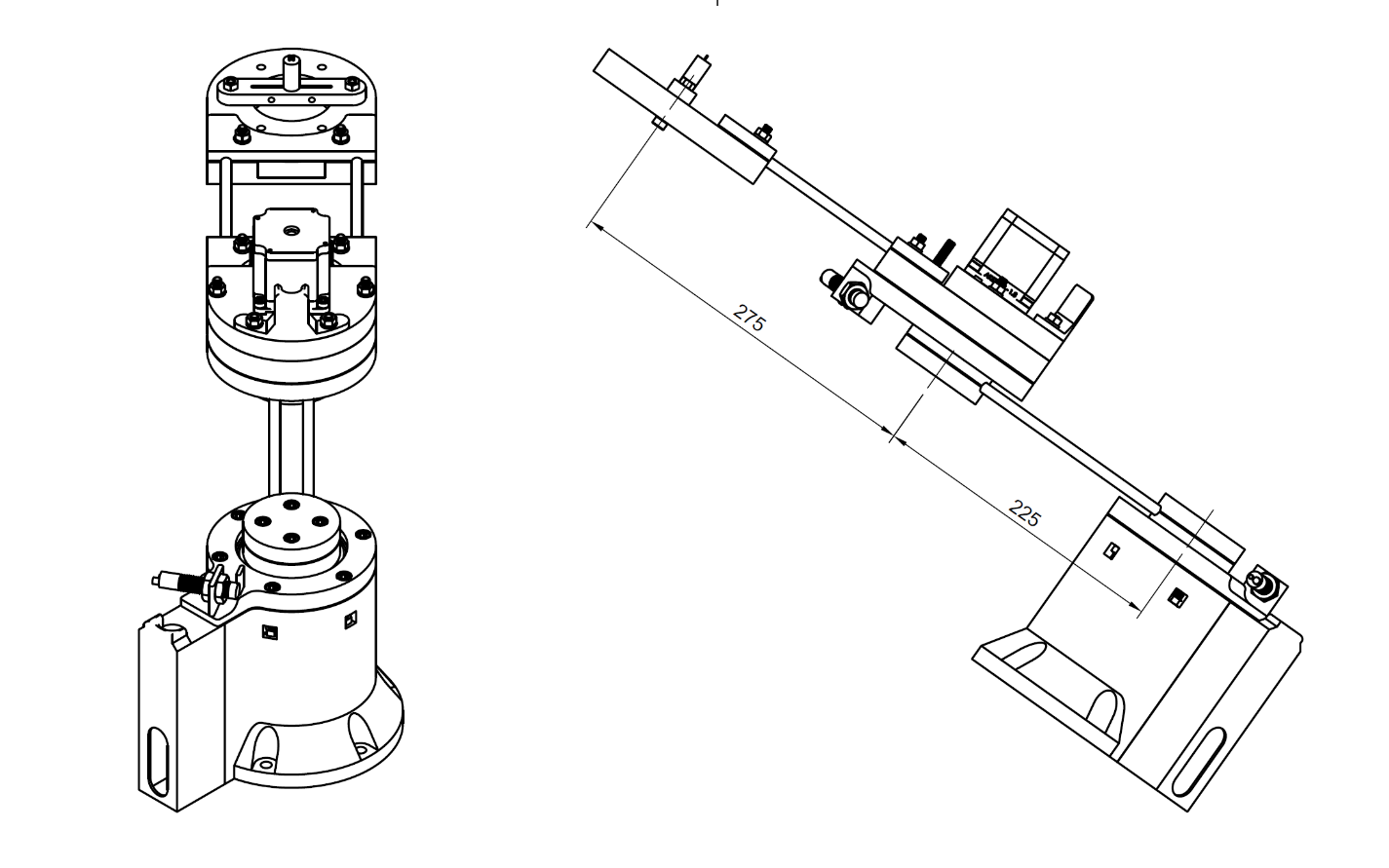
# Anexa – Scheme tehnice



Figură 2 Placa cu drivere pas cu pas TMC2660



Figură 3 Placa de bază



Figură 4 Schema tehică generală a robotului

Figură 5 Plăcuța cu Mircocontroler STM32F411CE