

Programmable PLC

Proiect realizat în colaborare cu studentul Anastasiu Alexandru Ioan, de la Facultatea de Inginerie Industrială și Robotică, anul II.

Introducere

Proiectul își propune transformarea unui microcontroller ESP32 într-un PLC care să poată controla roboți industriali, mai exact Mitsubishi RV-E3J și RV-E2, prin intermediul programelor scrise conform standardului IEC 61131-3. Ideea a pornit de la existența celor doi roboți și lipsa unui PLC pentru a îi programa. Cum unul din comerț poate deveni destul de scump, ne-am propus să facem unul propriu. Consider că este un proiect util în primul rând din scop educațional, deoarece îmi permite să pun în practică multe dintre cunoștințele obținute la facultate.

Descriere generală

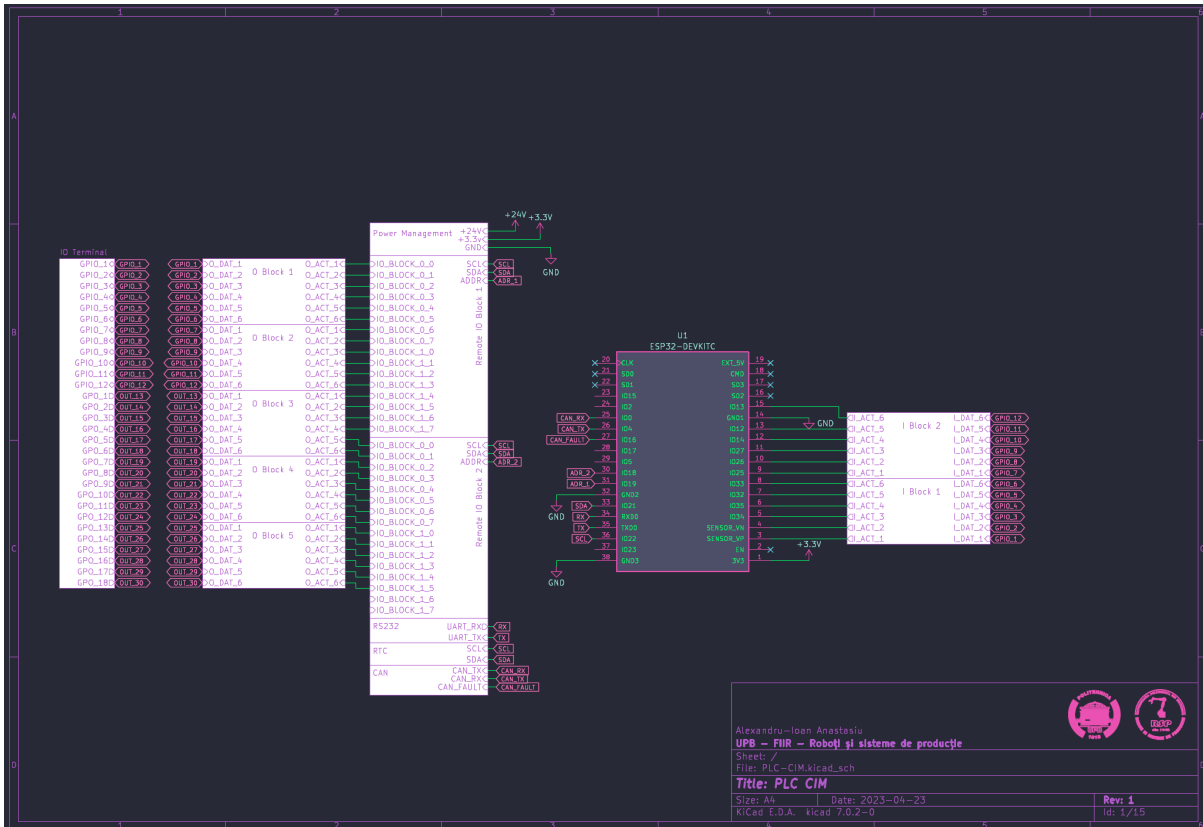
La modulul ESP32 am adăugat două block-uri de extindere de I/O, pe care le vom folosi ca Output. Am păstrat câțiva pini pe microcontroller pentru Input. Interpretarea programelor în Structured Text se va desfășura în felul următor: programatorul va scrie codul conform standardului (la care am adăugat mici modificări pentru a ne permite scrierea și citirea pe pini plăcuței), după care acesta va fi trecut printr-un parser care va scrie cod echivalent în Web Assembly. După aceea, codul nou obținut va fi compilat, iar rezultatul compilării va fi încărcat în memoria plăcuței și interpretat.

Hardware Design

Componente Hardware:

- 1x ESP32
- 1x eeprom
- 1x RTC
- 2x I2C I/O extender
- 32x relee solid-state
- 12x amplificatoare operationale
- Rezistori si condensatori de decuplare

Pentru a avea același număr de intrări și ieșiri ca un PLC normal, am folosit block-uri de extindere de

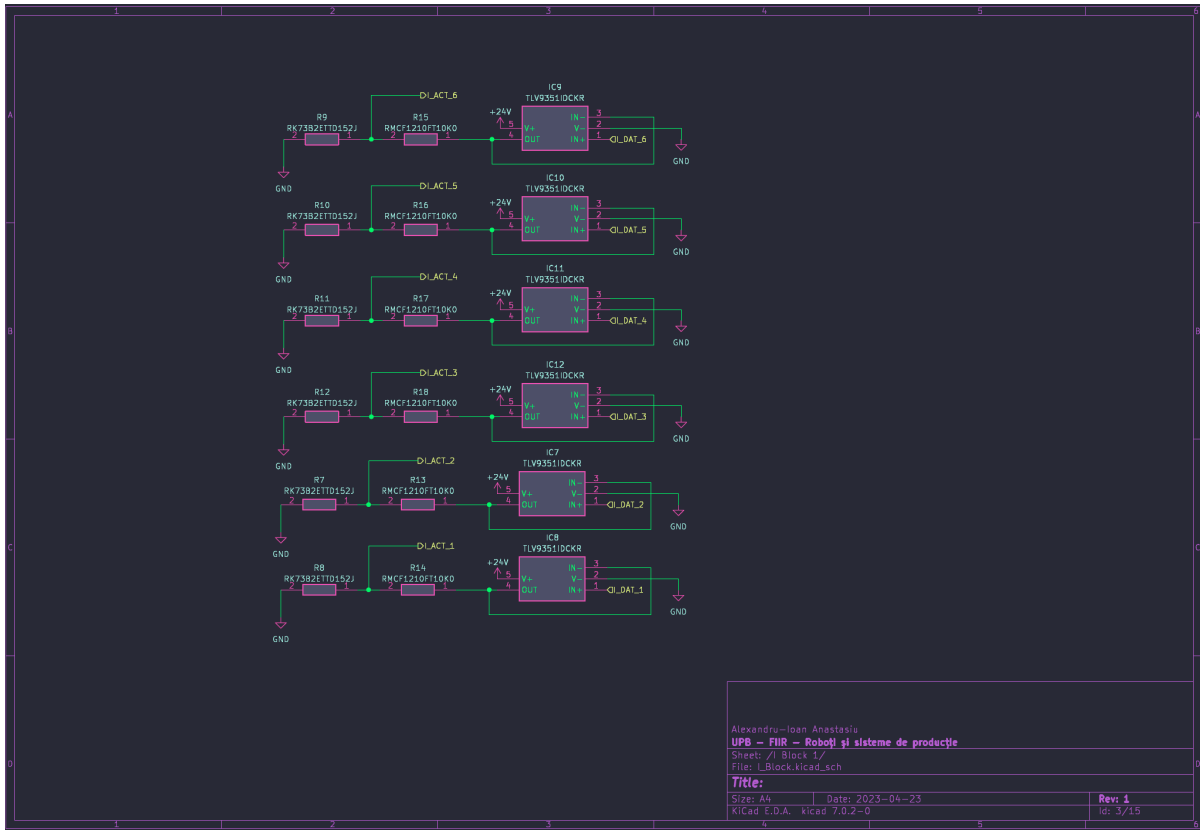


I/O configuration

Main



Output



Block
Block

Input

Software Design

Mediu de dezvoltare: -Intellij, cu plugin-ul de ANTLR
 -Visual Studio Code
 -Wasm3
 -FreeRTOS

Interpreter de Structured Text

Structured Text este unul din limbajele acceptate pentru programarea PLC-urilor. Pentru a ramane fideli conceptului de PLC, am ales sa il folosim si noi in scrierea de programe pentru PLC-ul nostru. Pentru ca microprocesorul de pe ESP32 sa il inteleaga, am creat, prin intermediul ANTLR, un parser pentru acest limbaj, care imi transfera codul in Web Assembly. Dupa aceasta transformare, este usor sa ii verific corectitudinea si sa il compilez. Pentru a incarca codul astfel obtinut pe placuta, ma voi folosi de modulul de WiFi al acesteia pentru a incarca fisierul si de wasm3 pentru a il interpreta. Aceasta este o solutie mult mai simpla decat dacda as fi incercat sa fac de la 0 un compilator pentru structured text.

Rezultate Obținute

In urma realizarii proiectului, parser-ul de Structured Text reuseste sa genereze cod de WebAssembly

```
1  VAR
2      STATUS : BOOL;
3      VAR1 : INT;
4  END_VAR;
5  VAR1 := 1;
6  WHILE VAR1 <= 4 DO
7      STATUS := INPUT_PIN_3;
8      OUTPUT_PIN_4 := STATUS;
9      OUTPUT_PIN_3 := TRUE;
10     VAR1 := VAR1 + 1;
11 END_WHILE;
```

```
1  (module
2      (import "IO" "getpin" (func $getpin (param $pin i32) (result i32)))
3      (import "IO" "setpin" (func $setpin (param $pin i32) (param $value i32)))
4      (func (export "run")
5          local $STATUS i32
6          local $VAR1 i32
7          i32.const 1
8          local.set $VAR1
9          block $endwhile0
10         loop $while0
11             local.get $V1
12             i32.const 4
13             i32.le
14             br_if $endwhile
15             i32.const 3
16             call $getpin
17             local.set $STATUS
18             local.get STATUS
19             i32.const 4
20             call $setpin
21             i32.const 1
22             i32.const 3
23             call $setpin
24             local.get VAR1
25             i32.const 1
26             i32.add
27             local.set $VAR1
28             br $while0
29         end $while0
30     end $endwhile0
31 )
32 )
```

Incarcarea

programului pe placuta ESP conduce la un boot loop, ramane sa investighez motivul.

Concluzii

In urma acestui proiect, am inceput sa stapandesc mai bine Web Assembly si partea de gramatica a unui limbaj de programare. In acelasi timp, m-am familiarizat cu Structured Text si standardul IEC 61131, utilizat in programarea PLC-urilor.

Download

Repo de git pentru parserul de Structured Text: https://github.com/mihaicostin34/ST_Parser

Bibliografie/Resurse

[Export to PDF](#)

Documentatie pentru Web Assembly Text:

https://developer.mozilla.org/en-US/docs/WebAssembly/Understanding_the_text_format

Standardul IEC 61131: https://en.wikipedia.org/wiki/IEC_61131

Datasheet ESP: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/ibradu/programmableplc>



Last update: **2023/05/31 10:53**