



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

<<Υλοποίηση PLC με λογισμικό ανοικτού κώδικα

OpenPLC σε Raspberry Pi, και εφαρμογές με χρήση Modbus/TCP

σε microcontroller ESP8266>>

Φοιτητής: Βοβούσας Γεώργιος

ΑΜ: 711130002

Εισηγητής: Κωσταντίνος Ευσταθίου, Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Φεβρουάριος 2023



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING DEPARTMENT
OF INFORMATICS AND COMPUTER ENGINEERING

Diploma Thesis

**<<PLC implementation with
OpenPLC open-source software in Raspberry Pi,
and applications using Modbus/TCP on ESP8266 microcontroller>>**

Student: Georgios Vovousas

Registration Number: 711130002

Supervisor: Constantinos Efstathiou, Professor

ATHENS-EGALEO, February 2023

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Κωνσταντίνος Ευστάθιου Καθηγητής	Ιωάννης Βογιατζής Καθηγητής	Ιωάννης Αμοργίνος Λέκτορας Εφαρμογών
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Copyright © 2023 Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Βοβούσας Γεώργιος του Σωκράτη, με αριθμό μητρώου 711130002 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΤΜΗΜΑΤΟΣ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ.

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Δεν επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας.

Ο Δηλών

Βοβούσας Γεώργιος



(Υπογραφή φοιτητή)

ΑΦΙΕΡΩΣΗ

Η διπλωματικής εργασία είναι αφιερωμένη στην σύζυγό μου Καίτη και στα παιδιά μου Σωκράτη-Νεκτάριο και Αγγελική, για την πίστη τους σε εμένα και την συνεχή υποστήριξη τους.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή μου και επιβλέποντα της παρούσας διπλωματικής εργασίας, κος Ευσταθίου Κωνσταντίνο για την πολύτιμη βοήθεια που μου παρείχε με την επιστημονική και συμβουλευτική καθοδήγηση σε όλη την διάρκεια της εκπόνηση αυτής της εργασίας.

Τέλος θα ήθελα να ευχαριστήσω τα μέλη της οικογενείας μου για την υπομονή και την συμπαράσταση τους.

Περίληψη

Σκοπός της διπλωματικής εργασίας είναι η εκμάθηση και η δημιουργία συγχρόνων συστημάτων αυτόματου ελέγχου. Θα παρουσιασθεί η δημιουργία ενός προγραμματιζόμενου λογικού ελεγκτή (PLC) χαμηλού κόστους, ο οποίος θα μπορεί να χρησιμοποιηθεί σε βιομηχανικά και οικιακά συστήματα αυτοματισμού. Ο συγκεκριμένος λογικός ελεγκτής θα υποστηρίζει την εκτέλεση προγραμμάτων από όλες τις γνωστές έως τώρα γλώσσες προγραμματισμού λογικών ελεγκτών (PLC) όπως Ladder, Function Block, Instruction List κ.α.

Για τον σκοπό αυτό θα χρησιμοποιήσουμε μια υπολογιστική μονάδα χαμηλού κόστους αλλά μεγάλης υπολογιστικής ισχύς, τον λεγόμενο υπολογιστή πλακέτας Raspberry Pi. Στη μονάδα αυτή θα εγκατασταθεί το λογισμικό ανοικτού κώδικα προσομοίωσης, προγραμματιζόμενου λογικού ελεγκτή (PLC) OpenPLC, το οποίο βασίζεται στο πρότυπο IEC 61131-3.

Επίσης θα παρουσιασθεί ο τρόπος επικοινωνίας OpenPLC με εξωτερικές μονάδες βασισμένες σε διάφορους μικροελεγκτές, όπως ο ESP8266, χρησιμοποιώντας το πρωτόκολλο επικοινωνίας Modbus TCP.

Συγχρόνως θα αναφερθούμε στη διασύνδεση των παραπάνω με το λογισμικό ScadaBR, ένα ανοιχτού κώδικα σύστημα SCADA (Supervisory Control And Data Acquisition), που έχει σκοπό την επεξεργασία και την οπτικοποίηση δεδομένων από βιομηχανικές ή οικιακές διεργασίες. Το εν λόγω σύστημα προσφέρει λειτουργίες, όπως η παρακολούθηση σε πραγματικό χρόνο η αποθήκευση ιστορικών δεδομένων και συμβάντων. Θα παρουσιαστεί το σύστημα ScadaBR που είναι βασισμένο σε γραφικό περιβάλλον με δυνατότητα για απομακρυσμένη πρόσβαση. Διαθέτει ενσωματωμένο διακομιστή ιστού WEB Server για το σκοπό αυτό. Θα αναφερθούμε τα οφέλη από τη χρήση του συστήματος ScadaBR το οποίο είναι πλήρως παραμετροποιήσιμο και μπορεί εύκολα να διασυνδεθεί με οποιοδήποτε σύστημα αυτοματισμού βασισμένο σε προγραμματιζόμενο λογικό εκλεκτή (PLC).

Λέξεις – κλειδιά

Προγραμματιζόμενος Λογικός Ελεγκτής, Λογισμικό ανοικτού κώδικα OpenPLC, OpenPLC Project, Modbus, IEC 61131-3, OpenPLC Runtime, OpenPLC Editor, OpenPLC σε Raspberry Pi, OpenPLC σε ESP8266.

Abstract

The purpose of this thesis is how to learn and create modern automatic control systems. We will present the creation of a low-cost programmable logic controller (PLC) that can be used in industrial and home automation systems. The specific logic controller will support the execution of programs from every logic controller (PLC) programming languages, which are known until now, such as Ladder, FunctionBlock, InstructionList etc.

For this purpose we will use a low-cost but powerful computing unit, the so-called Raspberry Pi board computer. The OpenPLC programmable logic controller (PLC) open source simulation software, which is based on the IEC 61131-3 standard, will be installed on this unit.

It will also show how OpenPLC communicates with external modules based on various microcontrollers, such as ESP8266, using the Modbus TCP communication protocol.

At the same time we will refer to the interface of all the above with the software ScadaBR., an open source SCADA system (Supervisory Control And Data Acquisition), which aims to process and visualize data from industrial or domestic processes.

The above system offers functions such as real-time monitoring and the storage of historical data and events. The ScadaBR system will be presented, a WEB-based interface for remote access, which has a built-in WEB Server web server for this purpose. Finally, we will refer to the benefits of using the ScadaBR system which is highly configurable and can be easily interfaced with any automation system based on a programmable logic controller (PLC).

Keywords

PLC, OpenPLC, OpenPLC Project, Modbus, IEC 61131-3, OpenPLC Runtime, OpenPLC Editor, OpenPLC with Raspberry Pi, OpenPLC with ESP8266

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος Εικόνων	16
Σκοπός και στόχοι	20
Καινοτομία.....	20
Δομή.....	20
ΚΕΦΑΛΑΙΟ 1.....	22
Εισαγωγή στα συστήματα προγραμματιζόμενων λογικών ελεγκτών PLC.....	22
1.1 Ιστορική Αναδρομή.	22
1.2 Τι είναι το PLC.....	23
1.3 Πλεονεκτήματα των PLC	24
1.4 Δομή ενός PLC.....	26
1.4.1 Πλαίσιο τοποθέτησης μονάδων.....	27
1.4.2 Μονάδα τροφοδοσίας.....	27
1.4.3 Κεντρική Μονάδα Επεξεργασίας	27
1.4.4 Μονάδες εισόδων - εξόδων.....	28
1.4.5 Οι προγραμματιζόμενοι λογικοί ελεγκτές της αγοράς.....	30
1.5 Αρχή λειτουργίας προγραμματιζόμενου λογικού ελεγκτή PLC.....	31
1.6 Προγραμματισμός PLC.....	33
1.7 Γλώσσες προγραμματισμού των προγραμματιζόμενων λογικών ελεγκτών	33
1.7.1 Γλώσσα διαγραμμάτων σκάλας LD	33
1.7.2 Γλώσσα λίστα εντολών ST.....	33
1.7.3 Γλώσσα λογικών γραφικών ή λογικού διαγράμματος FBD	33
1.8 Πρωτόκολλο επικοινωνίας Modbus	34
1.9 IEC 61131-3.....	37
1.10 OpenPLC Project.....	40
1.11 Οργανισμός PLCopen	43
ΚΕΦΑΛΑΙΟ 2.....	44
Εγκατάσταση Raspberry Pi 4 και OpenPLC-Project Runtime.....	44
2.1 Περιγραφή υλικού Raspberry Pi.....	44

2.2	Εγκατάσταση λειτουργικού συστήματος στο Raspberry Pi	46
2.3	Ενεργοποίηση VNC/SSH στο Raspberry Pi.	52
2.4	Εγκατάσταση VNC Viewer σε Η/Υ	52
2.5	Εγκατάσταση WinSCP σε Η/Υ	56
2.6	Εγκατάσταση OpenPLC-Project σε Raspberry Pi	59
	ΚΕΦΑΛΑΙΟ 3	65
	Εγκατάσταση και παρουσίαση του επεξεργαστή OpenPLC Editor	65
3.1	Εγκατάσταση OpenPLC Editor	65
3.2	Περιγραφή του OpenPLC Editor	67
3.3	Γλώσσες προγραμματισμού σε OpenPLC Editor	73
3.3.1	Διάγραμμα Σκάλας (LD)	73
3.3.2	Δομημένο Κείμενο (ST).....	73
3.3.3	Μπλοκ Λειτουργιών (FBD).....	74
3.3.4	Διαγράμματα Διαδοχικών Συναρτήσεων (SFC).....	74
3.3.5	Λίστα Οδηγιών (IL)	75
3.4	Μεταγλώττιση Προσομοίωση Φόρτωση προγράμματος στον OpenPLC	75
3.4.1	Προσομοίωση.	76
3.4.2	Φόρτωση προγράμματος στο OpenPLC Runtime.	78
	ΚΕΦΑΛΑΙΟ 4	80
4.1	Γλώσσα Ladder Diagram (LD)	80
4.2	Ανοικτή Επαφή	82
4.3	Κλειστή Επαφή	82
4.4	Επαφή ανερχομένου παλμού	82
4.5	Επαφή κατερχομένου παλμού	82
4.6	Πηνίο	83
4.7	Ανάστροφο Πηνίο	83
4.8	Πηνίο Αυτομανδάλωσης	83
4.9	Πηνίο Απομανδάλωσης	83
4.10	Πηνίο Απομανδάλωσης	84
4.11	Μπλοκ Αριθμητικών & Λογικών πράξεων	84

4.12	Μπλοκ Χρονικού	84
4.13	Μπλοκ Απαριθμητικού	85
4.14	Μπλοκ Τυμπανοειδούς Ελεγκτή	85
4.15	Μπλοκ Βηματικού Απαριθμητή	86
4.16	Μόνιμες Μεταβλητές	86
4.17	Εσωτερικά bits	86
	ΚΕΦΑΛΑΙΟ 5	88
5.1	Σύνδεση NodeMCU ESP8266 σε OpenPLC με χρήση Modbus TCP	88
5.2	Περιγραφή υλικού τύπου ESP8266	88
5.3	Εγκατάσταση Modbus TCP σε NodeMCU ESP8266	89
5.4	Σύνδεση NodeMCU ESP8266 με OpenPLC	92
	ΚΕΦΑΛΑΙΟ 6	95
	Εγκατάσταση του λογισμικού HMI ScadaBR για οπτικοποίηση προγραμμάτων PLC	95
6.1	Οπτικοποίηση λειτουργίας OpenPLC με ScadaBR	95
6.2	Τι σημαίνει ο όρος SCADA	95
6.3	Στοιχεία ενός συστήματος SCADA	95
6.4	Το ScadaBR	96
6.5	Εγκατάσταση ScadaBR	97
6.6	Ρύθμιση και χρήση του ScadaBR	100
6.6.1	Εισαγωγή πηγών δεδομένων Data sources	101
6.6.2	Εισαγωγή σημείων δεδομένων Data point.....	103
6.6.3	Οπτικοποίηση των στοιχείων πηγής	104
	Συμπεράσματα	108
	Βιβλιογραφία	109
	Διαδικτυακές Πηγές	110

Κατάλογος Εικόνων

Εικόνα 1.1 : Δομή modular PLC.	26
Εικόνα 1.2 : Compact PLC Siemens Logo	30
Εικόνα 1.3 : Modular PLC Siemens Simatic S7-1200	30
Εικόνα 1.4 : Βήματα λειτουργίας PLC.	31
Εικόνα 1.5 : Κύκλος λειτουργίας PLC.	32
Εικόνα 1.6 : Πεδίο Modbus.	34
Εικόνα 1.7 : Τοπολογίες δικτύου Modbus.	35
Εικόνα 1.8 : Πλαίσιο Modbus χρησιμοποιώντας Modbus TCP	36
Εικόνα 1.9 : Δομή το προτύπου IEC 61131-3.	38
Εικόνα 1.10 : Raspberry Pi PLC Unipi 1.1.	42
Εικόνα 1.11 : Επισκόπηση OpenPLC.	42
Εικόνα 1.12 : Η Δομή Οργανισμού PLCopen.	43
Εικόνα 2.1 : Raspberry Pi 4.	44
Εικόνα 2.2 : Raspberry Pi Zero 2W.	45
Εικόνα 2.3 : OpenPLC Raspberry Pi GPIO map.	46
Εικόνα 2.4 : Εκκίνηση του λογισμικού εγκατάστασης Raspberry Pi Imager.	46
Εικόνα 2.5 : Τέλος εγκατάστασης του λογισμικού Raspberry Pi Imager.	47
Εικόνα 2.6 : Επιλογή του λειτουργικού συστήματος.	47
Εικόνα 2.7 : Προτεινόμενα λειτουργικά συστήματα για το Raspberry Pi.	47
Εικόνα 2.8 : Διαθέσιμα Storage Device.	48
Εικόνα 2.9 : Επιλογή επιπλέον ρυθμίσεων.	48
Εικόνα 2.10 : Ορισμός hostname και ενεργοποίηση απομακρυσμένης πρόσβασης.	48
Εικόνα 2.11 : Ορισμός ονόματος χρήστη και κωδικού πρόσβασης και	49
Εικόνα 2.12 : Ενημέρωση στοιχείων σύνδεσης σε Wifi.	49
Εικόνα 2.13 : Επιλογή Time zone και keyboard layout.	49
Εικόνα 2.14 : Εκκίνηση της εγκατάστασης.	50
Εικόνα 2.15 : Προετοιμασία εγγραφής του λειτουργικού συστήματος στην microSD card.	50
Εικόνα 2.16: Εγγραφή του λειτουργικού συστήματος στην microSD card.	50
Εικόνα 2.17 : Έλεγχος ορθής εγγραφής του λειτουργικού συστήματος στην microSD card.	51
Εικόνα 2.18 : Ενεργοποίηση VNC/SSH.	52
Εικόνα 2.19 : Εγκατάσταση λογισμικού VNC Viewer.	53
Εικόνα 2.20 : Εγκατάσταση λογισμικού VNC Viewer.	53
Εικόνα 2.21 : Εγκατάσταση λογισμικού VNC Viewer.	53
Εικόνα 2.22 : Εγκατάσταση λογισμικού VNC Viewer.	54
Εικόνα 2.23 : Εγκατάσταση λογισμικού VNC Viewer.	54
Εικόνα 2.24 : VNC Viewer.	54

Εικόνα 2.25 : Διαμόρφωση VNC Viewer.	55
Εικόνα 2.26 : Εισαγωγή στοιχείων πρόσβασης.	55
Εικόνα 2.27 : Επιφάνια εργασίας Raspberry Pi OS.	56
Εικόνα 2.28 : Εγκατάσταση WinSCP.	56
Εικόνα 2.29 : Εγκατάσταση WinSCP.	57
Εικόνα 2.30 : Εγκατάσταση WinSCP.	57
Εικόνα 2.31 : Εγκατάσταση WinSCP.	57
Εικόνα 2.32 : Εγκατάσταση WinSCP.	58
Εικόνα 2.33 : Εγκατάσταση WinSCP.	58
Εικόνα 2.34 : Παράθυρο σύνδεσης με το Raspberry Pi.	58
Εικόνα 2.35 : Σύνδεση Η/Υ με Raspberry Pi με το WinSCP.	59
Εικόνα 2.36 : Εγκατάσταση Git.	60
Εικόνα 2.37 : Εγκατάσταση του OpenPLC Runtime.	60
Εικόνα 2.38 : Εγκατάσταση του OpenPLC Runtime.	61
Εικόνα 2.39 : Εγκατάσταση βιβλιοθήκης WiringPI.	61
Εικόνα 2.40 : Εγκατάσταση βιβλιοθήκης WiringPI.	62
Εικόνα 2.41 : Αρχική σελίδα σύνδεσης OpenPLC.	62
Εικόνα 2.42 : Πίνακας ελέγχου χρόνου εκτέλεσης OpenPLC.	63
Εικόνα 2.43 : Επιλογή υλικού για την εκτέλεση του Open PLC Runtime.	63
Εικόνα 2.44 : Ρυθμίσεις επικοινωνίας OpenPLC Runtime.	64
Εικόνα 3.1 : Ο ιστοτοπος του OpenPLC Project.	65
Εικόνα 3.2 : Εγκατάσταση OpenPLC Editor.	66
Εικόνα 3.3 : Εγκατάσταση OpenPLC Editor.	66
Εικόνα 3.4 : Εγκατάσταση OpenPLC Editor.	66
Εικόνα 3.5 : Αρχική σελίδα OpenPLC Editor.	67
Εικόνα 3.6 : OpenPLC Editor με πρόγραμμα σε γλώσσα Ladder Logic.	67
Εικόνα 3.7 : Αρχική οθόνη του OpenPLC Editor.	68
Εικόνα 3.8 : Μενού File.	68
Εικόνα 3.9 : Μενού Edit.	69
Εικόνα 3.10 : Μενού Display.	69
Εικόνα 3.11 : Μενού Help.	69
Εικόνα 3.12 : Μενού Library.	69
Εικόνα 3.13 : Δομή Project.	70
Εικόνα 3.14 : Ιδιότητες προγράμματος.	70
Εικόνα 3.15 : Χώρος εργασίας / δημιουργίας προγράμματος.	70
Εικόνα 3.16 : Καρτέλα ρυθμίσεων του προγράμματος.	71
Εικόνα 3.17 : Γραμμή εικονιδίων OpenPLC Editor.	71
Εικόνα 3.18 : Γραμμή εικονιδίων για γλώσσα Ladder Logic.	72

Εικόνα 3.19 : Γραμμή εικονιδίων για γλώσσα Structured Text.	72
Εικόνα 3.20 : Γραμμή εικονιδίων για γλώσσα Function Block Diagram.	72
Εικόνα 3.21 : Γραμμή εικονιδίων για γλώσσα Sequential Function Chart.	72
Εικόνα 3.22 : Γραμμή εικονιδίων για γλώσσα Instruction List.	73
Εικόνα 3.23 : Παράδειγμα προγράμματος Διαγράμματος Σκάλας (LD).	73
Εικόνα 3.24 : Παράδειγμα προγράμματος Δομημένου Κειμένου (ST).	73
Εικόνα 3.25 : Παράδειγμα προγράμματος Μπλοκ Λειτουργιών (FBD).	74
Εικόνα 3.26 : Παράδειγμα προγράμματος Διαγραμμάτων Διαδοχικών Συναρτήσεων (SFC).	74
Εικόνα 3.27 : Παράδειγμα προγράμματος Λίστας Οδηγιών (IL).	75
Εικόνα 3.28 : Δημιουργία «εκτελέσιμου» προγράμματος για το OpenPLC Runtime.	75
Εικόνα 3.29 : Αποθήκευση του μεταγλωττισμένου αρχείου.	76
Εικόνα 3.30 : Παράθυρο προσομοίωσης OpenPLC Editor.	76
Εικόνα 3.31 : Προσομοίωση προγράμματος.	77
Εικόνα 3.32 : Προσομοίωση προγράμματος	77
Εικόνα 3.33 : Φόρτωση προγράμματος στο OpenPLC Runtime.	78
Εικόνα 3.34 : Ονοματοδοσία προγράμματος στο OpenPLC Runtime.	78
Εικόνα 3.35 : Μεταγλώττιση προγράμματος στο OpenPLC Runtime.	79
Εικόνα 4.1 : Γραμμικό ηλεκτρολογικό σχέδιο (α) και το ισοδύναμο σε γλώσσα Ladder Diagram (β).	80
Εικόνα 4.2 : Η εκτέλεση των εντολών από το PLC, είναι όπως διαβάζουμε ενα βιβλίο.	81
Εικόνα 5.1 : NodeMCU ESP8266.	88
Εικόνα 5.2 : Λήψη του αρχείου Blank Project.	90
Εικόνα 5.3 : OpenPLC Editor	90
Εικόνα 5.4 : OpenPLC Editor compile Project.	90
Εικόνα 5.5 : Αποστολή του λογισμικού OpenPLC Modbus στο NodeMCU ESP8266.	91
Εικόνα 5.6 : Καρτέλα διαχειρίσεις Slave μονάδων στο OpenPLC Runtime στο Raspberry Pi.	92
Εικόνα 5.7 : Προσθήκη νέας μονάδας Modbus/TCP στο OpenPLC Runtime στο Raspberry Pi.	92
Εικόνα 5.8 : Αποθήκευση νέας μονάδας Modbus TCP στο OpenPLC Runtime στο Raspberry Pi.	93
Εικόνα 5.9 : Πίνακας με Slave μονάδες συνδεδεμένες με το OpenPLC Runtime στο Raspberry Pi.	93
Εικόνα 5.10 : NodeMCU ESP8266 PinOut	94
Εικόνα 6.1 : Εγκατάσταση λογισμικού ScadaBR	97
Εικόνα 6.2 : Εγκατάσταση λογισμικού ScadaBR.	98
Εικόνα 6.3 : Εγκατάσταση λογισμικού ScadaBR.	98
Εικόνα 6.4 : Εγκατάσταση λογισμικού ScadaBR.	98
Εικόνα 6.5 : Εγκατάσταση λογισμικού ScadaBR.	99
Εικόνα 6.6 : Εγκατάσταση λογισμικού ScadaBR.	99
Εικόνα 6.7 : ScadaBR – Apache Tomcat Properties	99
Εικόνα 6.8 : Σελίδα σύνδεσης στο ScadaBR.	100
Εικόνα 6.9 : Σελίδα καλωσορίσματος του ScadaBR.	100

Εικόνα 6.10 : Αρχική (default) σελίδα σύνδεσης στο ScadaBR.	101
Εικόνα 6.11 : Data sources icon.	101
Εικόνα 6.12 : Επιλογή τύπου εισαγωγής δεδομένων στο ScadaBR.	102
Εικόνα 6.13 : Σύνδεση OpenPLC Raspberry Pi με το ScadaBR.	102
Εικόνα 6.14 : Εισαγωγή Data point για την λήψη / αποστολή δεδομένων στο PLC	103
Εικόνα 6.15 : Σελίδα για την δημιουργία Graphic views στο ScadaBR.	104
Εικόνα 6.16 : Παράθυρο ρυθμίσεις σελίδας οπτικοποίησης.	105
Εικόνα 6.17 : Διαθέσιμοι τύποι αντικειμένων για οπτικοποίηση στο ScadaBR.	105
Εικόνα 6.18 : Εισαγωγή αντικειμένου οπτικοποίησης Button στο ScadaBR.	106
Εικόνα 6.19 : Αντιστοίχιση αντικειμένου σε λειτουργία.	106
Εικόνα 6.20 : Διαμόρφωση αντικειμένου.	106
Εικόνα 6.21 : Τελική μορφή αντικειμένου σύμφωνα με τις παραπάνω ρυθμίσεις.	106
Εικόνα 6.22 : Επιλογή διαγραφής αντικειμένου.	106
Εικόνα 6.23 : Η σελίδα οπτικοποίησης του ScadaBR.	107

Σκοπός και στόχοι

Ο σκοπός της εργασίας είναι η παρουσίαση και η ανάπτυξη ενός συστήματος προγραμματιζόμενου λογικού ελεγκτή (PLC), βασισμένο στο λογισμικό ανοιχτού κώδικα OpenPLC. Το σύστημα αυτό θα είναι προσαρμόσιμο σε μια μεγάλη γκάμα εφαρμογών βιομηχανικού και οικιακού αυτοματισμού.

Ο κύριος στόχος της εργασίας είναι η ανάδειξη της δυνατότητας δημιουργίας εναλλακτικών συστημάτων προγραμματιζόμενου λογικού ελεγκτή (PLC), τα οποία χαρακτηρίζονται από το χαμηλό κόστος κατασκευής, την υψηλή αξιοπιστία, τις μεγάλες δυνατότητες επεκτασιμότητας και παραμετροποιήσεις τους.

Αυτό επιτυγχάνεται μέσα από την ανάλυση των παρακάτω:

- Παρουσίαση των υφιστάμενων συστημάτων PLC.
- Παρουσίαση του συστήματος OpenPLC, του πρωτοτύπου IEC 61131-3 καθώς και του βιομηχανικού πρωτοκόλλου αυτοματισμού Modbus.
- Εγκατάσταση, προγραμματισμός και χειρισμός ενός προγραμματιζόμενου λογικού ελεγκτή βασισμένο στο OpenPLC με χρήση υλικού Raspberry Pi.
- Εγκατάσταση, προγραμματισμός και χειρισμός ενός προγραμματιζόμενου λογικού ελεγκτή βασισμένο στο OpenPLC Modbus TCP με χρήση υλικού ESP8266.
- Εγκατάσταση, και παρουσίαση συστήματος οπτικοποίησης OpenPLC με χρήση λογισμικού ανοικτού κώδικα ScadaBR.

Καινοτομία

Το OpenPLC είναι ένα λογισμικό ανοιχτού κώδικα προγραμματιζόμενου λογικού ελεγκτή PLC που επιτρέπει στους χρήστες να προγραμματίζουν συστήματα βιομηχανικού και οικιακού ελέγχου χρησιμοποιώντας τυπικές γλώσσες προγραμματισμού PLC. Έχει σχεδιαστεί για να είναι ευέλικτο και φιλικό προς τον χρήστη και μπορεί να χρησιμοποιηθεί σε ποικίλες βιομηχανικές και οικιακές εφαρμογές. Η χρήση λογισμικού ανοιχτού κώδικα σε συστήματα βιομηχανικού και οικιακού ελέγχου αντιπροσωπεύει μια καινοτομία στον τομέα του αυτοματισμού, καθώς επιτρέπει μεγαλύτερη προσβασιμότητα, προσαρμογή και αποδοτικότητα σε σύγκριση με τα παραδοσιακά συστήματα PLC.

Δομή

- **Κεφάλαιο 1** : Εξετάζεται το θεωρητικό τμήμα της διπλωματικής εργασίας. Γίνεται ιστορική αναδρομή στους προγραμματιζόμενους λογικούς ελεγκτές PLC. Αναφέρονται τα μέρη από τα οποία αποτελείται ο προγραμματιζόμενος λογικός ελεγκτής PLC, η αρχή λειτουργίας και ο προγραμματισμός τους. Παρουσιάζονται οι δημοφιλέστερες γλώσσες προγραμματισμού για PLC, το πρωτόκολλο επικοινωνίας Modbus το οποίο χρησιμοποιείτε για επικοινωνία μεταξύ των PLC και PLC με συστήματα HMI (SCADA). Το πρότυπο ICE 61131-3 στο οποίο στηρίζεται η δημιουργία του OpenPLC Project.
- **Κεφάλαιο 2** : Εξετάζεται το υλικό (hardware) Raspberry Pi. Επίσης αναλύονται τα βήματα

εγκατάστασης του λειτουργικού συστήματος του Raspberry Pi OS (32 bit) σε μια κάρτα micro SD, τα βήματα εγκατάστασης των απαιτούμενων λογισμικών για την δυνατότητα απομακρυσμένης διαχείριση του Raspberry Pi, VNC Viewer και WinSCP σε H/Y, καθώς και τα βήματα εγκατάστασης του λογισμικού ανοικτού κώδικα OpenPLC Runtime στο Raspberry Pi. Αναφερόμαστε στον έλεγχο καλής εγκατάστασης και στη παραμετροποίηση του λογισμικού OpenPLC Runtime και πρώτη δοκιμή.

- **Κεφάλαιο 3** : Εξετάζεται η εγκατάσταση του λογισμικού OpenPLC Editor σε H/Y, το οποίο είναι απαραίτητο για την συγγραφή και μετάφραση των προγραμμάτων που θα ανεβαίνουν στο OpenPLC Runtime, καθώς και η αναλυτική παρουσίαση του. Γίνεται συνοπτική παρουσίαση των υποστηριζόμενων γλωσσών προγραμματισμού για προγραμματιζόμενους λογικούς ελεγκτές PLC, όπως Διάγραμμα Σκάλας – Ladder Diagram, Δομημένο Κείμενο - Structured Text, Μπλοκ Λειτουργιών - Function Block Diagram, Διαγράμματα Διαδοχικών Συναρτήσεων - Sequential Function Chart, Λίστα Οδηγιών - Instruction List.
- **Κεφάλαιο 4** : Γίνεται αναλυτική παρουσίαση της γλώσσας προγραμματισμού Διάγραμμα Σκάλας – Ladder Diagram.
- **Κεφάλαιο 5** : Εξετάζεται η δυνατότητα επικοινωνίας ενός προγραμματισμένου λογικού ελεγκτή PLC βασισμένου στην χρήση του λογισμικού OpenPLC Project με εξωτερικές μονάδες. Θα χρησιμοποιηθεί το δημοφιλές πρωτόκολλο Modbus TCP σε συνδυασμό με τον μικροελεγκτή ESP8266 μέσω σύνδεσης WLAN.
- **Κεφάλαιο 6** : Εξετάζεται το σύστημα οπτικοποίησης HMI - SCADA δεδομένων από συστήματα αυτοματισμού με χρήση του λογισμικού ανοικτού κώδικα ScadaBR, το οποίο επιτρέπει να απεικονίσουμε τις διεργασίες που εκτελούνται σε έναν προγραμματιζόμενο λογικό ελεγκτή PLC μέσω του Modbus, σε έναν H/Y.

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή στα συστήματα προγραμματιζόμενων λογικών ελεγκτών PLC

1.1 Ιστορική Αναδρομή

Ήδη από τη δεκαετία του 1960 οι μηχανικοί άρχισαν να σκέφτονται τρόπους για το πως θα εκμεταλλευόταν τις δυνατότητες των υπολογιστών για βιομηχανικές εφαρμογές. Από τις πρώτες εφαρμογές των υπολογιστών στη βιομηχανία ήταν οι αυτόματες εργαλειομηχανές όπως τόννοι, φρέζες κ.α., οι οποίες μέχρι τότε χρησιμοποιούσαν κυρίως μηχανολογικούς και λιγότερο ηλεκτρολογικούς αυτοματισμούς. Η επιτυχία αυτή οδήγησε τους μηχανικούς να σκέφτονται την αντικατάσταση των αυτοματισμών με ηλεκτρονόμους από έναν υπολογιστή. Μέχρι όμως την δεκαετία του 1980 αυτό δεν ήταν δυνατό, λόγω του ότι οι υπολογιστές ήταν πολύ ακριβοί και δύσκολη στην χρήση τους.

Το έτος 1975 γίνεται η επανάσταση της πληροφορικής με την κατασκευή του πρώτου μικροϋπολογιστή. Πολλά από όσα σήμερα θεωρούμε δεδομένα στους υπολογιστές δημιουργήθηκαν μετά το 1980. Η ανάπτυξη της τεχνολογίας άλλαξε την πορεία σε όλους τους τομείς της ζωής μας.

Η βιομηχανία μέχρι και η δεκαετία του 1980 χρησιμοποιούσε ελάχιστα τα ηλεκτρονικά κυκλώματα. Οι αυτοματισμοί ήταν με ηλεκτρονόμους σε ποσοστό πάνω από 90%. Τα ηλεκτρονικά χρησιμοποιούνταν κυρίως σε ευφυείς εργασίες. Οι ηλεκτρονικές πλακέτες τοποθετούνταν μέσα στους πίνακες των ηλεκτρονόμων.

Από της αρχές δεκαετίας του 1980 οι εταιρείες παραγωγής ηλεκτρολογικού υλικού αρχίζουν να παρουσιάζουν στους μηχανικούς και τεχνικούς της βιομηχανίας ένα νέο προϊόν για τον αυτοματισμό, το οποίο ονόμασαν PLC, Programmable Logic Controller - Προγραμματιζόμενος Λογικός Ελεγκτής. Για να μην προκαλέσουν φόβο στους τεχνικούς της βιομηχανίας για αυτό το νέο προϊόν, τα πρώτα χρόνια κυκλοφορία των PLC οι εταιρείες παραγωγής τους δεν χρησιμοποίησαν την πλήρη ονομασία του προϊόντος, αλλά το ανέφεραν απλά ως PLC.

Το PLC είναι ένας μικροϋπολογιστής, που χρησιμοποιείται στη λειτουργία των αυτοματισμών. Τα συστήματα PLC σχεδιάστηκαν προς αντικατάσταση του κλασικού πίνακα αυτοματισμού με ηλεκτρονόμους τα γνωστά ρελέ. Είναι εύκολα αντιληπτό ότι βρισκόμαστε πλέον σε μια τεράστια αλλαγή σε σχέση με τον μέχρι τότε τρόπο που δούλευε η βιομηχανία, δηλαδή έπρεπε να πραγματοποιηθεί η μετάβαση από τους ηλεκτρονόμους στους υπολογιστές. Οι εταιρίες σχεδιασμού και κατασκευής PLC προσάρμοσαν τον τρόπο χρήσης τους, στον τρόπο που δούλευε μέχρι τότε η βιομηχανία. Επίσης απέφυγαν να αναφέρουν λέξεις που θα δημιουργούσαν φόβο στο τεχνικό προσωπικό της βιομηχανίας, όπως για παράδειγμα υπολογιστής, προγραμματισμός κ.α. Ακόμη και το όνομα του νέου προϊόντος απέφευγαν να το χρησιμοποιήσουν ολοκληρωμένο και προτιμούσαν να το αναφέρουν ως PLC. Επίσης κατέβαλαν προσπάθειες να μην αλλάξουν τον μέχρι τότε τρόπο εργασίας των τεχνικών στον τομέα του αυτοματισμού, δηλαδή δεν άλλαξαν τίποτα από τα μέχρι τότε βήματα που ακολουθούσαν για τον σχεδιασμό ενός συστήματος αυτοματισμού. Ανέφεραν απλά στους τεχνικούς, ότι αυτό το σχέδιο αντί να το δώσετε στον ηλεκτρολόγο για να το υλοποιήσει, θα το κάνετε

εσείς με τον τρόπο που θα σας δείξουμε. Στην πραγματικότητα όμως τους μάθαιναν προγραμματισμό.

Με αποτέλεσμα την επιτυχή και ομαλή είσοδο του PLC στην βιομηχανία. Ο κλασικός αυτοματισμός με ηλεκτρονόμους σήμερα καλύπτει ένα πολύ μικρό τμήμα των αυτοματισμών στην βιομηχανία, ενώ όλες οι καινούργιες εγκαταστάσεις χρησιμοποιούν PLC.

Τα PLC τα τελευταία χρόνια έχουν εξελιχτεί και αναβαθμιστεί σε σύγκριση με τα μοντέλα που παρουσιάστηκαν αρχικά. Το τεχνικό προσωπικό της βιομηχανίας έχει εκπαιδευτεί στον χειρισμό και τον προγραμματισμό τους. Σήμερα ένας ηλεκτρολόγος πρέπει να έχει στοιχειώδεις γνώσεις στα ηλεκτρονικά και στους ηλεκτρονικούς υπολογιστές, διαφορετικά θα είναι πολύ δύσκολο να καταλάβει ακόμη και το πιο απλό εγχειρίδιο ενός PLC.

Η υιοθέτηση των PLC παρέχει πάρα πολλά πλεονεκτήματα σε σύγκριση με τον κλασικό αυτοματισμό. Η αποδοχή των PLC όμως δεν οφείλεται μόνο στα πλεονεκτήματα που παρέχουν στον τελικό χρήστη.

Η χρήση των PLC σε σύγκριση με τον κλασικό αυτοματισμό ωφελεί πρώτιστα τις εταιρείες που κατασκευάζουν υλικά αυτοματισμού. Αν κάποιος υπολογίσει το κόστος που θα είχε για την παραγωγή ηλεκτρολογικού εξοπλισμού και την παραγωγή ενός τεράστιου αριθμού ηλεκτρονόμων, χρονικών και απαριθμητών, συγκριτικά με το κόστος των υλικών αυτοματισμού που αφορά τη χρήση τους σε αυτοματισμό με PLC όπως μονάδες εισόδου και εξόδου, θα διαπιστώσουμε ότι το κέρδος τους είναι αρκετά μεγάλο.

Θα μπορούσαμε να πούμε ότι, ενώ σε όλους τους τομείς της παραγωγής περάσαμε από τις ηλεκτρολογικές συσκευές, στις συσκευές με λυχνίες, μετά στις συσκευές με τρανζίστορ και τέλος φθάσαμε στους μικροϋπολογιστές, και στις ψηφιακές συσκευές, στον τομέα των αυτοματισμών περάσαμε σχεδόν κατευθείαν από τον κλασικό ηλεκτρολογικό αυτοματισμό στους ψηφιακούς αυτοματισμούς με PLC.

1.2 Τι είναι το PLC

Στον κλασικό αυτοματισμό με ηλεκτρονόμους τα στάδια εργασίας από το σχεδιασμό και την κατασκευή ενός αυτοματισμού μέχρι το σημείο της πλήρους λειτουργίας είναι τα εξής:

1. Περιγραφή του αυτοματισμού.
2. Ανάπτυξη του λειτουργικού σχεδίου του αυτοματισμού.
3. Ανάπτυξη του σχεδίου καλωδίωσης του πίνακα.
4. Κατασκευή του πίνακα της εγκατάστασης.
5. Εγκατάσταση και σύνδεση στους ακροδέκτες του πίνακα των αισθητήρων που δίνουν τις εντολές και των αποδεκτών που εκτελούν τις εργασίες.
6. Δοκιμή λειτουργίας της εγκατάστασης.
7. Πλήρης λειτουργία του αυτοματισμού.

Ο Προγραμματιζόμενος Λογικός Ελεγκτής PLC είναι μια συσκευή, η οποία θα αντικαταστήσει τον

ηλεκτρολογικό πίνακα του κλασικού αυτοματισμού, δηλαδή τον πίνακα με τους βοηθητικούς ηλεκτρονόμους, χρονικά, απαριθμητές, και όλα τα στοιχεία του κλασικού αυτοματισμού. Αντί δηλαδή να δημιουργήσουν έναν πίνακα αυτοματισμού με πολύπλοκες συνδεσμολογίες και χρήση υλικών αυτοματισμού που χρησιμοποιούσαν για τον κλασικό αυτοματισμό, θα κατασκεύαζαν έναν πίνακα αυτοματισμού που περιέχει μόνο το PLC και τις απαραίτητες μονάδες εισόδου – εξόδου. Με την χρησιμοποίηση του PLC ο προγραμματισμός του αυτοματισμού γίνονταν με την χρήση ειδικής συσκευής που ονομαζόταν προγραμματιστής ή με την χρήση ηλεκτρονικού υπολογιστή και του ειδικού λογισμικού προγραμματισμού που παρείχε ο κατασκευαστής του PLC.

Τα στάδια του σχεδιασμού και της κατασκευής ενός αυτοματισμού με προγραμματιζόμενη λογική είναι τα εξής:

1. Περιγραφή του αυτοματισμού.
2. Ανάπτυξη του σχεδίου καλωδίωσης του πίνακα.
3. Κατασκευή του πίνακα της εγκατάστασης.
4. Ανάπτυξη του προγράμματος λειτουργίας του αυτοματισμού και εισαγωγή του προγράμματος στο PLC μέσω του προγραμματιστή.
5. Τοποθέτηση στους ακροδέκτες του πίνακα των συσκευών εισόδου που παρέχουν πληροφορίες και των συσκευών εξόδου που επιτελούν τις εργασίες.
6. Δοκιμή λειτουργίας της εγκατάστασης.
7. Πλήρης λειτουργία του αυτοματισμού.

Βλέπουμε τα στάδια, τα οποία αλλάζουν στις εργασίες του σχεδιασμού και της κατασκευής ενός συστήματος αυτοματισμού, όταν χρησιμοποιούμε PLC, είναι τα στάδια 2, 3 και 4. Άρα αντί να έχουμε την κατασκευή ενός κλασικού πίνακα αυτοματισμού, με πολύπλοκες καλωδιακές συνδέσεις και μεγάλο πλήθος ηλεκτρολογικού υλικού, έχουμε την κατασκευή ενός πίνακα αυτοματισμού μόνο με υλικά εισόδων και εξόδων, με απλές καλωδιακές συνδέσεις, καθώς και τον προγραμματισμό του PLC για τον επιθυμητό τρόπο εργασίας του συστήματος. Ο χρόνος, που απαιτείται για τον προγραμματισμό και την κατασκευή του πίνακα αυτοματισμού με PLC, είναι αρκετά μικρότερος σε σύγκριση με τον χρόνο, που απαιτείται για τη μελέτη και την κατασκευή του αντίστοιχου πίνακα κλασικού αυτοματισμού.

1.3 Πλεονεκτήματα των PLC

Τα πρώτα μεγάλα πλεονεκτήματα των PLC αφορούν τους κατασκευαστές εξοπλισμού αυτοματισμών και πινάκων αυτοματισμού τα οποία έχουν ήδη αναφερθεί:

1. Το κόστος υλοποίησης ενός πίνακα αυτοματισμού με PLC είναι σημαντικά μικρότερο από το κόστος κατασκευής ενός κλασικού πίνακα αυτοματισμού με μεγάλο πλήθος βοηθητικών ηλεκτρονόμων, χρονικών, απαριθμητών κ.α.
2. Ο χρόνος υλοποίησης του πίνακα αυτοματισμού με PLC είναι ελάχιστος σε σύγκριση με τον χρόνο υλοποίησης ενός κλασικού πίνακα αυτοματισμού.

Πολλά πλεονεκτήματα από την χρήση του PLC έχουν να κάνουν με τον τελικό χρήστη, δηλαδή τις

βιομηχανίες που χρησιμοποιούν τους αυτοματισμούς για την λειτουργία τους.

Τα σημαντικότερα πλεονεκτήματα είναι τα εξής:

- Τα PLC ελαχιστοποιούν το κόστος συντήρησης του πίνακα αυτοματισμού. Το κόστος αυτό αφορά, τη συχνότητα στην οποία εμφανίζονται βλάβες στο πίνακα αυτοματισμού, τον χρόνο εντοπισμού της βλάβης και τον χρόνο αποκατάστασης αυτής. Δηλαδή, όταν υπάρχει μια βλάβη στον πίνακα μιας εγκατάστασης κλασικού αυτοματισμού, υπάρχει καθυστέρηση στην παραγωγή της βιομηχανίας μέχρι να εντοπιστεί η βλάβη. Αφού εντοπιστεί το πρόβλημα, θα πρέπει να υπάρχει άμεσα διαθέσιμο το ανταλλακτικό στην αποθήκη, γιατί διαφορετικά θα υπάρξει σημαντική καθυστέρηση, αφού θα χρειαστεί να γίνει η σχετική παραγγελία και η προμήθεια. Στους πίνακες αυτοματισμού με PLC η συχνότητα εμφάνισης βλαβών είναι μικρή σε σχέση με τον κλασικό αυτοματισμό. Λόγο της απλούστερης κατασκευής του πίνακα αυτοματισμού ο χρόνος εντοπισμού και επιδιόρθωσης της βλάβης είναι ελάχιστος. Επίσης ο χρόνος εγγύησης των PLC είναι πολύ μεγάλος.
- Τα PLC μας παρέχουν ευελιξία στην τροποποίηση η και αλλαγή της λειτουργίας του αυτοματισμού. Δηλαδή, αν θέλουμε να κάνουμε μια αλλαγή – διόρθωση στον πίνακα αυτοματισμού, αυτή μπορεί να γίνει σε ελάχιστο χρόνο, είτε με ένα νέο πρόγραμμα που θα φορτωθεί στο PLC, είτε με την προσθήκη επιπλέον μονάδων εισόδων ή εξόδων. Ενώ σε έναν πίνακα κλασικού αυτοματισμού, αυτού του είδους οι αλλαγές είναι πολύ δύσκολες και χρονοβόρες.
- Ο αυτοματισμός με PLC επεκτείνεται πάρα πολύ εύκολα. Αυτό μπορεί να γίνει είτε αντικαθιστώντας το πρόγραμμα που έχει φορτωθεί στο PLC με ένα νέο, είτε με την προσθήκη επιπλέον μονάδων εισόδων και εξόδων. Η επέκταση σε πίνακες κλασικού αυτοματισμού είναι περίπλοκη και δύσκολη έως και μερικές φορές αδύνατη.
- Ο αυτοματισμός με PLC μας δίνει δυνατότητες να δημιουργούμε εύκολα, πολύπλοκες και έξυπνες επεξεργασίες, που για τον κλασικό αυτοματισμό είναι εξαιρετικά δύσκολο να σχεδιαστούν και να υλοποιηθούν.
- Σε μια εγκατάσταση, που χρησιμοποιεί αυτοματισμούς με PLC, σήμερα παρέχονται δυνατότητες σύνδεσης με κεντρικό ηλεκτρονικό υπολογιστή, καθώς είναι δυνατή η σύνδεση με άλλα τμήματα της βιομηχανίας όπως αποθήκη, λογιστήριο κ.α.
- Ο πίνακας αυτοματισμού με PLC καταλαμβάνει ελάχιστο χώρο σε σχέση με έναν αντίστοιχο πίνακα κλασικού αυτοματισμού.

Από τη χρήση των PLC έχουμε κατά κύριο λόγο πλεονεκτήματα. Αλλά υπάρχει όμως και ένα σημαντικό μειονέκτημα αυτό της μη επαρκούς ενημέρωσης των τεχνικών όλων των βαθμίδων. Ειδικά στην Ελλάδα, παρατηρείται δυσκολία στην αποδοχή και στην εφαρμογή των PLC.

Η τελική ερώτηση που προκύπτει είναι, Πόσο κοστίζουν τελικά τα PLC συμφέρει να χρησιμοποιούμε σε κάθε εγκατάσταση PLC; Η απάντηση είναι ότι οι τιμές μειώνονται και οι εταιρείες βγάζουν συνεχώς νέα μοντέλα που καλύπτουν μεγάλο εύρος εφαρμογών. Παρ' όλα αυτά δεν συμφέρει ακόμη

η χρήση του PLC, όταν έχουμε να υλοποιήσουμε εφαρμογές με κινητήρες άρα και πολλούς ηλεκτρονόμους ισχύος, ενώ το σύστημα αυτοματισμού είναι απλό αποτελούμενο από λίγους βοηθητικούς ηλεκτρονόμους, χρονικά και απαριθμητές.

1.4 Δομή ενός PLC

Στην αγορά υπάρχουν σήμερα πάρα πολλά μοντέλα PLC, κατασκευασμένα από διάφορες εταιρείες.

Γενικά, τα PLC μπορούμε να τα διακρίνουμε στα ακόλουθα μέρη:

- Την κεντρική μονάδα επεξεργασίας - Central Processing Unit, CPU, που αποτελεί τον εγκέφαλο του PLC.
- Την μονάδα τροφοδοσίας.
- Τις μονάδες εισόδων - εξόδων - I/O modules.

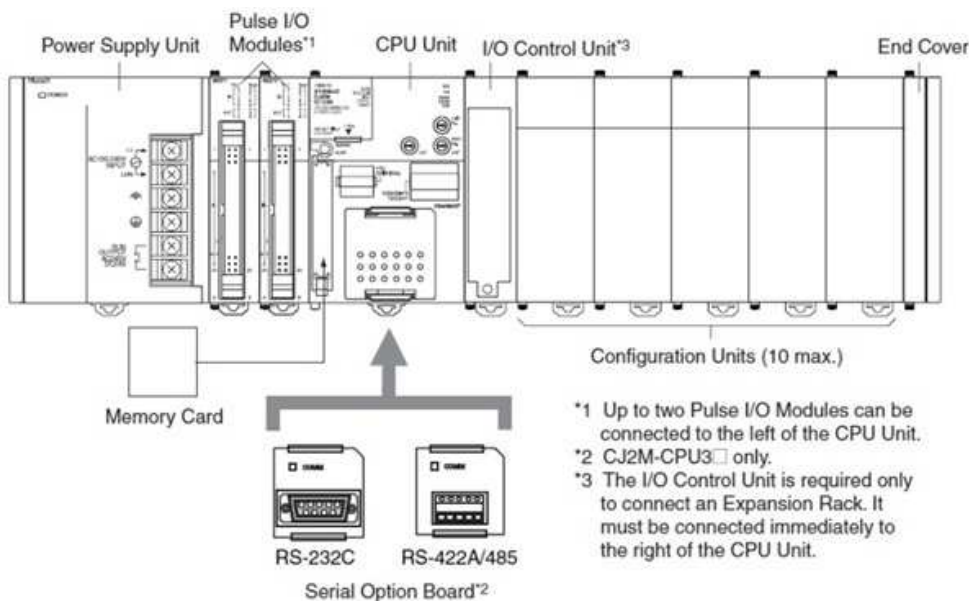
Το κύριο μέρος του PLC αποτελείται από την μονάδα τροφοδοσίας, τη κεντρική μονάδα επεξεργασίας, και τις μονάδες εισόδων - εξόδων.

Σε πολλά μοντέλα, κυρίως στα μικρά μοντέλα των εταιρειών, οι τρεις παραπάνω μονάδες βρίσκονται ενσωματωμένες σε μια συσκευή.

Σε ένα αρθρωτό PLC είναι ακόμη απαραίτητα:

- Το πλαίσιο ή τα πλαίσια για την τοποθέτηση των μονάδων και των τυχόν επεκτάσεών τους.
- Τη συσκευή προγραμματισμού του PLC.

Ο προγραμματιστής είναι μια ξεχωριστή συσκευή, η οποία χρησιμοποιείται για την εισαγωγή του προγράμματος στο PLC, την παρακολούθηση της ροής του προγράμματος αυτοματισμού μέσα από την οθόνη που διαθέτει. Με ένα μόνο προγραμματιστή μπορούμε να χειριζόμαστε όλες τις μονάδες PLC μιας αυτοματοποιημένης εγκατάστασης αρκεί να είναι της ίδιας εταιρείας.



Εικόνα 1.1 : Δομή modular PLC.

Παρακάτω θα δούμε αναλυτικά τις μονάδες από τις οποίες αποτελείται ένα PLC.

1.4.1 Πλαίσιο τοποθέτησης μονάδων

Όταν έχουμε ένα μεγάλο μεγέθους PLC, αυτό τοποθετείται σε ένα κεντρικό πλαίσιο. Στο πλαίσιο αυτό είναι ενσωματωμένο ένα σύστημα αγωγών μέσω των οποίων επικοινωνούν οι διάφορες μονάδες με την Κεντρική Μονάδα Επεξεργασίας του PLC.

Αν οι θέσεις του κεντρικού πλαισίου, που διατίθεται, δεν επαρκούν για να τοποθετηθούν οι μονάδες εισόδων και εξόδων που απαιτούνται για την συγκεκριμένη εφαρμογή, τότε χρησιμοποιούνται επιπλέον πλαίσια επέκτασης για την τοποθέτηση των πρόσθετων μονάδων. Κάθε πλαίσιο επέκτασης διασυνδέεται με το κεντρικό πλαίσιο ή με τα άλλα πλαίσια επέκτασης μέσω ειδικής μονάδας διασύνδεσης και καλωδίου. Ανάλογα με το τύπο του PLC κάθε εταιρεία διαθέτει το αντίστοιχο πλαίσιο.

1.4.2 Μονάδα τροφοδοσίας

Η μονάδα τροφοδοσίας ενός συστήματος PLC είναι υπεύθυνη τροφοδοσίας του με την κατάλληλη τάση και ένταση (ρεύμα). Δηλαδή δημιουργεί την απαιτούμενη τάση δικτύου που χρειάζεται για την τροφοδοσία τόσο της μονάδας όσο και των περιφερικών της. Όπως για παράδειγμα τις απαραίτητες εσωτερικές τάσεις, που απαιτούνται για την τροφοδοσία των ηλεκτρονικών στοιχείων της όπως τρανζίστορ, ολοκληρωμένα κυκλώματα κ.α. καθώς και την τροφοδοσία των μονωδών εισόδου - εξόδου. Οι τυπικές τάσεις των μονάδων τροφοδοσίας του PLC είναι συνήθως: DC 5V, DC 9V, DC 24V.

Σε ορισμένα μοντέλα, όταν το PLC δεν τροφοδοτείται από το δίκτυο, η μονάδα τροφοδοσίας διατηρεί το περιεχόμενο της μνήμης του PLC με την βοήθεια μπαταρίας που διαθέτει. Σε άλλα μοντέλα PLC η μπαταρία βρίσκεται στην Κεντρική Μονάδα Επεξεργασίας.

Επίσης μεγάλη προσοχή πρέπει να δοθεί κατά την επιλογή και τοποθέτηση του τροφοδοτικού ώστε να μην προκαλέσουμε βλάβη ή και ολοκληρωτική καταστροφή του τροφοδοτικού ή ολοκλήρου του συστήματος PLC. Πάντα πριν την επιλογή και εγκατάσταση του διαβάζουμε τα τεχνικά φυλλάδια του κατασκευαστή.

1.4.3 Κεντρική Μονάδα Επεξεργασίας

Είναι η κύρια μονάδα ενός PLC, είναι υπεύθυνη για την εκτέλεση του προγράμματος του αυτοματισμού. Η κεντρική μονάδα επεξεργασίας είναι στην ουσία ένας μικροϋπολογιστής. Διακρίνουμε σ' αυτή όλα τα κύρια μέρη του μικροϋπολογιστή, όπως ο επεξεργαστής, η μνήμη, τα I/O.

Η κεντρική μονάδα επεξεργασίας είναι ένα ολοκληρωμένο κύκλωμα, το οποίο αποτελεί τον εγκέφαλο του PLC. Ο επεξεργαστής του PLC είναι υπεύθυνος για όλες τις λειτουργίες του PLC. (Τον τρόπο λειτουργίας του PLC θα το δούμε σε επόμενη παράγραφο).

Η μνήμη της κύριας μονάδας ενός PLC χωρίζεται σε μνήμη RAM, ROM και EEPROM.

Μνήμη RAM. Η μνήμη RAM - Random Access Memory, μνήμη τυχαίας προσπέλασης είναι εκείνη στην οποία μπορούμε να γράφουμε και να σβήνουμε κατά την διάρκεια λειτουργίας του PLC. Τα δεδομένα της όμως χάνονται μόλις σταματήσει τροφοδοσία της με ρεύμα. Στη μνήμη RAM η κεντρική μονάδα αποθηκεύει μια σειρά από πληροφορίες σε ξεχωριστές περιοχές εργασίας.

Μπορούμε να τις διακρίνουμε ως εξής περιοχές:

- Περιοχή μνήμης όπου αποθηκεύονται οι καταστάσεις των εισόδων και των εξόδων. Η περιοχή αυτή ονομάζεται για τις εισόδους, εικόνα εισόδων και για τις εξόδους, εικόνα εξόδων.
- Περιοχή μνήμης όπου αποθηκεύονται οι ενδιάμεσες πληροφορίες, που αφορούν τη λειτουργία του αυτοματισμού.
- Περιοχή μνήμης των χρονικών.
- Περιοχή μνήμης των απαριθμητών.
- Περιοχή μνήμης όπου αποθηκεύονται τα προγράμματα του χρήστη, δηλαδή τα προγράμματα που λειτουργούν ένα συγκεκριμένο αυτοματισμό.

Τα περιεχόμενα της μνήμη RAM μηδενίζουν μόλις σταματήσει η τροφοδοσία της με ρεύμα. Όμως το πρόγραμμα που λειτουργεί τον αυτοματισμό πρέπει να παραμένει αναλλοίωτο στη μνήμη και αφού κλείσουμε την τροφοδοσία του PLC. Γι' αυτό το λόγο η μνήμη RAM παραμένει σε τροφοδοσία με τη βοήθεια μιας μπαταρίας. Αλλά και πάλι το πρόβλημα δεν λύνεται επαρκώς μιας πάντα υπάρχει η πιθανότητα η μπαταρία να αστοχήσει.

Μνήμη EEPROM. Τα PLC δεν χρησιμοποιούν μόνο τον παραπάνω τρόπο, της πάντα τροφοδοτούμενης μνήμης RAM, για να διατηρήσουν το πρόγραμμα στη μνήμη. Αλλά χρησιμοποιούν τη μνήμη EEPROM - Eclectically Erasable Programmable Read Only Memory. Πρόκειται για μνήμη που δε χάνονται τα δεδομένα του PLC, όταν μείνει χωρίς τροφοδοσία. Στην μνήμη EEPROM μπορούμε να γράφουμε, να σβήνουμε και να ξαναγράφουμε δεδομένα με χρήση ειδικού μηχανήματος. Σε πολλά PLC η EEPROM χρησιμοποιείται σαν σκληρός δίσκος για την εύκολη αλλαγή του τρόπου λειτουργίας του αυτοματισμού από ένα απλό χειριστή. Δηλαδή έχουμε αποθηκευμένο το εναλλακτικό πρόγραμμα σε ένα chip EEPROM και απλά όταν θέλουμε να αλλάξουμε ο πρόγραμμα λειτουργίας του αυτοματισμού, αλλάζουμε τα EEPROM chips στην ηλεκτρονική πλακέτα του PLC.

Μνήμη ROM. Στη μνήμη ROM – Read Only Memory ο κατασκευαστής του PLC αποθηκεύει το λειτουργικό σύστημα του PLC, δηλαδή το πρόγραμμα για όλες τις βασικές λειτουργίες που είναι απαραίτητες για να δουλέψει το PLC.

Εξωτερικά σε μια κεντρική μονάδα επεξεργασίας συνήθως υπάρχουν:

- Θέση σύνδεσης της συσκευής προγραμματισμού.
- Θέση σύνδεσης επεκτάσεων.
- Διακόπτης δύο θέσεων ο οποίος θέτει το PLC σε κατάσταση RUN ή STOP, δηλαδή σε κατάσταση λειτουργίας RUN ή τερματισμού STOP.(όχι σε όλα τα PLC).
- Λυχνίες ένδειξης, όπως λυχνία ένδειξης ότι το PLC είναι σε τροφοδοσία, λυχνία ένδειξης ότι το PLC είναι σε κατάσταση RUN, λυχνία ένδειξης όπως το PLC είναι σε κατάσταση STOP και λυχνία που δείχνει, εάν έχει πρόβλημα το PLC.

1.4.4 Μονάδες εισόδων - εξόδων

Οι μονάδες των εισόδων και των εξόδων αποτελούν τις μονάδες επικοινωνίας της κεντρικής μονάδας

με τον έξω κόσμο, δηλαδή με τους αισθητήρες, τους διακόπτες, τα μπουτόν, που δίνουν τις πληροφορίες (εντολές), καθώς και με τους ηλεκτρονόμους ισχύος των κινητήρων, ηλεκτρομαγνητικές βαλβίδες, ενδεικτικές λυχνίες και γενικά τους αποδέκτες που εκτελούν τις εντολές του αυτοματισμού.

Η κεντρική μονάδα μπορεί να δεχτεί ψηφιακά σήματα εισόδου και εξόδου χαμηλής τάσης και πολύ μικρού ρεύματος. Η τάση που δέχεται η κεντρική μονάδα είναι συνήθως 0V για το λογικό 0 και 5V για το λογικό 1. Το ρεύμα εισόδου καθώς και το ρεύμα εξόδου δεν μπορεί να περάσει τα μερικά mA. Οι μονάδες εισόδων και εξόδων αναλαμβάνουν να προσαρμόσουν τα σήματα εισόδου και εξόδου, που έχουμε στον αυτοματισμό, με τα σήματα που μπορεί να δεχτεί η κεντρική μονάδα, τόσο από άποψη τάσεων όσο και από άποψη ρευμάτων.

Η προσαρμογή αυτή γίνεται με τη χρήση ηλεκτρονικών στοιχείων ισχύος, δηλαδή τρανζίστορ ισχύος, θυρίστορ και τριας, είτε ακόμη με τη χρήση κατάλληλων μικροηλεκτρονόμενων.

Κάθε σύστημα PLC καταλήγει πάντα σε ακροδέκτες. Οι ακροδέκτες ανήκουν στις μονάδες εισόδων και εξόδων του PLC. Στους ακροδέκτες εισόδων καταλήγουν οι αγωγοί που έρχονται από αισθητήρες τερματικούς διακόπτες, πιεζοστάτες, διακόπτες, μπουτόν κ.α. Στους ακροδέκτες των εξόδων καταλήγουν οι αγωγοί που τροφοδοτούν πηνία ηλεκτρονόμων ισχύος, ηλεκτρομαγνητικές βαλβίδες, λυχνίες ένδειξης και λοιπούς αποδέκτες.

Αν και ανάλογα με τον τύπο του PLC οι μονάδες εισόδων και εξόδων έχουν διαφορετικά χαρακτηριστικά.

Γενικά ισχύουν τα παρακάτω:

- Οι μονάδες εισόδων ή εξόδων μπορούν να λειτουργήσουν είτε με συνεχή τάση (DC) είτε με εναλλασσόμενη τάση (AC). Τυπικές τάσεις που συναντούμε στα PLC είναι: DC 24V, 48V, 60V και AC 24V, 48V, 115V, 230V, με συνηθέστερες τις DC 24V και AC 115V και 230V.
- Η τάση για την λειτουργία όλου του συστήματος αυτοματισμού δεν παρέχεται συνήθως από τη μονάδα τροφοδοσίας του PLC. Πρέπει να τη δώσουμε προσθέτοντας μια νέα μονάδα τροφοδοσίας.
- Τα κυκλώματα και οι τάσεις των εισόδων είναι τελείως ανεξάρτητα από τα κυκλώματα και τις τάσεις των εξόδων. Οπότε η τάση για τις εισόδους μπορεί να είναι διαφορετική από την τάση για τις εξόδους. Αν η τάση εξόδων είναι η ίδια με την τάση των εισόδων μπορεί να χρησιμοποιηθεί το ίδιο τροφοδοτικό (για τάσεις DC), ή μετασχηματιστής χειρισμού (για τάσεις AC) για τις εισόδους και τις εξόδους.
- Η τάση εισόδων δηλαδή η τάση που θα φθάσει σε μια είσοδο, όταν ενεργοποιηθεί ο αντίστοιχος αισθητήρας διαχωρίζεται συνήθως γαλβανικά από το υπόλοιπο εσωτερικό κύκλωμα του PLC. Τα ίδια ισχύουν και για τις εξόδους. Αν σε κάποιες μονάδες εισόδων ή εξόδων δεν έχουμε γαλβανική απομόνωση πρέπει να προσέξουμε ιδιαίτερα το θέμα των γειώσεων.

1.4.5 Οι προγραμματιζόμενοι λογικοί ελεγκτές της αγοράς

Η εικόνα που έχει διαμορφωθεί στην αγορά από τις εταιρείες, που κατασκευάζουν PLC, είναι η εξής:

Modular PLC (εικόνα 1.3).

Στην περίπτωση αυτή το PLC πωλείται σε modular μορφή, δηλαδή σε κομμάτια. Τα βασικά κομμάτια ενός modular PLC είναι:

- Η μονάδα τροφοδοσίας.
- Η κεντρική μονάδα επεξεργασίας, η οποία έχει τη δυνατότητα να οδηγήσει ένα ανώτατο αριθμό εισόδων και εξόδων. π.χ. το PLC SIMATIC S7-1200 SIEMENS με 1024 I/O.
- Οι μονάδες εισόδων και εξόδων. Στα modular PLC οι μονάδες αυτές πωλούνται σε κομμάτια. Κάθε μονάδα εισόδων ή εξόδων μπορεί να έχει 4, 8, 16 ή 32 εισόδους ή εξόδους. Έτσι έχουμε την δυνατότητα να επιλέγουμε το πλήθος και τον τύπο, μιας μονάδας εισόδων ή εξόδων με τα τεχνικά χαρακτηριστικά που χρειαζόμαστε. Επίσης σε ένα modular PLC μπορούμε να έχουμε μονάδες εισόδων ή εξόδων που να λειτουργούν σε διαφορετικές τάσεις τροφοδοσίας.



Εικόνα 1.2 : Compact PLC Siemens Logo



Εικόνα 1.3 : Modular PLC Siemens Simatic S7-1200

Compact PLC (εικόνα 1.2).

Όλες οι εταιρείες διαθέτουν compact PLC, στα οποία όλες οι μονάδες τους όπως, τροφοδοσίας, κεντρική μονάδα και μονάδες εισόδων - εξόδων είναι ενσωματωμένες σε μια συσκευή. Σε αυτού του είδους τα PLC είσοδοι και έξοδοι είναι συνήθως μέχρι 20 και όλες οι είσοδοι ή έξοδοι έχουν τα ίδια τεχνικά χαρακτηριστικά.

Το σημείο που πρέπει κάποιος να προσέξει σχετικά με τις εισόδους και εξόδους είναι ότι κάθε είσοδος ή έξοδος είναι για το PLC ακριβώς καθορισμένη, δηλαδή έχει καθορισμένο όνομα με το οποίο αναφέρεται και στο πρόγραμμα. Στα compact PLC σε κάθε ακροδέκτη αναγράφεται το όνομα της εισόδου ή της εξόδου. Στα modular PLC υπάρχει σαφές σύστημα με το οποίο αναγνωρίζουμε το όνομα της εισόδου ή εξόδου σε κάθε ακροδέκτη μιας μονάδας εισόδων ή εξόδων.

1.5 Αρχή λειτουργίας προγραμματιζόμενου λογικού ελεγκτή PLC

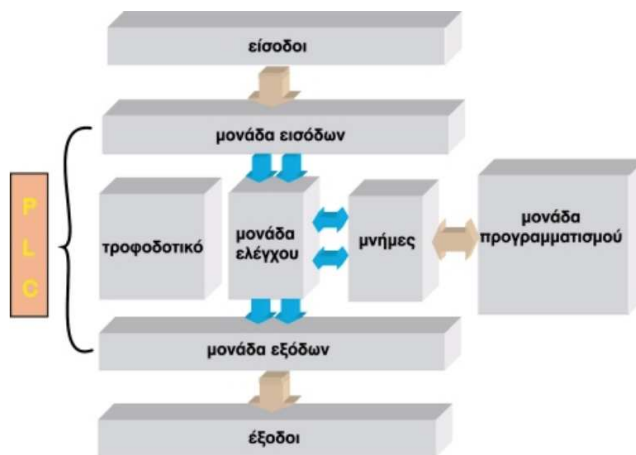
Όταν ένα PLC βρίσκεται σε κατάσταση λειτουργίας - RUN. Τα βήματα που ακολουθεί κατά τη λειτουργία του είναι τα εξής:

Βήμα 1^ο : Η κύρια μονάδα επεξεργασίας – CPU, διαβάζει τις εισόδους. Αυτό σημαίνει ότι για κάθε είσοδο ελέγχει αν έχει υψηλή τάση λογικό 1 ή χαμηλή τάση λογικό 0. Η τιμή για κάθε είσοδο αποθηκεύεται σε ειδική περιοχή μνήμης η οποία ονομάζεται εικόνα εισόδων. Η εικόνα εισόδων είναι σαν έναν πίνακα, όπου η κύρια μονάδα σημειώνει τις τιμές, που διάβασε από τις εισόδους. π.χ. είσοδος %IX0.0=1,%IX0.1=0, %IX0.0=0.

Βήμα 2^ο : Στη συνέχεια η κύρια μονάδα – CPU, χρησιμοποιώντας σαν δεδομένα τις τιμές των εισόδων, που διάβασε, εκτελεί βήμα – βήμα τις εντολές του προγράμματος που της έχουμε φορτώσει, το οποίο λειτουργεί τον αυτοματισμό. Το πρόγραμμα αυτό στην ουσία περιέχει μια σειρά από λογικές πράξεις.

Με την εκτέλεση του προγράμματος θα πάρουμε τα αποτελέσματα στις εξόδους. Τα αποτελέσματα αυτά αποθηκεύονται στην ειδική περιοχή της μνήμης που ονομάζεται εικόνα εξόδων. Όπως η εικόνα εισόδων, η εικόνα εξόδων περιέχει την τιμή 0 ή 1 για κάθε έξοδο, π.χ. %QX0.0=1, %QX0.1=1, %QX0.3=0, οι τιμές αυτές προκύπτουν από την εκτέλεση των λογικών πράξεων του προγράμματος και μόνο.

Βήμα 3^ο : Στη συνέχεια η κύρια μονάδα – CPU, στέλνει τιμές της εικόνας εξόδων στις φυσικές εξόδους του PLC. Αυτό σημαίνει ότι θα δοθεί τάση σε όποια έξοδο έχει 1 και θα δεν θα δοθεί τάση σε όποια έξοδο έχει 0.



Εικόνα 1.4 : Βήματα λειτουργίας PLC.

Με τη συμπλήρωση του 3ου βήματος συμπληρώνεται ένας πλήρης κύκλος λειτουργίας και η διαδικασία ξαναρχίζει από την αρχή. Ο κύκλος λειτουργίας εκτελείται συνεχώς όσο το PLC βρίσκεται σε κατάσταση RUN. Δηλαδή το PLC εκτελεί τα βήματα του κύκλου λειτουργίας συνέχεια.

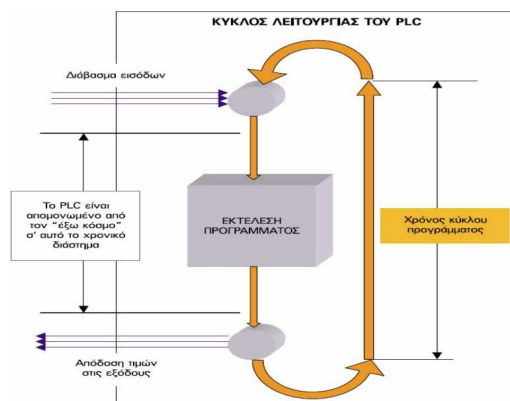
Ο χρόνος που χρειάζεται για να εκτελέσει το PLC ένα πλήρη κύκλο λειτουργίας ονομάζεται χρόνος κύκλου και εξαρτάται από τη “ταχύτητα” του μικροεπεξεργαστή του PLC, αλλά και από τον αριθμό και το είδος των εντολών του προγράμματος. Δηλαδή στο ίδιο PLC για ένα μεγαλύτερο πρόγραμμα

έχουμε μεγαλύτερο χρόνο κύκλου. Ο χρόνος που διαρκεί ο κύκλος λειτουργίας ενός PLC αποτελεί και το μέτρο σύγκρισης μεταξύ των PLC. Για να μπορούν να συγκριθούν τα PLC ως προς τη ταχύτητα εκτέλεσης ενός προγράμματος, ορίζουμε το μέσο χρόνο κύκλου, σαν το χρόνο κύκλου ενός προγράμματος που περιλαμβάνει 1 KB δυαδικές εντολές. Ακόμη όμως και στην χειρότερη περίπτωση, σε ένα αργό PLC ο χρόνος κύκλου δεν ξεπερνά μερικές εκατοντάδες χιλιοστά του δευτερολέπτου.

Θα θέλαμε στο σημείο αυτό να τονίσουμε την ουσιαστική διαφορά στη λειτουργία ενός αυτοματισμού με PLC από ένα κλασικό αυτοματισμό με ηλεκτρονόμους.

Στην περίπτωση του κλασικού αυτοματισμού, όταν έχουμε αλλαγή της κατάστασης ενός διακόπτη εισόδου, η αλλαγή αυτή προκαλεί εκείνη τη στιγμή διαδοχικές αλλαγές στα στοιχεία του κυκλώματος που τροφοδοτούνται από το συγκεκριμένο διακόπτη. Δηλαδή έχουμε εκτέλεση διαδικασίας σε πραγματικό χρόνο.

Αν μελετήσουμε τον κύκλο λειτουργίας του PLC, θα δούμε ότι το PLC δεν βλέπει συνεχώς την εξωτερική εγκατάσταση, παρά μόνο κατά τα χρονικά διαστήματα που διαβάζει τις εισόδους και αποδίδει τιμές στις εξόδους. Στον υπόλοιπο χρόνο του κύκλου, το PLC είναι ένας υπολογιστής ο οποίος εκτελεί πράξεις λογικές ή αριθμητικές απομονωμένο από τον έξω κόσμο. Για να γίνει αυτό κατανοητό υποθέστε ότι αλλάζει η κατάσταση μιας εισόδου, κατά την διάρκεια του χρόνου κατά τον οποίον εκτελούνται οι εντολές προγράμματος. Στην περίπτωση αυτή στο τέλος του κύκλου το PLC θα δώσει αποτελέσματα στις εξόδους, στα οποία δεν θα έχει ληφθεί υπ' όψη η αλλαγή στην κατάσταση της συγκεκριμένης εισόδου. Αυτό γιατί το PLC θα ενημερώσει την εικόνα των εισόδων για την αλλαγή στην κατάσταση της συγκεκριμένης εισόδου στην αρχή του επόμενου κύκλου επεξεργασίας του προγράμματος. Το PLC θα δώσει αποτελέσματα στις εξόδους, στα οποία θα έχει ληφθεί υπ' όψη η αλλαγή στην κατάσταση της παραπάνω εισόδου στο τέλος του επόμενου κύκλου επεξεργασίας του προγράμματος. Λαμβάνοντας τα παραπάνω υπ' όψη θα έλεγε κάποιος ότι τελικά το PLC ανταποκρίνεται πολύ καθυστερημένα στις αλλαγές μιας αυτοματοποιημένης διαδικασίας. Όμως, αυτό δεν είναι η πραγματικότητα, αφού ο χρόνος πραγματοποίησης ενός κύκλου προγράμματος από ένα PLC είναι πάρα πολύ μικρός, το πολύ 300ms σε πολύπλοκες εγκαταστάσεις αυτοματισμού.



Εικόνα 1.5 : Κύκλος λειτουργίας PLC.

1.6 Προγραμματισμός PLC

Το βασικότερο τμήμα σε ένα σύστημα αυτοματισμού με PLC είναι το πρόγραμμα που υλοποιεί τον αυτοματισμό που θέλουμε. Το οποίο αναπτύσσεται σε μια γλώσσα προγραμματισμού. Δυστυχώς στα PLC δεν υπήρξε τυποποίηση σε κανέναν τομέα, λόγω του ανταγωνισμού των εταιρειών, ούτε βέβαια στο θέμα των γλωσσών προγραμματισμού. Δηλαδή δεν υπάρχουν γλώσσες προγραμματισμού για PLC που να ισχύουν ανεξάρτητα από εταιρεία, όπως για παράδειγμα συμβαίνει στον προγραμματισμό των ηλεκτρονικών υπολογιστών. Ακόμη και έτσι οι γλώσσες των PLC των εταιρειών μοιάζουν αρκετά οπότε μπορούμε να μιλάμε για μια τυποποίηση της αγοράς.

1.7 Γλώσσες προγραμματισμού των προγραμματιζόμενων λογικών ελεγκτών

Τρεις είναι σήμερα οι κυριότερες κατηγορίες γλωσσών προγραμματισμού για PLC, τις οποίες συναντούμε με μικρές διαφορές στα PLC όλων των εταιρειών:

1.7.1 Γλώσσα διαγραμμάτων σκάλας Ladder Diagram

Είναι η πρώτη γλώσσα που αναπτύχθηκε ιστορικά. Η γλώσσα Ladder Diagram στην ουσία επιτρέπει τη μεταφορά του ηλεκτρολογικού σχεδίου, μέσω της συσκευής προγραμματισμού στο PLC. Με τη γλώσσα αυτή η εκπαίδευση των τεχνικών, που ήταν συνηθισμένοι στον κλασικό αυτοματισμό, γινόταν εύκολα και γρήγορα, αφού δεν άλλαζε ουσιαστικά την εργασία σχεδιασμού του αυτοματισμού. Η γλώσσα Ladder Diagram χρησιμοποιεί την προτυποποίηση των Η.Π.Α, στο σχεδιασμό των ηλεκτρικών επαφών, αυτό οφείλεται στο γεγονός ότι τα πρώτα PLC αναπτύχθηκαν στην Αμερική. Όμως στη συνέχεια ο τρόπος αυτός σχεδιασμού διατηρήθηκε και από τις ευρωπαϊκές εταιρείες, με αποτέλεσμα σήμερα να είναι πλέον καθιερωμένος.

1.7.2 Γλώσσα λίστα εντολών Structured Text

Η γλώσσα αυτή αναπτύχθηκε σχεδόν ταυτόχρονα με τη γλώσσα Ladder, αν και οι εταιρείες έδειξαν στην αρχή δισταγμό στο να την προωθήσουν, φοβούμενες μην τρομάζουν το τεχνικό κατεστημένο της βιομηχανίας. Η γλώσσα αυτή δημιουργεί λίστα προγράμματος με εντολές, οι οποίες αντιστοιχούν στις λογικές πύλες AND, OR, NOT κ.α. Στις αρχικές εκδόσεις η γλώσσα ST ήταν φτωχή έχοντας μόνο τις βασικές λογικές εντολές, οι οποίες αντιστοιχούσαν άμεσα στις γραφικές εντολές της γλώσσας Ladder. Σήμερα οι γλώσσες αυτές έχουν εξελιχθεί πάρα πολύ και συναντά κανείς σε αυτές στοιχεία από τις γλώσσες των υπολογιστών και κυρίως των γλωσσών Assembly. Ο προγραμματισμός σε λίστα εντολών απαιτεί από τον προγραμματιστή του PLC να έχει έστω στοιχειώδεις γνώσεις προγραμματισμού.

1.7.3 Γλώσσα λογικών γραφικών ή λογικού διαγράμματος Function Block Diagram

Και εδώ έχουμε άλλη μια γλώσσα που συντάσσεται με γραφικό τρόπο, αλλά αντί του ηλεκτρολογικού σχεδίου, χρησιμοποιεί τα λογικά κυκλώματα. Η γλώσσα αυτή είναι νεότερη σε σχέση με την γλώσσα Ladder και δεν χρησιμοποιούνταν από όλες τις εταιρείες κατασκευής PLC.

Πρέπει να σημειώσουμε εδώ ότι το ηλεκτρολογικό σχέδιο ενός αυτοματισμού δεν είναι απαραίτητο

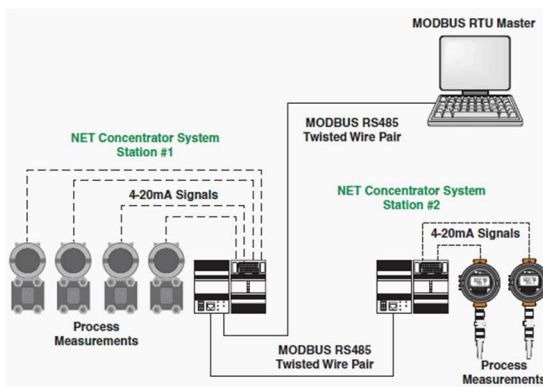
προκειμένου να αναπτυχθεί πρόγραμμα σε PLC. Ένας σχετικά έμπειρος στον προγραμματισμό τεχνικός δεν περνά από το ηλεκτρολογικό σχέδιο του αυτοματισμού, προκειμένου να αναπτύξει το πρόγραμμα. Σε σύνθετους αυτοματισμούς η ανάπτυξη του ηλεκτρολογικού σχεδίου για χρήση σε αυτοματισμό είναι πολύ δύσκολη σε σχέση με την ανάπτυξη ενός προγράμματος αυτοματισμού. Σε ορισμένες περιπτώσεις είναι ευκολότερο να γράψουμε το πρόγραμμα αυτοματισμού με χρήση των δεδομένων, από το να βασιστούμε στο ηλεκτρολογικό σχέδιο του.

Ένα πρόβλημα που υπήρχε σχετικά με τον προγραμματισμό των PLC στο παρελθόν πριν την υιοθέτηση του προτύπου IEC 61131-3 είναι ότι οι γλώσσες προγραμματισμού των PLC δεν ήταν τυποποιημένες, αλλά διαφέραν από εταιρεία σε εταιρεία. Διαφέραν ακόμη και μεταξύ των μοντέλων της ίδιας εταιρείας. Όμως η λογική των γλωσσών προγραμματισμού σε όλα τα PLC είναι ίδια. Έτσι, όποιος μάθαινε να χρησιμοποιεί πολύ καλά τις γλώσσες προγραμματισμού ενός μοντέλου PLC, αρκετά εύκολα μαθαίνει τις γλώσσες προγραμματισμού ενός άλλου μοντέλου, εντοπίζοντας γρήγορα τις διαφορές.

1.8 Πρωτόκολλο επικοινωνίας Modbus

Το Modbus παρουσιάστηκε το 1979 από την Modicon σήμερα Schneider Electric, ως μέσο επικοινωνίας με πολλές συσκευές (έως 127) σε ένα και μόνο σύρμα συνεστραμμένου ζεύγους. Το δίκτυο Modbus αποτελεί ένα, ανοικτό πρότυπο επικοινωνιών συσκευών, το οποίο υποστηρίζεται από ένα μεγάλο αριθμό προϊόντων και κατασκευαστών στην αγορά μέχρι και σήμερα. Με περισσότερα από 7 εκατομμύρια κόμβους σε Βόρειο Αμερική και Ευρώπη. Το Modbus αποτελεί το de facto βιομηχανικό πρότυπο για την δικτύωση συσκευών και συστημάτων πολλών κατασκευαστών βιομηχανικού υλικού και εξοπλισμού.

Αρχικά το Modbus έτρεχε πάνω από το πρωτόκολλο RS232, αλλά γρήγορα προσαρμόστηκε να τρέχει στο RS485, κερδίζοντας σε ταχύτητα επικοινωνίας, ενώ μπορούσε να συνδέσει διάφορες μονάδες σε μεγαλύτερες αποστάσεις από πριν. Το Modbus πολύ σύντομα έγινε το de facto πρότυπο για την επικοινωνία συστημάτων αυτοματισμού στην βιομηχανία. Η εταιρία Modicon κυκλοφόρησε το Modbus ως πρωτόκολλο ανοικτού κώδικα, δηλαδή μπορούσε να το χρησιμοποιήσει όποιος ήθελε χωρίς να χρειάζεται να λάβει την άδεια χρήσης από την Modicon.



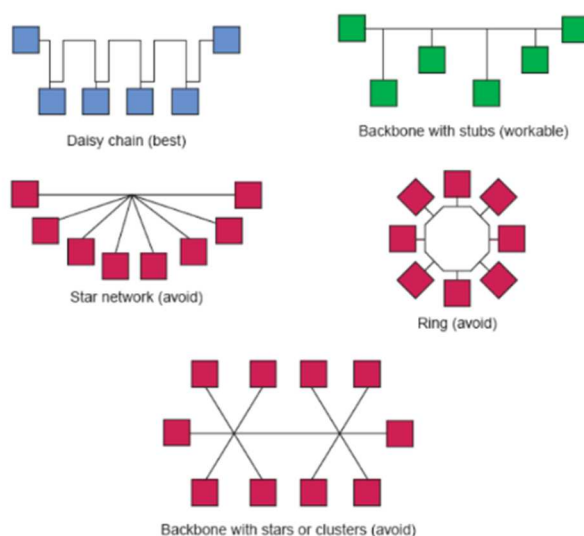
Εικόνα 1.6 : Πεδίο Modbus.

Ακόμη ένας λόγος που το Modbus είναι δημοφιλές για χρήση σε βιομηχανικά περιβάλλοντα, είναι

ο σχετικά εύκολος τρόπος επικοινωνίας, ανάπτυξης και συντήρησης εφαρμογών σε σύγκριση με άλλα πρότυπα, επίσης θέτει ελάχιστους περιορισμούς στη μορφή των δεδομένων που πρόκειται να μεταδοθούν. Σε ένα τυπικό δίκτυο Modbus, η αποστολή μηνυμάτων γίνεται σειριακά μέσω RS232/RS485 με βάση το πρότυπο EIA. Το πρωτόκολλο Modbus είναι βασισμένο στην αρχιτεκτονική master/slave. Ο κύριος κόμβος ονομάζεται master και είναι αυτός που στέλνει μηνύματα στους άλλους κόμβους του πεδίου που ονομάζονται slaves. Οι slaves κόμβοι μόνο απαντούν στα μηνύματα του master. Σημαντικό πλεονέκτημα του Modbus είναι ότι μπορεί να τρέξει στα περισσότερα μέσα επικοινωνίας, συμπεριλαμβανομένων καλωδίων συνεστραμμένου ζεύγους, ασύρματων, οπτικών ινών, Ethernet, τηλεφωνικών μόντεμ, κινητών τηλεφώνων και μικροκυμάτων. Αυτό μας δίνει την δυνατότητα σύνδεσης μιας συσκευής με Modbus σε μια νέα ή υπάρχουσα μονάδα πολύ εύκολα. Οπότε μπορούμε με την χρήση του πρωτόκολλου Modbus να παρέχουμε ψηφιακές επικοινωνίες ακόμη και σε παλαιότερες εγκαταστάσεις, χρησιμοποιώντας υπάρχουσες καλωδιώσεις συνεστραμμένου ζεύγους. Το Modbus είναι πρωτόκολλο ανεξάρτητο από το φυσικό επίπεδο δικτύου, η σειριακή γραμμή Modbus RTU μπορεί να ενσωματωθεί απ' ευθείας με τα δίκτυα Modbus TCP.

Όταν η Master Modbus RTU συσκευή θέλει πληροφορίες από μια άλλη συσκευή του δικτύου, αποστέλλει ένα μήνυμα που περιέχει τη διεύθυνση της, τα δεδομένα που επιθυμεί και ένα αθροιστικό άθροισμα για την ανίχνευση σφαλμάτων. Όλες οι συσκευές του δικτύου βλέπουν το μήνυμα, αλλά θα απαντήσει μόνο η συσκευή στην οποία απευθύνεται το μήνυμα. Slaves Modbus RTU δεν μπορούν να στέλνουν μηνύματα επικοινωνίας προς άλλες συσκευές του πεδίου αλλά μόνο να απαντούν σε μηνύματα που δέχονται από την Master συσκευή του πεδίου.

Οι Slaves Modbus RTU είναι συχνά συσκευές πεδίου, όλες οι οποίες συνδέονται στο δίκτυο σε τοπολογία πολλαπλών διαδρομών.



Εικόνα 1.7 : Τοπολογίες δικτύου Modbus.

Στην πραγματικότητα, μπορούμε με μια νέα εφαρμογή για το Modbus να παρέχουμε ψηφιακές επικοινωνίες σε παλαιότερες εγκαταστάσεις, χρησιμοποιώντας υπάρχουσες καλωδιώσεις συνεστραμμένου ζεύγους. “Χρήση του MODBUS για τον έλεγχο διαδικασιών και τον αυτοματισμό (1)”

Οι τρεις εκδόσεις Modbus που χρησιμοποιούνται σήμερα είναι:

- Modbus ASCII
- Modbus RTU
- Modbus TCP/IP

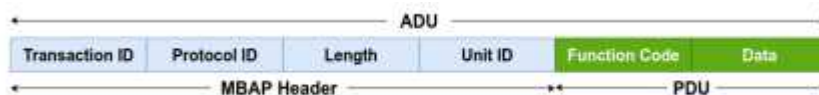
Όλα τα μηνύματα Modbus αποστέλλονται με την ίδια μορφή. Η μόνη διαφορά μεταξύ των τριών τύπων Modbus είναι ο τρόπος με τον οποίο κωδικοποιούνται τα μηνύματα.

Στο Modbus ASCII, όλα τα μηνύματα κωδικοποιούνται σε δεκαεξαδικό σύστημα αριθμήσεις, χρησιμοποιώντας χαρακτήρες ASCII 4-bit. Για κάθε byte πληροφοριών, χρειάζονται δύο byte επικοινωνίας, διπλάσια από αυτά με το Modbus RTU ή του Modbus TCP/IP.

Επομένως, το Modbus ASCII είναι το πιο αργό από τα τρία πρωτόκολλα. Χρησιμοποιείται όταν έχουμε δίκτυα επικοινωνίας βασισμένα σε τηλεφωνικά μόντεμ ή σύνδεση με RF. Αυτό συμβαίνει γιατί ο κώδικας ASCII χρησιμοποιεί χαρακτήρες για να οριοθετήσει το μήνυμα. Λόγω αυτού του τρόπου οριοθέτησης του μηνύματος, οι τυχόν καθυστερήσεις που προέρχονται από το μέσο μετάδοσης δεν θα προκαλέσουν λανθασμένη ερμηνεία του μηνύματος από τη συσκευή λήψης. Αυτό είναι σημαντικό όταν πρόκειται για αργά modems, κινητά τηλέφωνα, επικοινωνία σε θορυβώδες περιβάλλον.

Στο Modbus RTU, τα δεδομένα κωδικοποιούνται σε δυαδικό σύστημα αρίθμησης, και απαιτεί μόνο ένα byte επικοινωνίας ανά byte δεδομένων. Αυτό είναι ιδανικό για χρήση σε δίκτυα RS232 ή RS485, σε ταχύτητες από 1.200 έως 115.000 baud. Οι ταχύτητες επικοινωνίας που κατά κύριο λόγο χρησιμοποιούνται στο Modbus RTU είναι από 9.600 έως 19.200 baud. Το Modbus RTU είναι το ευρύτερα χρησιμοποιούμενο βιομηχανικό πρωτόκολλο.

Το Modbus TCP/IP γνωστό και ως Modbus TCP, είναι απλά το Modbus RTU «φορτωμένο» σε πρωτόκολλο Ethernet. Αντί να χρησιμοποιεί διευθύνσεις συσκευών για επικοινωνία με εξαρτημένες συσκευές, χρησιμοποιούνται διευθύνσεις IP. Με την χρήση του Modbus TCP IP, τα δεδομένα αποστολής του πρωτοκόλλου, είναι απλά ενσωματωμένα σε ένα πακέτο TCP/IP. Ως εκ τούτου, οποιοδήποτε δίκτυο Ethernet που υποστηρίζει TCP/IP μπορεί να υποστηρίξει και το Modbus TCP/IP. Το Modbus TCP/IP χρησιμοποιεί για τις συνδέσεις του, την TCP θύρα 502.



Εικόνα 1.8 : Πλαίσιο Modbus χρησιμοποιώντας Modbus TCP

Το ADU πακέτο αποτελείται από την κεφαλίδα του πρωτοκόλλου Modbus TCP/IP (MBAP) η οποία αποτελείται από ένα αναγνωριστικό μετάδοσης (Transaction ID) 2 byte, αναγνωριστικό πρωτοκόλλου (Protocol ID) 2 byte, μήκος (Length) 2 byte και το πεδίο (Unit ID) 1 byte.

Το αναγνωριστικό πρωτοκόλλου υποδεικνύει το πρωτόκολλο εφαρμογής που ενθυλακώνεται από την κεφαλίδα MBAP που το μηδέν για το Modbus. Το πεδίο μήκους υποδεικνύει το μήκος σε byte

των υπόλοιπων πεδίων (αναγνωριστικό μονάδας και PDU). Το πεδίο Unit ID είναι το αναγνωριστικό που δηλώνει την slave μονάδα που σχετίζεται με τη συναλλαγή. Το MBAP, το μήκος καθορίζεται στα επτά byte, άρα το μέγιστο μήκος ενός Modbus TCP/IP πακέτου δεδομένων είναι 260 byte.

Σήμερα ο οργανισμός Modbus - www.modbus.org, είναι μια ομάδα ανεξάρτητων χρηστών και προμηθευτών συσκευών αυτοματισμού που επιδιώκει να προωθήσει και την υιοθέτηση της σουίτας πρωτοκόλλων επικοινωνίας Modbus. Ο οργανισμός Modbus παρέχει επίσης την υποδομή για τη λήψη και την ανταλλαγή πληροφοριών σχετικά με τα πρωτόκολλα, την εφαρμογή και την πιστοποίησή τους για την απλούστευση της εφαρμογής από τους χρήστες με αποτέλεσμα τη μείωση του κόστους, ενώ συνεχίζει να υποστηρίζει το πρωτόκολλο Modbus παγκοσμίως. Το Modbus είναι ένα σύστημα master - slave, όπου ο master επικοινωνεί με έναν ή περισσότερους slaves. Το master είναι συνήθως PLC ή ένα PC ή ένα DCS Distributed Control System.

1.9 IEC 61131-3

Το πρότυπο IEC 61131-3 είναι η πρώτη πραγματική προσπάθεια τυποποίησης των γλωσσών προγραμματισμού που χρησιμοποιούνται σε βιομηχανικούς αυτοματισμούς. Με την παγκόσμια υποστήριξη που λαμβάνει, καταφέρνει να είναι ανεξάρτητο από οποιαδήποτε εταιρεία.

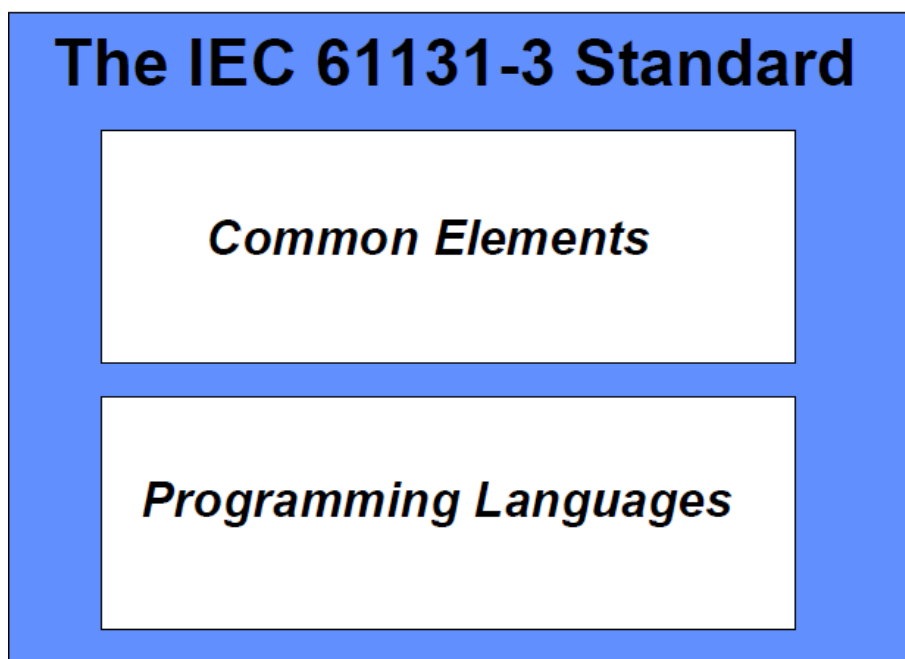
Το IEC 61131-3 είναι το τρίτο μέρος της οικογένειας IEC 61131, προδιαγράφει το συντακτικό και την σημασιολογία μιας ενιαίας σουίτας γλωσσών προγραμματισμού και περιλαμβάνει το συνολικό μοντέλο λογισμικού και μια δομημένη γλώσσα προγραμματισμού.

Αποτελείται από:

- Μέρος 1ο: Γενική Επισκόπηση.
- Μέρος 2ο: Απαιτήσεις εξοπλισμού και Διαγνωστικά σενάρια.
- Μέρος 3ο: Γλώσσες προγραμματισμού.
- Μέρος 4ο: Οδηγίες Χρήστη.
- Μέρος 5ο: Επικοινωνίες.
- Μέρος 6ο: Οδηγίες ασφαλείας.
- Μέρος 7ο: Προγραμματισμός Fuzzy Control.
- Μέρος 8ο: Οδηγίες εφαρμογής.
- Μέρος 9ο: Διεπαφή επικοινωνίας.

Από μια άλλη οπτική θα μπορούσε το πρότυπο να χωρισθεί σε δύο μέρη:

- Κοινά στοιχεία
- Γλώσσες προγραμματισμού



Εικόνα 1.9 : Δομή το προτύπου IEC 61131-3.

Κοινά Στοιχεία

▪ Τύποι Δεδομένων

Οι τύποι δεδομένων καθορίζονται στο μέρος των κοινών στοιχείων. Με τους τύπους δεδομένων αποφεύγονται τα λάθη σε πρωταρχικό στάδιο. Χρησιμοποιούνται για να καθορίσουν τον τύπο κάθε παραμέτρου. Για παράδειγμα με τον καθορισμό των τύπων δεδομένων αποφεύγεται η διαίρεση μίας ημερομηνίας με έναν ακέραιο αριθμό.

Οι κοινοί τύποι δεδομένων είναι:

- Boolean
- Ακέραιοι
- Πραγματικοί
- Byte
- Word
- Date
- Αλφαριθμητικά

Με βάση τους παραπάνω τύπους δεδομένων παρέχεται η δυνατότητα σε κάποιον να δημιουργήσει νέους τύπους δεδομένων γνωστών και ως εξαγόμενους. Με αυτό τον τρόπο κάποιος θα μπορούσε να καθορίσει ένα αναλογικό κανάλι εισόδου ως τύπο δεδομένων και να το επαναχρησιμοποιήσει.

▪ Μεταβλητές

Οι μεταβλητές αποδίδονται μόνο σε συγκεκριμένες διευθύνσεις υλικού (π.χ. είσοδοι και έξοδοι) σε παραμετροποιήσεις, πηγές ή προγράμματα. Με αυτό τον τρόπο δημιουργείται μια υψηλού επιπέδου ανεξαρτησία από το υλικό, υποστηρίζοντας έτσι την επαναχρησιμοποίηση του λογισμικού. Η εμβέλεια των μεταβλητών συνήθως περιορίζεται στην οργανική μονάδα στην οποία δηλώνονται π.χ.

τοπικές.

Αυτό σημαίνει ότι τα ονόματα των μεταβλητών μπορούν να επαναχρησιμοποιηθούν σε άλλα τμήματα χωρίς να δημιουργούνται προβλήματα. Αν πρέπει οι μεταβλητές να έχουν καθολική εμβέλεια (global) θα πρέπει να δηλωθούν ως καθολικές. Αρχικές τιμές μπορούν να καθοριστούν κατά την αρχικοποίηση ή μετά από επανεκκίνηση διασφαλίζοντας έτσι ότι θα έχουν σωστές τιμές εξ αρχής.

■ Παραμετροποίηση, Πόροι και Διεργασίες

Για να καταλάβουμε το παραπάνω μοντέλο πρέπει να δούμε το παρακάτω σχήμα όπως καθορίζεται στο πρότυπο.

Σε υψηλό επίπεδο, ολόκληρο το σύστημα που απαιτείται για την λύση διαχείρισης ενός συγκεκριμένου προβλήματος, μπορεί να αναπαρασταθεί ως παραμετροποίηση περιλαμβάνοντας την κατανομή του υλικού, διευθύνσεις μνήμης για τα κανάλια εισόδου/εξόδου και τις δυνατότητες του συστήματος.

Σε μια παραμετροποίηση υπάρχει η δυνατότητα καθορισμού ενός οι περισσότερων πόρων. Μπορούμε να αντιληφθούμε τους πόρους σαν μια μονάδα επεξεργασίας που έχει την δυνατότητα εκτέλεσης προγραμμάτων IEC.

Μέσα σε έναν πόρο μπορούν να καθοριστούν μία ή περισσότερες διεργασίες. Οι διεργασίες καθορίζουν την ελεγχόμενη εκτέλεση προγραμμάτων ή και συναρτήσεων. Αυτά μπορούν να εκτελούνται περιοδικά ή μετά την εμφάνιση μια καθορισμένης συνθήκης π.χ. αλλαγή της τιμής μιας μεταβλητής. Τα προγράμματα δημιουργούνται από έναν αριθμό διαφορετικών προγραμματιστικών στοιχείων γραμμένα με οποιαδήποτε από τις γλώσσες προγραμματισμού που καθορίζονται στο πρότυπο IEC. Τυπικά ένα πρόγραμμα αποτελείται από ένα σύνολο συναρτήσεων (όπως ADD(ition), SIN(us) και COS(inus)) και μπλοκ συναρτήσεων, που έχουν την δυνατότητα ανταλλαγής δεδομένων. Οι συναρτήσεις και τα μπλοκ συναρτήσεων είναι τα βασικά δομικά στοιχεία που περιλαμβάνουν μια δομή δεδομένων και έναν αλγόριθμο.

Τα μπλοκ συναρτήσεων όπως και οι αλγόριθμοι περιέχουν δεδομένα, δίνοντας την δυνατότητα καταγραφής των αλλαγών που συντελούνται. Έχουν μια καλά καθορισμένη διεπαφή και αποκρύπτουν εσωτερικές διεργασίες όπως ακριβώς και ένα ολοκληρωμένο κύκλωμα ή ένα μαύρο κουτί. Με αυτό τον τρόπο επιτυγχάνεται ένας καθαρός διαχωρισμός μεταξύ προγραμματιστών διαφορετικών επιπέδων. Ένας βρόχος ελέγχου θερμοκρασίας (PID) είναι ένα εξαιρετικό παράδειγμα μπλοκ συνάρτησης. Αφού καθορισθεί, μπορεί να επαναχρησιμοποιηθεί είτε στο ίδιο πρόγραμμα είτε σε διαφορετικά προγράμματα ακόμα και σε διαφορετικά έργα. Τα μπλοκ συναρτήσεων μπορούν να δημιουργηθούν χρησιμοποιώντας οποιαδήποτε γλώσσα από το πρότυπο IEC, εντούτοις τις περισσότερες φορές χρησιμοποιείται η C ή C++.

Ένα συμβατικό PLC περιλαμβάνει έναν πόρο ο οποίος τρέχει μια διεργασία, ελέγχοντας ένα πρόγραμμα που τρέχει σε έναν κλειστό βρόχο. Το πρότυπο IEC 61131-3 προσφέρει αρκετά σε αυτό καθιστώντας το ευμετάβλητο για το μέλλον. Ένα μέλλον το οποίο περιλαμβάνει ήδη την πολύ-επεξεργασία και προγράμματα που εκτελούνται με βάση γεγονότα (event driven). Το πρότυπο IEC

61131-3 είναι κατάλληλο για ένα πλήθος εφαρμογών, χωρίς την απαίτηση εκμάθησης πρόσθετων γλωσσών προγραμματισμού.

1.10 OpenPLC Project

Το OpenPLC Project είναι ένας προγραμματιζόμενος λογικός ελεγκτής ανοιχτού κώδικα που βασίζεται στο λογισμικό OpenPLC Runtime. Είναι το πρώτο πλήρως τυποποιημένο ανοιχτού κώδικα PLC, τόσο σε λογισμικό όσο και σε υλικό. Το OpenPLC Project δημιουργήθηκε σύμφωνα με το πρότυπο IEC 61131-3, το οποίο ορίζει τη βασική αρχιτεκτονική λογισμικού και τις γλώσσες προγραμματισμού για τα PLC.

Το έργο αναπτύχθηκε για πρώτη φορά από τους προγραμματιστές Edouard Tisserant και Mario de Sousa στη Βραζιλία το 2010. Το λογισμικό αρχικά γράφτηκε σε C και ήταν διαθέσιμο για πλατφόρμες Windows και Linux. Με τα χρόνια, το έργο μεγάλωσε και εξελίχθηκε, με νέες δυνατότητες και υποστήριξη για πρόσθετες πλατφόρμες, όπως το Raspberry Pi, που προστέθηκαν. Η ομάδα OpenPLC εισήγαγε επίσης υποστήριξη για νέα πρωτόκολλα και συσκευές, συμπεριλαμβανομένων των Modbus, IEC 60870-5-104 και MQTT.

Η ανάπτυξη του OpenPLC Project είναι ανοιχτή σε οποιονδήποτε ενδιαφέρεται να συνεισφέρει και έχει μια ισχυρή και ενεργή κοινότητα προγραμματιστών και χρηστών. Το λογισμικό κυκλοφορεί με την άδεια GPLv3, που σημαίνει ότι είναι δωρεάν για χρήση και τροποποίηση.

Το έργο έχει υιοθετηθεί ευρέως σε εκπαιδευτικά και βιομηχανικά περιβάλλοντα και έχει χρησιμοποιηθεί για τον έλεγχο μιας ευρείας γκάμας βιομηχανικών συστημάτων αυτοματισμού, όπως οι μεταφορικοί μάντες, οι αντλίες και οι κινητήρες. Η δημοτικότητα του OpenPLC Project έχει επίσης αυξηθεί με την άνοδο του Internet of Things (IoT) και την ανάγκη για χαμηλού κόστους, εύκολα προγραμματιζόμενους ελεγκτές για μικρής κλίμακας έργα βιομηχανικού αυτοματισμού.

Ορισμός του PLC σύμφωνα με το πρότυπο IEC 61131-3:

Ένα PLC είναι ένα ψηφιακά λειτουργικό ηλεκτρονικό σύστημα για χρήση σε βιομηχανικά περιβάλλοντα με προγραμματιζόμενη μνήμη για εσωτερική αποθήκευση οδηγιών ελέγχου προσαρμοσμένων στο χρήστη για την υλοποίηση συγκεκριμένων λειτουργιών όπως λογικός έλεγχος, έλεγχος ακολουθίας, χρονισμός, μέτρηση και αριθμητικές συναρτήσεις για έλεγχο διαφόρων τύπων μηχανών και διεργασιών μέσω ψηφιακών ή αναλογικών σημάτων εισόδου και εξόδου.

Το OpenPLC Project χρησιμοποιείται κυρίως σε βιομηχανικούς και οικιακούς αυτοματισμούς, στο διαδίκτυο των πραγμάτων IoT και στην διασύνδεση με SCADA.

Το OpenPLC Project αποτελείται από δύο μέρη τον Runtime και τον Editor. Το Runtime είναι ένα φορητό λογισμικό σχεδιασμένο να τρέχει από μικρούς μικροελεγκτές όπως Raspberry Pi, Arduino έως σε ισχυρούς διακομιστές στο Cloud. Είναι υπεύθυνος για την εκτέλεση των προγραμμάτων PLC που δημιουργείτε χρησιμοποιώντας το Editor. Επί του παρόντος, το OpenPLC Runtime υποστηρίζεται επίσημα στις ακόλουθες πλατφόρμες:

- Arduino Uno / Nano / Leonardo / Micro

- Arduino Mega / Due
- Arduino Nano Every / IoT / BLE
- Arduino RB2040 Connect
- Arduino Mkr / Zero / WiFi
- Arduino Pro (Machine Control and EDGE)
- Controllino Maxi / Automation / Mega / Mini
- Productivity Open P1AM
- ESP8266 (nodemcu)
- ESP32
- Raspberry Pi 2 / 3 / 4
- PiXtend
- UniPi Industrial Platform
- Neuron PLC
- FreeWave Zumlink
- FreeWave ZumIQ
- Windows (generic target as a soft-PLC)
- Linux (generic target as a soft-PLC)

Ο OpenPLC Editor είναι το λογισμικό που εκτελείται στον υπολογιστή, χρησιμοποιείται για τη δημιουργία των προγραμμάτων για το PLC. Υποστηρίζει και τις πέντε γλώσσες που ορίζονται στο πρότυπο IEC 61131-3:

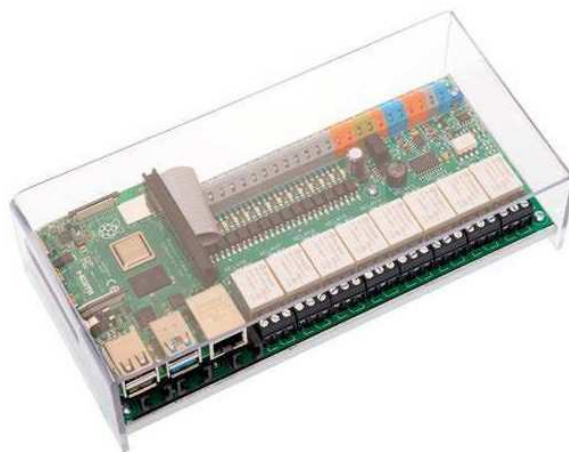
- Ladder Logic (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Structured Text (ST)
- Sequential Function Chart (SFC).

Ο επεξεργαστής είναι πολύ απλός στη χρήση και υποστηρίζει και τις πέντε γλώσσες που ορίζονται στο πρότυπο IEC 61131-3. Το πρότυπο IEC 61131-3 είναι ένα κανονιστικό έγγραφο που παρέχεται από τον οργανισμό προτύπων IEC - Διεθνής Ηλεκτροτεχνική Επιτροπή και περιγράφει ένα πρότυπο για το σχεδιασμό προγραμματιζόμενων ελεγκτών. Το μέρος 3 αυτού του εγγράφου (κοινή ονομασία IEC 61131-3) καθορίζει τη σύνταξη και τη σημασιολογία της γλώσσας προγραμματισμού για τα PLC.

Τα έργα PLC αποθηκεύονται σε φακέλους, όπου το κύριο αρχείο έργου XML ακολουθεί το σχήμα PLCopen TC6-XML.

Στην αγορά διατίθενται διάφορα έτοιμα PLC βασισμένα στο OpenPLC για βιομηχανική ή οικιακή χρήση. Ένα μεγάλο πλεονέκτημα του προγραμματισμού PLC είναι ότι ο προγραμματιστής δεν χρειάζεται να μάθει τις λεπτομέρειες υλικού των γραμμών I/O. Οι αναλογικές και ψηφιακές έξοδοι

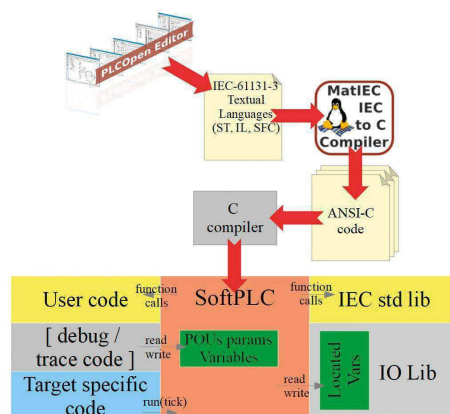
και εισοδοι είναι σαν μεταβλητές. Αυτό είναι επίσης δυνατό με μονάδες που είναι συνδεδεμένες στις υποστηριζόμενες πλατφόρμες υλικού μέσω δικτύου χρησιμοποιώντας το πρωτόκολλο Modbus TCP/IP. Ένα άλλο πλεονέκτημα είναι η συμβατότητα μεταξύ διαφορετικών συστημάτων PLC. Τα προγράμματα για ένα PLC από τον κατασκευαστή Α μπορούν γενικά να χρησιμοποιηθούν για PLC από τον κατασκευαστή Β χωρίς μεγάλη προσπάθεια.



Εικόνα 1.10 : Raspberry Pi PLC Unipi 1.1.

Στην εικόνα 1.10 βλέπουμε ένα PLC της εταιρίας Unipi Technology έκδοση 1.1 με εισόδους 24V και εξόδους ρελέ συνδεδεμένα σε ένα Raspberry Pi 4.

Τα περισσότερα συστήματα PLC υποστηρίζουν γλώσσες προγραμματισμού γραφικών και κειμένου. Το OpenPLC Project υποστηρίζει πλήρως το πρότυπο IEC 61131-3, το οποίο ορίζει τη βασική αρχιτεκτονική λογισμικού και τις γλώσσες προγραμματισμού για PLC.



Εικόνα 1.11 : Επισκόπηση OpenPLC.

Το σύστημα αποτελείται από ένα στοιχείο χρόνου εκτέλεσης, OpenPLC Runtime, το οποίο είναι βασικά το λογισμικό που είναι εγκατεστημένο σε μια από τις υποστηριζόμενες πλατφόρμες υλικού. Αυτό εκτελεί το πρόγραμμα PLC. Το πρόγραμμα επεξεργασίας OpenPLC Editor για τα προγράμματα που θα φορτώσουμε στο PLC εγκαθίσταται στον υπολογιστή είτε με λειτουργικό σύστημα Windows ή Linux για τη σύνταξη του προγράμματος σύμφωνα με το πρότυπο IEC 61131-3.

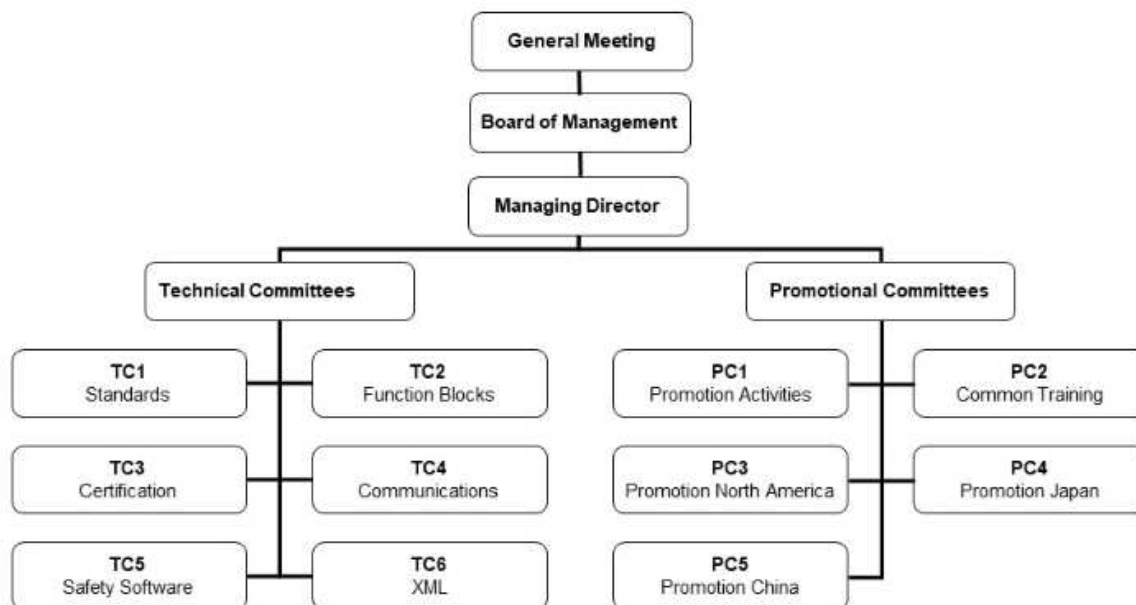
Η παρούσα διπλωματική εργασία θα ασχοληθεί με την δημιουργία ενός PLC με βάση το OpenPLC

Project σε Raspberry Pi 4. Το οποίο είναι απόλυτα κατάλληλο για χρήση ως PLC λόγω της αρχιτεκτονικής του με την υποδοχή GPIO, καθώς και της χαμηλής τιμής.

1.11 Οργανισμός PLCopen

Ο PLCopen (<https://plcopen.org>) είναι ένας ανεξάρτητος παγκόσμιος οργανισμός που παρέχει υποστήριξη στο OpenPLC λογισμικό ανοικτού κώδικα για χρήση σε βιομηχανικό ή οικιακό αυτοματισμό με βάση τις ανάγκες των μελών του. Ιδρύθηκε το 1992 και έχει την έδρα του στην Ολλανδία με υποστηρικτικά γραφεία στις ΗΠΑ, την Ιαπωνία και την Κίνα.

Ο οργανισμός PLCopen ελέγχεται από τα μέλη του μέσω της Γενικής Συνέλευσης, τα οποία μαζί παράγουν τα «περιουσιακά στοιχεία» του οργανισμού PLCopen. Εντός του, όλα τα μέλη αντιμετωπίζονται επί ίσοις όροις, ανεξάρτητα από το μέγεθος της εταιρείας που μετέχει και την χρηματοδοτική συνεισφορά ή την επιχειρησιακή συνεισφορά που προσφέρει κάθε μέλος. Η μόνη διαφορά έγκειται στην επιλογή μεταξύ με δικαίωμα ψήφου ή χωρίς δικαίωμα ψήφου. Αντανακλώντας αυτό το δικαίωμα ψήφου σε θέματα στη Γενική Συνέλευση του οργανισμού PLCopen.



Εικόνα 1.12 : Η Δομή Οργανισμού PLCopen.

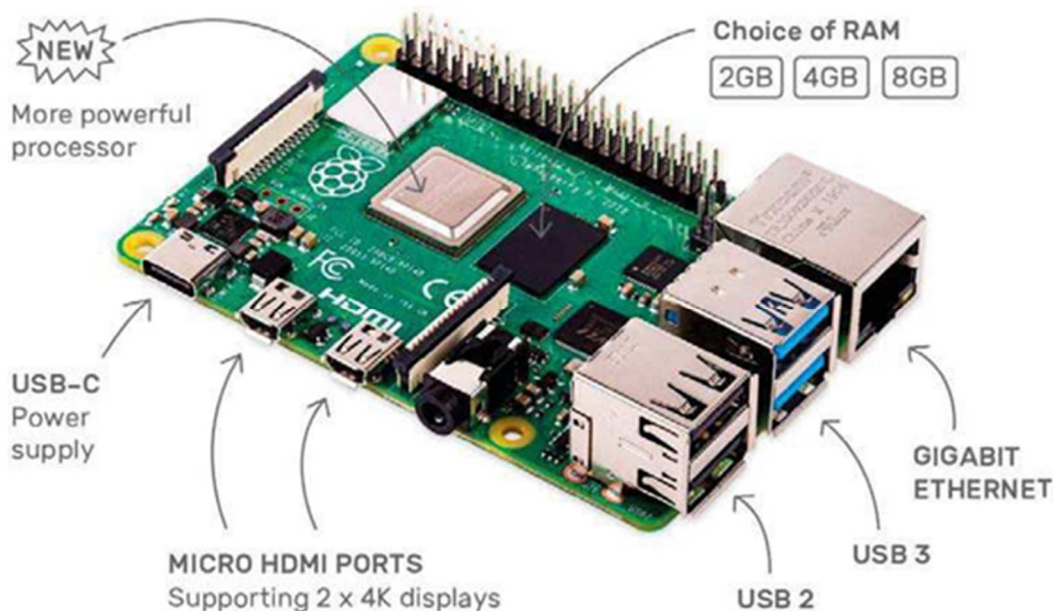
Ο οργανισμός PLCopen ορίζει τα διεθνή πρότυπα για διάφορα θέματα που σχετίζονται με τον προγραμματισμό του OpenPLC σύμφωνα με το IEC 61131-3. Ο στόχος της έκτης π.χ. τεχνικής επιτροπής PLCopen (TC6) είναι να ορίσει μια τυπική μορφή αρχείου, βασισμένη σε XML, για την αποθήκευση προγραμμάτων IEC 61131-3. Ένα ενοποιημένο πρότυπο επιτρέπει στους χρήστες να μπορούν να εισάγουν και να εξάγουν τα προγράμματα PLC που έχουν δημιουργηθεί σε διαφορετικές πλατφόρμες υλικού.

ΚΕΦΑΛΑΙΟ 2

Εγκατάσταση Raspberry Pi 4 και OpenPLC-Project Runtime

2.1 Περιγραφή υλικού Raspberry Pi

Το Raspberry Pi γεννήθηκε ως ιδέα το 2006, στο πανεπιστήμιο του Cambridge, στο τμήμα Computer Laboratory από μια ομάδα εργαζομένων που είχε διαπιστώσει το μικρό ενδιαφέρον των φοιτητών για την πληροφορική. Αυτή η ομάδα σκέφτηκε να δημιουργήσει ένα μικρό υπολογιστή που θα ήταν οικονομικά προσιτός. Με ένα τέτοιο οικονομικό μικροϋπολογιστή, θα μπορούσε να διδάσκεται η πληροφορική στα σχολεία, έχοντας μεγαλύτερο ενδιαφέρον από τους μαθητές. Για τον λόγο αυτό στην αρχή σχεδιαστήκαν πολλά πρωτότυπα του Raspberry Pi. Όμως το 2006 υπήρχαν αρκετοί περιορισμοί για την υλοποίηση του σχεδίου, κυρίως λόγω του υψηλού κόστους και της μικρής επεξεργαστικής ισχύς των μικροεπεξεργαστών. Μετά την άφιξη των smartphones, το κόστος της παραγωγής μικροεπεξεργαστών άρχισε να μειώνεται αρκετά ενώ η επεξεργαστική τους ισχύ αυξανόταν. Τελικά το 2008, δημιουργήθηκε το φιλανθρωπικό ίδρυμα Raspberry Pi Foundation και άρχισε η κατασκευή του πρώτου Raspberry Pi. Σήμερα έχουμε φτάσει στην έκδοση Raspberry Pi 4. Παρά το μικρό του όγκο, το Raspberry Pi 4 διαθέτει τετραπύρηνο επεξεργαστή 64bit τεχνολογίας ARM v8, είναι συγχρονισμένος 1,5GHz, μνήμη RAM από 1 έως 8 GB, δυο θύρες USB 2 και δυο USB 3, δυο εξόδους Micro HDMI 4K, Ethernet, Wifi, BT και GPIO 40 pins γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά. Τροφοδοτείται μέσω USB Type C με 5V DC 2,5A. Το κύριο λειτουργικό του σύστημα που χρησιμοποιείται είναι το Raspberry Pi OS και βασίζεται στο Linux.



Εικόνα 2.1 : Raspberry Pi 4.

Λόγω των αρκετών διεπαφών που διαθέτει το Raspberry Pi 4 και της επεξεργαστικής του ισχύς, μπορεί να χρησιμοποιηθεί ως PLC το οποίο θα έχει πολύ μικρό κόστος σε σχέση με τα κλασικά PLC, ρελέ, κουμπιά, διακόπτες κ.λπ. που μπορούν να συνδεθούν σε αυτόν κατάλληλων διεπαφών.

Οι πτυχές υλικού δεν θα αναλυθούν περαιτέρω στη παρούσα διπλωματική εργασία, διότι ιστότοπος του Raspberry Pi φιλοξενεί πολλά λεπτομερή στοιχεία σχετικά με αυτόν τον μίνι υπολογιστή.

Επίσης μια ακόμη επιλογή για μια πιο συμπαγή λύση, είναι η χρήση του Raspberry Pi Zero 2W. Το οποίο σε δοκιμή που έγινε κατά την διάρκεια αυτής της διπλωματικής εργασίας λόγω έλλειψης του Raspberry Pi 4 από την αγορά, έδειξε ότι όλα λειτουργούσαν κανονικά χωρίς να υπάρχει κανένα πρόβλημα κατά την εγκατάσταση του λογισμικού και των δοκιμών. Οπότε μπορούμε να συμπεράνουμε ότι το OpenPLC σε συνδυασμό με το Pi Zero 2W, αποτελεί μια καλή εναλλακτική λύση με ακόμη χαμηλότερη τιμή εν συγκρίσει με το Raspberry Pi 4.

Επίσης είναι δυνατό να χρησιμοποιηθούν παλαιότερες εκδόσεις υλικού του Raspberry Pi όπως Pi 3 και Pi Model B του 2011. Αυτό βέβαια εξαρτάται από το έργο που πρέπει να υλοποιηθεί, δηλαδή το πλήθος των εισόδων/εξόδων που χρειαστεί για να υλοποιηθεί το έργο, καθώς και τον κύκλο προγράμματος, δηλαδή τον χρόνο εκτέλεσης εντολών που προγράμματος που θα υλοποιηθεί.

Για να χρησιμοποιήσουμε το Raspberry Pi 4 ή Pi Zero 2W ως PLC, θα πρέπει πρώτα να εγκαταστήσουμε το λειτουργικό σύστημα Raspberry Pi OS.



Εικόνα 2.2 : Raspberry Pi Zero 2W.

Πέρα από το χαμηλό κόστος που έχει μια υλοποίηση PLC με υλικό (hardware) Raspberry Pi 4 είναι οι διεπαφές GPIO. Οι GPIO είναι ακίδες που μπορούμε να τις χρησιμοποιήσουμε ως ψηφιακές εισόδους/εξόδους, ενώ μερικές από αυτές ως αναλογικές, οι οποίες μπορούν να εκτελούν διαφορετικές λειτουργίες παράλληλα. Χρησιμοποιούνται έτσι και για άλλους λόγους, όπως η επικοινωνία με μια περιφερειακή συσκευή. Οι ακίδες δέχονται 3.3 V και 16 mA η καθεμία, με εξαίρεση τις ακίδες των 5V, που δέχονται το πολύ 500 mA. Η τροφοδοσία των 5V τραβάει ρεύμα από την τροφοδοσία του Raspberry Pi, με αποτέλεσμα να χρησιμοποιείται ότι απομένει από το ρεύμα που τραβάει ο μίνι υπολογιστής (η συνήθης τροφοδοσία της 2,5A μπορεί να δώσει 500mA). Οι GPIO ορίζονται σε δεκατέσσερις εισόδοι (%IX0.0 έως %IX1.5), έντεκα εξόδοι (%QX0.0 έως %QX1.2) και μια αναλογική (%QW0.0) από το λογισμικό OpenPLC Runtime σύμφωνα με την ακόλουθη (εικόνα 2.3).



Εικόνα 2.3 : OpenPLC Raspberry Pi GPIO map.

Επίσης θα πρέπει να δοθεί ιδιαίτερη προσοχή στην σύνδεση των στοιχείων που θα συνδεθούν στα pins του GPIO για την αποφυγή καταστροφής του Raspberry Pi. Εργαλεία όπως πολύμετρο, προσεκτικός έλεγχος των καλωδίων και της συνδεσμολογίας, χρήση antistatic wrist strap για την αποφυγή στατικού ηλεκτρισμού είναι αναγκαία.

2.2 Εγκατάσταση λειτουργικού συστήματος στο Raspberry Pi

Κατεβάζουμε το Raspberry Pi Imager ([Raspberry Pi OS – Raspberry Pi](#)) ανάλογα με την τύπο (Windows, Linux, Mac OS) του λειτουργικού συστήματος που έχουμε και αποθηκεύουμε ο αρχείο εγκατάστασης στον H/Y. Κάνουμε διπλό κλικ στο αρχείο που κατεβάσαμε και ακολουθούμε τα παρακάτω βήματα. Το λογισμικό Raspberry Pi OS μπορεί να το εγκαταστήσουμε είτε microSD Card είτε σε USB Stick. Προτείνεται η εγκατάσταση του σε microSD Card, οπότε είναι πιθανό να χρειαστεί να συνδεθεί στον H/Y μια SD Card Reader συσκευή.



Εικόνα 2.4 : Εκκίνηση του λογισμικού εγκατάστασης Raspberry Pi Imager.

Στο παράθυρο που εμφανίζεται επιλέγουμε το πλήκτρο [Install] για να αρχίσει η εγκατάσταση του προγράμματος Raspberry Pi Imager. Ο Imager είναι το λογισμικό το οποίο θα εγκαταστήσει στην microSD κάρτα το λειτουργικό σύστημα του Raspberry Pi.



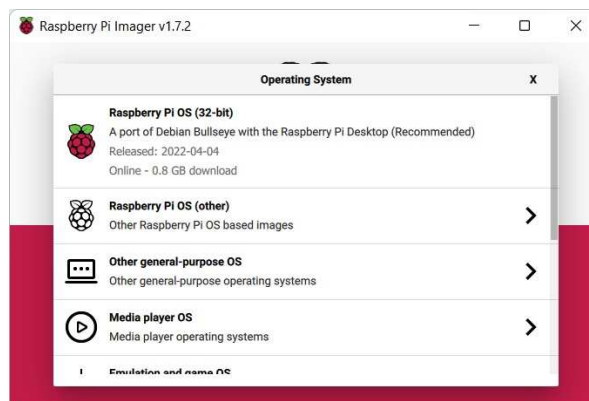
Εικόνα 2.5 : Τέλος εγκατάστασης του λογισμικού Raspberry Pi Imager.

Στο παράθυρο που εμφανίζεται τσεκάρουμε το Run Raspberry Pi Imager και κάνουμε κλικ στο πλήκτρο [Finish].



Εικόνα 2.6 : Επιλογή του λειτουργικού συστήματος.

Επιλέγουμε κάνοντας κλικ στο πλήκτρο [CHOOSE OS] το λειτουργικό σύστημα που θα εγκαταστήσουμε στο Raspberry Pi, (εικόνα 2.4). Από το πλήκτρο [CHOOSE STORAGE] επιλέγουμε την microSD Card που θα το εγκαταστήσουμε, (εικόνα 2.5) και κάνουμε κλικ στο πλήκτρο [WRITE] για να ξεκινήσει η εγκατάσταση, (εικόνα 2.6) στις (εικόνες 2.7 και 2.8) βλέπουμε την πορεία εγκατάστασης.



Εικόνα 2.7 : Προτεινόμενα λειτουργικά συστήματα για το Raspberry Pi.

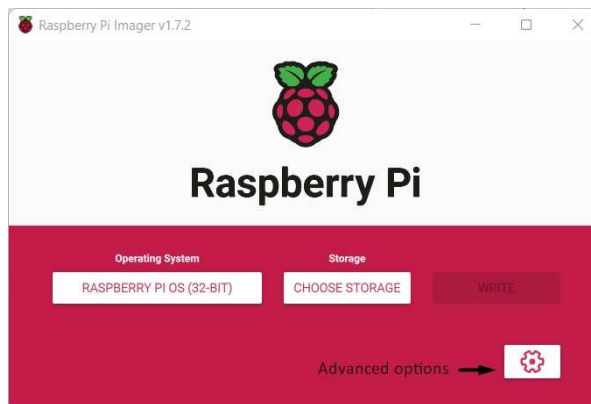
Επιλεγούμε το Raspberry Pi OS (32-bit)



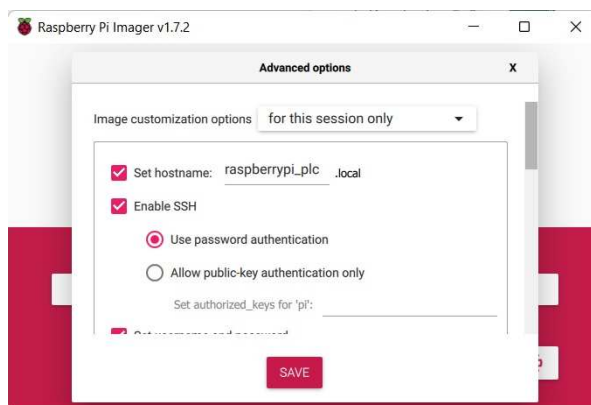
Εικόνα 2.8 : Διαθέσιμα Storage Device.

Επιλεγούμε το Mass Storage Device USB Device- 31.0 GB. Η περιγραφή του storage device είναι υποκειμενική και έχει να κάνει με τον τύπο και το μέγεθος της microSD Card που έχουμε τοποθετήσει στον Η/Υ.

Επιπλέον κατά την διαδικασία εγκατάστασης του λειτουργικού συστήματος του Raspberry Pi μέσω του πλήκτρου (εικόνα 2.6) Advanced options, μας δίνεται η δυνατότητα να ρυθμίσουμε το hostname της συσκευής, ενεργοποίηση απομακρυσμένης πρόσβασης με χρήση του πρωτοκόλλου SSH, το password του χρήστη pi, την πρόσβαση σε ασύρματο δίκτυο Wifi. Καθώς και τις τοπικές ρυθμίσεις όπως το time zone και layout του πληκτρολογίου.

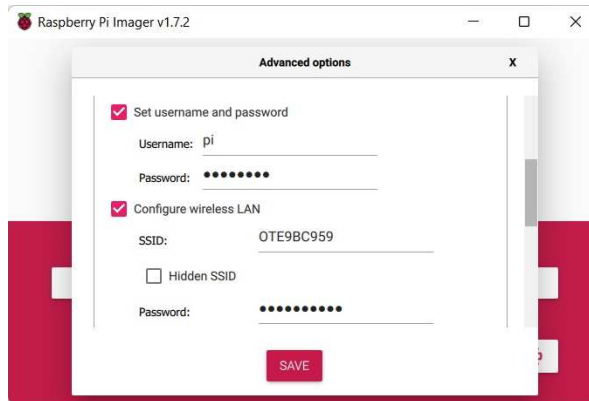


Εικόνα 2.9 : Επιλογή επιπλέον ρυθμίσεων.



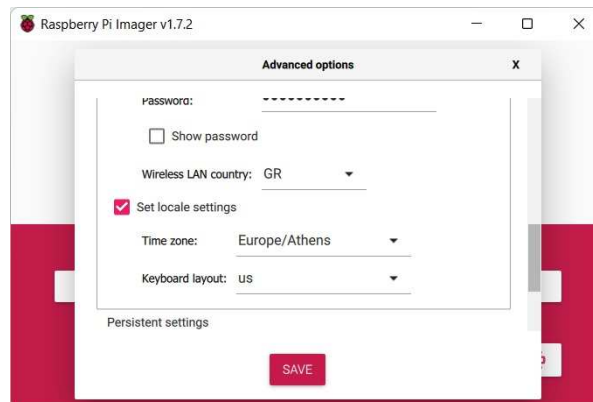
Εικόνα 2.10 : Ορισμός hostname και ενεργοποίηση απομακρυσμένης πρόσβασης.

Δίνουμε το όνομα που θα έχει το Raspberry Pi μετά την εγκατάσταση του λειτουργικού συστήματος. Επιλέγουμε επίσης αν θέλουμε να ενεργοποιήσουμε το πρωτόκολλο SSH για να έχουμε την δυνατότητα απομακρυσμένης διαχείρισης του.

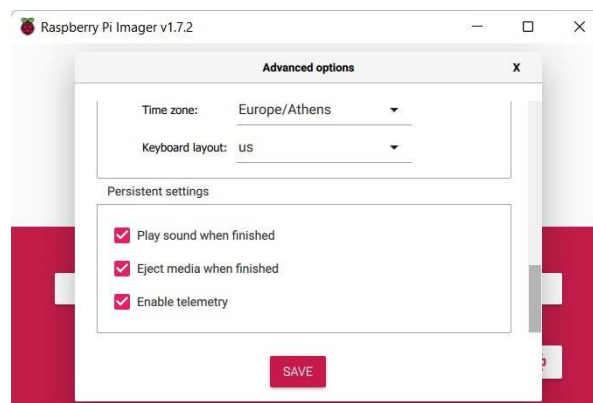


Εικόνα 2.11 : Ορισμός ονόματος χρήστη και κωδικού πρόσβασης και ενημέρωση στοιχείων σύνδεσης σε Wifi.

Επιλέγοντας το “Configure wireless LAN” πληκτρολογούμε το όνομα του ασύρματου δικτύου “SSID” τον κωδικό του “Password”. Τα στοιχεία αυτά θα πρέπει να είναι αληθή και η τοποθέτηση του Raspberry Pi να είναι εντός της εμβέλειας του ασύρματου δικτύου.

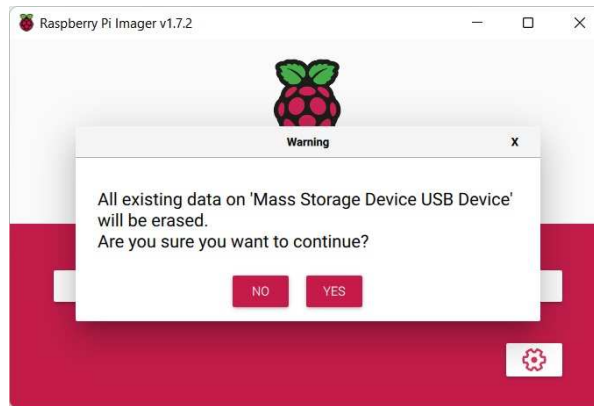


Εικόνα 2.12 : Ενημέρωση στοιχείων σύνδεσης σε Wifi.



Εικόνα 2.13 : Επιλογή Time zone και keyboard layout.

Μετά την συμπλήρωση των παραπάνω στοιχείων κάνουμε κλικ στο πλήκτρο [SAVE] για την αποθήκευση των ρυθμίσεων.

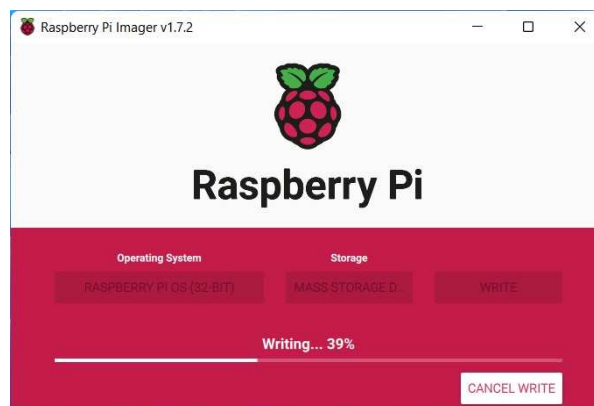


Εικόνα 2.14 : Εκκίνηση της εγκατάστασης.

Επιλέγουμε το πλήκτρο [YES] για την εκκίνηση της εγκατάστασης του λειτουργού στην microSD Card.

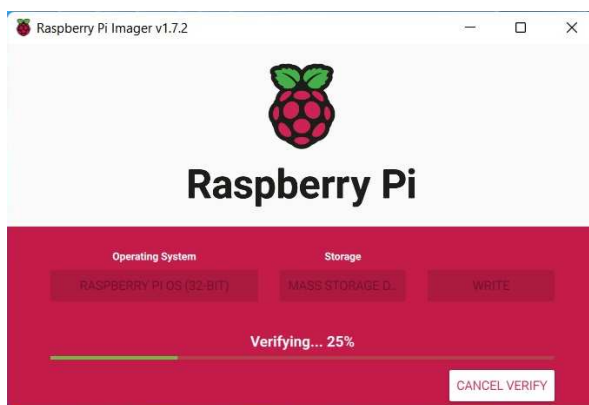


Εικόνα 2.15 : Προετοιμασία εγγραφής του λειτουργικού συστήματος στην microSD card.



Εικόνα 2.16: Εγγραφή του λειτουργικού συστήματος στην microSD card.

Τώρα γίνεται λήψη του λειτουργικού συστήματος και εγγραφή του στην microSD Card. Αυτή η εργασία μπορεί να διαρκέσει έως και 30 λεπτά ανάλογα με την ταχύτητα πρόσβασης στο διαδίκτυο και τον Η/Υ που εκτελεί την εργασία.



Εικόνα 2.17 : Έλεγχος ορθής εγγραφής του λειτουργικού συστήματος στην microSD card.

Για την έξοδο από το Raspberry Pi Imager κάνουμε κλικ στο πλήκτρο [CONTINUE] που θα εμφανιστεί μετά το πέρας της εγκατάστασης του λειτουργικού συστήματος.

Το επόμενο βήμα είναι να τοποθετήσουμε την προγραμματισμένη κάρτα microSD Card στο Raspberry Pi. Στη συνέχεια πρέπει να συνδέσουμε οθόνη, πληκτρολόγιο, ποντίκι, καλώδιο δικτύου αν υπάρχει για να προχωρήσουμε στην αρχική ρύθμιση του Raspberry Pi. Συνδέουμε το τροφοδοτικό 5V DC (2Amp) στο Raspberry Pi. Σε λίγο το λειτουργικό σύστημα του Raspberry Pi θα εκκινήσει.

Μετά την αρχική εκκίνηση, το λειτουργικό σύστημα θα ζητήσει να επιλέξουμε την χώρα και την ζώνη ώρας την οποία θα έχει το σύστημα μας. Μετά από αυτό, θα πρέπει στην αρχική οθόνη να πληκτρολογήσουμε τα στοιχεία πρόσβασης. Το αρχικό όνομα χρήστη είναι pi, και ο κωδικός πρόσβασης είναι raspberry.

Εάν δεν έχουμε σύνδεση Ethernet ή κατά την δημιουργία της της εγκατάστασης του λειτουργικού συστήματος δεν ρυθμίσαμε τη σύνδεση με το Wifi δίκτυο (εικόνα 2.11), τότε μπορούμε να ρυθμίσουμε την πρόσβαση του Raspberry Pi στο δίκτυο Wifi. Πηγαίνοντας στην επάνω δεξιά γωνία της οθόνης δίπλα στο εικονίδιο του ηχείου, κάνουμε κλικ στο εικονίδιο του Wifi. Επιλέγουμε το δίκτυο που θέλουμε να συνδεθούμε, εισαγάγουμε τον κωδικό πρόσβασης σε αυτό. Μετά από λίγο, ολοκληρώνεται η σύνδεση με το Wifi δίκτυο μας.

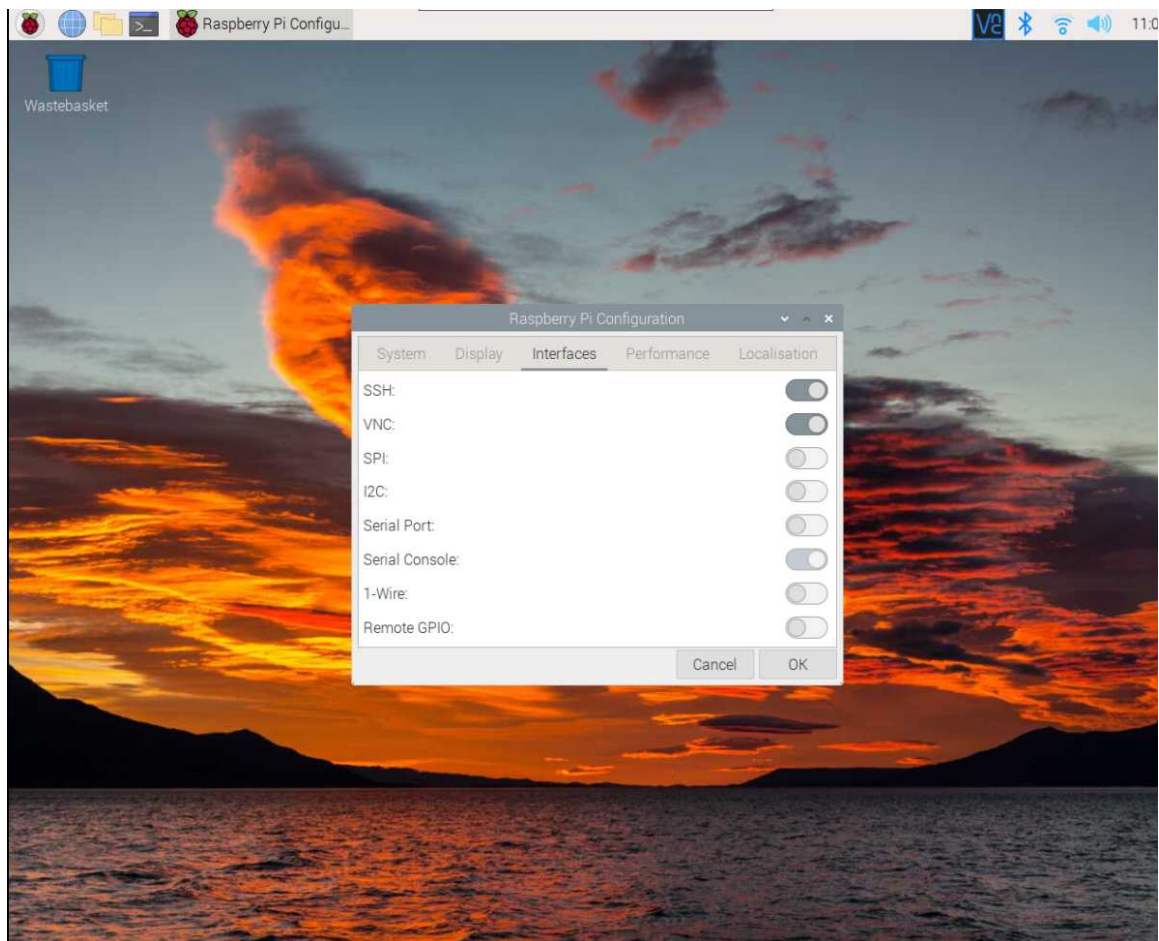
Συνεχίζουμε με την εγκατάσταση των ενημερώσεων που έχει λάβει το λειτουργικό σύστημα. Αυτή η εργασία μπορεί να διαρκέσει αρκετή ώρα ανάλογα με την ταχύτητα του δικτύου μας και το hardware του μοντέλου που έχει υλοποιηθεί το Raspberry Pi. Στο τέλος της εγκατάστασης των αναβαθμίσεων, θα πραγματοποιηθεί επανεκκίνηση του Raspberry Pi.

Μετά την ολοκλήρωση της επανεκκίνησης και την είσοδο μας στο σύστημα, μπορούμε να δούμε την IP διεύθυνση που έχει λάβει το Raspberry Pi κάνοντας κλικ στο εικονίδιο Wifi. Σημειώνουμε την διεύθυνση IP έτσι ώστε, μετά την ενεργοποίηση του λογισμικού απομακρυσμένης διαχείρισης μέσω γραφικού περιβάλλοντος (VNC) ή κονσόλας (SSH), όταν το Raspberry Pi χρησιμοποιηθεί ως PLC, να είναι ευκολότερη η πρόσβαση σε αυτό χωρίς την χρήση περιφερειακού εξοπλισμού.

2.3 Ενεργοποίηση VNC/SSH στο Raspberry Pi.

Ακολουθήστε: Menu → Preference → Raspberry Pi Configuration.

Στη συνέχεια, στο tab Interfaces, ενεργοποιούμε το VNC και το SSH (αν δεν το έχουμε ενεργοποιήσει κατά την διάρκεια της εγκατάστασης του λειτουργικού συστήματος (εικόνα 2.7) κάνοντας κλικ στο αντίστοιχο κουμπί που βρίσκετε στην δεξιά πλευρά του παραθύρου κάθε επιλογής.

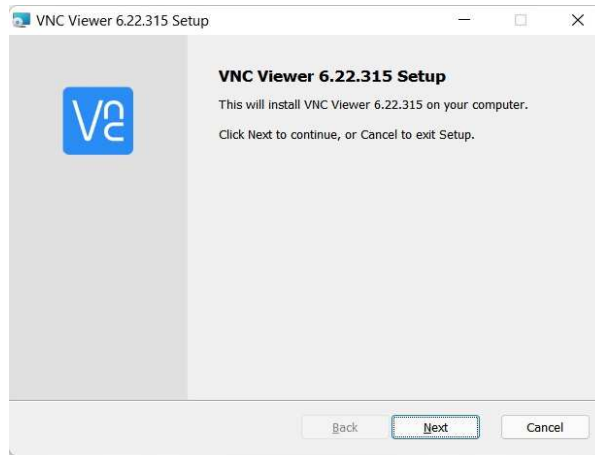


Εικόνα 2.18 : Ενεργοποίηση VNC/SSH.

2.4 Εγκατάσταση VNC Viewer σε Η/Υ

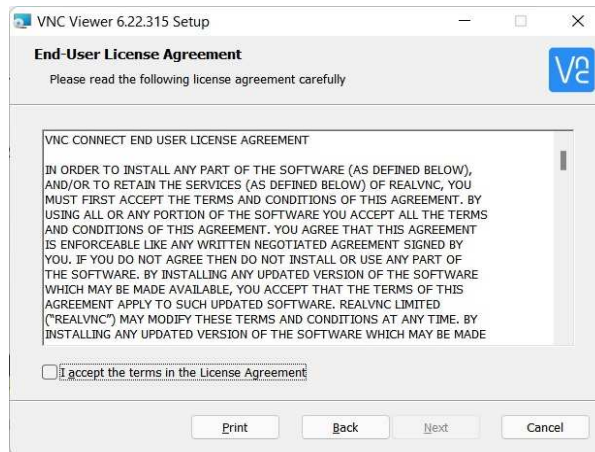
Το VNC είναι ένα λογισμικό που μας επιτρέπει να διαχειριζόμαστε από απόστασεως έναν υπολογιστή μέσω γραφικού περιβάλλοντος, φέρνοντας την επιφάνεια εργασίας του στο δικό μας Η/Υ. Σε αυτήν την περίπτωση του Raspberry Pi VNC Server από άλλον υπολογιστή χρησιμοποιώντας το VNC Viewer. Το VNC Viewer μεταφέρει πληροφορία από το πληκτρολόγιο, ποντίκι του Η/Υ στον VNC Server του Raspberry Pi και λαμβάνει την εικόνα της επιφάνειας εργασίας του Raspberry Pi. Κατεβάζουμε και εγκαθιστούμε στον υπολογιστή μας το VNC Viewer από την διεύθυνση: <https://www.realvnc.com/en/connect/download/viewer/>. Μπορούμε να κατεβάσουμε το κατάλληλο πρόγραμμα προβολής. Χρησιμοποιούμε την πιο πρόσφατη έκδοση για Windows, Linux, MacOS αλλά είναι επίσης διαθέσιμη και για iOS και Android.

Μετά τη λήψη του αρχείου, θα προχωρήσουμε με την εγκατάσταση του λογισμικού, από τον φάκελο download του Η/Υ.



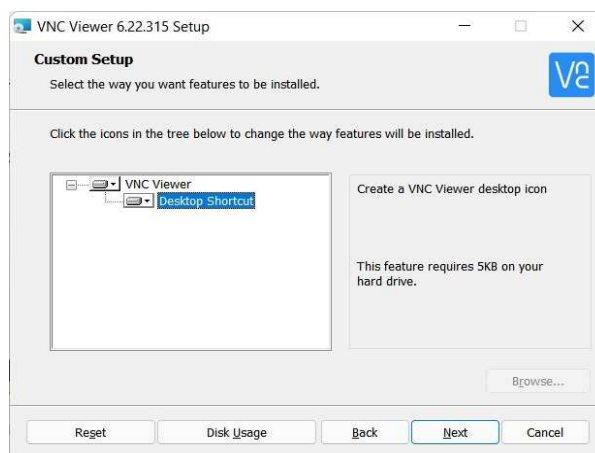
Εικόνα 2.19 : Εγκατάσταση λογισμικού VNC Viewer.

Κάνουμε κλικ στο πλήκτρο [Next].



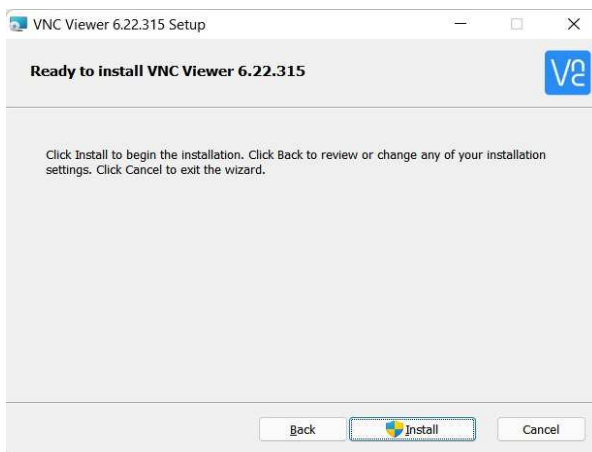
Εικόνα 2.20 : Εγκατάσταση λογισμικού VNC Viewer.

Κάνουμε αποδοχή του License Agreement και κάνουμε κλικ στο πλήκτρο [Next].



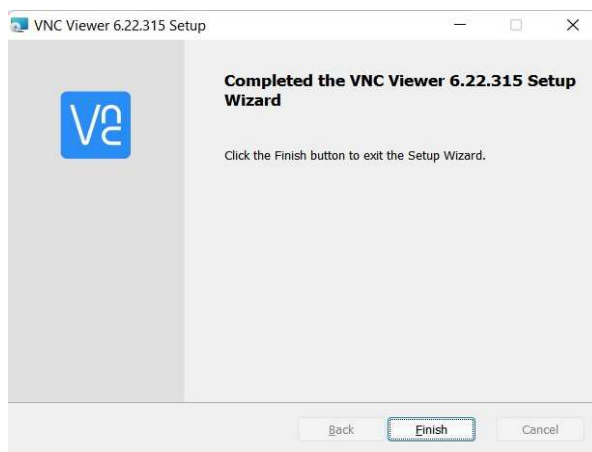
Εικόνα 2.21 : Εγκατάσταση λογισμικού VNC Viewer.

Κάνουμε κλικ στο πλήκτρο [Next].



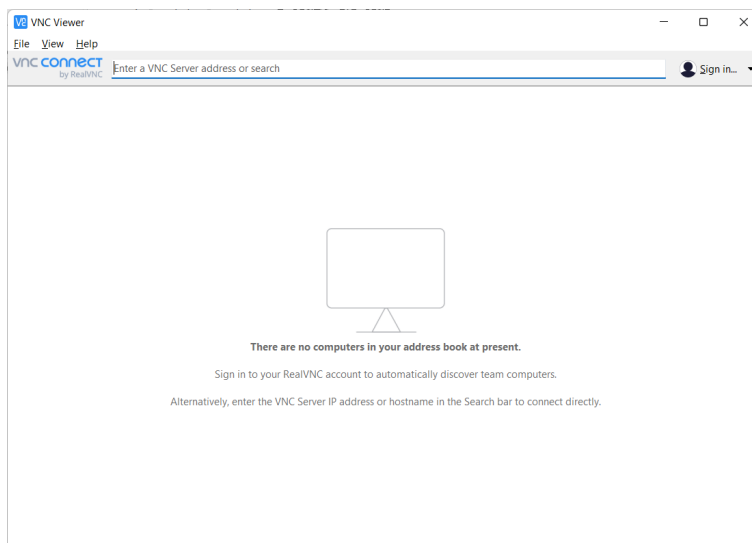
Εικόνα 2.22 : Εγκατάσταση λογισμικού VNC Viewer.

Κάνουμε κλικ στο πλήκτρο [Install].



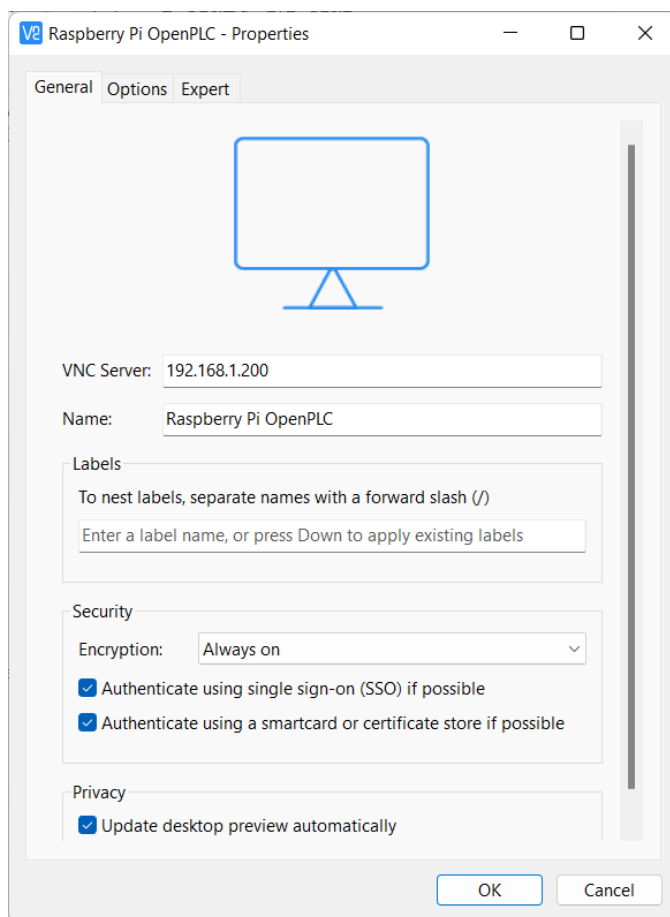
Εικόνα 2.23 : Εγκατάσταση λογισμικού VNC Viewer.

Κάνουμε κλικ στο πλήκτρο [Finish]. Ανοίγουμε το λογισμικό VNC Viewer από την επιφάνεια εργασίας.



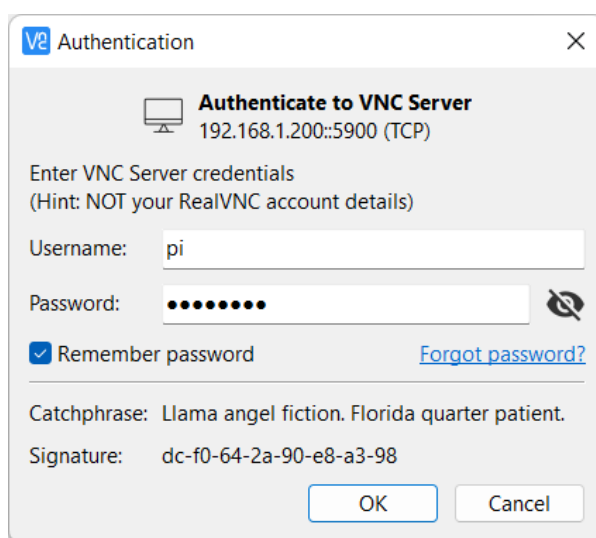
Εικόνα 2.24 : VNC Viewer.

Πηγαίνουμε στο File → New Connection ή πατάμε Ctrl+N από το πληκτρολόγιο.



Εικόνα 2.25 : Διαμόρφωση VNC Viewer.

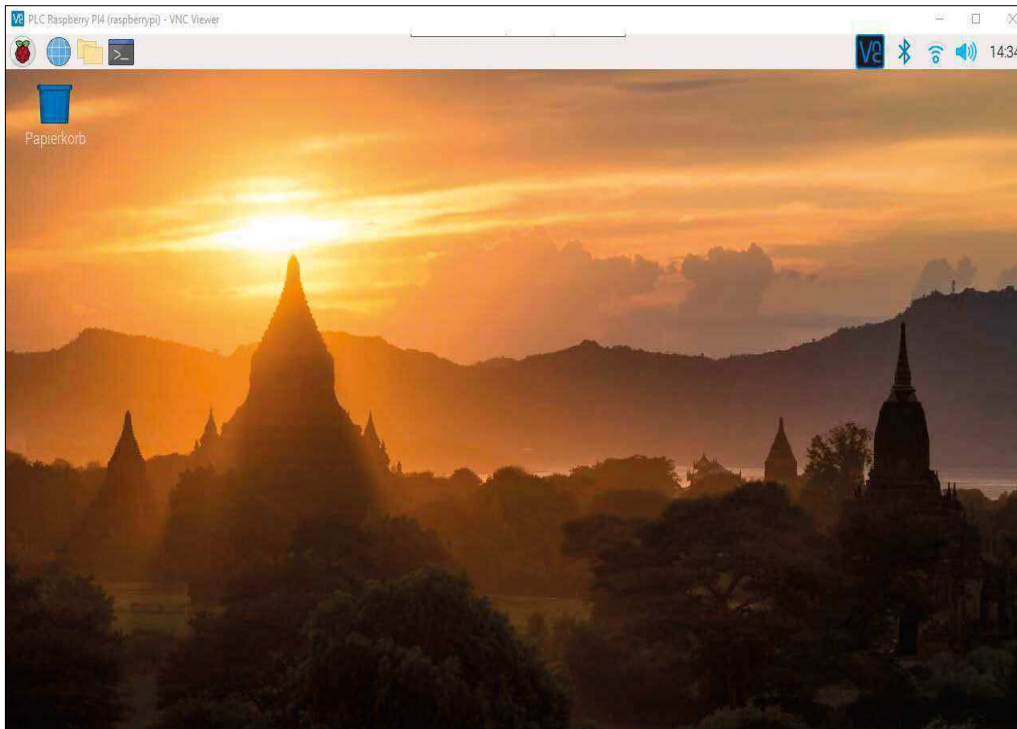
Εισάγουμε την IP διεύθυνση του Raspberry Pi στο πεδίο VNC Server, στο πεδίο Name μπορούμε αν θέλουμε να βάλουμε μια περιγραφή ή όνομα για την σύνδεση η οποία θα εμφανίζεται στην αρχική οθόνη του VNC Viewer. Στο τμήμα Security στο πεδίο Encryption επιλέγουμε το Always on, και κάνουμε κλικ στο πλήκτρο [OK].



Εικόνα 2.26 : Εισαγωγή στοιχείων πρόσβασης.

Εισάγουμε το username και password, αν θέλουμε μπορούμε να επιλέξουμε να αποθηκεύσει το όνομα χρήστη και τον κωδικό πρόσβασης για επόμενες συνδέσεις (προτείνεται μόνο αν στον Η/Υ

έχουμε μόνο εμείς πρόσβαση) και κάνουμε κλικ στο πλήκτρο [OK] .



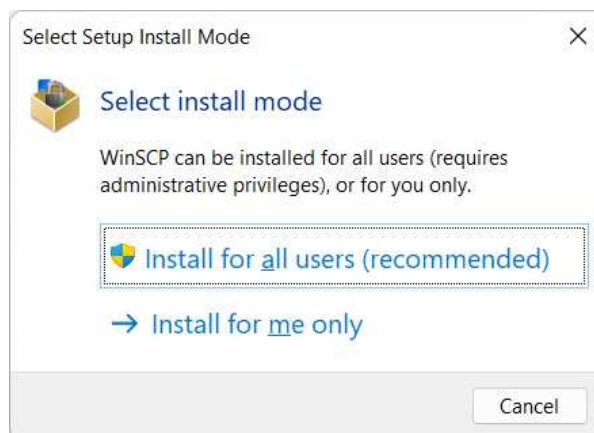
Εικόνα 2.27 : Επιφάνια εργασίας Raspberry Pi OS.

2.5 Εγκατάσταση WinSCP σε Η/Υ

Το WinSCP είναι ένα λογισμικό πελάτη για SFTP και FTP. Είναι ανοιχτού κώδικα για χρήση σε περιβάλλον Windows, και μέσω αυτού θα γίνετε η μεταφορά αρχείων μεταξύ το υπολογιστή και του Raspberry Pi.

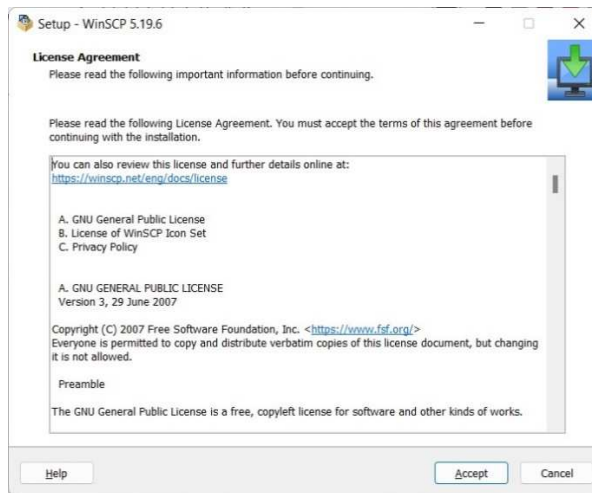
Κατεβάζουμε και εγκαθιστούμε στον υπολογιστή μας το WinSCP από την διεύθυνση: <https://winscp.net/eng/downloads.php>

Μετά τη λήψη του αρχείου, θα προχωρήσουμε με την εγκατάσταση του λογισμικού, από τον φάκελο download του Η/Υ.



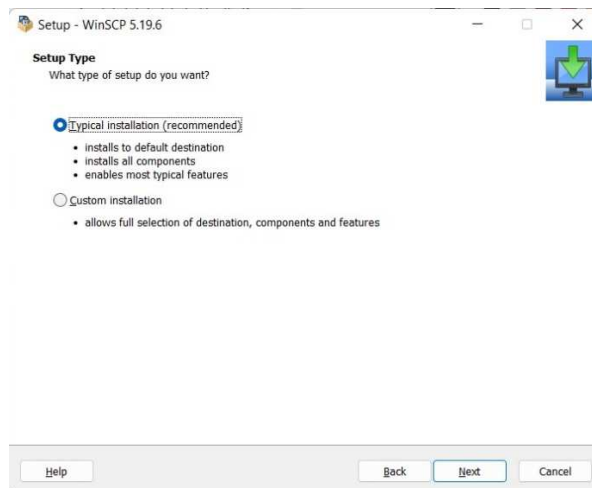
Εικόνα 2.28 : Εγκατάσταση WinSCP.

Κάνουμε κλικ στην επιλογή Install for all users (recommended)

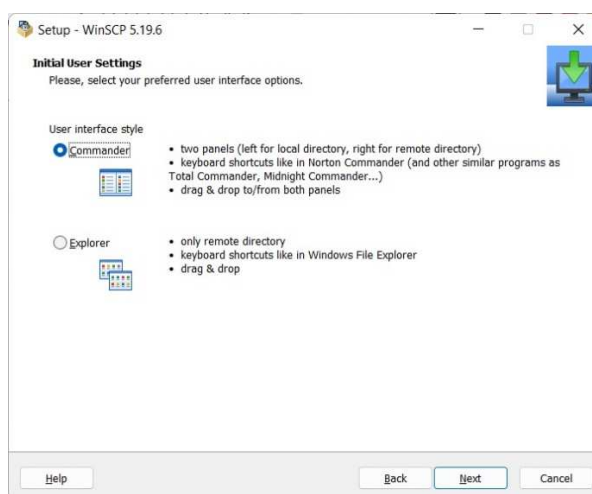


Εικόνα 2.29 : Εγκατάσταση WinSCP.

Κάνουμε κλικ στο πλήκτρο [Accept] για να αποδεχτούμε την άδεια χρήσης του λογισμικού.

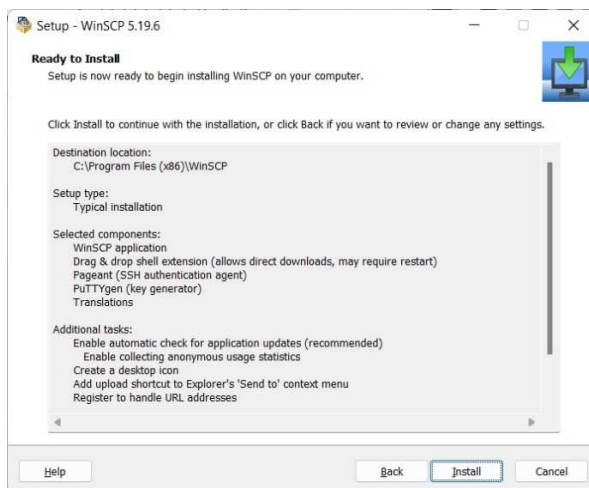


Εικόνα 2.30 : Εγκατάσταση WinSCP.



Εικόνα 2.31 : Εγκατάσταση WinSCP.

Κάνουμε κλικ στο πλήκτρο [Next] (εικόνα 1.29), στο επόμενο παράθυρο επιλέγουμε στο πεδίο User interface style το Commander και κάνουμε κλικ στο πλήκτρο [Next] (εικόνα 1.30).



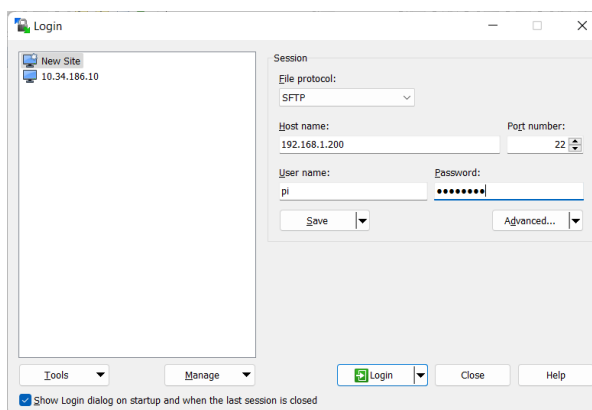
Εικόνα 2.32 : Εγκατάσταση WinSCP.

Κάνουμε κλικ στο πλήκτρο [Install].



Εικόνα 2.33 : Εγκατάσταση WinSCP.

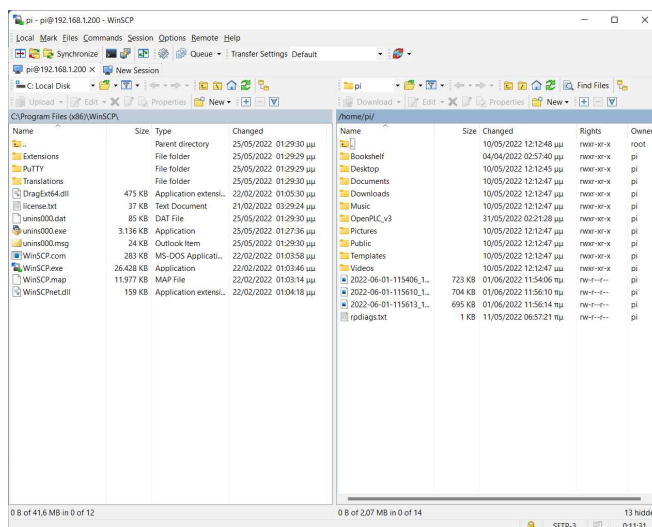
Κάνουμε κλικ στο πλήκτρο [Finish].



Εικόνα 2.34 : Παράθυρο σύνδεσης με το Raspberry Pi.

Εκκινούμε το WinSCP για να ρυθμίσουμε μια νέα σύνδεση με τη διεύθυνση του Raspberry Pi από τον Η/Υ μας. Για παράδειγμα, η διεύθυνση IP του Raspberry Pi είναι η 192.168.1.200 με στοιχεία πρόσβασης username και Password ίδια με αυτά που έχουμε από την εγκατάσταση του Raspberry Pi OS, αν δεν τα έχετε αλλάξει και είναι pi, raspberry αντίστοιχα. Κάνοντας κλικ στο πλήκτρο

[Login] βλέπουμε τους καταλόγους αρχείων του Raspberry Pi (εικόνα 2.35). Επίσης μπορούμε να αποθηκεύσουμε τα στοιχεία πρόσβασης στο Raspberry Pi κάνοντας κλικ στο πλήκτρο [Save].



Εικόνα 2.35 : Σύνδεση H/Y με Raspberry Pi με το WinSCP.

2.6 Εγκατάσταση OpenPLC-Project σε Raspberry Pi

Το OpenPLC Project είναι ένας προγραμματιζόμενος λογικός ελεγκτής (PLC) που χρησιμοποιεί λογισμικό ανοικτού κώδικα για προγραμματισμό και έλεγχο συστημάτων βιομηχανικού και οικιακού αυτοματισμού. Η αρχιτεκτονική του OpenPLC περιλαμβάνει την κεντρική μονάδα επεξεργασίας (CPU), διεπαφές εισόδου/εξόδου (I/O) και την μνήμη προγράμματος.

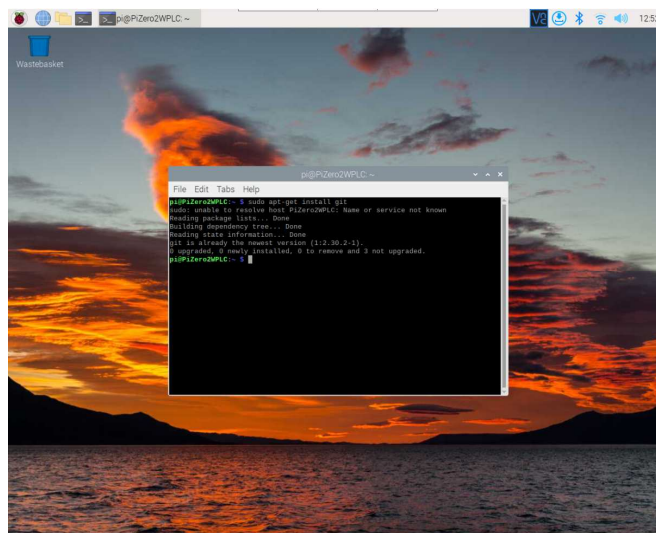
Το λογισμικό OpenPLC είναι χτισμένο σε μια αρθρωτή δομή, η οποία επιτρέπει στους χρήστες να προσθέτουν ή να αφαιρούν λειτουργίες όπως απαιτείται. Είναι επίσης ανεξάρτητο από την πλατφόρμα εγκατάστασής του και μπορεί να τρέξει σε μια μεγάλη ποικιλία υλικού, συμπεριλαμβανομένων των υπολογιστών πλακέτας, όπως το Raspberry Pi ή το Arduino.

Ένα από τα βασικά χαρακτηριστικά του OpenPLC είναι η χρήση λογισμικού ανοικτού κώδικα, το οποίο επιτρέπει μεγαλύτερη ευελιξία και προσαρμογή σε σύγκριση με τα κλασικά PLC. Επιτρέπει στους χρήστες να έχουν πρόσβαση και να τροποποιούν τον πηγαίο κώδικα, διευκολύνοντας τη δημιουργία προσαρμοσμένων λειτουργιών και την ενοποίηση εξωτερικών βιβλιοθηκών.

Σε σύγκριση με τα κλασικά PLC, τα OpenPLC Device είναι γενικά λιγότερο ακριβά και πιο προσαρμόσιμα. Ωστόσο, ενδέχεται να μην έχουν το ίδιο επίπεδο απόδοσης και αξιοπιστίας με τα κλασικά PLC και μπορεί να μην έχουν το ίδιο επίπεδο υποστήριξης και διαθέσιμους πόρους. Επιπλέον, η χρήση των OpenPLC Device μπορεί επίσης να απαιτεί υψηλότερο επίπεδο τεχνικής κατάρτισης και επιπλέον γνώσεις προγραμματισμού H/Y.

Συνολικά, τα OpenPLC Device προσφέρουν μια οικονομικά αποδοτική και ευέλικτη εναλλακτική λύση στα παραδοσιακά PLC, αλλά μπορεί να μην είναι κατάλληλα για όλες τις εφαρμογές βιομηχανικού αυτοματισμού. Η επιλογή χρήσης του OpenPLC εξαρτάται από τις συγκεκριμένες απαιτήσεις του έργου και το επίπεδο τεχνογνωσίας της ομάδας που το χρησιμοποιεί.

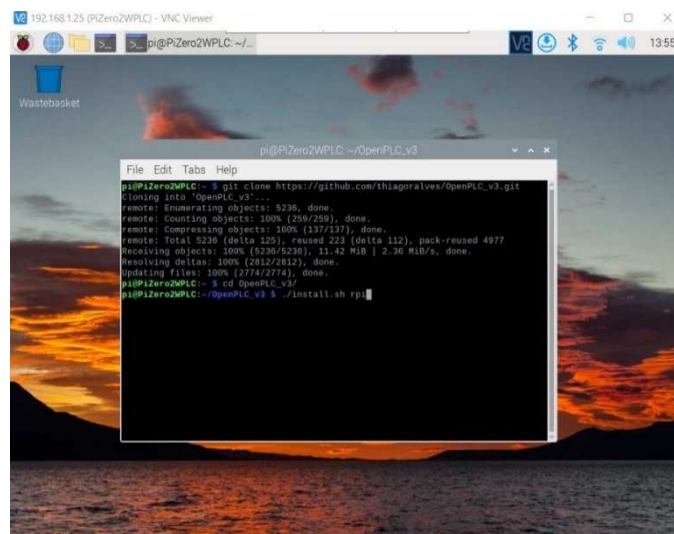
Το επόμενο σημαντικό βήμα είναι να εγκαταστήσουμε το OpenPLC Runtime. Ο ευκολότερος τρόπος για να μεταφέρετε αρχεία OpenPLC στο Raspberry Pi σας είναι να χρησιμοποιήσετε το Git. Κανονικά, το Git είναι προ-εγκατεστημένο στο Raspberry Pi OS. Εάν για κάποιο λόγο δεν έχει εγκατασταθεί το Git στο σύστημά, μπορεί να προστεθεί πληκτρολογώντας κάποιες εντολές. Για να γίνει αυτό, ανοίγουμε ένα παράθυρο Terminal είτε τοπικά στο Raspberry Pi είτε απομακρυσμένα μέσω του προγράμματος προβολής VNC. Πληκτρολογώντας την ακόλουθη εντολή: *sudo apt-get install git*.



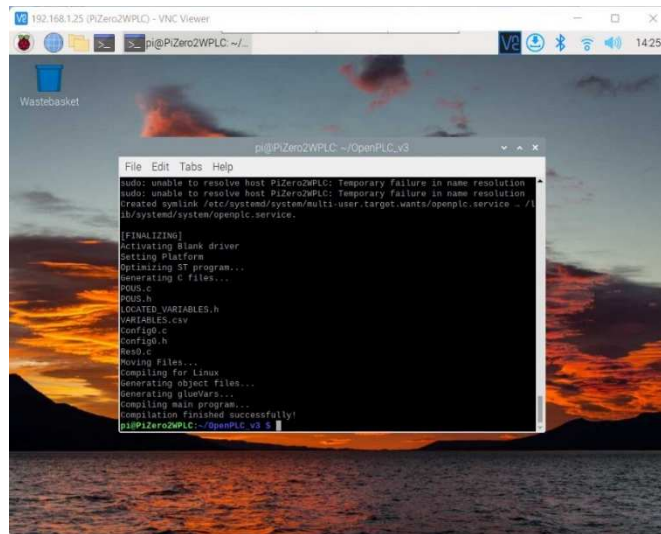
Εικόνα 2.36 : Εγκατάσταση Git.

```
git clone https://github.com/thiagoralves/OpenPLC_v3.git
cd OpenPLC_v3
./install.sh rpi
```

Για την εγκατάσταση του λογισμικού OpenPLC Runtime θα χρειαστείτε τις παραπάνω εντολές. Σε ένα παράθυρο Terminal μπορείτε να πληκτρολογήσετε ή να κάνετε αντιγραφή / επικόλληση τις εντολές με το δεξί πλήκτρο του ποντικού.



Εικόνα 2.37 : Εγκατάσταση του OpenPLC Runtime.

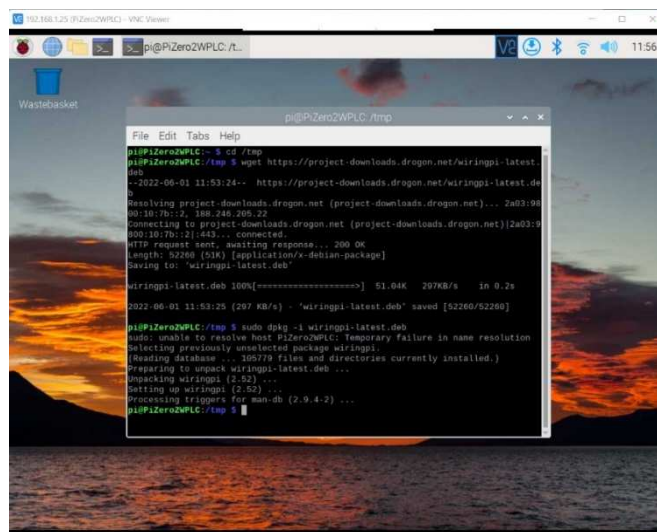


Εικόνα 2.38 : Εγκατάσταση του OpenPLC Runtime.

Η διαδικασία εγκατάστασης του λογισμικού OpenPLC Runtime θα χρειαστεί κάποιο χρόνο για να ολοκληρωθεί. Μόλις ολοκληρωθεί η εγκατάσταση του λογισμικού OpenPLC Runtime θα πρέπει να εγκαταστήσουμε την βιβλιοθήκη WiringPI που είναι υπεύθυνη για την διαχείριση του GPO του Raspberry Pi.

Ακολουθούν οι εντολές για την εγκατάσταση της βιβλιοθήκης WiringPI.

```
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
./install.sh rpi
```



Εικόνα 2.39 : Εγκατάσταση βιβλιοθήκης WiringPI.

Μετά από την εγκατάσταση της βιβλιοθήκης WiringPI θα πρέπει να βεβαιωθούμε ότι λειτουργεί

Στο παράθυρο του Terminal πληκτρολογούμε την εντολή `gpio -v` και θα πρέπει να εμφανίσει την παρακάτω εικόνα.

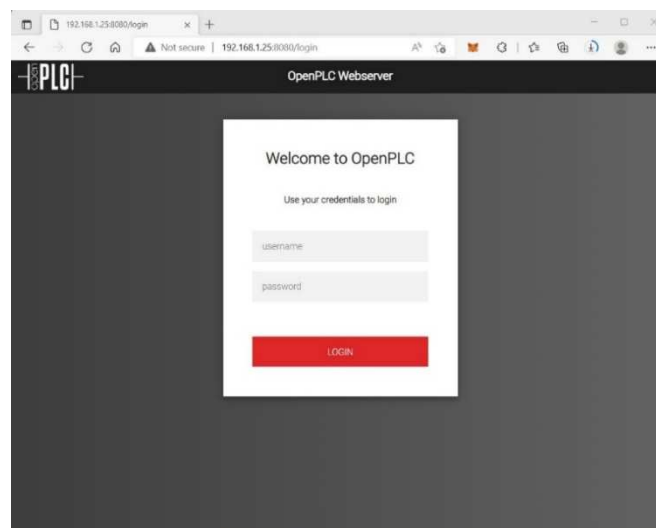
```

192.168.1.25 (PiZero2WPLC) - VNC Viewer
pi@PiZero2WPLC /tmp
File Edit Tabs Help
[HTTP request sent, awaiting response... 200 OK
Length: 52266 (51K) [application/x-debian-package]
Saving to: 'wiringpi-latest.deb'
wiringpi-latest.deb 100% [#####] 51.04K 297KB/s in 0.2s
2022-06-01 11:53:25 (297 KB/s) - 'wiringpi-latest.deb' saved [52266/52266]
pi@PiZero2WPLC /tmp # sudo dpkg -i wiringpi-latest.deb
sudo: unable to resolve host PiZero2WPLC: Temporary failure in name resolution
Selecting previously unselected package wiringpi.
Reading database ... 3057/9 files and directories currently installed.)
Preparing to unpack wiringpi-latest.deb ...
Unpacking wiringpi (2.52) ...
Setting up wiringpi (2.52) ...
Processing triggers for man-db (2.9.4-2) ...
pi@PiZero2WPLC /tmp # gpio -v
gpio version: 2.52
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
Type: Unknown, Revision: 00, Memory: 512MB, Maker: Sony
* Device tree is enabled.
* -> Raspberry Pi Zero 2 W Rev 1.0
* This Raspberry Pi supports user-level GPIO access.
pi@PiZero2WPLC /tmp #
    
```

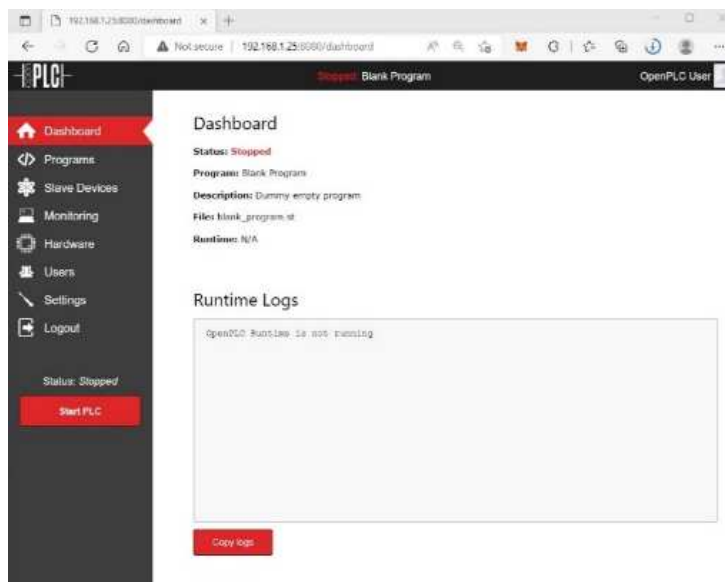
Εικόνα 2.40 : Εγκατάσταση βιβλιοθήκης WiringPi.

Μετά την ολοκλήρωση της εγκατάστασης, θα πρέπει να γίνει επανεκκίνηση του Raspberry Pi. Το OpenPLC Runtime διαθέτει ενσωματωμένο διακομιστή WEB που μας επιτρέπει να έχουμε πρόσβαση στην διαμόρφωση του OpenPLC Runtime καθώς και να ανεβάζουμε τα προγράμματα μας σε αυτό, και να τα θέσουμε σε λειτουργία. Για να συνδεθούμε στο περιβάλλον διαχείρισης του OpenPLC χρειαζόμαστε ένα πρόγραμμα περιήγησης ιστού Web Browser στον Η/Υ μας, εισάγοντας τη διεύθυνση IP του Raspberry Pi και την θύρα 8080 στην γραμμή διεύθυνση (URL). Για παράδειγμα, εάν το Raspberry Pi έχει την IP 192.168.1.25 στο δίκτυό μας, πρέπει να εισαγάγουμε στο πρόγραμμα περιήγησης ιστού τη διεύθυνση (URL) <http://192.168.1.25:8080>. Αν έχουμε πρόσβαση τοπικά στο Raspberry Pi μπορούμε να πληκτρολογήσουμε <http://localhost:8080>



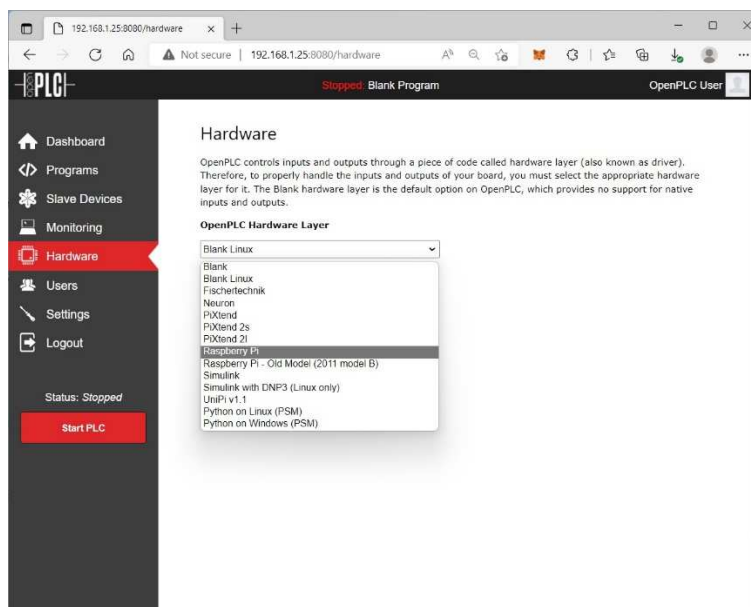
Εικόνα 2.41 : Αρχική σελίδα σύνδεσης OpenPLC.

Το προεπιλεγμένο username και password είναι και στα δύο η λέξη *openplc*. Έχουμε επίσης την δυνατότητα να τα αλλάξουμε μετά τη σύνδεση μας στο περιβάλλον του OpenPLC, εάν το επιθυμούμε ή να προσθέσουμε χρήστη από την επιλογή Users.



Εικόνα 2.42 : Πίνακας ελέγχου χρόνου εκτέλεσης OpenPLC.

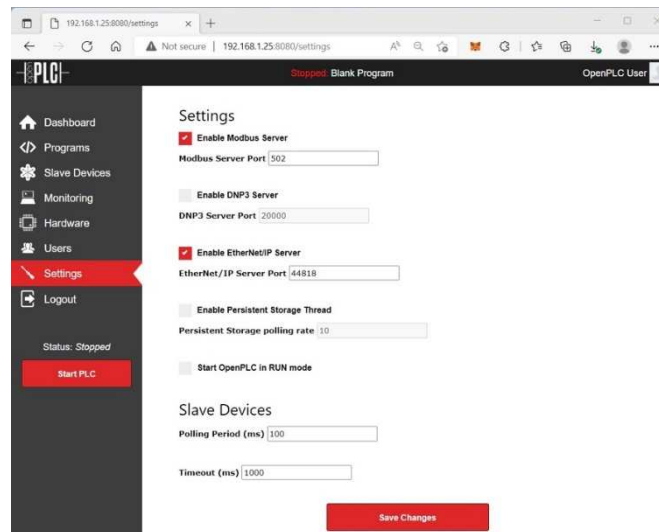
Μετά από μια επιτυχημένη σύνδεση, εμφανίζεται το μενού έναρξης. Τώρα μπορούν να γίνουν όλες οι απαραίτητες ρυθμίσεις στο Raspberry Pi. Από αυτή την σελίδα μπορούμε να διαχειριστούμε όλες τις εργασίες που αφορούν το PLC, όπως να φορτώσουμε τα δικά μας προγράμματα PLC, να ξεκινήσουμε ή να τερματίσουμε την λειτουργία ενός προγράμματος στο PLC, να διαχειριστούμε τη διαμόρφωση των εξωτερικών μονάδων του PLC καθώς και η παρακολούθηση του τρέχοντος προγράμματος.



Εικόνα 2.43 : Επιλογή υλικού για την εκτέλεση του Open PLC Runtime.

Από την επιλογή Hardware επιλέγουμε το υλικό στο οποίο θα εκτελεί το OpenPLC Runtime. Από την λίστα με τις διαθέσιμες επιλογές υλικού εμείς θα επιλέξουμε το Raspberry Pi. Αποθηκεύουμε

τη ρύθμιση με κλικ στο πλήκτρο [Save Changes].



Εικόνα 2.44 : Ρυθμίσεις επικοινωνίας OpenPLC Runtime.

Σε αυτή τη διπλωματική εργασία το DNP3 δεν θα χρησιμοποιηθεί οπότε μπορούμε να το απενεργοποιήσουμε. Αφήνουμε τον διακομιστή Ethernet/IP και Modbus εγκατεστημένα τα οποία θα απαιτηθούν αργότερα για την σύνδεση του Raspberry Pi με το ESP8266 καθώς και για το λογισμικό οπτικοποίησης ScadaBR. Επίσης αν θέλουμε το Raspberry Pi να εκτελεί το πρόγραμμα που του έχουμε φορτώσει μετά από επανεκκίνηση του, μπορούμε να κάνουμε κλικ στο Settings → Start OpenPLC in RUN Mode. Αποθηκεύουμε τις ρυθμίσεις με κλικ στο πλήκτρο [Save Changes] η εγκατάσταση του OpenPLC Runtime έχει ολοκληρωθεί, πλέον μπορούμε να ασχοληθούμε με τον προγραμματισμό του PLC.

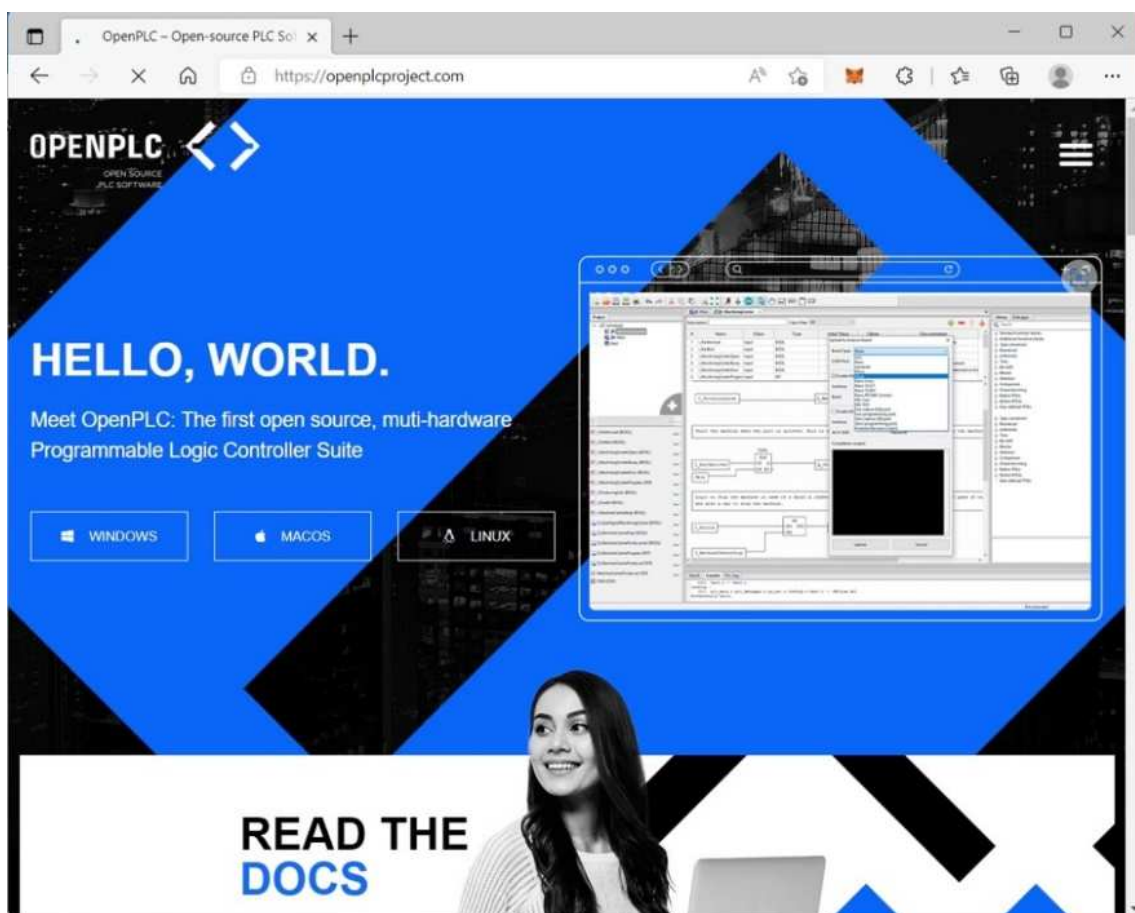
ΚΕΦΑΛΑΙΟ 3

Εγκατάσταση και παρουσίαση του επεξεργαστή OpenPLC Editor

3.1 Εγκατάσταση OpenPLC Editor

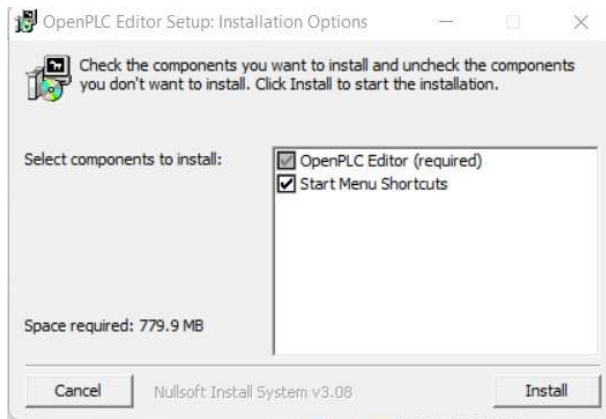
Από τον ιστότοπο <https://openplcproject.com/download/> μπορούμε να κατεβάσουμε το λογισμικό OpenPLC Editor. Το λογισμικό εκτελείται στον Η/Υ μας και μέσω αυτού δημιουργούμε τα προγράμματα για τα PLC. Διαθέσιμες εκδόσεις του λογισμικού υπάρχουν για Windows, Linux και MacOS. Στην παρούσα διπλωματική εργασία θα ασχοληθούμε μόνο με την εγκατάσταση του λογισμικού σε λειτουργικό σύστημα Windows.

Το λογισμικό OpenPLC Editor είναι πολύ απλό στη χρήση και υποστηρίζει, τις πέντε γλώσσες που ορίζονται στο πρότυπο IEC 61131-3 όπως: Ladder Diagram (LD), Function Block Diagram (FBD), Instruction List (IL), Structured Text (ST), και Sequential Function Chart (SFC).



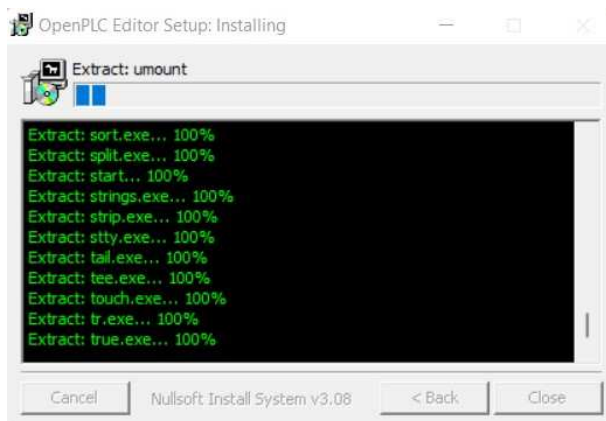
Εικόνα 3.1 : Ο ιστοτοπος του OpenPLC Project.

Μετά τη λήψη του αρχείου προχωράμε στην εγκατάσταση του λογισμικού OpenPLC Editor (το εκτελέσιμο αρχείο της εγκατάστασης θα το βρούμε στον φάκελο Λήψης ή Download του Η/Υ μας). Κάνουμε διπλό κλικ στο αρχείο *OpenPLC Editor for Windows.exe* και η διαδικασία εγκατάστασης του λογισμικού θα ξεκινήσει.

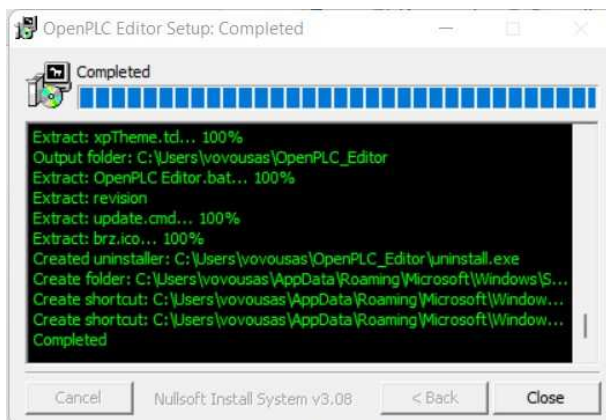


Εικόνα 3.2 : Εγκατάσταση OpenPLC Editor.

Κάνουμε κλικ στο πλήκτρο [Install].



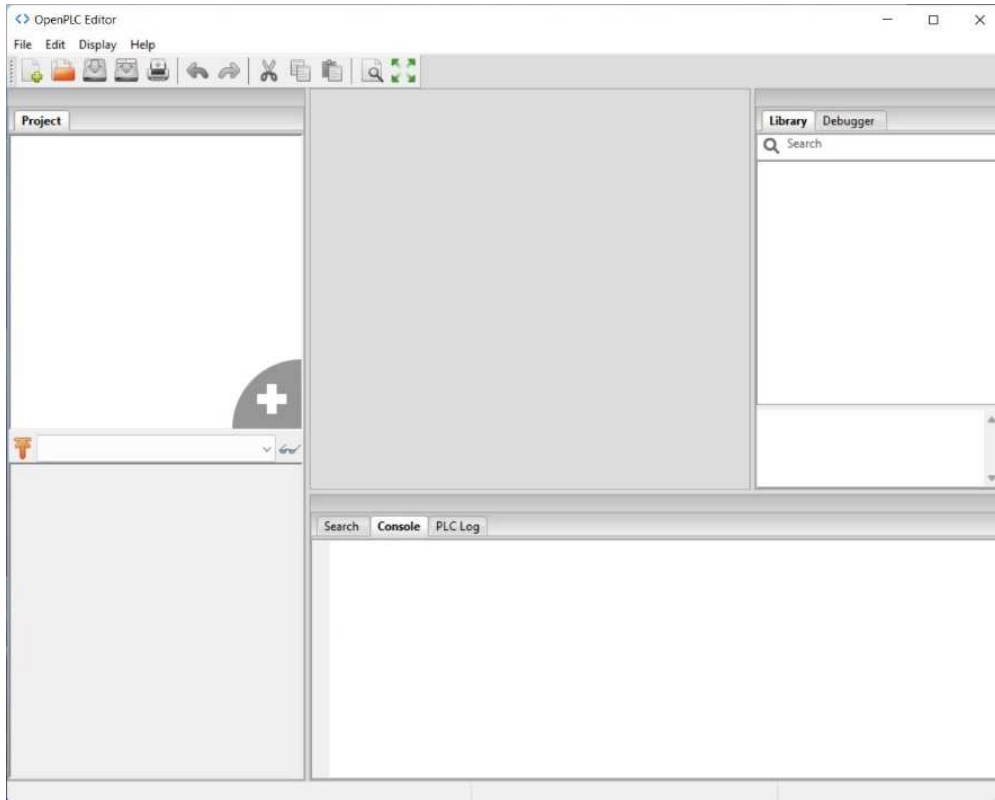
Εικόνα 3.3 : Εγκατάσταση OpenPLC Editor.



Εικόνα 3.4 : Εγκατάσταση OpenPLC Editor.

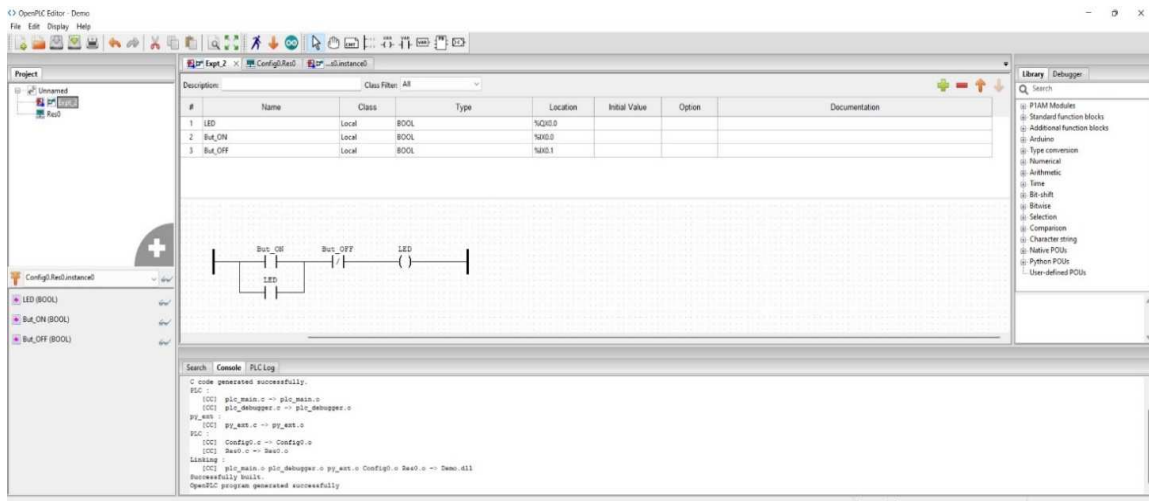
Κάνουμε κλικ στο πλήκτρο [Close].

Έχουμε τελειώσει την εγκατάσταση του OpenPLC Editor. Τώρα μπορούμε να ανοίξουμε το λογισμικό κάνοντας διπλό κλικ στο εικονίδιο που έχει δημιουργηθεί στην επιφάνεια εργασίας του Η/Υ.



Εικόνα 3.5 : Αρχική σελίδα OpenPLC Editor.

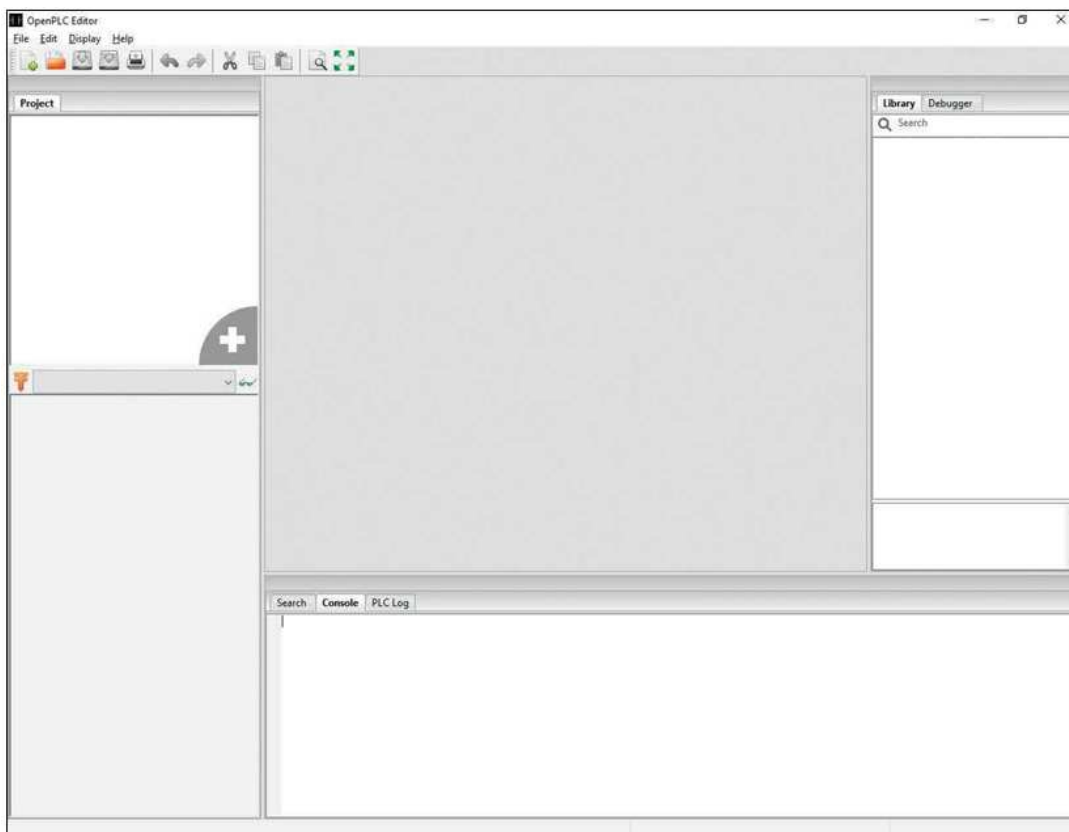
Στο πρόγραμμα επεξεργασίας μπορούμε στη συνέχεια να κάνουμε κλικ στον κατάλογο κάτω από το "File Open" και να επιβεβαιώσουμε με "OK". Τώρα το πρόγραμμα φορτώνεται.



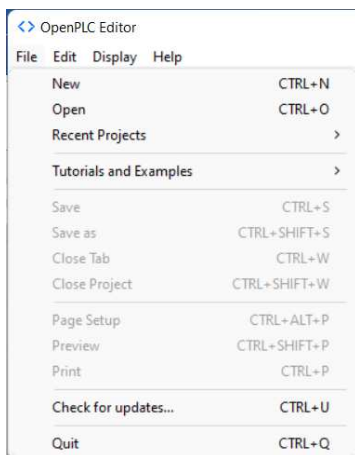
Εικόνα 3.6 : OpenPLC Editor με πρόγραμμα σε γλώσσα Ladder Diagram.

3.2 Περιγραφή του OpenPLC Editor

Τον OpenPLC Editor τον χρησιμοποιούμε για να γράψουμε τα προγράμματα που θα φορτωθούν στο Raspberry Pi OpenPLC και όχι μόνο. Εφόσον το OpenPLC είναι λογισμικό ανοικτού κώδικα μπορεί να εγκατασταθεί σε διαφορετικές συσκευές και λειτουργικά συστήματα. Παρακάτω ακολουθεί περιγραφή των μενού, της γραμμής εικονιδίων και των υποστηριζόμενων γλωσσών προγραμματισμού του λογισμικού OpenPLC Editor.



Εικόνα 3.7 : Αρχική οθόνη του OpenPLC Editor.



Εικόνα 3.8 : Μενού File.

Μενού File μπορούμε να:

New CTRL+N: Δημιουργούμε ένα νέο πρόγραμμα

Open CTRL+O: Φορτώνουμε ένα ήδη υπάρχων πρόγραμμα

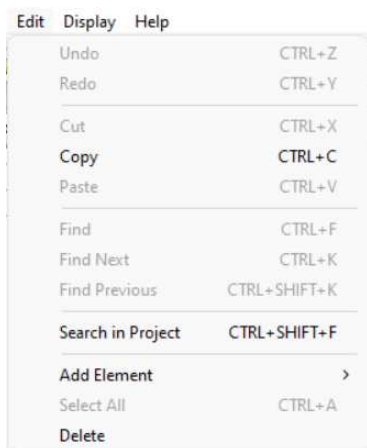
Recent Project: Μας δείχνει τα πρόσφατα προγράμματα που έχουμε δημιουργήσει και τα οποία μπορούμε να φορτώσουμε.

Tutorials and Examples: Εκπαιδευτικά προγράμματα και παραδείγματα.

Check for updates CTRL+U: Έλεγχος για αναβαθμίσεις του OpenPLC Editor.

Quit CTRL+Q: Τερματισμός του OpenPLC Editor.

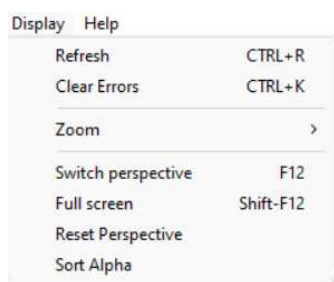
Οι γκριζαρισμένες επιλογές ενεργοποιούνται όταν δημιουργήσουμε ένα νέο πρόγραμμα ή ανοίξουμε ένα υπάρχων. Αφορούν την διαχείριση του προγράμματος όπως αποθήκευση, κλείσιμο, εκτύπωση κλπ.



Εικόνα 3.9 : Μενού Edit.

Μενού Edit, μπορούμε να κάνουμε αντιγραφή, επικόλληση, διαγραφή, προσθήκη, αναζήτηση διάφορων αντικειμένων που έχουμε στο πρόγραμμα το οποίο επεξεργαζόμαστε.

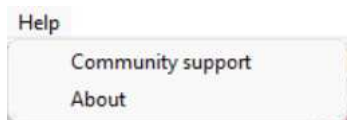
Οι παραπάνω επιλογές ενεργοποιούνται όταν δημιουργήσουμε ένα νέο πρόγραμμα ή ανοίξουμε ένα υπάρχων.



Εικόνα 3.10 : Μενού Display.

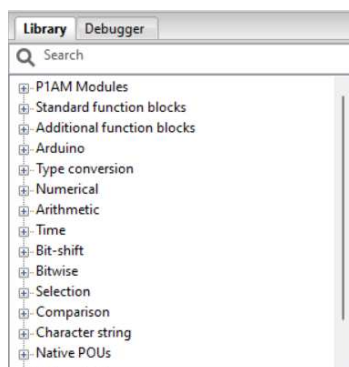
Μενού Display, μπορούμε να αλλάξουμε την προβολή του OpenPLC Editor σε πλήρη οθόνη και πάλι σε κανονική. Να αλλάξουμε το μέγεθος της εμφάνισης του προγράμματος μας.

Οι παραπάνω επιλογές ενεργοποιούνται όταν δημιουργήσουμε ένα νέο πρόγραμμα ή ανοίξουμε ένα υπάρχων.



Εικόνα 3.11 : Μενού Help.

Μενού Help, στο Community support μπορούμε να βρούμε την ιστοσελίδα της κοινότητας που υποστηρίζει την ανάπτυξη και μετάφραση του user interface για το OpenPLC Editor. Στο About βλέπουμε διάφορα στοιχεία για το λογισμικό OpenPLC Editor όπως π.χ. τον αριθμό της έκδοσης κ.α.



Εικόνα 3.12 : Μενού Library.

Μενού Library, μπορούμε να επιλέξουμε πολλά έτοιμα μπλοκ λειτουργιών για τα προγράμματα μας.

Τα παραπάνω ενεργοποιούνται όταν δημιουργήσουμε ένα νέο πρόγραμμα ή ανοίξουμε ένα υπάρχων.



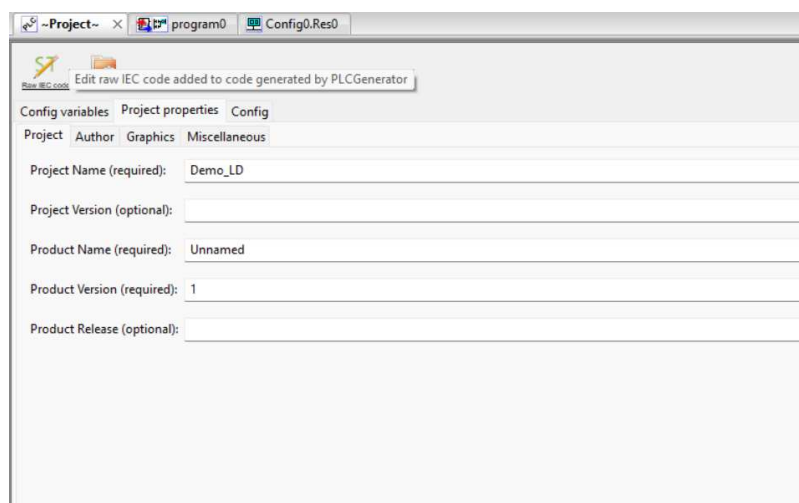
Εικόνα 3.13 : Δομή Project.

Από την καρτέλα Project μπορούμε κάνοντας διπλό κλικ στο όνομα του προγράμματος (π.χ. Demo_LD) να δούμε την καρτέλα με τις ρυθμίσεις του προγράμματος μας, να εισάγουμε διάφορες πληροφορίες γ' αυτό όπως στοιχεία εταιρίας, προγραμματιστή, η έκδοση του προγράμματος για την τεκμηρίωση του κ.α.

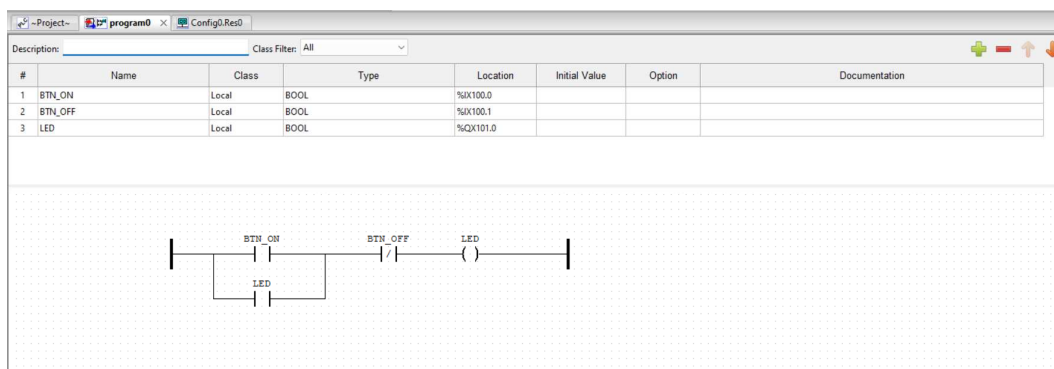
Με διπλό κλικ στο program0 βλέπουμε την καρτέλα στην οποία μπορούμε να γράψουμε το πρόγραμμα μας.

Με διπλό κλικ στο Res0 βλέπουμε την καρτέλα στην οποία μπορούμε να ρυθμίσουμε τον κύκλο του προγράμματος π.χ. 20 ms που σημαίνει ότι το πρόγραμμα μας θα εκτελείτε 5 φορές ανά δευτερόλεπτο.

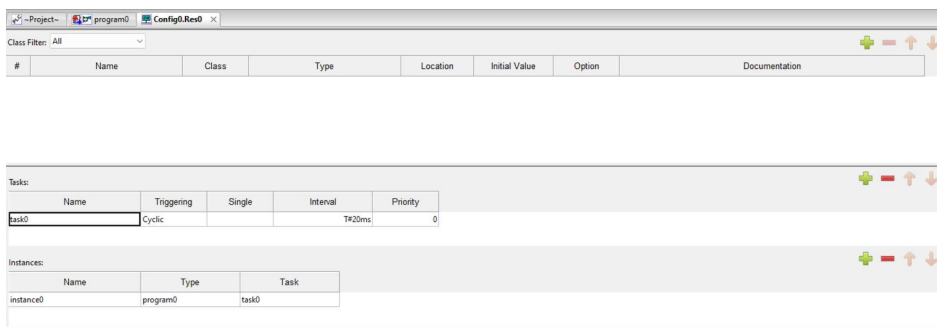
(Βλέπε τις παρακάτω εικόνες 3.15, 3.16, 3.17)



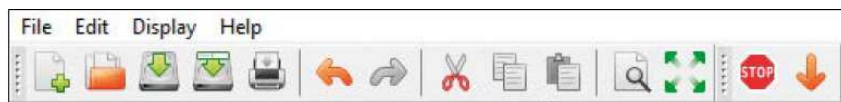
Εικόνα 3.14 : Ιδιότητες προγράμματος.



Εικόνα 3.15 : Χώρος εργασίας / δημιουργίας προγράμματος.



Εικόνα 3.16 : Καρτέλα ρυθμίσεων του προγράμματος.



Εικόνα 3.17 : Γραμμή εικονιδίων OpenPLC Editor.

Εικονίδιο	Περιγραφή	Λειτουργία εικονιδίου
	Νέο πρόγραμμα	Δημιουργούμε ένα νέο πρόγραμμα
	Άνοιγμα προγράμματος	Ανοίγουμε ένα υπάρχων πρόγραμμα
	Αποθήκευση προγράμματος	Αποθηκεύουμε το τρέχων πρόγραμμα
	Αποθήκευση ως	Αποθηκεύουμε το τρέχων πρόγραμμα με άλλο όνομα
	Εκτύπωση	Εκτυπώστε το τρέχων πρόγραμμα
	Αναίρεση	Αναίρεση της τελευταίας αλλαγής στην επιφάνεια εργασίας
	Επανάληψη	Επαναφορά αλλαγής στην επιφάνεια εργασίας
	Αποκοπή	Αποκοπή των επιλεγμένων αντικειμένων στην επιφάνεια εργασίας
	Αντιγραφή	Αντιγραφή αντικειμένου
	Επικόλληση	Επικόλληση αντικειμένων από το πρόχειρο στο πρόγραμμα εργασίας
	Αναζήτηση	Αναζήτηση αντικειμένων στο πρόγραμμα
	Προσομοίωση προγράμματος	Προσομοίωση του προγράμματος που έχουμε αναπτύξει
	Δημιουργία εκτελέσιμου κώδικα για το OpenPLC Runtime.	Μεταφράζει το πρόγραμμα σε αρχείο ST

Παραπάνω έχουμε την επεξήγηση των βασικών εικονιδίων του λογισμικού OpenPLC Editor. Το

λογισμικό μας δίνει τη δυνατότητα να γράψουμε το πρόγραμμά μας σε όλες τις διαθέσιμες γλώσσες προγραμματισμού που υπάρχουν για τα PLC σύμφωνα με το πρότυπο IEC 61131-3 (βλέπε παρακάτω πίνακα). Κατά την δημιουργία ενός νέου προγράμματος μας δίνεται η δυνατότητα να επιλέξουμε τη γλώσσα που θα γράψουμε το πρόγραμμά μας. Ανάλογα με την γλώσσα μας εμφανίζει μαζί με την γραμμή των βασικών εικονιδίων και τα εικονίδια – λειτουργίες της γλώσσας που θα χρησιμοποιήσουμε στο πρόγραμμά μας.

Εικονίδιο	Συντομογραφία	Πλήρες όνομα
	LD	Ladder Diagram (Διάγραμμα σκάλας)
	FBD	Function Block (Μπλοκ λειτουργιών)
	IL	Instruction List (Λίστα οδηγιών)
	ST	Structured Text (Δομημένο κείμενο)
	SFC	Sequential Function Chart (Διαδοχικό διάγραμμα λειτουργιών)

Ακολουθούν η γραμμή βασικών και επιπλέον εικονιδίων για κάθε μια από τις πέντε γλώσσες προγραμματισμού PLC που υποστηρίζονται από τον OpenPLC Editor.



Εικόνα 3.18 : Γραμμή εικονιδίων για γλώσσα Ladder Logic.



Εικόνα 3.19 : Γραμμή εικονιδίων για γλώσσα Structured Text.



Εικόνα 3.20 : Γραμμή εικονιδίων για γλώσσα Function Block Diagram.



Εικόνα 3.21 : Γραμμή εικονιδίων για γλώσσα Sequential Function Chart.

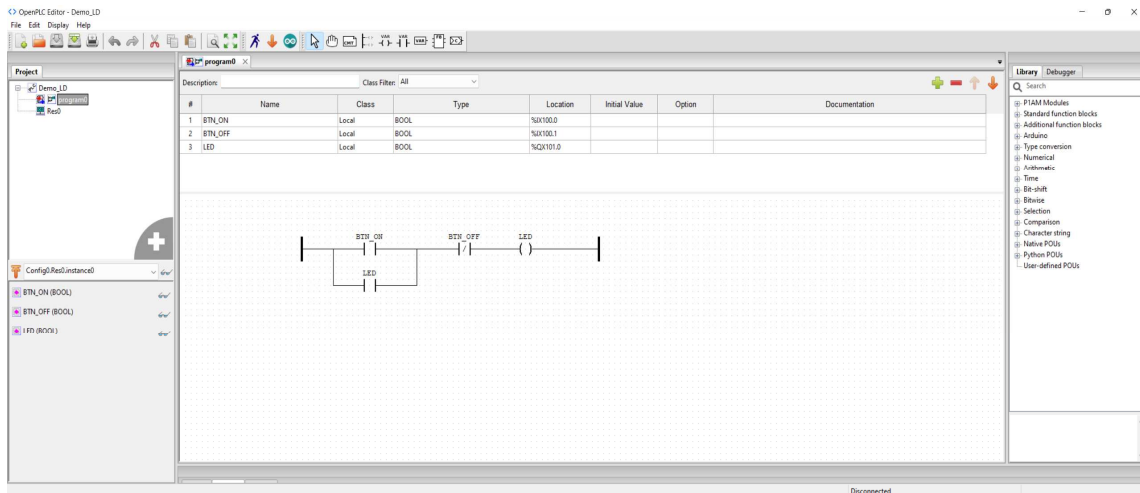


Εικόνα 3.22 : Γραμμή εικονιδίων για γλώσσα Instruction List.

3.3 Γλώσσες προγραμματισμού σε OpenPLC Editor

Οι γλώσσες προγραμματισμού PLC που υποστηρίζονται από τον OpenPLC Editor είναι οι ακόλουθες.

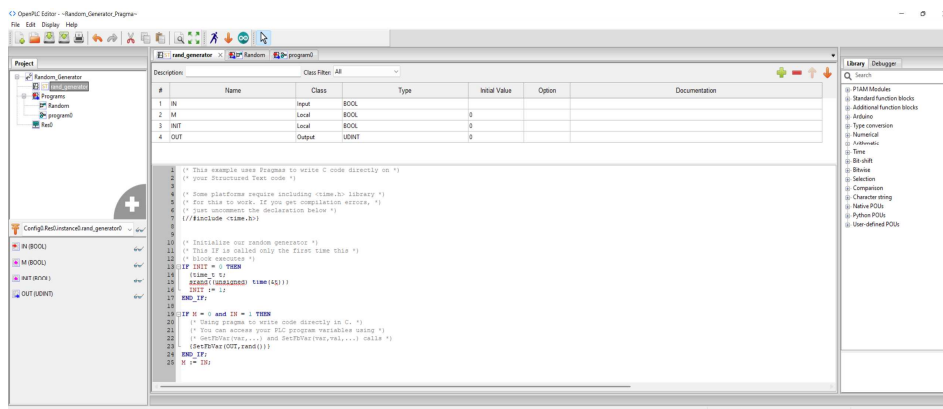
3.3.1 Διάγραμμα Σκάλας (LD)



Εικόνα 3.23 : Παράδειγμα προγράμματος Διαγράμματος Σκάλας (LD).

Χρησιμοποιώντας την γλώσσα προγραμματισμού LD, το πρόγραμμα μας αναπαρίστανται με γραφικά σύμβολα, όμοια με αυτά των ηλεκτρολογικών κυκλωμάτων. Όπως μπάρες τροφοδοσίας επαφές και πηνία. Ένα πρόγραμμα μπορεί να αποτελείται από πολλά τέτοια δίκτυα που μπορούν να αλληλοεπιδρούν μεταξύ τους. Υπάρχουν επίσης πολλά μπλοκ λειτουργιών που μπορούμε να προσθέσουμε στο πρόγραμμα μας. Αυτά τα βρίσκουμε στην καρτέλα Library του OpenPLC Editor.

3.3.2 Δομημένο Κείμενο (ST)

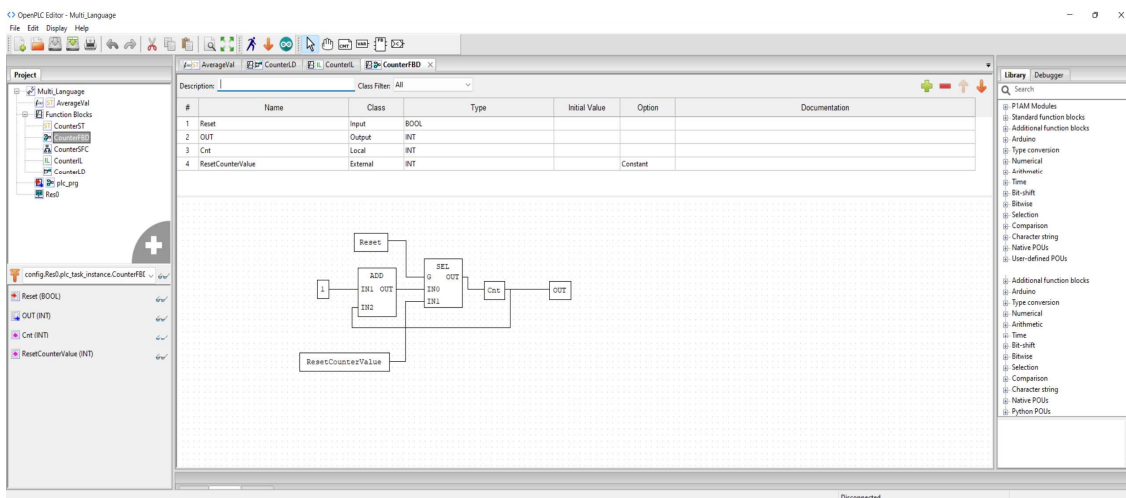


Εικόνα 3.24 : Παράδειγμα προγράμματος Δομημένου Κειμένου (ST).

Η γλώσσα προγραμματισμού ST, είναι σε μεγάλο βαθμό όμοια με την γλώσσα προγραμματισμού Pascal. Είναι μια σύγχρονη γλώσσα προγραμματισμού και με την χρήση της μπορούμε να δημιουρ-

γήσουμε σαφή και δομημένα προγράμματα. Εκτός από το "IF THEN" υπάρχουν επίσης δομές ελέγχου "DO WHILE" και "REPEAT UNTIL" για τα προγράμματα.

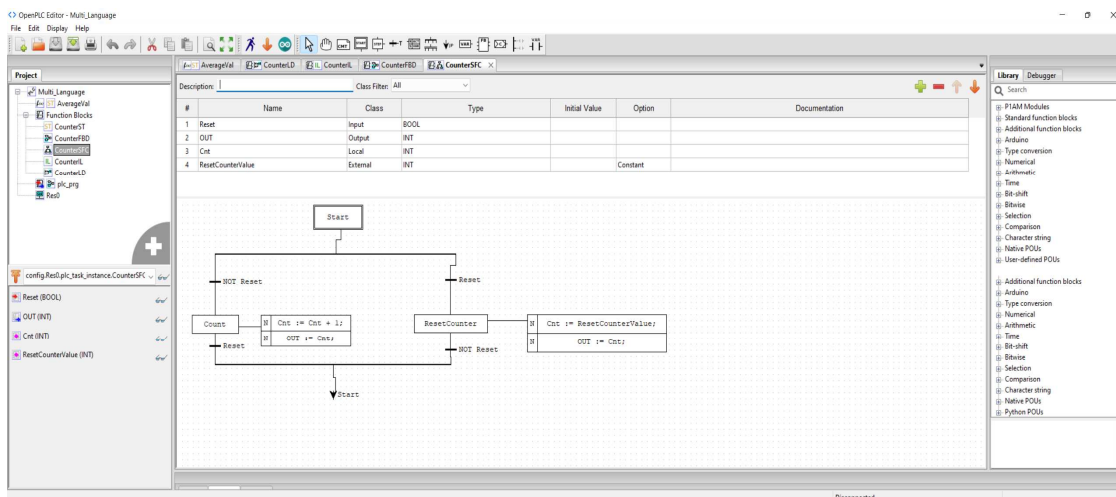
3.3.3 Μπλοκ Λειτουργιών (FBD)



Εικόνα 3.25 : Παράδειγμα προγράμματος Μπλοκ Λειτουργιών (FBD).

Η γλώσσα προγραμματισμού FBD είναι επίσης μια γλώσσα γραφικού προγραμματισμού όπως και η LD. Πρωτοεμφανίστηκε στην Γαλλία όπως και η SFC. Στην FBD μπορούμε να χρησιμοποιήσουμε μπλοκ όπως π.χ. AND, OR, χρονόμετρα κ.α. τα οποία είναι διαθέσιμα στην καρτέλα Library του OpenPLC Editor. Στην FBD δεν χρειάζεται να χρησιμοποιούμε μπάρες τροφοδοσίας, επαφές ή πηνία όπως στην LD.

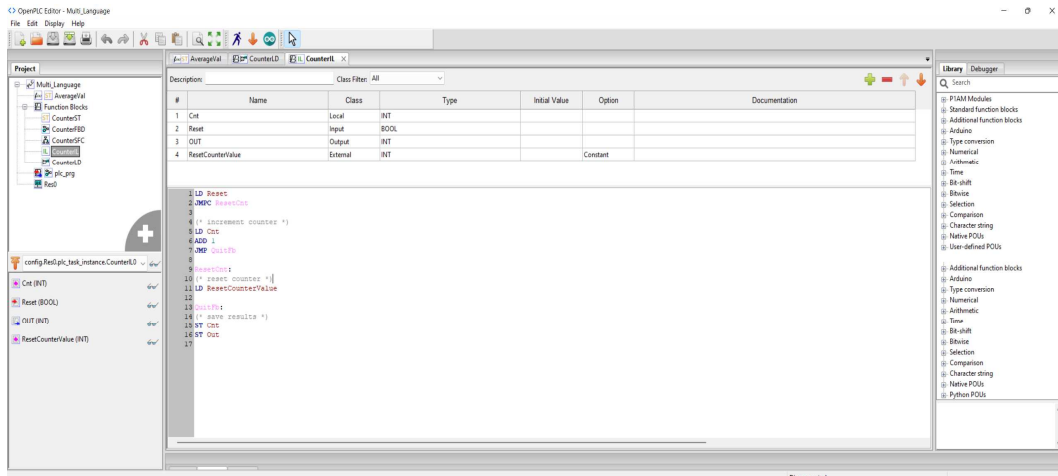
3.3.4 Διαγράμματα Διαδοχικών Συναρτήσεων (SFC)



Εικόνα 3.26 : Παράδειγμα προγράμματος Διαγραμμάτων Διαδοχικών Συναρτήσεων (SFC).

Η γλώσσα προγραμματισμού SFC είναι γλώσσα γραφικού προγραμματισμού την οποία χρησιμοποιούμε συνήθως σε συστήματα διαδοχικού ελέγχου. Υπάρχουν βήματα, μεταβάσεις, ενέργειες και κλάδοι σε ακολουθίες προγράμματος. Όσον αφορά τις ενέργειες είναι δυνατή η εκτέλεσή τους μαζί με χρονόμετρα. Η SFC είναι γλώσσα που την χρησιμοποιούμε κύριος σε εφαρμογές όπως ταινίες μεταφοράς, γκαραζόπορτες και γραμμές παραγωγής.

3.3.5 Λίστα Οδηγιών (IL)



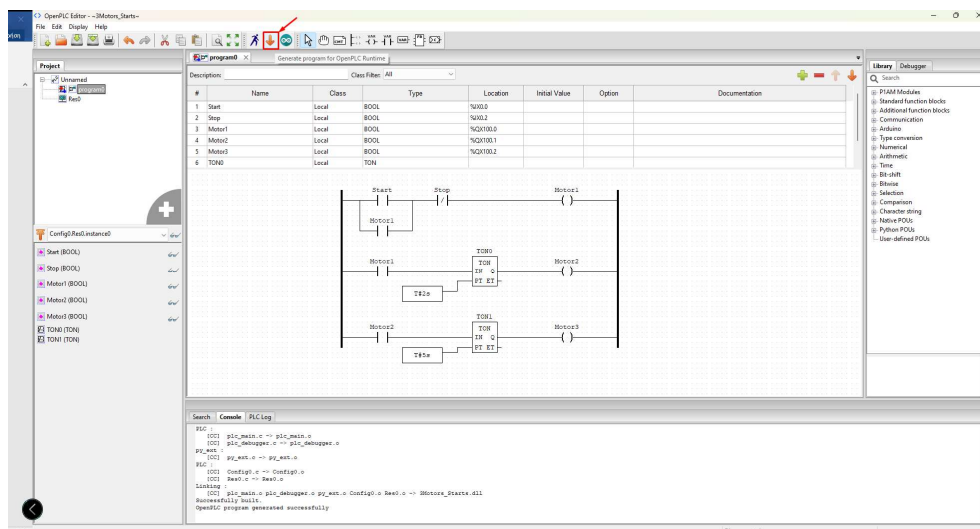
Εικόνα 3.27 : Παράδειγμα προγράμματος Λίστας Οδηγιών (IL).

Η γλώσσα προγραμματισμού IL είναι μια από τις παλαιότερες γλώσσες για προγραμματισμό PLC. Τα PLC ήρθαν στην Ευρώπη αργότερα από ότι στις Η.Π.Α, οπότε προέκυψε η ανάγκη να δημιουργηθούν νέες γλώσσες προγραμματισμού εκτός της LD, που θα ήταν κατανοητές από τη νέα γενιά μηχανικών. Πρωτοεμφανίστηκε στην Γερμανία, βασίζεται σε κείμενο και μοιάζει σε μεγάλο βαθμό με τη γλώσσα Assembly των επεξεργαστών. Οι οδηγίες επεξεργάζονται με τη σειρά που εμφανίζονται στο πρόγραμμα. Τα προγράμματα της IL μπορούν να χρησιμοποιηθούν συναρτήσεις και μπλοκ συναρτήσεων.

3.4 Μεταγλώττιση Προσομοίωση Φόρτωση προγράμματος στον OpenPLC

Για να μεταφέρουμε ένα πρόγραμμα από τον OpenPLC Editor στον OpenPLC Runtime θα πρέπει να ακολουθηθούν τα παρακάτω βήματα.

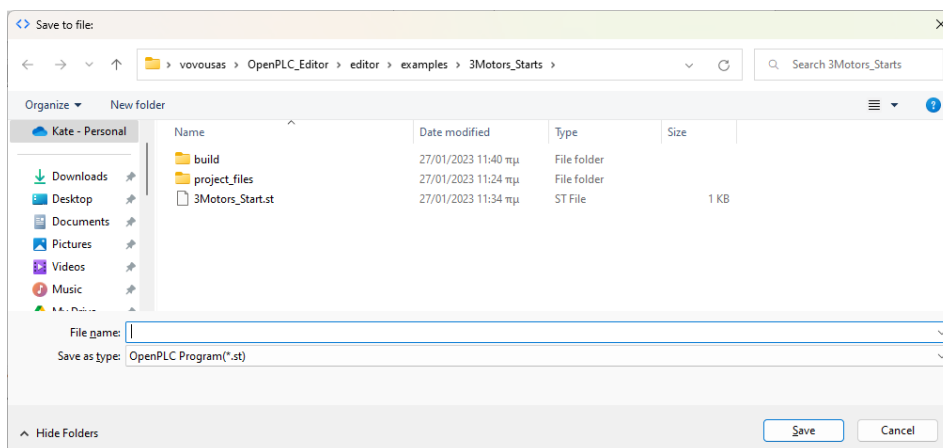
Εφόσον ολοκληρωθεί ο πρόγραμμα, πρέπει να μεταγλωττιστεί το αρχείο του προγράμματος σε τέτοια μορφή που να μπορεί να φορτωθεί στον OpenPLC Runtime.



Εικόνα 3.28 : Δημιουργία «εκτελέσιμου» προγράμματος για το OpenPLC Runtime.

Για την μεταγλώττιση του προγράμματος κάνουμε κλικ στο εικονίδιο ↓ όπως φαίνεται στην (εικόνα



3.28). Μετά την ολοκλήρωση της μεταγλώττισης το πρόγραμμα θα μας ζητήσει την θέση στην οποία θα αποθηκεύσει το αρχείο που δημιουργήθηκε. Θα πρέπει να δώσουμε ένα όνομα στο αρχείο και να κάνουμε κλικ στο πλήκτρο [Save]. Το αρχείο μας θα πρέπει να έχει την κατάληξη (.st) έτσι ώστε να μπορεί να φορτωθεί στο OpenPLC Runtime (εικόνα 3.29).

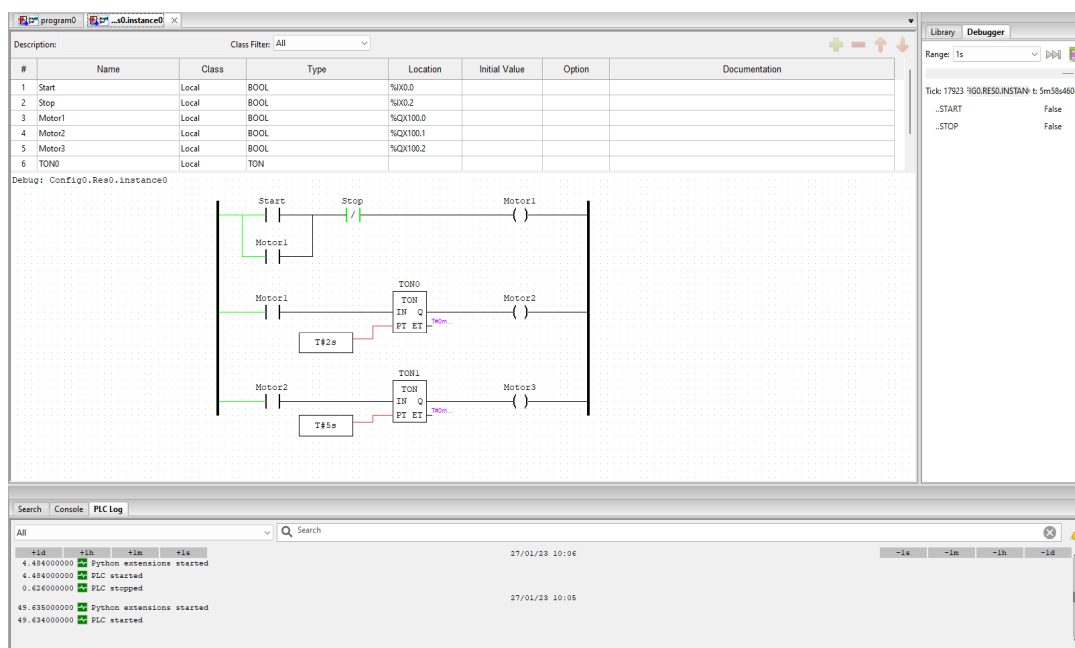


Εικόνα 3.29 : Αποθήκευση του μεταγλωττισμένου αρχείου.

3.4.1 Προσομοίωση.

Να αναφέρουμε εδώ ότι ο OpenPLC Editor δίνει την δυνατότητα πριν φορτωθεί ένα πρόγραμμα στο OpenPLC Runtime να γίνει προσομοίωση του ώστε να εξασφαλίσουμε ότι έχει την επιθυμητή συμπεριφορά.

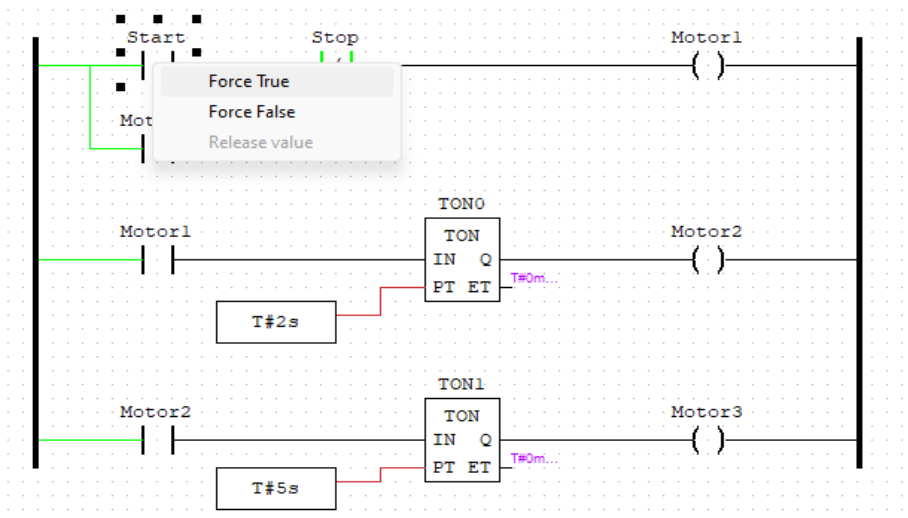
Για να ξεκινήσουμε την λειτουργία προσομοίωσης κάνουμε κλικ στο πλήκτρο  σε ελάχιστο χρονικό διάστημα θα ολοκληρωθεί ο έλεγχος του προγράμματος προς εξομοίωση. Αν όλα είναι σωστά κάνοντας κλικ στο εικονίδιο  θα μας εμφανιστεί η (εικόνα 3.30). Παρατηρούμε οι γραμμές του κυκλώματος έχουν αλλάξει κατάσταση, ουσιαστικά τροφοδοτούνται με ρεύμα.



Εικόνα 3.30 : Παράθυρο προσομοίωσης OpenPLC Editor.

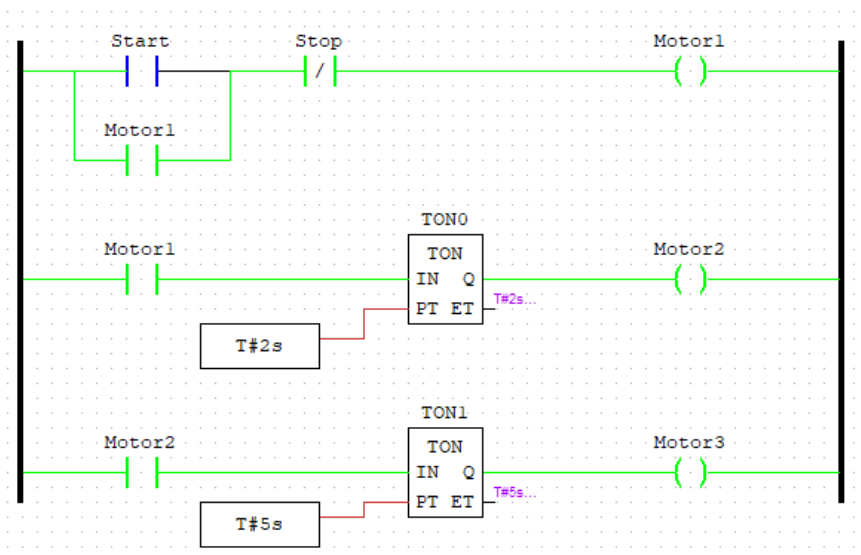
Για να την ενεργοποίηση του κυκλώματος κάνουμε δεξιά κλικ στην επαφή Start και από μενού που

εμφανίζεται επιλέγουμε το Force True για την ενεργοποίηση του προγράμματος. Στην περίπτωση αυτή αφορά την εκκίνηση τριών κινητήρων με χρονική καθυστέρηση στην ενεργοποίηση του καθενός. Ο πρώτος κινητήρας θα εκκινήσει άμεσα, ο δεύτερος κινητήρας θα εκκινήσει μετά από δυο δευτερόλεπτα από την εκκίνησή του πρώτου, και ο τρίτος κινητήρας μετά από 5 δευτερόλεπτα από την εκκίνησή του δεύτερου κινητήρα. Στην (εικόνα 3.32) βλέπουμε το κύκλωμα σε πλήρη λειτουργία.



Εικόνα 3.31 : Προσομοίωση προγράμματος.

Επαναφέρουμε την λειτουργία της επαφής Start στην αρχική της κατάσταση κάνοντας το παραπάνω βήμα που περιεγράφηκε αλλά αυτή την φορά επιλέγουμε Force False.



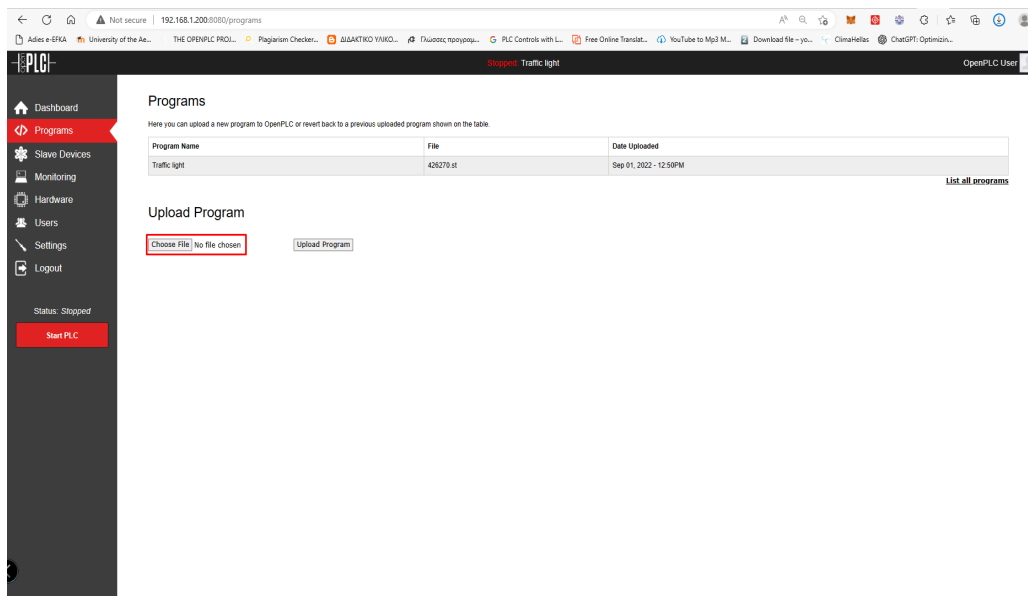
Εικόνα 3.32 : Προσομοίωση προγράμματος

Για την διακοπή λειτουργίας του κυκλώματος κάνουμε τα παραπάνω βήματα αλλά αυτή την φορά για την επαφή Stop.

Για να τερματίσουμε την λειτουργίας προσομοίωσης κάνουμε κλικ στο εικονίδιο .

3.4.2 Φόρτωση προγράμματος στο OpenPLC Runtime.

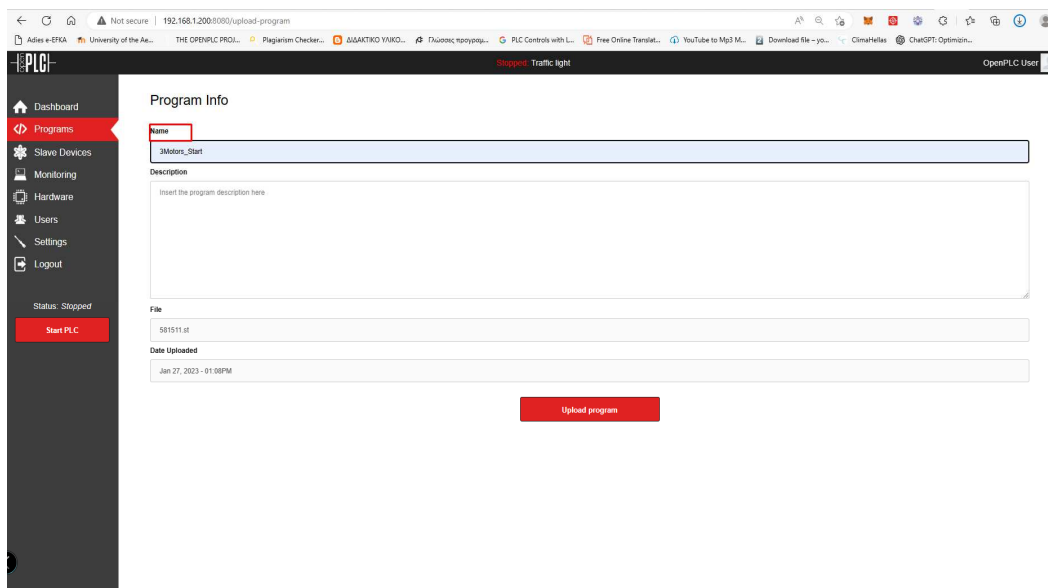
Για να φορτώσουμε το πρόγραμμα στο OpenPLC Runtime θα πρέπει να συνδεθούμε το WEB interface του Raspberry Pi OpenPLC και από την επιλογή Programs κάνουμε κλικ στο Choose files (εικόνα 3.33).



Εικόνα 3.33 : Φόρτωση προγράμματος στο OpenPLC Runtime.

Από το πλαίσιο που εμφανίζεται επιλέγουμε το αρχείο που μεταφράσαμε (με κατάληξη .st) από τον OpenPLC Editor και κάνουμε κλικ στο πλήκτρο [Open]. Έπειτα κάνουμε κλικ στο πλήκτρο [Upload Program] για να ξεκινήσει η διαδικασία φόρτωσης του προγράμματος.

Στο παράθυρο που εμφανίζεται δίνουμε όνομα πρόγραμμα που θα φορτώσουμε. Πληκτρολογούμε το όνομα στο πλαίσιο κάτω από το Name, κάνουμε κλικ στο πλήκτρο [Upload program].

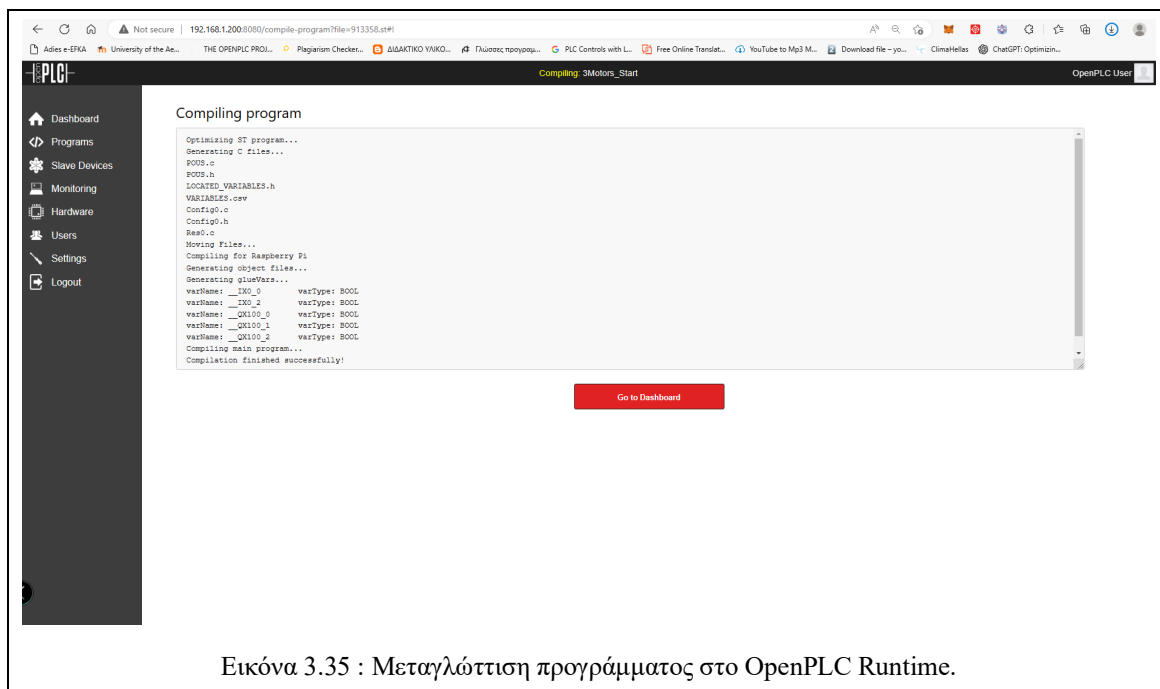


Εικόνα 3.34 : Ονοματοδοσία προγράμματος στο OpenPLC Runtime.

Το πρόγραμμα πλέον μεταφέρετε στο OpenPLC Runtime, και αρχίζει η μεταγλώττιση του .st αρχείου σε αρχεία γλώσσας προγραμματισμού C. Ωστε να μπορούν να «τρέξουν» στο περιβάλλον του

Raspberry Pi (εικόνα 3.35).

Κάνουμε κλικ στο πλήκτρο [Go to Dashboard] για να επιστρέψουμε στην αρχική οθόνη. Πλέον μπορούμε να θέσουμε σε λειτουργία το PLC κάνοντας κλικ στο πλήκτρο [Start PLC.]



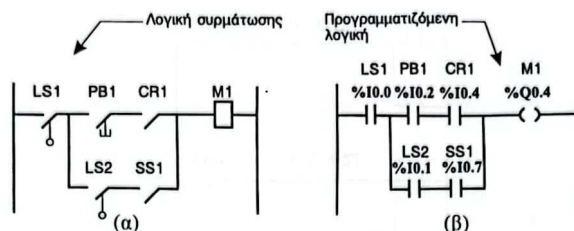
Εικόνα 3.35 : Μεταγλώττιση προγράμματος στο OpenPLC Runtime.

ΚΕΦΑΛΑΙΟ 4

Παρουσίαση της γλώσσας προγραμματισμού για PLC Ladder Diagram

4.1 Γλώσσα Ladder Diagram (LD)

Η γλώσσα προγραμματισμού Ladder είναι μία εύκολη στη χρήση γραφική γλώσσα προγραμματισμού με την βοήθεια της οποίας μπορεί να γίνει απευθείας μετατροπή του ηλεκτρολογικού σχεδίου σε γλώσσα κατανοητή από το PLC. Με τη χρήση γραφικών εργαλείων όπως επαφών, πηνίων, καλωδιώσεων, χρονικών κ.α., μπορούμε να δομήσουμε ένα λογικό πρόγραμμα, ικανό να ακολουθήσει την λογική συνδεσμολογίας ενός κλασικού αυτοματισμού όπως είδαμε στο ηλεκτρολογικό σχέδιο. Οι δυνατότητες που παρέχονται, είναι περισσότερες, μιας και μπορούμε να εκτελέσουμε λειτουργίες χρόνου, σύγκρισης, μεταφοράς και μαθηματικής επεξεργασίας δεδομένων. Στην παρακάτω εικόνα βλέπουμε την διαφορά ενός γραμμικού ηλεκτρολογικού σχεδίου και το αντίστοιχο διάγραμμα Ladder.



Εικόνα 4.1 : Γραμμικό ηλεκτρολογικό σχέδιο (α) και το ισοδύναμο σε γλώσσα Ladder Diagram (β).

Όπως βλέπουμε, η κύρια διαφορά μεταξύ της λογικής συρμάτωσης του ηλεκτρολογικού σχεδίου και της προγραμματιζόμενης λογικής σε Ladder Diagram, είναι ότι οι εισοδοί αντικαθίστανται με τα σύμβολα επαφών $-||-$ και οι εξοδοί μετά σύμβολα πηνίων $-(-)-$ της γλώσσας Ladder Diagram. Στο συγκεκριμένο παράδειγμα, οι διευθύνσεις (%I0.0, %Q0.4) πάνω από κάθε επαφή $-||-$ ή πηνίο $-(-)-$ στην γλώσσα Ladder Diagram, προσδιορίζουν τις εισόδους και εξόδους του PLC, πάνω στους οποίους συνδέουμε τα στοιχεία εισόδου (διακόπτη LS1) ή τα στοιχεία εξόδου (πηνίο M1).

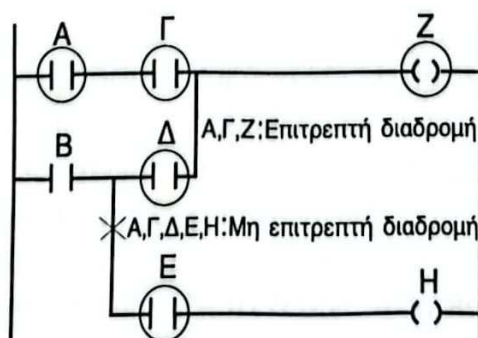
Ένα πρόγραμμα γραμμένο σε γλώσσα Ladder Diagram αποτελείται από rungs, δηλαδή ένα σύνολο από γραφικές εντολές, οι οποίες είναι σχεδιασμένες και τοποθετημένες μεταξύ δύο κάθετων γραμμών, που αντιπροσωπεύουν η μεν $-|$ την αριστερή γραμμή τροφοδοσίας, η $-|$ την δε δεξιά γραμμή επιστροφής. Τα rungs εκτελούνται το ένα μετά το άλλο από την CPU του PLC, προκειμένου να γίνει η επεξεργασία των δεδομένων.

Τα διάφορα γραφικά σύμβολα - εντολές που υπάρχουν σε ένα rung παριστάνουν:

- Τις εισόδους και εξόδους του PLC (π.χ. διακόπτες, μπουτόν, αισθητήρια, ρελέ).
- Τις λειτουργίες του PLC (π.χ. χρονικά, μετρητές).
- Τις μαθηματικές και λογικές πράξεις (π.χ. πρόσθεση, αφαίρεση, λογικό AND, λογικό OR).
- Πράξεις συγκρίσεως και αριθμητικές λειτουργίες (π.χ. $A < B$, $A = B$).
- Εσωτερικές μεταβλητές του PLC (π.χ. bits, words).

Τα γραφικά σύμβολα συνδέονται με οριζόντιες και κάθετες γραμμές για να καταλήξουν σε μία ή περισσότερες εξόδους ή σε στοιχεία που εκτελούν διαφορετικές λειτουργίες όπως να αθροίσουμε το περιεχόμενο δύο καταχωρητών και το αποτέλεσμα να το στείλουμε σε ένα άλλο.

Προσοχή ένα rung δεν μπορεί να υποστηρίξει περισσότερο από μία ομάδα εντολών συνδεδεμένων μεταξύ τους. Οι εντολές σε ένα rung εκτελούνται πάντοτε από αριστερά προς τα δεξιά και από επάνω προς τα κάτω. Αν θεωρούμε ότι τα στοιχεία που βρίσκονται σε κύκλο στην παρακάτω εικόνα, είναι σε λογική κατάσταση 1 δηλαδή περνάει ρεύμα. Το πηνίο Z θα ενεργοποιηθεί μέσω των A και Γ, ενώ το πηνίο H δεν θα ενεργοποιηθεί, μέσω της διαδρομής A, Γ, Δ, και E επειδή όπως έχουμε αναφέρει οι εντολές από δεξιά προς τα αριστερά δεν μπορούν να εκτελεστούν.



Εικόνα 4.2 : Η εκτέλεση των εντολών από το PLC, είναι όπως διαβάζουμε ενα βιβλίο.

Τα γραφικά σύμβολα τα οποία χρησιμοποιούμε στη γλώσσα προγραμματισμού Ladder Diagram είναι πάρα πολλά και δεν είναι δυνατό να τα αναλύσουμε όλα διεξοδικά. Για το λόγο αυτό, στις επόμενες παραγράφους θα γίνει παρουσίαση των πιο βασικών στοιχείων που συναντάμε σε αυτοματισμούς. Θα πρέπει να μελετηθεί με προσοχή το εγχειρίδιο χρήσης και προγραμματισμού του PLC, εάν επιθυμούμε να γνωρίσουμε όλες τις εντολές.

Τα γραφικά στοιχεία τα οποία θεωρούνται βασικά και τα οποία θα αναλύσουμε, αλλά και ταυτόχρονα θα δώσουμε παραδείγματα για να φανεί ο τρόπος λειτουργίας τους, είναι αυτά που παρουσιάζονται παρακάτω:

- Ανοιχτή επαφή.
- Κλειστή επαφή.
- Επαφή ανερχόμενου παλμού.
- Επαφή κατερχόμενου παλμού.
- Πηνίο.
- Ανάστροφο πηνίο.
- Πηνίο αυτομανδάλωσης.
- Πηνίο απομανδάλωσης.
- Λειτουργικό μπλοκ αριθμητικών και λογικών πράξεων.
- Χρονικά
- Μετρητές.
- Βηματικός μετρητής.

- Τυμπανοειδής ελεγκτής.
- Ρολόι πραγματικού χρόνου.
- Βοηθητικά πηνία.

4.2 Ανοικτή Επαφή

Όνομα : Ανοικτή Επαφή / Normal Open

Σύμβολο: --| |--

Περιγραφή:

Σε κατάσταση ηρεμίας, λογικό 0 η επαφή είναι ανοικτή και δεν περνάει ρεύμα. Σε κατάσταση λειτουργίας, λογικό 1 η επαφή είναι κλειστή και περνάει ρεύμα. Η επαφή αυτού του τύπου χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εισόδους του PLC, καθώς και τα εσωτερικά bits.

4.3 Κλειστή Επαφή

Όνομα : Κλειστή Επαφή / Normal Close

Σύμβολο: --|/|--

Περιγραφή:

Σε κατάσταση ηρεμίας, λογικό 0 η επαφή είναι κλειστή και περνάει ρεύμα. Σε κατάσταση λειτουργίας, λογικό 1 η επαφή είναι ανοικτή και δεν περνάει ρεύμα. Η επαφή αυτού του τύπου χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εισόδους του PLC, καθώς και τα εσωτερικά bits.

4.4 Επαφή ανερχομένου παλμού

Όνομα : Επαφή ανερχομένου παλμού / Rising Edge Contact

Σύμβολο: --|P|--

Περιγραφή:

Η επαφή ανερχόμενου παλμού ή επαφή στιγμιαίας διέγερσης κατά την ενεργοποίηση ενός σήματος είναι σε λογική κατάσταση 1 την ώρα που ενεργοποιείται μία είσοδος του PLC. Η επαφή αυτή παραμένει σε λογική κατάσταση 1 όσο διαρκεί ένας κύκλος λειτουργίας του PLC.

4.5 Επαφή κατερχομένου παλμού

Όνομα : Επαφή κατερχομένου παλμού / Falling Edge Contact

Σύμβολο: --|N|--

Περιγραφή:

Η επαφή ανερχόμενου παλμού ή επαφή στιγμιαίας διέγερσης κατά την ενεργοποίηση ενός σήματος είναι σε λογική κατάσταση 1 την ώρα που ενεργοποιείται μία είσοδος του PLC. Η επαφή αυτή παραμένει σε λογική κατάσταση 1 όσο διαρκεί ένας κύκλος λειτουργίας του PLC. Η επαφή

αυτού του τύπου χρησιμοποιείται για να αντιπροσωπεύσει μόνο πραγματικές εισόδους, του PLC.

4.6 Πηνίο

Όνομα : Πηνίο /Coil

Σύμβολο: --()--

Περιγραφή:

Σε κατάσταση ηρεμίας, λογικό 0 το πηνίο είναι ανοικτό και δεν περνάει ρεύμα. Σε κατάσταση λειτουργίας, λογικό 1 το πηνίο είναι κλειστό και περνάει ρεύμα. Το πηνίο αυτού του τύπου χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εξόδους του PLC, καθώς και τα εσωτερικά bits.

4.7 Ανάστροφο Πηνίο

Όνομα : Ανάστροφο Πηνίο / Inverse Coil

Σύμβολο: --(/)--

Περιγραφή:

Σε κατάσταση ηρεμίας, λογικό 0 το πηνίο είναι κλειστό και περνάει ρεύμα. Σε κατάσταση λειτουργίας, λογικό 1 το πηνίο είναι ανοικτό και δεν περνάει ρεύμα. Το πηνίο αυτού του τύπου χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εξόδους του PLC, καθώς και τα εσωτερικά bits.

4.8 Πηνίο Αυτομανδάλωσης

Όνομα : Πηνίο αυτομανδάλωσης / Set Coil

Σύμβολο: --(S)--

Περιγραφή:

Το πηνίο πριν πάρει τάση στα άκρα του δεν είναι οπλισμένο. Μόλις εφαρμοστεί τάση στα άκρα του, τότε το πηνίο οπλίζει και παραμένει οπλισμένο - μανδαλωμένο συνέχεια, ανεξάρτητα από το αν συνεχίζει να τροφοδοτείται ή όχι με τάση. Ο μόνος τρόπος να από μανδαλωθεί, είναι με το πηνίο RESET. Το πηνίο Set χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εξόδους του PLC, καθώς και τα εσωτερικά bits. Θα πρέπει να διευκρινίσουμε ότι, όταν αναφέρεται η έκφραση “τάση στα άκρα του πηνίου”, εννοούμε ότι το πηνίο είναι ενεργό.

4.9 Πηνίο Απομανδάλωσης

Όνομα : Πηνίο Απομανδάλωσης / Reset Coil

Σύμβολο: --(R)--

Περιγραφή:

Το πηνίο πριν πάρει τάση στα άκρα του δεν είναι οπλισμένο έχει λογικό 0. Μόλις εφαρμοστεί μία στιγμιαία τάση στα άκρα του, τότε το πηνίο οπλίζει και ταυτόχρονα απενεργοποιεί ένα μανδαλωμένο πηνίο Set Coil, που έχει την ίδια διεύθυνση με αυτό. Το πηνίο Reset Coil χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εξόδους του PLC, καθώς και τα εσωτερικά bits.

4.10 Πηνίο Απομανδάλωσης

Όνομα : Πηνίο Απομανδάλωσης / Reset Coil

Σύμβολο: --(R)--

Περιγραφή:

Το πηνίο πριν πάρει τάση στα άκρα του δεν είναι οπλισμένο έχει λογικό 0. Μόλις εφαρμοστεί μία στιγμιαία τάση στα άκρα του, τότε το πηνίο οπλίζει και ταυτόχρονα απενεργοποιεί ένα μανδαλωμένο πηνίο Set Coil, που έχει την ίδια διεύθυνση με αυτό. Το πηνίο Reset Coil χρησιμοποιείται για να αντιπροσωπεύσει τις πραγματικές εξόδους του PLC, καθώς και τα εσωτερικά bits.

4.11 Μπλοκ Αριθμητικών & Λογικών πράξεων

Όνομα : Μπλοκ Αριθμητικών & Λογικών πράξεων / Operate Block

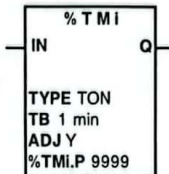
Σύμβολο: 

Περιγραφή:

Ένα Μπλοκ Αριθμητικών & Λογικών πράξεων χρησιμοποιείται για την αναπαράσταση μιας σειράς λογικών ή αριθμητικών πράξεων που εκτελούνται με συγκεκριμένη σειρά. Το αποτέλεσμα αυτών των πράξεων αποθηκεύεται σε εσωτερικές μεταβλητές του PLC αντίστοιχου τύπου και μεγέθους.

4.12 Μπλοκ Χρονικού

Όνομα : Μπλοκ Χρονικού / Timer Function Block

Σύμβολο: 

Περιγραφή:

Υπάρχουν τρεις τύποι χρονικών:

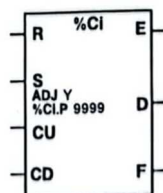
- TON: Αυτός ο τύπος χρονικού χρησιμοποιείται ως χρονικό Τύπου ON DELAY. Ο χρόνος καθυστέρησης μπορεί να καθοριστεί από τον χρήστη ή μέσα από το πρόγραμμα.
- TOF: Αυτός ο τύπος χρονικού χρησιμοποιείται ως χρονικό τύπου OFF DELAY. Ο χρόνος καθυστέρησης μπορεί να καθοριστεί από τον χρήστη ή μέσα από το πρόγραμμα.

- TP : Αυτός ο τύπος χρονικού χρησιμοποιείται για να δημιουργεί παλμούς συγκεκριμένης διάρκειας. Ο χρόνος διάρκειας του παλμού μπορεί να καθοριστεί από τον χρήστη ή μέσα από το πρόγραμμα.

4.13 Μπλοκ Απαριθμητικού

Όνομα : Μπλοκ Απαριθμητικού / Counter Function Block

Σύμβολο:



Περιγραφή:

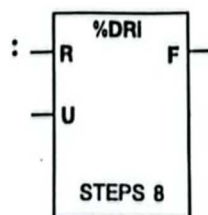
Υπάρχουν τρεις τύποι απαριθμητών.

- UP Counter: Αυτός ο τύπος απαριθμητού χρησιμοποιείται για να μετράει γεγονότα με αθροιστικό τρόπο 1, 2, 3, 4, κλπ. όταν δεχθεί σήμα στην είσοδο CU.
- DOWN Counter: Αυτός ο τύπος απαριθμητού χρησιμοποιείται για να μετράει γεγονότα με αφαιρετικό τρόπο 10, 9, 8, 7, κλπ. όταν δεχθεί σήμα στην είσοδο CD.
- UP & DOWN Counter: Αυτός ο τύπος απαριθμητού χρησιμοποιείται για να μετράει ταυτόχρονα δύο γεγονότα με αθροιστικό και αφαιρετικό τρόπο, όταν δεχθεί σήματα στις εισόδους CU και CD αντίστοιχα.

4.14 Μπλοκ Τυμpanοειδούς Ελεγκτή

Όνομα : Μπλοκ Τυμpanοειδούς Ελεγκτή / Drum Function Block

Σύμβολο:



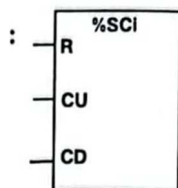
Περιγραφή:

Το μπλοκ του τυμpanοειδούς ελεγκτή λειτουργεί με την ίδια αρχή που λειτουργεί και ένα ηλεκτρομηχανικό τύμpano, το οποίο αλλάζει βήμα σύμφωνα με τα εξωτερικά ερεθίσματα που παίρνει. Ένα ηλεκτρομηχανικό τύμpano έχει ανάγλυφη επιφάνεια και καθώς περιστρέφεται ενεργοποιεί ή απενεργοποιεί διάφορες επαφές που είναι στερεωμένες στο ακίνητο μέρος. Στην περίπτωση του τυμpanοειδούς ελεγκτή του PLC μπορούμε να ορίσουμε σε κάθε βήμα του ελεγκτή ποια bits θα ενεργοποιήσουμε και ποια θα απενεργοποιήσουμε. Τα bits που σχετίζονται με τα βήματα μπορούν να είναι στοιχεία εξόδου ή εσωτερικά bits. Ο τυμpanοειδής ελεγκτής του PLC αποτελείται από 8 βήματα και κάθε βήμα μπορεί να ελέγξει 16 bits, τα οποία ονομάζονται bits ελέγχου control bits.

4.15 Μπλοκ Βηματικού Απαριθμητή

Όνομα : Μπλοκ Βηματικού Απαριθμητή / Step Counter Function Block

Σύμβολο:



Περιγραφή:

Το μπλοκ του βηματικού απαριθμητή αποτελείται από 256 βήματα και κάθε ένα από αυτά, μπορεί να χρησιμοποιηθεί για να εκτελεί συγκεκριμένη λειτουργία. Προκειμένου να κινηθούμε στο επόμενο ή στο προηγούμενο βήμα, αρκεί να ενεργοποιηθεί μία από τις δύο εισόδους CU και CD. Κάθε φορά που ένα βήμα είναι ενεργό, το bit που σχετίζεται με το βήμα αυτό αποκτά λογική κατάσταση 1. Κάθε φορά μόνο ένα βήμα μπορεί να είναι ενεργό.

Η διαφορά μεταξύ ενός απαριθμητή και ενός βηματικού απαριθμητή. Ο απαριθμητής χρησιμοποιείται, κυρίως, για την καταμέτρηση αντικειμένων όπως για παράδειγμα σε μία γραμμή παραγωγής για την καταμέτρηση μπουκαλιών. Ένας βηματικός απαριθμητής, δε θα μπορούσε να κάνει την παραπάνω εργασία, εκτός και αν ο αριθμός των μπουκαλιών είναι μικρότερος ή ίσος με τον αριθμό των βημάτων που διαθέτει, δηλαδή μικρότερος ή ίσος από 256. Ο βηματικός απαριθμητής χρησιμοποιείται εκεί όπου θέλουμε να συσχετίσουμε το κάθε βήμα του με μία συγκεκριμένη δράση.

4.16 Μόνιμες Μεταβλητές

Όνομα : Μόνιμες Μεταβλητές / Constant Words

Σύμβολο: %Mx.x όπου x.x διεύθυνση μνήμης

Περιγραφή:

Οι μόνιμες μεταβλητές είναι λέξεις των 16 bits και χρησιμοποιούνται για να αποθηκεύουν σταθερές τιμές ή αλφαριθμητικούς χαρακτήρες. Τα περιεχόμενα τους μπορούν να γραφούν ή να τροποποιηθούν. Οι τιμές των μεταβλητών αυτών μπορούν να έχουν δεκαδική, δεκαεξαδική, δυαδική ή ASCII μορφή. Πολλές φορές είναι απαραίτητο να έχουμε πρόσβαση σε συγκεκριμένο bit μίας λέξης. Οι μόνιμες μεταβλητές ανήκουν σε αυτές τις λέξεις, οι οποίες μας δίνουν το δικαίωμα να εισχωρήσουμε σε οποιοδήποτε από τα 16 bits που διαθέτουν.

4.17 Εσωτερικά bits

Όνομα : Εσωτερικά bits / Internal bits.

Σύμβολο: %Bx.x όπου x.x διεύθυνση μνήμης

Περιγραφή:

Τα εσωτερικά bits είναι εσωτερικές περιοχές αποθήκευσης του PLC και είναι διαθέσιμα στον χρήστη για τις ανάγκες του προγράμματος. Τα bits αυτού του τύπου χρησιμοποιούνται ως πηνία, για τον έλεγχο βοηθητικών λειτουργιών. Αν θέλαμε να τα παραλληλίσουμε με ένα αντίστοιχο εξάρτημα του κλασικού αυτοματισμού, θα λέγαμε ότι έχουν την ιδιότητα των βοηθητικών ρελέ. Συνεπώς, όπως τα βοηθητικά ρελέ χρησιμοποιούνται για τον έλεγχο βοηθητικών λειτουργιών του κυκλώματος, έτσι και τα εσωτερικά bits, χρησιμοποιούνται για τον έλεγχο βοηθητικών λειτουργιών του προγράμματος. Το κάθε ένα εσωτερικό bit μπορούμε να το καλέσουμε όσες φορές επιθυμούμε μέσα στο ίδιο πρόγραμμα, κάτι που δεν μπορούμε να κάνουμε στον κλασικό αυτοματισμό με τις βοηθητικές επαφές.

ΚΕΦΑΛΑΙΟ 5

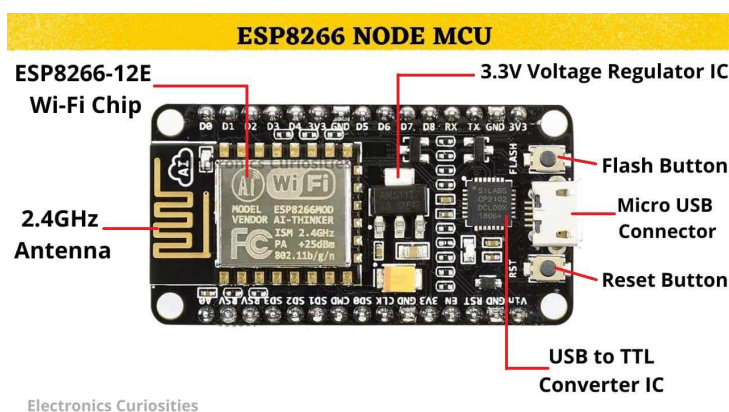
Το OpenPLC σε συνεργασία με το Modbus TCP

5.1 Σύνδεση NodeMCU ESP8266 σε OpenPLC με χρήση Modbus TCP

Τόσο στα συστήματα βιομηχανικού ελέγχου όσο και στον οικιακό αυτοματισμό, είναι όλο και πιο αποδεκτή πρακτική ο έλεγχος και η παρακολούθηση της λειτουργίας ενός PLC μέσω των λεγόμενων «οθονών». Αυτή η λειτουργία υποστηρίζεται και από το OpenPLC. Στο Raspberry Pi με OpenPLC Runtime, υπάρχει εγκατεστημένος ένας διακομιστής Modbus TCP που λειτουργεί στο παρασκήνιο, και μας δίνει τη δυνατότητα να δημιουργηθεί σύνδεση και άλλων συσκευών με το πρόγραμμα που τρέχει στο OpenPLC. Έτσι ώστε να διαχειρίζονται οι μεταβλητές που δηλώνονται στο πρόγραμμα του OpenPLC μέσω Ethernet ή WLAN σε άλλες συσκευές όπως το NodeMCU ESP8266 η και στο ίδιο το Raspberry Pi. Επίσης μπορούν να παρακολουθούνται και να προσομοιώνετε οι έξοδοι / είσοδοι του προγράμματος που τρέχει στο OpenPLC μέσα από λογισμικά HMI Scada. Όλα τα παραπάνω είναι εφικτά μέσα από την χρησιμοποίηση του πρωτόκολλου Modbus TCP/IP.

5.2 Περιγραφή υλικού τύπου ESP8266

Ο ESP8266 είναι ένας χαμηλού κόστους μικροελεγκτής με Wifi που αναπτύχθηκε από την εταιρία Espressif Systems. Έγινε διαθέσιμος στην αγορά για πρώτη φορά το 2014. Έγινε πολύ δημοφιλής μεταξύ των κατασκευαστών και των απλών χρηστών, για την χαμηλή του τιμή το μικρό του μέγεθος, τη χαμηλή κατανάλωση ενέργειας και την ικανότητα εύκολης σύνδεσης με δίκτυα Wifi. Χρησιμοποιείται συχνά σε έργα Internet of Things (IoT) και ως μικροελεγκτής με ενσωματωμένες δυνατότητες Wifi. Μπορεί να προγραμματιστεί με το Arduino IDE και μπορεί να χρησιμοποιηθεί σε ποικίλες εφαρμογές όπως οικιακός αυτοματισμός και δίκτυα αισθητήρων κ.α.



Εικόνα 5.1 : NodeMCU ESP8266.

Διαθέτει επεξεργαστή RISC 32 bit, με την μνήμη να είναι ενσωματωμένη στο τσιπ, καθώς και πομποδέκτη Wifi. Υποστηρίζει και τις δύο λειτουργίες Station (STA) και Access Point (AP) και μπορεί να ρυθμιστεί ώστε να συνδέεται σε ένα υπάρχον δίκτυο Wifi ή να δημιουργεί το δικό του. Διαθέτει επίσης μια ενσωματωμένη στοιβία TCP/IP, που διευκολύνει τη σύνδεση στο διαδίκτυο για την αποστολή και λήψη δεδομένων.

Είναι επίσης εξοπλισμένος με μια σειρά από περιφερειακές διεπαφές, συμπεριλαμβανομένων των UART, SPI, I2C και I2S, καθώς και με έναν αριθμό ακίδων GPIO. Αυτές οι διεπαφές μπορούν να χρησιμοποιηθούν για σύνδεση με εξωτερικούς αισθητήρες και συσκευές, επιτρέποντας του να συλλέγει και να μεταδίδει δεδομένα μέσω του διαδικτύου.

Για τον προγραμματισμό του χρησιμοποιείται συνήθως το Arduino IDE και το Espressif SDK. Είναι συμβατό με μια ποικιλία γλωσσών προγραμματισμού, συμπεριλαμβανομένων των C και Lua κ.α. Υπάρχει επίσης ένας μεγάλος αριθμός βιβλιοθηκών και παραδειγμάτων διαθέσιμα στο διαδίκτυο, γεγονός που καθιστά εύκολο να ξεκινήσει κάποιος την ενασχόληση του με τον ESP8266.

Ο ESP8266 υπάρχει σε διάφορες εκδόσεις, ESP8266EX, ESP32, το ESP8266EX είναι η πιο συνηθισμένη και φθηνότερη έκδοση του ESP8266. Έχει ορισμένους περιορισμούς όσον αφορά την ισχύ επεξεργασίας και τη μνήμη. Ο ESP32 είναι πιο ισχυρός έχοντας περισσότερη μνήμη και διεπαφές, αλλά είναι και πιο ακριβός.

5.3 Εγκατάσταση Modbus TCP σε NodeMCU ESP8266

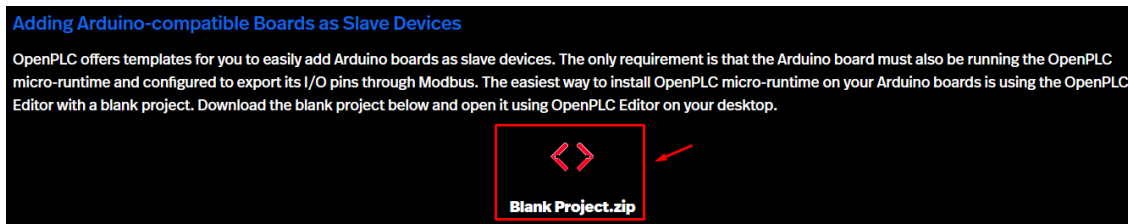
Τόσο στα συστήματα βιομηχανικού ελέγχου όσο και στον οικιακό αυτοματισμό, ο έλεγχος και η παρακολούθηση της ροής λειτουργίας του προγράμματος που εκτελείτε στο PLC γίνεται μέσω «οθονών». Αυτή η λειτουργία υποστηρίζεται και από το OpenPLC. Με την εγκατάσταση του Runtime OpenPLC στο Raspberry Pi, λειτουργεί στο παρασκήνιο ο διακομιστής TCP Modbus, που επιτρέπει την σύνδεση και επικοινωνία διάφορων συσκευών με το OpenPLC Runtime. Έτσι έχουμε την δυνατότητα να παρακολουθούνται οι μεταβλητές που δηλώνονται στο πρόγραμμα μέσω του PLC Ethernet ή WLAN και να προσομοιώνονται διακόπτες. Για το σκοπό αυτό χρησιμοποιείται το πρωτόκολλο Modbus TCP.

Όπως έχουμε αναφέρει και παραπάνω το Modbus TCP είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται για συστήματα βιομηχανικού αυτοματισμού και ελέγχου. Βασίζεται στο πρωτόκολλο Modbus, το οποίο αναπτύχθηκε αρχικά για σειριακή επικοινωνία, αλλά το Modbus TCP χρησιμοποιεί δικτύωση TCP/IP για να επιτρέπει την επικοινωνία μεταξύ συσκευών μέσω δικτύου. Το Modbus TCP χρησιμοποιείται συνήθως σε βιομηχανικά περιβάλλοντα για τη σύνδεση προγραμματιζόμενων λογικών ελεγκτών (PLC), απομακρυσμένων τερματικών μονάδων (RTU) και άλλων συσκευών με το μοντέλο Master - Slave. Ο Master κόμβος στέλνει ένα αίτημα στον Slave και ο Slave επιστρέφει το αποτέλεσμα. Στο Modbus TCP, οι σχετικοί όροι είναι Client και Server. Εδώ ο Client είναι ο Master και ο Server είναι ο Slave.

Πριν την εγκατάσταση του OpenPLC Modbus, συνιστάται να ελέγξετε την τεκμηρίωση ESP8266 και την τεκμηρίωση OpenPLC για να κατανοήσετε καλύτερα τη διαδικασία και να αποφύγετε προβλήματα.

Το Modbus θα χρησιμοποιείτε για να έχετε πρόσβαση στο πρόγραμμα PLC που εκτελείται στο Raspberry Pi. Κατά συνέπεια το Raspberry Pi, λειτουργεί ως πελάτης Modbus, ενώ στο NodeMCU ESP8266 θα λειτουργεί ως διακομιστής.

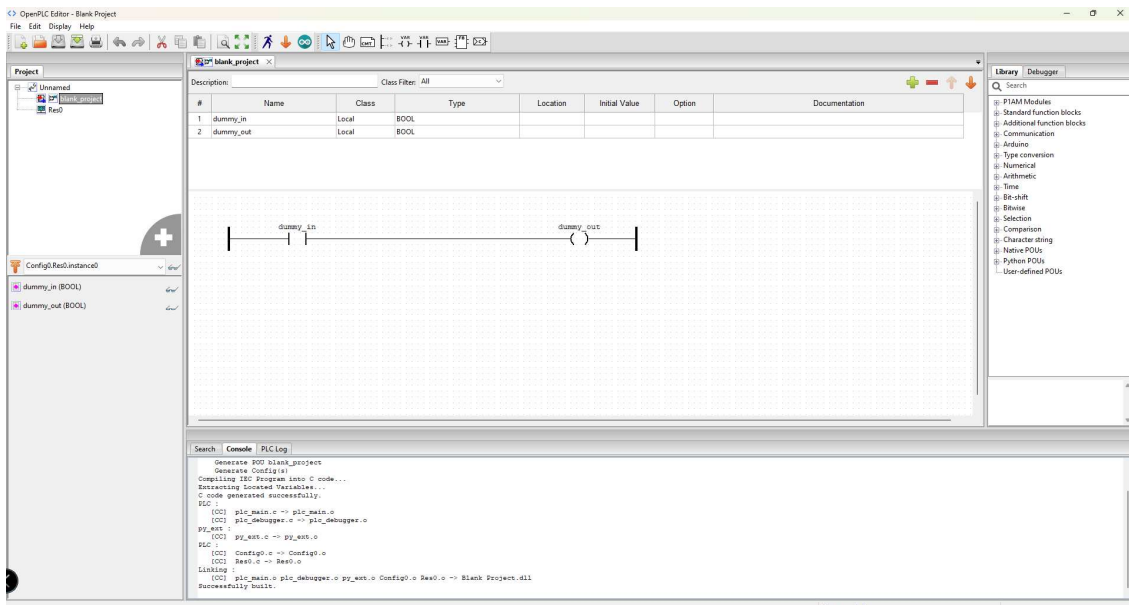
Τα βήματα που πρέπει να ακολουθηθούν για την εγκατάσταση του λογισμικού OpenPLC Modbus στον NodeMCU ESP8266 και την σύνδεση του με το OpenPLC του Raspberry Pi είναι:



Εικόνα 5.2 : Λήψη του αρχείου Blank Project.

Από την διεύθυνση <https://openplcproject.com/docs/2-6-slave-devices/> κάνουμε λήψη το αρχείο Blank Project.zip, (εικόνα 5.2) οποίο βρίσκεται στο μέσω της σελίδας.

Αποσυμπιέζουμε το αρχείο στον φάκελο με τα project της εφαρμογής OpenPLC Editor και ανοίγουμε μέσω αυτής το αρχείο Blank Project (εικόνα 5.3).

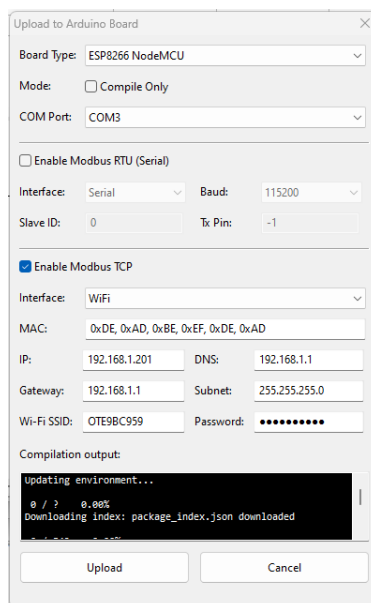


Εικόνα 5.3 : OpenPLC Editor.



Εικόνα 5.4 : OpenPLC Editor compile Project.

Κάνουμε κλικ στο εικονίδιο με το κόκκινο περίγραμμα (εικόνα 5.4)



Εικόνα 5.5 : Αποστολή του λογισμικού OpenPLC Modbus στο NodeMCU ESP8266.

Στο παράθυρο που μας εμφανίζεται (εικόνα 5.5) επιλέγουμε τον τύπο του μικροελεγκτή που έχουμε από το πλαίσιο Board Type, καθώς και την σειριακή θύρα στην οποία έχει συνδεθεί.

Κάνουμε τσεκ το Enable Modbus TCP και επιλέγουμε στο Interface το Wifi. Συμπληρώνουμε τα πεδία με τα αντίστοιχα στοιχεία που μας ζητούνται. Τα στοιχεία που δίνουμε εδώ είναι τα στοιχεία του δικτύου στο οποίο θα συνδεθεί μικροελεγκτής NodeMCU ESP 8266.

Όταν ολοκληρώσουμε κάνουμε κλικ στο πλήκτρο [Upload]. Πλέον μεταφέρεται το λογισμικό OpenPLC Modbus με τα στοιχεία που δώσαμε στο NodeMCU ESP8266 μετά το πέρας της εργασίας αποστολής του λογισμικού κάνουμε επανεκκίνηση της μονάδας. Πλέον είναι έτοιμη για να συνδεθεί με το Raspberry Pi.

Θα πρέπει εδώ να τονίσουμε ότι το πρωτόκολλο Modbus είναι ένα από τα πιο δημοφιλή μεταξύ των PLC και των συστημάτων SCADA. Ειδικά η έκδοση TCP του πρωτοκόλλου, σήμερα είναι πιο δημοφιλής από τη σειριακή RTU έκδοση, η οποία χρησιμοποιείται ευρύτερα στον χώρο του αυτοματισμού έχοντας όμως δυστυχώς κληρονομήσει όλα τα τρωτά σημεία του πρωτοκόλλου TCP σε επίπεδο ασφάλειας.

Η επικοινωνία μεταξύ των συστημάτων που γίνεται με τη χρήση του Modbus TCP κάνει όλη την διαδικασία επικοινωνίας πιο περίπλοκη, αλλά και τα κενά ασφάλειας του πρωτοκόλλου TCP, προσθέτουν θέματα ασφάλειας και αξιοπιστίας των επικοινωνιών μεταξύ των συσκευών, τα οποία μπορούν να εκμεταλλευτούν κακόβουλοι χρήστες ώστε να επιτεθούν στο σύστημα αυτοματισμού.

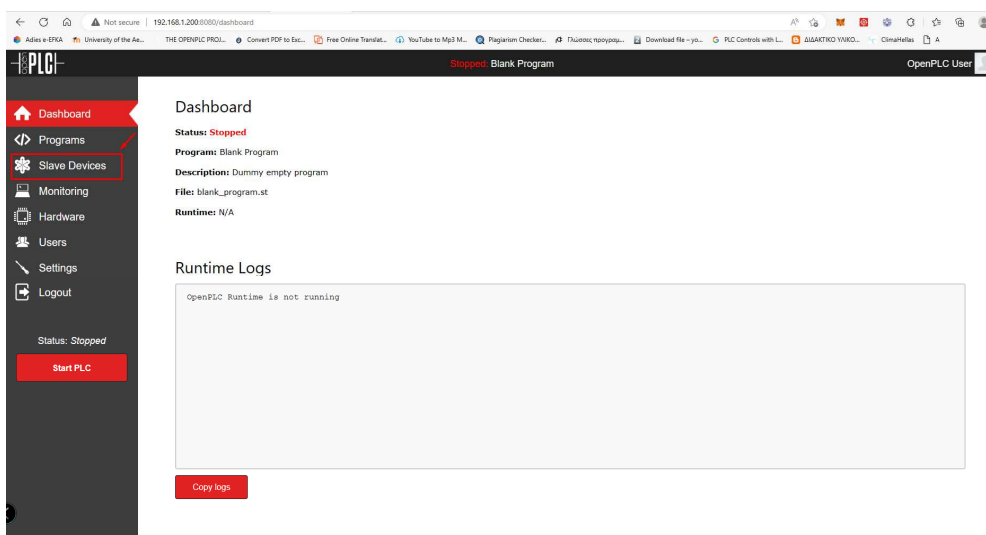
Το Modbus TCP δεν είναι ασφαλές πρωτόκολλο, δεν διαθέτει μηχανισμούς προστασίας για την εμπιστευτικότητα και την ακεραιότητα των μηνυμάτων που μεταδίδονται μεταξύ των μελών του συστήματος. Δεν υπάρχει τρόπος να ανακαλύψουμε εάν το μήνυμα που μεταδόθηκε, έχει την αρχική του μορφή ή εάν έχει τροποποιηθεί από κάποιον. Επίσης το Modbus δεν διαθέτει μηχανισμό ελέγχου ταυτότητας για το master και το slave. Έτσι μια παραβιασμένη συσκευή θα μπορούσε να προ-

σποηθεί ότι είναι η master και να στείλει ψευδείς εντολές στις slave συσκευές. Τα βασικά προβλήματα ασφάλειας του πρωτοκόλλου Modbus TCP είναι η έλλειψη ακεραιότητας, έλλειψη διαθεσιμότητας, έλλειψη εμπιστευτικότητας και έλλειψη ελέγχου ταυτότητας.

Τα παραπάνω προβλήματα δεν μας απασχολούν σε μεγάλο βαθμό όταν τα δίκτυα αυτοματισμού είναι κλειστά, δηλαδή δεν έχει δοθεί σε αυτά πρόσβαση στο διαδίκτυο ή επικοινωνία με αλλά δίκτυα εντός του χώρου όπου βρίσκονται. Όταν είναι αναγκαία η σύνδεση τους στο διαδίκτυο ή στο εταιρικό δίκτυο, τότε πάντα θα πρέπει να βρίσκονται πίσω από τείχος προστασίας firewall.

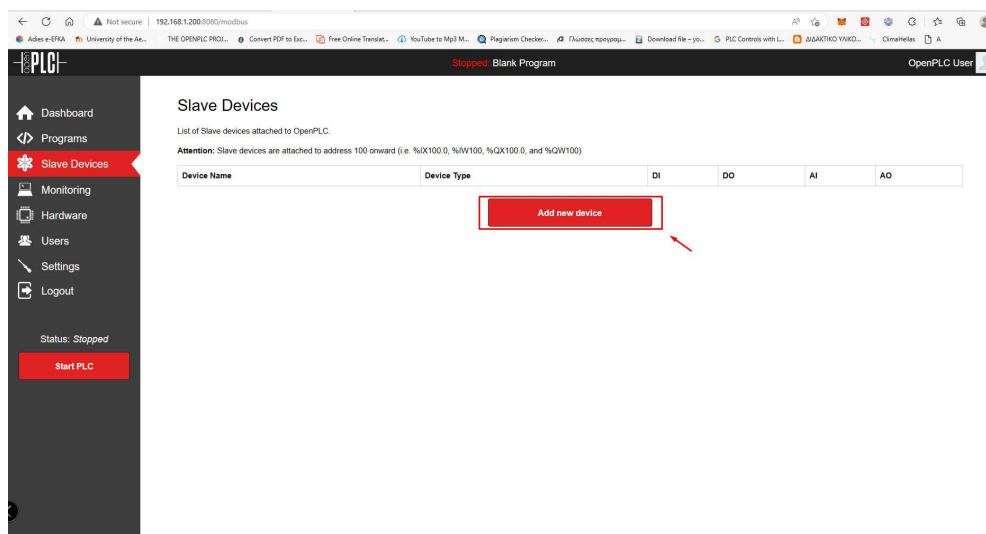
5.4 Σύνδεση NodeMCU ESP8266 με OpenPLC

Συνδεόμαστε στο web περιβάλλον του OpenPLC Raspberry Pi δίνοντας την IP διεύθυνση του και το port στο οποίο έχει ρυθμιστεί ο web server του Raspberry Pi, καθώς και στοιχεία πρόσβασης όνομα χρήστη και κωδικός χρήστη.



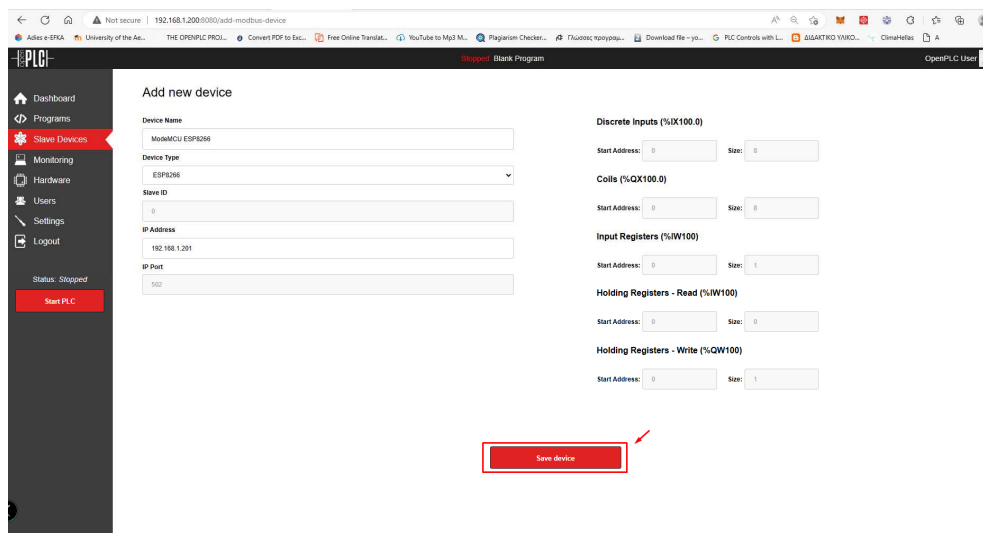
Εικόνα 5.6 : Καρτέλα διαχειρίσεις Slave μονάδων στο OpenPLC Runtime στο Raspberry Pi.

Κάνουμε κλικ στην επιλογή Slave Device (εικόνα 5.6).



Εικόνα 5.7 : Προσθήκη νέας μονάδας Modbus/TCP στο OpenPLC Runtime στο Raspberry Pi.

Κάνουμε κλικ στην επιλογή Add new device (εικόνα 5.7).



Εικόνα 5.8 : Αποθήκευση νέας μονάδας Modbus TCP στο OpenPLC Runtime στο Raspberry Pi.

Πλέον βρισκόμαστε στην σελίδα στην οποία μπορούμε να προσθέσουμε μια συσκευή διακομιστή (server) με ModbusTCP. Βέβαια σε αυτή την σελίδα μπορούμε να προσθέσουμε όλες τις υποστηριζόμενες Modbus συσκευές από το OpenPLC Runtime

Συμπληρώνουμε τα πεδία με τα αντίστοιχα στοιχεία όπως το όνομα τον τύπο και την IP διεύθυνση της συσκευής (αυτή πρέπει να είναι η ίδια με αυτή που δώσαμε στην συσκευή σύμφωνα με την εικόνα 5.5).

Κάνουμε κλικ στην επιλογή Save device και επιστρέφουμε στην σελίδα της (εικόνα 5.8).

Slave Devices

List of Slave devices attached to OpenPLC.

Attention: Slave devices are attached to address 100 onward (i.e. %IX100.0, %IW100, %QX100.0, and %QW100)

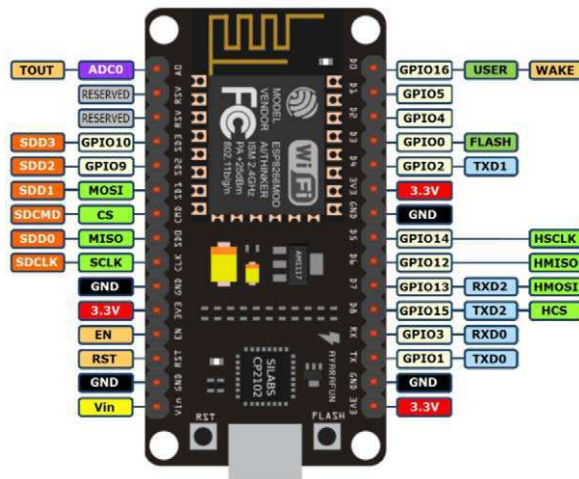
Device Name	Device Type	DI	DO	AI	AO
ModemCU ESP8266	ESP8266	%IX100.0 to %IX100.7	%QX100.0 to %QX100.7	%IW100 to %IW100	%QW100 to %QW100

[Add new device](#)

Εικόνα 5.9 : Πίνακας με Slave μονάδες συνδεδεμένες με το OpenPLC Runtime στο Raspberry Pi.

Πλέον η μονάδα έχει δηλωθεί και από την (εικόνα 5.9) βλέπουμε τις διεύθυνσης που έχουν οι εισόδοι και οι εξόδοι της συσκευής NodeMCU ESP8266. Από τον παραπάνω πίνακα μπορείτε να δείτε τις διευθύνσεις που δίνει το OpenPLC Runtime για κάθε συσκευή, δηλαδή από ποια διεύθυνση ξεκινά και σε ποια τελειώνει. Όπως π.χ. στην (εικόνα 5.9) βλέπουμε ότι η συσκευή ESP8266 έχει 8 ψηφιακές εισόδους, 8 ψηφιακές εξόδους, 1 αναλογική είσοδο και 1 αναλογική έξοδο. Επομένως, η συσκευή έχει τις 8 ψηφιακές εισόδους αντιστοιχισμένες στο %IX100.0 έως %IX100.7, τις 8 εξόδους αντιστοιχισμένες στο %QX100.0 έως %QX100.7, μια αναλογική είσοδο %IW100 και μια αναλογική έξοδο %QW100.

Το OpenPLC Runtime μας δίνει την δυνατότητα να προσθέσουμε αρκετές (127) slave συσκευές συνδεδεμένες με το πρωτόκολλο Modbus TCP είτε με το Modbus RTU. Οπότε πρέπει να δίνουμε την δέουσα προσοχή στο εύρος και στη αντιστοίχιση των εισόδων / εξόδων αυτών από το OpenPLC.



Εικόνα 5.10 : NodeMCU ESP8266 PinOut

ΚΕΦΑΛΑΙΟ 6

Εγκατάσταση του λογισμικού HMI ScadaBR για οπτικοποίηση προγραμμάτων PLC.

6.1 Οπτικοποίηση λειτουργίας OpenPLC με ScadaBR

Το HMI σημαίνει Human Machine Interface, το οποίο είναι ένας τύπος λογισμικού που επιτρέπει στους χειριστές να αλληλοεπιδρούν με συστήματα βιομηχανικού ελέγχου όπως τα συστήματα SCADA (Supervisory Control And Data Acquisition). Το ScadaBR είναι ένα σύστημα SCADA ανοιχτού κώδικα που αναπτύσσεται και διατηρείται από μια κοινότητα χρηστών. Έχει σχεδιαστεί για να είναι ευέλικτο και εύκολο στη χρήση και μπορεί να χρησιμοποιηθεί για την παρακολούθηση και τον έλεγχο ενός ευρέος φάσματος βιομηχανικών συστημάτων και διαδικασιών.

6.2 Τι σημαίνει ο όρος SCADA

Το SCADA (Supervisory Control And Data Acquisition) ονομάζεται ο Εποπτικός Έλεγχος και Απόκτηση Δεδομένων. Είναι ένας τύπος συστήματος λογισμικού και υλικού που χρησιμοποιείται για την παρακολούθηση και τον έλεγχο βιομηχανικών διεργασιών, όπως αυτές σε εργοστάσια παραγωγής, σταθμούς ηλεκτροπαραγωγής, αντλιοστάσια, εγκαταστάσεις επεξεργασίας λυμάτων κ.α. Τα συστήματα SCADA περιλαμβάνουν συνήθως έναν κεντρικό υπολογιστή ή διακομιστή που συλλέγει δεδομένα από αισθητήρες και συσκευές σε μια εγκατάσταση και χρησιμοποιεί αυτά τα δεδομένα για τον έλεγχο και την αυτοματοποίηση των διαφόρων διαδικασιών. Συχνά παρέχουν επίσης μια διεπαφή χρήστη που επιτρέπει στους χειριστές να προβάλλουν και να ελέγχουν το σύστημα από απόσταση. Τα συστήματα SCADA είναι ζωτικής σημασίας σε πολλές βιομηχανίες και χρησιμοποιούνται για τη διασφάλιση της ασφαλούς και αποτελεσματικής λειτουργίας διαφόρων τύπων εξοπλισμού και εγκαταστάσεων.

Μπορούμε να πούμε ότι ένα SCADA λειτουργεί ως το «κεντρικό νευρικό σύστημα» ενός συστήματος αυτοματισμού, παρακολουθώντας όλες τις συσκευές και προσφέροντας οργανωμένη πρόσβαση στα χειριστήρια και τις παραμέτρους τους.

6.3 Στοιχεία ενός συστήματος SCADA

Τα κύρια στοιχεία ενός συστήματος SCADA είναι:

- Απομακρυσμένες τερματικές μονάδες RTU ή προγραμματιζόμενοι λογικοί ελεγκτές PLC που βρίσκονται στην τοποθεσία του πεδίου και συλλέγουν δεδομένα από αισθητήρες και άλλες συσκευές πεδίου.
- Ένα σύστημα επικοινωνίας, όπως ένα ενσύρματο ή ασύρματο δίκτυο, που χρησιμοποιείται για τη μετάδοση δεδομένων από τις συσκευές πεδίου στον κεντρικό υπολογιστή ή διακομιστή.
- Ο κεντρικός υπολογιστής ή διακομιστής που χρησιμοποιείται για την επεξεργασία των δεδομένων και τη λήψη αποφάσεων σχετικά με τον τρόπο ελέγχου της βιομηχανικής διαδικασίας.

- Λογισμικό Human Machine Interface HMI που χρησιμοποιείται για την εμφάνιση των δεδομένων και επιτρέπει στους χειριστές να ελέγχουν τη διαδικασία.

Τα συστήματα SCADA χρησιμοποιούνται σε ένα ευρύ φάσμα βιομηχανικών εφαρμογών, όπως η παραγωγή ενέργειας, η επεξεργασία νερού και λυμάτων, η παραγωγή πετρελαίου και φυσικού αερίου και η κατασκευή. Επιτρέπουν την παρακολούθηση και τον έλεγχο των βιομηχανικών διεργασιών σε πραγματικό χρόνο, που μπορούν να βελτιώσουν την απόδοση, να μειώσουν το κόστος και να αυξήσουν την ασφάλεια.

Τα τελευταία χρόνια, με την άνοδο του IoT, τα συστήματα SCADA έχουν ενσωματωθεί με την τεχνολογία IoT για να επιτρέψουν την απομακρυσμένη παρακολούθηση και έλεγχο των βιομηχανικών διαδικασιών, καθώς και τη βελτίωση των δυνατοτήτων ανάλυσης δεδομένων και λήψης αποφάσεων.

6.4 Το ScadaBR

Το ScadaBR είναι ένα σύστημα SCADA ανοιχτού κώδικα που αναπτύχθηκε από το Εθνικό Εργαστήριο Επιστημονικών Υπολογιστών της Βραζιλίας LNCC και μπορεί να χρησιμοποιηθεί για την παρακολούθηση και τον έλεγχο ενός ευρέος φάσματος βιομηχανικών διεργασιών. Μπορεί να εγκατασταθεί σε λειτουργικό σύστημα Windows είτε Linux και χρησιμοποιεί μια διαδικτυακή διεπαφή για την οπτικοποίηση και έλεγχο δεδομένων.

Μερικά από τα βασικά χαρακτηριστικά του ScadaBR περιλαμβάνουν:

- Υποστήριξη για ένα ευρύ φάσμα συσκευών: Το ScadaBR μπορεί να συνδεθεί με μια ποικιλία συσκευών, συμπεριλαμβανομένων PLC, RTU και αισθητήρων, και μπορεί να χρησιμοποιήσει μια ποικιλία πρωτοκόλλων, όπως Modbus, OPC και SNMP, για να επικοινωνήσει μαζί τους.
- Οπτικοποίηση δεδομένων: Το ScadaBR παρέχει μια ποικιλία εργαλείων για οπτικοποίηση δεδομένων, συμπεριλαμβανομένων γραφημάτων σε πραγματικό χρόνο, γραφημάτων τάσεων, διαχείρισης συναγερμών κ.α.
- Διαχείριση συναγερμών: Το ScadaBR διαθέτει ενσωματωμένο σύστημα διαχείρισης συναγερμού που μπορεί να χρησιμοποιηθεί για την παρακολούθηση του συστήματος για μη φυσιολογικές συνθήκες και τη λήψη των κατάλληλων μέτρων.
- Αναφορές και δεδομένα ιστορικού: Το ScadaBR μπορεί να δημιουργήσει αναφορές και να αποθηκεύσει δεδομένα ιστορικού για ανάλυση και αντιμετώπιση προβλημάτων.
- Ανοιχτού κώδικα και προσαρμόσιμο: Το ScadaBR είναι λογισμικό ανοιχτού κώδικα, που σημαίνει ότι μπορεί να τροποποιηθεί για την κάλυψη συγκεκριμένων αναγκών.

Το ScadaBR χρησιμοποιείται ευρέως στη Βραζιλία και σε άλλες χώρες και θεωρείται ένα από τα καλύτερα συστήματα SCADA ανοιχτού κώδικα που διατίθενται στην αγορά. Η εγκατάσταση του ScadaBR μπορεί να είναι μια απλή διαδικασία, αλλά απαιτεί ένα ορισμένο επίπεδο τεχνικών γνώσεων σε συστήματα αυτοματισμού, πρωτόκολλα μεταφοράς δεδομένων και H/Y.

6.5 Εγκατάσταση ScadaBR

Συνολικά, η διαδικασία εγκατάστασης του ScadaBR απαιτεί ορισμένες τεχνικές γνώσεις και κατανόηση των διακομιστών ιστού και των συστημάτων βιομηχανικού ελέγχου. Συνιστάται να συμβουλευόμαστε την επίσημη τεκμηρίωση και τις οδηγίες για να το πως θα πρέπει να εγκατασταθεί στο λειτουργικό σύστημα που θα επιλεγεί να γίνει η εγκατάσταση του.

Στα πλαίσια αυτής της διπλωματικής εργασίας η εγκατάσταση του λογισμικού ScadaBR θα γίνει σε περιβάλλον Windows.

Προϋποθέσεις: Πριν εγκαταστήσετε το ScadaBR, θα χρειαστεί να έχετε έναν υπολογιστή με λειτουργικό σύστημα Windows και περιβάλλον εκτέλεσης Java (JRE) έκδοση 8 ή νεότερη. Θα χρειαστεί επίσης να έχετε εγκατεστημένο και διαμορφωμένο έναν διακομιστή Web, όπως ο Apache Tomcat 9.0.

Ακολουθεί μια γενική επισκόπηση των βημάτων που απαιτούνται για την εγκατάσταση του ScadaBR.

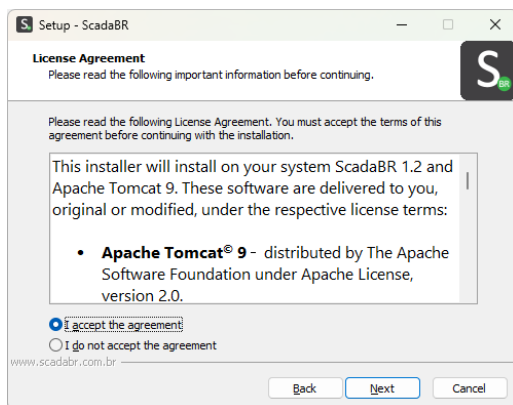
Από τον ιστότοπο GitHub (<https://github.com/ScadaBR/ScadaBR/releases>) πραγματοποιούμε την λήψη της πιο πρόσφατης έκδοσης του λογισμικού ScadaBR, αυτή είναι η έκδοση 1.2 (20/05/21). Διατίθεται με τη μορφή αρχείου εγκατάστασης (installation), το οποίο μπορεί εύκολα να εγκατασταθεί σε Windows.

Πριν προχωρήσουμε στην εγκατάσταση του λογισμικού ScadaBR πρέπει να βεβαιωθούμε ότι είναι εγκατεστημένη στον Η/Υ η έκδοση java 8 (java JRE 8 Update 351 και άνω). Σε περίπτωση που δεν είναι εγκατεστημένη μπορούμε να την κατεβάσουμε από το επίσημο ιστότοπο της java (<https://www.java.com/en/download/manual.jsp>).



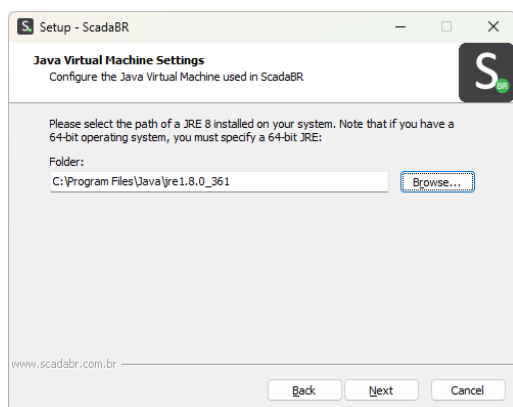
Εικόνα 6.1 : Εγκατάσταση λογισμικού ScadaBR.

Κάνουμε κλικ στο πλήκτρο [Next]



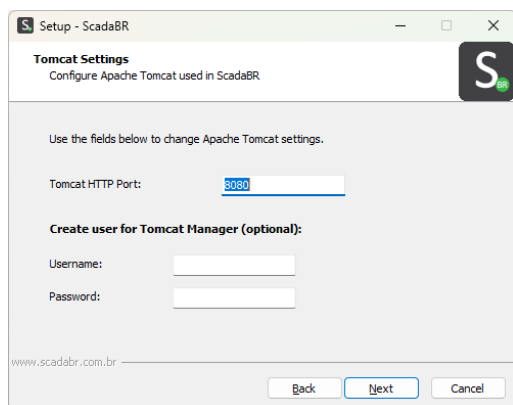
Εικόνα 6.2 : Εγκατάσταση λογισμικού ScadaBR.

Κάνουμε αποδοχή των όρων χρήσεως του λογισμικού, και κάνουμε κλικ στο πλήκτρο [Next]



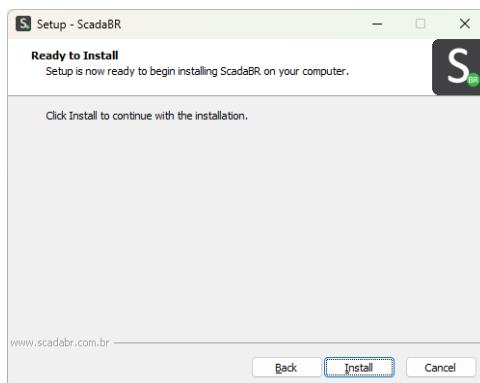
Εικόνα 6.3 : Εγκατάσταση λογισμικού ScadaBR.

Σε αυτό το σημείο πρέπει να δηλώσουμε το path στο οποίο είναι εγκατεστημένη η java. Προσοχή αν δηλώσουμε λάθος path ή path με διαφορετική έκδοση της java το λογισμικό ScadaBR δεν θα λειτουργήσει. Εφόσον δηλώσουμε το path κάνουμε κλικ στο πλήκτρο [Next].



Εικόνα 6.4 : Εγκατάσταση λογισμικού ScadaBR.

Σε αυτό το σημείο μπορούμε να δηλώσουμε διαφορετική θύρα (port) από την προτεινομένη, στην οποία θα απαντά το λογισμικό ScadaBR. Προσοχή το port πρέπει να είναι μέσα στο εύρος του TCP (από 1 έως 65535) προτείνεται να κρατήσουμε τη θύρα που μας προτείνει η εγκατάσταση. Για τα πεδία username και password η συμπλήρωση τους είναι προαιρετική. Εφόσον επιλέξουμε θύρα κάνουμε κλικ στο πλήκτρο [Next].



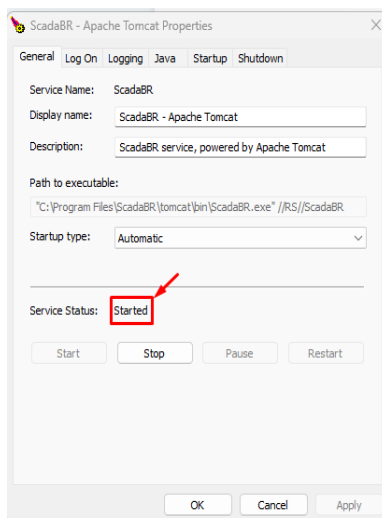
Εικόνα 6.5 : Εγκατάσταση λογισμικού ScadaBR.

Κάνουμε κλικ στο πλήκτρο [Install] και το πρόγραμμα εγκατάστασης προχωρά στην εγκατάσταση του λογισμικού ScadaBR.



Εικόνα 6.6 : Εγκατάσταση λογισμικού ScadaBR.

Κάνουμε κλικ στο πλήκτρο [Finish]. Η εγκατάσταση λογισμικού ScadaBR έχει ολοκληρωθεί. Τώρα θα πρέπει να προχωρήσουμε στον έλεγχο και στην τροποποίηση των ρυθμίσεων εγκατάστασης αν αυτό είναι απαραίτητο. Αυτό μπορεί να γίνει από την εφαρμογή ScadaBR – Apache Tomcat Properties. Ο κύριος έλεγχος που πρέπει να γίνει είναι να ελέγξουμε αν το services του Apache Tomcat είναι σε λειτουργία Started (εικόνα 6.7).



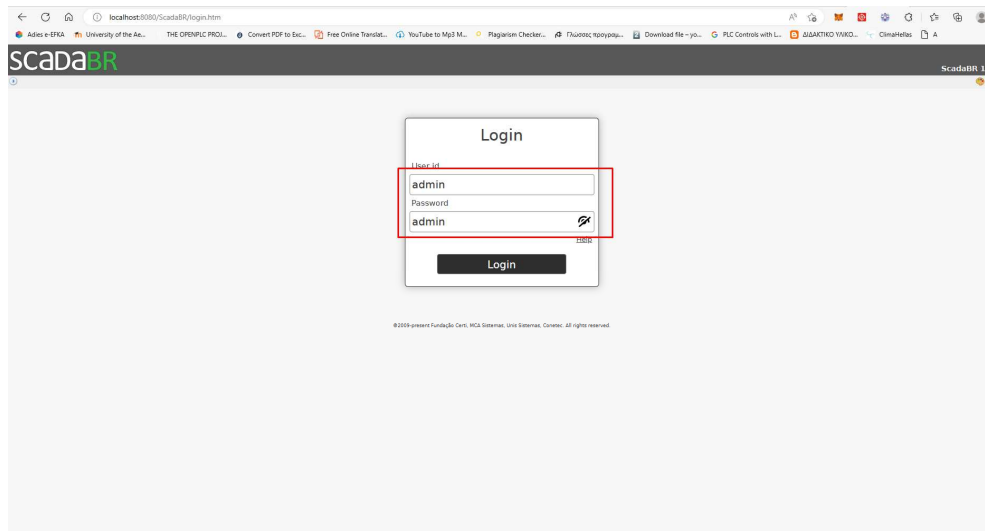
Εικόνα 6.7 : ScadaBR – Apache Tomcat Properties

6.6 Ρύθμιση και χρήση του ScadaBR

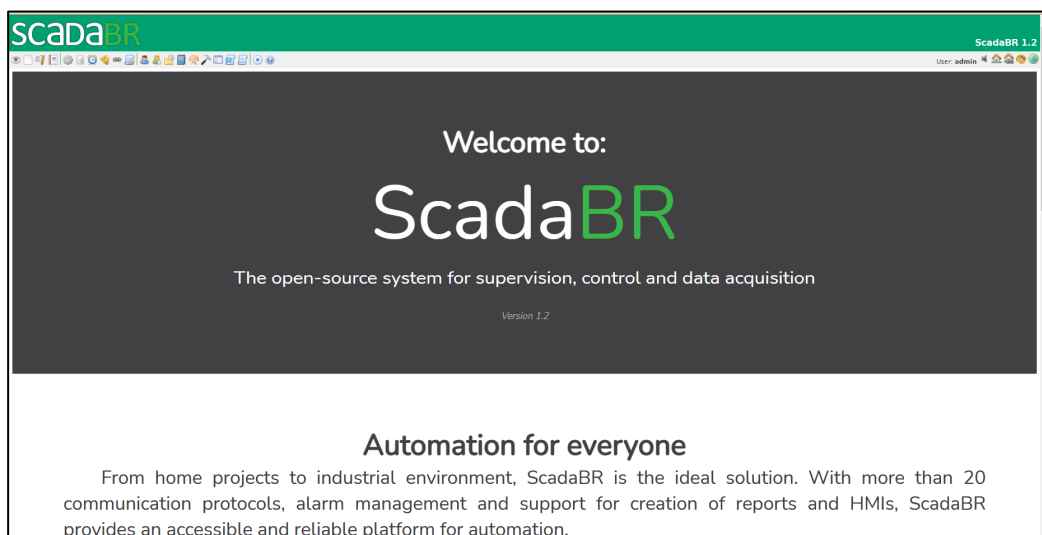
Μετά την ολοκλήρωση της εγκατάστασης του ScadaBR, θα πρέπει να προχωρήσουμε στην ρύθμιση των βασικών παραμέτρων του λογισμικού ώστε να είναι εφικτή η σύνδεση του με τον προγραμματιζόμενο λογικό ελεγκτή. Αυτό περιλαμβάνει τη ρύθμιση της επικοινωνίας με τα OpenPLC Raspberry Pi, καθώς και με όλες τις άλλες συσκευές και αισθητήρες που πιθανόν να υπάρχουν στο σύστημα, καθώς και την διαμόρφωση συναγερμών, συμβάντων κ.α.

Ακολουθεί μια γενική επισκόπηση των βημάτων που απαιτούνται για την ρύθμιση του ScadaBR.

Ανοίξτε ένα πρόγραμμα περιήγησης ιστού και εισαγάγετε την ακόλουθη διεύθυνση URL: <http://localhost:8080/ScadaBR/>

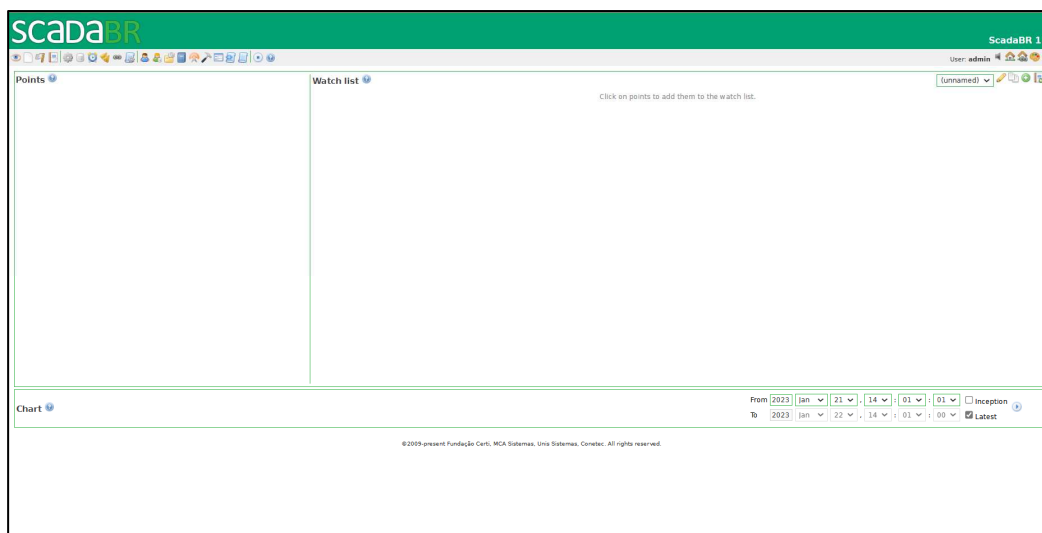


Εικόνα 6.8 : Σελίδα σύνδεσης στο ScadaBR.



Εικόνα 6.9 : Σελίδα καλωσορίσματος του ScadaBR.

Για να συνδεθούμε στο ScadaBR θα πρέπει να δώσουμε τα στοιχεία πρόσβασης. Το προεπιλεγμένο User-ID και ο κωδικός πρόσβασης είναι λέξη admin. Έπειτα κάνουμε κλικ στο πλήκτρο [Login].




Εικόνα 6.10 : Αρχική (default) σελίδα σύνδεσης στο ScadaBR.

Κατά την πρώτη σύνδεση στο περιβάλλον του ScadaBR θα εμφανιστή η σελίδα καλωσορίσματος. Ενώ σε όλες τις επόμενες συνδέσεις, θα ανοίξει η σελίδα Watch list, η οποία άλλη σελίδα έχει οριστεί ως αρχική (default) σελίδα (εικόνα 6.10). Στην σελίδα αυτή βλέπουμε όλα τα event που υπάρχουν στο ScadaBR. Στην αρχική εκκίνηση του προγράμματος η σελίδα είναι κενή γιατί ακόμη δεν έχουμε προσθέσει κάτι.




Εικόνα 6.11 : Data sources icon.

6.6.1 Εισαγωγή πηγών δεδομένων Data sources

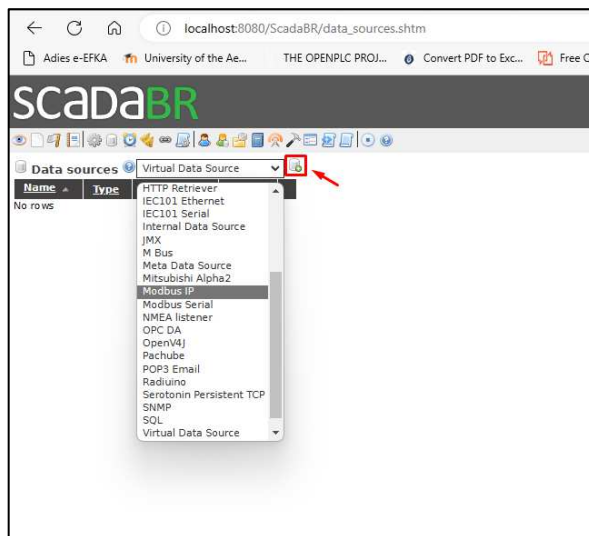
Για να προσθέσουμε τον εξοπλισμό που θα διαχειριστούμε μέσα από το ScadaBR θα πρέπει να μεταβούμε στην σελίδα Data sources κάνοντας κλικ στο εικονίδιο  (εικόνα 6.11) το εικονίδιο μοιάζει με το εικονίδιο ενός σκληρού δίσκου.

Από εδώ προσθέσουμε τις συσκευές PLC, τους αισθητήρες και ότι άλλο θα χρειαστούμε για να δημιουργήσουμε ένα περιβάλλον ελέγχου μέσα από το ScadaBR.

Το ScadaBR υποστηρίζει ένα αρκετά μεγάλο πλήθος πρωτοκόλλων, συνδέσεις εξοπλισμού αυτοματισμού όπως PLC, αισθητήρια κ.α. Δεν είναι δυνατό στα πλαίσια της παρούσας διπλωματικής να παρουσιασθούν και αναλυθούν όλα. Οπότε θα ασχοληθούμε μόνο με τον τρόπο σύνδεσης του πρωτοκόλλου Modbus TCP.

Κάνοντας κλικ στην drop-down list δίπλα στο Data sources. Μπορούμε να βρούμε και να επιλέξουμε το πρωτόκολλο που επιθυμούμε από μια μεγάλη λίστα διαφορετικών πρωτοκόλλων. Εδώ βρίσκουμε και επιλέγουμε το πρωτόκολλο Modbus IP και κάνουμε κλικ στο εικονίδιο προσθήκης  (εικόνα 6.12). Αυτό μας μεταφέρει σε ένα νέο παράθυρο στο οποίο θα ρυθμίσουμε την πηγή

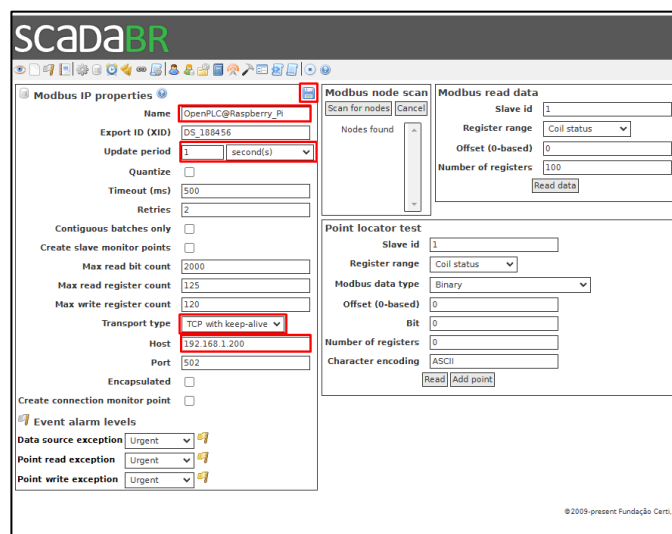
δεδομένων μας, έτσι ώστε το ScadaBR να μπορεί να συνδεθεί μαζί της για να διαβάσει τις πληροφορίες.



Εικόνα 6.12 : Επιλογή τύπου εισαγωγής δεδομένων στο ScadaBR.

Στο πεδίο Name θα ονομάσουμε τη συσκευή που θα συνδέσουμε στο ScadaBR π.χ. Open-PLC@Raspberry_Pi.

Στο πεδίο Update period θα επιλέξουμε 1 second. Εδώ ορίζουμε το πόσο συχνά θα ενημερώνεται το ScadaBR με τα δεδομένα από την συσκευή που θα προσθέσουμε.



Εικόνα 6.13 : Σύνδεση OpenPLC Raspberry Pi με το ScadaBR.


Στο πεδίο Transport type επιλέγουμε το TCP with keep-alive. Θα μπορούσαμε να επιλέξουμε το TCP. Η μόνη διαφορά μεταξύ των δυο είναι ότι το TCP κλείνει την σύνδεση μετά από κάθε ερώτημα, ενώ το TCP with keep-alive κρατά ανοικτή την σύνδεση για όσο χρονικό διάστημα είναι συνδεδεμένη η συσκευή με το ScadaBR.

Τέλος στο πεδίο Host θα προσθέσουμε την IP Address της συσκευής που συνδέουμε με το ScadaBR. Το port μένει το 502 που είναι και το default port για το πρωτόκολλο Modbus TCP.

Πλέον μπορούμε να ελέγξουμε από το πλαίσιο Modbus read data αν έχουμε διαμορφώσει σωστά

τις ρυθμίσεις του ScadaBR με την συσκευή.


Επιλέγοντας στο πεδίο Register range to Coil status και κλικ στο πλήκτρο Read data διαβάζουμε την κατάσταση των πηνίων. Με τον ίδιο τρόπο επιλέγοντας Input status διαβάζουμε την κατάσταση των εισόδων, με Holding register την κατάσταση των καταχωρητών μνήμης, με Input register τους καταχωρητές εισόδου.

Αποθηκεύουμε τις ρυθμίσεις που έχουμε κάνει κάνοντας κλικ στο εικονίδιο  που βρίσκεται στο πλαίσιο Modbus IP properties.

Το επόμενο βήμα είναι να προσθέσουμε τα data point (σημεία δεδομένων) από την συσκευή που συνδέσαμε. Με τον όρο αυτό εννοούμε ότι έχουμε ορίσει στο πρόγραμμα του PLC από (εισόδους, εξόδους, απαριθμητές, χρονικά, καταχωρητές κ.α.) τα αισθητήρια και γενικά όποια πληροφορία μπορούμε να πάρουμε από την συνδεδεμένη συσκευή για να την εμφανίσουμε στο ScadaBR.

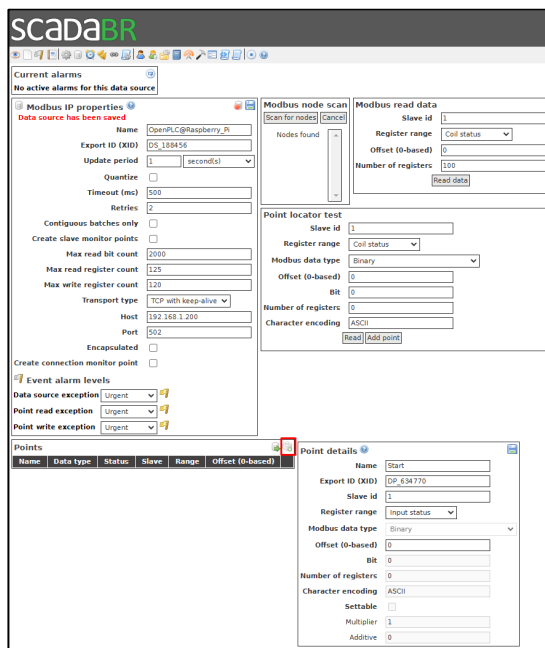
Προτείνεται όταν αρχίσουμε να προσθέτουμε τα σημεία να έχουν την εικόνα του προγράμματος με τα στοιχεία αυτά είτε σε χαρτί είτε στον H/Y. Δίπλα - δίπλα τον OpenPLC Editor με το ScadaBR, ώστε να βλέπουμε τον τύπο και την διεύθυνση τους.

6.6.2 Εισαγωγή σημείων δεδομένων Data point

Για να προσθέσουμε τα Data point κάνουμε κλικ στο εικονίδιο  που βρίσκετε στην καρτέλα Points και μας εμφανίζεται πλαίσιο Point details.

Στο πεδίο Name θα ονομάσουμε το στοιχείο data point.


Στο πεδίο Register range θα επιλέγουμε τον τύπο του στοιχείου. Προσοχή όταν εισάγουμε καταχωρητή θα πρέπει από το πεδίο Modbus data type να επιλέγουμε και τον τύπο του καταχωρητή, δηλαδή το πλήθος των bytes από τα όποια αποτελείται καθώς και τον τύπο int, float κ.α.






Εικόνα 6.14 : Εισαγωγή Data point για την λήψη / αποστολή δεδομένων στο PLC.

Στο πεδίο Offset θα δώσουμε την διεύθυνση του στοιχείου όπως αυτή μας δίνετε από το πρόγραμμα


του PLC ή του αισθητήριου, π.χ. διεύθυνση πηνίου (coil) %QX0.0 στο πεδίο αυτό θα δώσουμε την τιμή 0 αν ήταν %QX0.1 θα δίνουμε την τιμή 1 κ.ο.κ.

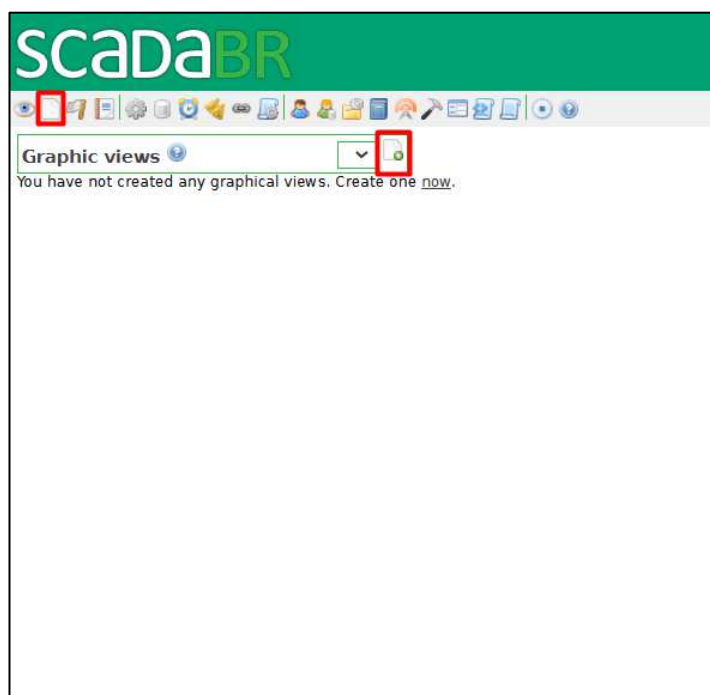
Όταν ολοκληρώσουμε με την ρύθμιση των παραμέτρων του στοιχείου που θα προσθέσουμε κάνουμε κλικ στο εικονίδιο  για να αποθηκεύσουμε το στοιχείο.

Αν θέλουμε να προσθέσουμε και άλλα στοιχεία ακολουθούμε τα παραπάνω βήματα.


Μετά την ολοκλήρωση της προσθήκης των στοιχείων κάνουμε κλικ στο εικονίδιο  για να ενεργοποιήσουμε όλα τα στοιχεία που προσθέσαμε ή στο εικονίδιο  που βρίσκετε στην στήλη Status για κάθε ένα από τα στοιχεία που προσθέσαμε. Επίσης κάνουμε κλικ στο εικονίδιο  στο πλαίσιο Modbus IP properties για την ενεργοποίηση της πηγής δεδομένων, ώστε αυτά να είναι διαθέσιμα στο ScadaBR.

6.6.3 Οπτικοποίηση των στοιχείων πηγής

Για να δημιουργήσουμε την οπτικοποίηση των στοιχείων που προσθέσαμε στο ScadaBR, καθώς και να τα διαχειριστούμε θα πρέπει να μεταβούμε στην σελίδα Graphical view κάνοντας κλικ στο εικονίδιο .

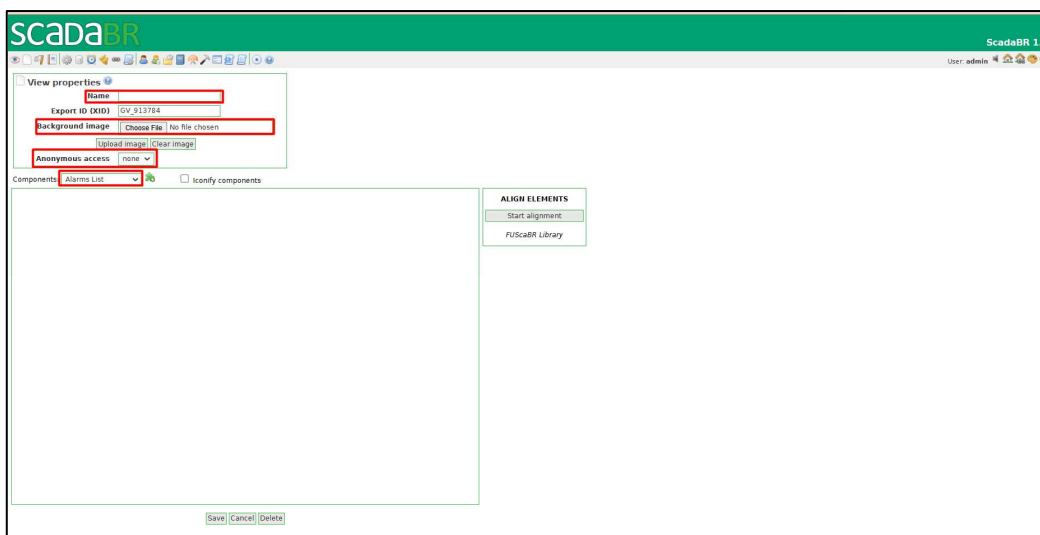


Εικόνα 6.15 : Σελίδα για την δημιουργία Graphic views στο ScadaBR.

Για να προσθέσουμε την αρχική σελίδα οπτικής παρουσίασης των στοιχείων μας κάνουμε κλικ στο εικονίδιο .

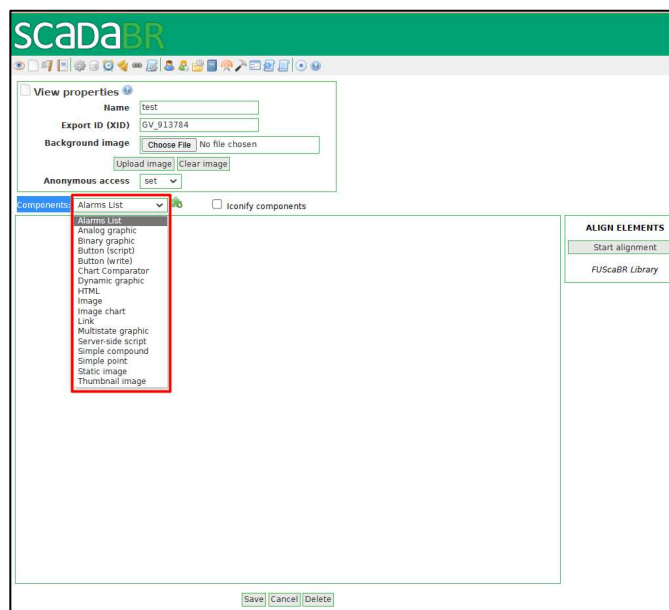
Στο πεδίο Name θα ονομάσουμε την σελίδα στο ScadaBR π.χ. OpenPLC@Raspberry_Pi_View.

Στο πεδίο Background image μπορούμε να εισάγουμε μια εικόνα φόντου. Προτείνεται να εισάγουμε μια εικόνα φόντου σχετικά με αντικείμενο της οπτικοποίησης π.χ. έναν ταινιόδρομο ή ένα σύνολο από μηχανές, βάνες, όργανα μέτρησης θερμοκρασίας κ.α. Τα όποια θα μας βοηθήσουν να έχουμε καλύτερο και πληρέστερο έλεγχο του συστήματος.




Εικόνα 6.16 : Παράθυρο ρυθμίσεις σελίδας οπτικοποίησης.

Το πεδίο Anonymous access μας δίνει την δυνατότητα να έχουν πρόσβαση στο οπτικοποιημένο και ουσιαστικά στο σύστημα μας, όλοι όσοι έχουν πρόσβαση στο δίκτυο στο οποίο βρίσκετε ο server του ScadaBR. Προτείνετε να αφήσουμε την επιλογή none για προφανή λόγο. Αλλά αν είμαστε σίγουροι ότι το δίκτυο αυτό είναι ένα κλειστό δίκτυο δηλαδή ένα δίκτυο με ελεγχόμενη πρόσβαση τότε μπορούμε να αλλάξουμε αυτή την επιλογή σε read ή set.



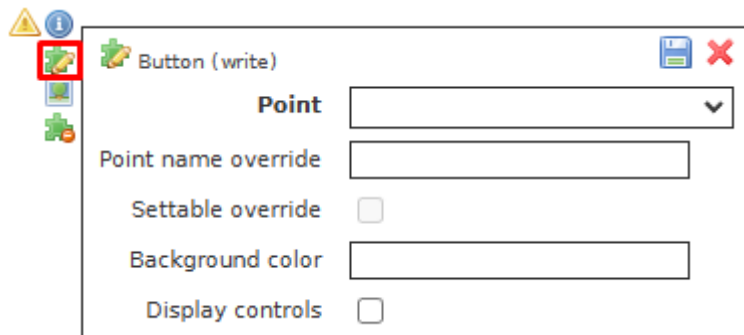
Εικόνα 6.17 : Διαθέσιμοι τύποι αντικειμένων για οπτικοποίηση στο ScadaBR.

Από το πεδίο Components με την βοήθεια του drop-down list (εικόνα 6.17) μπορούμε να προσθέσουμε τα στοιχεία που θέλουμε στην οθόνη οπτικοποίησης, πχ Button (write), Analog graphic κ.α. Κάνοντας κλικ στο εικονίδιο , το αντικείμενο που επιλέξαμε τοποθετείτε στην οθόνη οπτικοποίησης για παραμετροποίηση (εικόνα 6.18). Οι επιλογές που έχουμε για την ρύθμιση του αντικειμένου είναι: α) σύνδεση αντικειμένου με λειτουργία π.χ. είσοδο, έξοδο, καταχωρητές κ.α. ή ενδείξεις αισθητήριου, β) επιλογή τύπου αντικειμένου π.χ. Led, μπουτόν, όργανο μέτρησης κ.α., καθώς και ορισμός των διαστάσεων του, γ) διαγραφή του αντικειμένου.

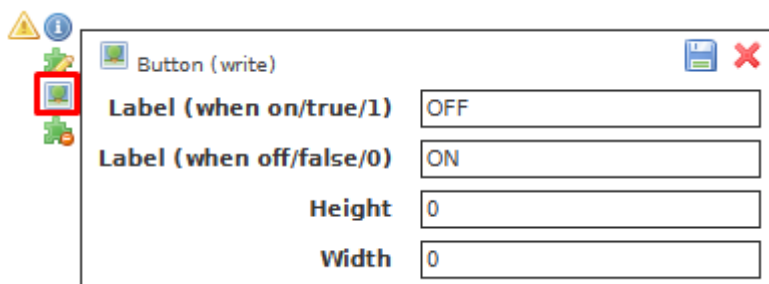
Για να προσθέσουμε και αλλά αντικείμενα ακολουθούμε την παραπάνω διαδικασία.



Εικόνα 6.18 : Εισαγωγή αντικειμένου οπτικοποίησης Button στο ScadaBR.



Εικόνα 6.19 : Αντιστοίχιση αντικειμένου σε λειτουργία.



Εικόνα 6.20 : Διαμόρφωση αντικειμένου.



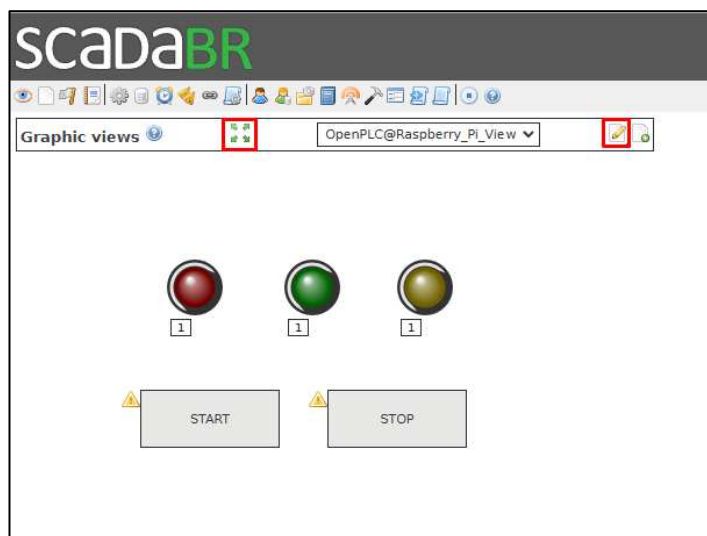
Εικόνα 6.21 : Τελική μορφή αντικειμένου σύμφωνα με τις παραπάνω ρυθμίσεις.



Εικόνα 6.22 : Επιλογή διαγραφής αντικειμένου.

Μετά το τέλος της τοποθέτησης των αντικειμένων, κάνουμε κλικ στο πλήκτρο [Save] στο κάτω μέρος του παραθύρου για την αποθήκευση τους και θα επιστρέψει στην προηγούμενη οθόνη. Η οθόνη αυτή πλέον θα έχει την μορφή και τα αντικείμενα που εισήγαμε κατά την παραπάνω διαδικασία (εικόνα 6.23). Με τις άλλες δυο επιλογές που μας δίνονται μπορούμε να ακυρώσουμε τις

ενέργειες που έχουμε κάνει κάνοντας κλικ στο πλήκτρο [Cancel]. Ενώ κάνοντας κλικ στο πλήκτρο [Delete] μπορούμε να διαγράψουμε την καρτέλα και να επιστρέψουμε στην προηγούμενη οθόνη.



Εικόνα 6.23 : Η σελίδα οπτικοποίησης του ScadaBR.

Συμπεράσματα

Ανακεφαλαιώνοντας διαπιστώνουμε ότι ο στόχος που τέθηκε, για τη δημιουργία ενός αξιόπιστου προγραμματισμένου λογικού ελεγκτή PLC μικρού κόστους, με δυνατότητες επέκτασης και επαναχρησιμοποίησης, χρησιμοποιώντας το λογισμικό ανοικτού κώδικα OpenPLC Project, καθώς και το λογισμικό διεπαφή ανθρώπου - μηχανής HMI ScadaBR, σε συνδυασμό με υλικό (hardware) όπως το Raspberry Pi και το NodeMCU ESP8266, είναι απόλυτα εφικτός.

Διαπιστώθηκε επίσης ότι σε ορισμένες περιπτώσεις χρήσης υπερτερούσε λόγω του άμεσου προγραμματισμού του, χωρίς να υπάρχει η ανάγκης χρήσης ειδικού εξοπλισμού, όπως εξωτερικός προγραμματιστής, ειδικά καλώδια σύνδεσης PLC - Η/Υ ή λογισμικά προγραμματισμού που χρειάζονται άδειες χρήσης κ.α.

Δεν προτείνετε η χρήση του σε περιβάλλοντα υψηλού κίνδυνου. Όπως για παράδειγμα ορυχεία, διυλιστήρια, και γενικά σε βιομηχανικά περιβάλλοντα που διαχειρίζονται υλικά μεγάλης επικινδυνότητας όπως εκρηκτικά, υγρά και αέρια καύσιμα, διότι το Raspberry Pi, δεν πληροί τις απαιτούμενες προδιαγραφές ασφαλείας για χρήση σε αυτά

Βιβλιογραφία

Νικόλαος Μαραντίδης (2000). *Αυτοματισμός με SIMATIC S7*. Έκδοση Siemens.

Μπερέτας Ιωάννης (2008). *Αυτοματισμός με χρήση PLC, Ολοκληρωμένες πρακτικές εφαρμογές*. Θεσσαλονίκη: ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ.

Collins Deniw – Lane Eamonn (1997). *Προγραμματιζόμενοι Ελεγκτές, Ένας Πρακτικός Οδηγός* (μτφ. Dr. Νικόλαος Κ. Κωνσταντινίδης). Θεσσαλονίκη: ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ (έτος έκδοσης πρωτοτύπου 1995).

Antosen Mejer Tom (2021). *PLC Controls with Ladder Diagram (LD), Monochrome: IEC 61131-3 and introduction to Ladder programming* (μτφ. Katrine Bay Madsen). Copenhagen: Books on Demand GmbH (έτος έκδοσης πρωτοτύπου 2020).

Gustavo Mendonça de Morais Rabelo Vieira (2018-2019). «Low-cost Industrial Controller based on the Raspberry Pi platform» Διπλωματική εργασία, School of Technology and Management of Bragança Portugal.

Διαδικτυακές Πηγές

Προγραμματιζόμενοι Λογικοί Ελεγκτές PLC (duth.gr)

<https://eclass.duth.gr/modules/document/file.php/TME134/%CE%A0%CE%A1%CE%9F%CE%93%CE%A1%CE%91%CE%9C%CE%9C%CE%91%CE%A4%CE%99%CE%96%CE%9F%CE%9C%CE%95%CE%9D%CE%9F%CE%99%CE%9B%CE%9F%CE%93%CE%99%CE%9A%CE%9F%CE%99%CE%95%CE%9B%CE%95%CE%93%CE%9A%CE%A4%CE%95%CE%A3%PLC.pdf>

OpenPLC Project.

<https://www.openplcproject.com/>
https://openplcproject.gitlab.io/openplc_v3/md_README.html

OpenPLC Forum.

<https://openplc.discussion.community/>

PLCOPEN.

<https://www.plcopen.org/downloads>

Raspberry PI.

<https://www.raspberrypi.org/>

VNC Viewer.

<https://www.realvnc.com/en/connect/download/viewer/windows/>

WinSCP.

<https://winscp.net/eng/download.php>

Modbus Web site.

<https://modbus.org/>

Modbus TCP/IP Specification User Guide.

https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

History of Modbus.

<http://www.summaryplanet.com/summary/History-of-Modbus-protocol.html>

Free library for Modbus.

<https://libmodbus.org/documentation/>

ScadaBR.

<https://www.scadabr.com.br/>

<https://github.com/ScadaBR/ScadaBR/releases>