

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ &  
ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**



**ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ**

**ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ ΕΝΕΡΓΕΙΩΝ**

**Ανάπτυξη Συστήματος Μέτρησης Θερμοκρασίας και Υγρασίας  
βασισμένο σε Arduino και WinCC\_OA**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΤΗΣ**

**Ανδρέου Δήμητρας**

**Επιβλέπων: ΤΣΙΠΟΛΙΤΗΣ ΓΕΩΡΓΙΟΣ  
ΚΑΘΗΓΗΤΗΣ ΕΜΠ**

**ΑΘΗΝΑ, ΙΑΝΟΥΑΡΙΟΣ 2015**



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ &  
ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ



ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ ΕΝΕΡΓΕΙΩΝ

**Ανάπτυξη Συστήματος Μέτρησης Θερμοκρασίας και Υγρασίας  
βασισμένο σε Arduino και WinCC\_OA**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΗΣ

**Ανδρέου Δήμητρας**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26/01/2015

.....  
Θεόδωρος Αλεξόπουλος  
Καθηγητής Ε.Μ.Π

.....  
Γεώργιος Τσιπολίτης  
Αν. Καθηγητής Ε.Μ.Π

.....  
Ευάγγελος Γαζής  
Καθηγητής Ε.Μ.Π

.....

**Δήμητρα Ν. Ανδρέου**

Διπλωματούχος Φυσικός Εφαρμογών, Σ.Ε.Μ.Φ.Ε., Ε.Μ.Π

© 2015, Εθνικό Μετσόβιο Πολυτεχνείο. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξολοκλήρου 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 τον συγγραφέα.

# Περίληψη

Στην παρούσα διπλωματική εργασία γίνεται μετάβαση του συστήματος αυτομάτου ελέγχου RD51 Slow Control System από το PVSS σε HVLab στο WinCC\_OA. Επιπλέον γίνεται ανάπτυξη του συστήματος HVLab με τη προσθήκη επιπλέον λειτουργιών. Το κύριο μέρος της εργασίας αποτελεί η ανάπτυξη ενός συστήματος μέτρησης της θερμοκρασίας και της υγρασίας με τη χρήση αισθητήρων που ελέγχονται από τη μικρουπολογιστική πλατφόρμα Arduino. Τα δεδομένα που λαμβάνονται από το Arduino μεταφέρονται μέσω ενός OPC Server για Arduino στο project HVLab του WinCC\_OA για επεξεργασία. Η εφαρμογή του παραπάνω συστήματος είναι ιδιαίτερα σημαντική καθώς επιτρέπει τον έλεγχο των περιβαλλοντικών παραμέτρων των ανιχνευτών κατά τη διάρκεια των testbeams.



# Abstract

This thesis consists of two main parts. The first one is the migration of the project RD51 Slow Control System from PVSS to WinCC\_OA. The migrated project was renamed to HVLab and developed in order to provide more functions and facilities. The second and most important part of this thesis is the development of a temperature and humidity measurement system based on Arduino and WinCC\_OA. This system is using sensors, controlled by the Arduino microcontroller, that provide the values of temperature and humidity. These values are transferred to WinCC\_OA via OPC Server for Arduino for elaboration. This application is very important because allows the users of the detectors to control the environmental parameters during the test beams.





## Ευχαριστίες

Ολοκληρώνοντας την παρούσα διπλωματική εργασία θα ήθελα να ευχαριστήσω, όλους όσους ήταν δίπλα μου κατά τη διάρκεια της εκπόνησής της.

Ένα μεγάλο ευχαριστώ στον καθηγητή μου, κ. Γεώργιο Τσιπολίτη που με εμπιστεύτηκε και μου έδωσε την ευκαιρία να συνεργαστώ μαζί του. Μέσα από αυτή τη συνεργασία, μου δόθηκε η δυνατότητα να εξερευνήσω τον κόσμο της Φυσικής Ύψηλών Ενεργειών και να εφαρμόσω στην πράξη θεωρητικές γνώσεις. Οι συμβουλές του θα με ακολουθούν σε όλη τη διάρκεια της μετέπειτα πορείας μου.

Ευχαριστώ επίσης τον Κωνσταντίνο Καρακώστα για τη βοήθειά του όπως επίσης και για το γεγονός ότι μέσω της δουλειάς του απέκτησα γνώσεις στον τομέα των συστημάτων αυτομάτου ελέγχου.

Θα ήθελα επίσης να ευχαριστήσω τους γονείς μου, Νίκο και Μαρουδιώ κυρίως για την ηθική αλλά και υλική υποστήριξη που μου παρέχουν και είναι δίπλα μου σε ό,τι χρειαστώ. Το μεγαλύτερο ευχαριστώ ανήκει στον παππού μου Γιάννη και την γιαγιά μου Πάτρα που είναι δίπλα μου σε κάθε μου βήμα και απόφαση.

Επιπλέον θα ήθελα να ευχαριστήσω το Στέργιο, για όλες τις στιγμές που είναι στο πλευρό μου και με στηρίζει καθώς και για τις καίριες συμβουλές και τη βοήθειά του κατά την εκπόνηση αυτής της εργασίας.



# Περιεχόμενα

Περίληψη	iii
Abstract	v
Ευχαριστίες	vii
Κατάλογος σχημάτων	x
<b>1 ΑΝΑΒΑΘΜΙΣΗ ΤΟΥ RD51 Slow Control System</b>	<b>1</b>
1.1 WinCC_OA . . . . .	1
1.1.1 Εισαγωγή . . . . .	1
1.1.2 Βασικά εργαλεία WinCC_OA . . . . .	2
1.1.3 JCOP Framework . . . . .	4
1.2 Εφαρμογή συστήματος ελέγχου RD51 Slow Control System στο PVSS . . . . .	5
1.2.1 SY2527 Multichannel Power Supply Mainframe System	5
1.2.2 Περιγραφή του συστήματος ελέγχου Slow Control System . . . . .	7
1.3 Μετάβαση του RD51 Slow Control System από PVSS σε HVLab στο WinCC_OA . . . . .	10
1.3.1 Εγκατάσταση WinCC_OA σε Windows 8.1 . . . . .	10
1.3.2 Εγκατάσταση διακομιστή της CAEN σε Windows 8.1 . . . . .	18
1.3.3 Αναβάθμιση του Slow Control System σε HVLab στο WinCC_OA . . . . .	27
1.3.4 Ανάπτυξη του HVLab στο WinCC_OA . . . . .	43
<b>2 ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΙΚΗ ΠΛΑΤΦΟΡΜΑ ARDUINO</b>	<b>47</b>
2.1 Περιβάλλον εργασίας Arduino . . . . .	49
2.1.1 Δημιουργία "σκέτςων" . . . . .	49
2.1.2 Sketchbook . . . . .	50

2.1.3	Tabs, Multiple Files, and Compilation . . . . .	50
2.1.4	Φόρτωση κώδικα στο Arduino (Uploading) . . . . .	50
2.1.5	Βιβλιοθήκες (Libraries) . . . . .	51
2.1.6	Σειριακή οθόνη (Serial Monitor) . . . . .	51
2.2	Πλακέτα Arduino Mega 2560 . . . . .	52
2.2.1	Παροχή Ενέργειας (Power) . . . . .	52
2.2.2	Μνήμη (Memory) . . . . .	53
2.2.3	Είσοδος και Έξοδος (Input and Output) . . . . .	53
2.2.4	ATmega2560-Arduino Pin Mapping . . . . .	55
2.3	Πλακέτα Arduino Ethernet Shield . . . . .	56
2.3.1	Περιγραφή Ethernet Shield . . . . .	57
2.4	OPC Server για Arduino . . . . .	59
2.4.1	Λογισμικό διασύνδεσης OPC . . . . .	59
2.4.2	Διακομιστής OPC για Arduino . . . . .	60
2.4.3	Χαρακτηριστικά . . . . .	60
<b>3</b>	<b>ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΜΕΤΡΗΣΗ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΑΙ ΥΓΡΑΣΙΑΣ</b>	<b>63</b>
3.1	Αισθητήρες θερμοκρασίας και υγρασίας, DHT22 - DHT11	63
3.1.1	Αισθητήρες DHT22 . . . . .	63
3.1.2	Αισθητήρες DHT11 . . . . .	65
3.2	Σύνδεση πλακέτας Arduino με αισθητήρες . . . . .	66
3.2.1	Σύνδεση Arduino με αισθητήρες DHT22 . . . . .	66
3.2.2	Σύνδεση Arduino με αισθητήρες DHT11 . . . . .	75
3.3	Επικοινωνία αισθητήρων DHT22 & Arduino με WinCC_OA μέσω OPC Server . . . . .	77
3.3.1	Εγκατάσταση OPC Server για Arduino . . . . .	77
3.3.2	Παραμετροποίηση OPC Server για Arduino . . . . .	78
3.4	Σύνδεση Arduino με OPC Server και Iconics OPC DataSpy	80
3.5	Σύνδεση Arduino με OPC Server & WinCC_OA . . . . .	84
3.5.1	Δημιουργία OPC πελάτη (OPC client) στο WinCC .	84
3.5.2	Σύνδεση OPC DA Client με OPC Server για Arduino	86
3.5.3	Δημιουργία datapoints . . . . .	88
3.6	Παράθυρο περιβαλλοντικών παραμέτρων στο project HVLab	91
	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	<b>95</b>

# Κατάλογος σχημάτων

1.1	SY2527 Multichannel Power Supply Mainframe . . . . .	5
1.2	Τμήματα του μηχανήματος SY2527 της CAEN . . . . .	6
1.3	Κύριο Παράθυρο Slow Control System . . . . .	7
1.4	Παράθυρο Ομαδοποίησης Slow Control System . . . . .	8
1.5	Παράθυρο Ρυθμίσεων Slow Control System . . . . .	8
1.6	Export Παράθυρο Slow Control System . . . . .	9
1.7	Έναρξη εγκατάστασης WinCC_OA μέσω της επιλογής Run as Administrator . . . . .	12
1.8	Επιβεβαίωση εγκατάστασης . . . . .	12
1.9	Προετοιμασία εγκατάστασης . . . . .	12
1.10	Παράθυρο ενημέρωσης για προετοιμασία εγκατάστασης WinCC_OA . . . . .	13
1.11	Παράθυρο εκκίνησης εγκατάστασης WinCC_OA . . . . .	13
1.12	Επιλογή αποδοχής όρων χρήσης WinCC_OA . . . . .	14
1.13	Συμπλήρωση στοιχείων χρήστη του WinCC_OA . . . . .	14
1.14	Επιλογή χαρακτηριστικών του WinCC_OA προς εγκατά- σταση . . . . .	15
1.15	Εγκατάσταση WinCC_OA . . . . .	15
1.16	Ενημέρωση κατάστασης εγκατάστασης WinCC_OA . . . . .	16
1.17	Ολοκλήρωση εγκατάστασης WinCC_OA . . . . .	16
1.18	Project HVLab στην κονσόλα του WinCC_OA . . . . .	17
1.19	Έναρξη εγκατάστασης CAENHVOPCServer 5.45 μέσω της επιλογής Run as Administrator . . . . .	18
1.20	Παράθυρο εκκίνησης εγκατάστασης CAENHVOPCServer 5.45 . . . . .	19
1.21	Επιλογή θέσης εγκατάστασης CAENHVOPCServer 5.45 . . . . .	19
1.22	Εγκατάσταση CAENHVOPCServer 5.45 . . . . .	20
1.23	Ολοκλήρωση εγκατάστασης CAENHVOPCServer 5.45 . . . . .	20
1.24	Έναρξη εγκατάστασης CAENHVOPCServerConfigurator 5.32 μέσω της επιλογής Run as Administrator . . . . .	21

1.25 Παράθυρο εκκίνησης εγκατάστασης CAENHVOPCServer Configurator 5.32 . . . . .	21
1.26 Επιλογή θέσης εγκατάστασης CAENHVOPCServerConfigurator 5.32 . . . . .	22
1.27 Εγκατάσταση CAENHVOPCServerConfigurator 5.32 . . . . .	22
1.28 Ολοκλήρωση εγκατάστασης CAENHVOPCServerConfigurator 5.32 . . . . .	23
1.29 Ένδειξη ανεπιτυχούς σύνδεσης στο παράθυρο ενημέρωσης του CAEN Server Configurator . . . . .	24
1.30 Ρυθμίσεις Δικτύου . . . . .	25
1.31 Ένδειξη επιτυχούς σύνδεσης στο παράθυρο ενημέρωσης του CAEN Server Configurator . . . . .	26
1.32 Αποσυμπίεση αρχείου framework5.0.1 στη σωστή θέση . . . . .	28
1.33 Αποσυμπίεση αρχείου fwInstallation- 7.1.2 στη σωστή θέση . . . . .	29
1.34 Επιλογή Vision στο παράθυρο Gedi . . . . .	29
1.35 Επιλογή του fwInstallation.pnl . . . . .	30
1.36 Δημιουργία φακέλου για την τοποθέτηση των Components του HVLab . . . . .	30
1.37 Παράθυρο επιλογής σύνδεσης με το FW System Configuration DB . . . . .	31
1.38 Επιλογή σωστού φακέλου στον οποίο θα αναζητηθούν τα απαραίτητα Components . . . . .	31
1.39 Παράθυρο Vision στο αριστερό μέρος του οποίου εμφανίζονται τα διαθέσιμα Components . . . . .	32
1.40 Επιλογή Components προς εγκατάσταση . . . . .	32
1.41 Εγκατάσταση επιλεγμένων Components . . . . .	33
1.42 Παράθυρο ενεργοποίησης ή μη ενός Driver . . . . .	33
1.43 Παράθυρο ειδοποίησης για την εγκατάσταση του OPC Server της CAEN . . . . .	34
1.44 Παράθυρο ενημέρωσης για την επανεκκίνηση του project . . . . .	34
1.45 Παρουσία JCOP Framework στην γραμμή εργαλείων του Gedi . . . . .	35
1.46 Προσθήκη συσκευής στο παράθυρο Device Editor and Navigator (DEN) . . . . .	35
1.47 Δήλωση χαρακτηριστικών της συσκευής . . . . .	36
1.48 Παράθυρο ενημέρωσης για την επιτυχή δήλωση της συσκευής . . . . .	36
1.49 Προσθήκη κάρτας στο παράθυρο Device Editor and Navigator (DEN) . . . . .	37
1.50 Δήλωση χαρακτηριστικών της κάρτας . . . . .	37
1.51 Παράθυρο ενημέρωσης για την επιτυχή δήλωση της κάρτας . . . . .	38

1.52 Παράθυρο ενημέρωσης για την επιτυχή δήλωση των κα- ναλιών της κάρτας . . . . .	38
1.53 Παράθυρο εμφάνισης των datapoints ( para: Datapoint Parametrization) . . . . .	39
1.54 Παράθυρο Dp-Type Editor για τη δημιουργία datapoints .	41
1.55 Main Panel (Κύριο Παράθυρο) . . . . .	44
1.56 Configuration Panel (Παράθυρο Παραμετροποίησης) . . . .	44
1.57 Configuration Panel (Παράθυρο Παραμετροποίησης ολο- κληρωμένο) . . . . .	45
1.58 Selection Panel (Παράθυρο Επιλογής) . . . . .	45
1.59 Archiving All Panel . . . . .	46
1.60 Export Panel . . . . .	46
2.1 Πλακέτα Arduino Mega με Ethernet Shield . . . . .	47
2.2 Μπροστινή όψη Arduino Mega . . . . .	52
2.3 Πίσω όψη Arduino Mega . . . . .	52
2.4 Διάγραμμα κεφαλών Arduino Mega . . . . .	55
2.5 Μπροστινή Όψη του Ethernet Shield . . . . .	56
2.6 Πίσω Όψη του Ethernet Shield . . . . .	56
2.7 Διάγραμμα OPC Επικοινωνίας . . . . .	59
2.8 OPC Server για Arduino . . . . .	60
3.1 Αισθητήρας θερμοκρασίας και υγρασίας DHT22 . . . . .	63
3.2 Διαστάσεις αισθητήρα DHT22 . . . . .	64
3.3 Αισθητήρας θερμοκρασίας και υγρασίας DHT11 . . . . .	65
3.4 Διαστάσεις αισθητήρα DHT11 . . . . .	66
3.5 Κεφαλές αισθητήρα DHT22 . . . . .	66
3.6 Διάγραμμα θερμοκρασιών - χρόνου DHT22 με 1kohm αντί- σταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22_1 + resistor, με κόκκινο του DHT22_2 + resistor, με πράσινο του DHT22_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22_4 + resistor αισθητήρα. . . . .	68
3.7 Διάγραμμα υγρασιών - χρόνου DHT22 με 1kohm αντί- σταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22_1 + resistor, με κόκκινο του DHT22_2 + resistor, με πράσινο του DHT22_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22_4 + resistor αισθητήρα. . . . .	68

3.8	Διάγραμμα θερμοκρασιών - χρόνου DHT22 με 4.7kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22_1 + resistor, με κόκκινο του DHT22_2 + resistor, με πράσινο του DHT22_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22_4 + resistor. . . . .	69
3.9	Διάγραμμα υγρασιών - χρόνου DHT22 με 4.7kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22_1 + resistor, με κόκκινο του DHT22_2 + resistor, με πράσινο του DHT22_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22_4 + resistor. . . . .	69
3.10	Διάγραμμα θερμοκρασιών - χρόνου DHT22 με 10kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22_1 + resistor, με κόκκινο του DHT22_2 + resistor, με πράσινο του DHT22_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22_4 + resistor. . . . .	70
3.11	Διάγραμμα υγρασιών - χρόνου DHT22 με 10kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22_1 + resistor, με κόκκινο του DHT22_2 + resistor, με πράσινο του DHT22_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22_4 + resistor. . . . .	70
3.12	Διάγραμμα θερμοκρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις του αισθητήρα DHT22_1 που συνδέεται με το board,(πυκνωτή 100nF και αντίσταση 10Kohm) αναπαριστώνται με μαύρο χρώμα, ενώ με κόκκινο, πράσινο και κίτρινο παρουσιάζονται οι μετρήσεις των αισθητήρων DHT22_2, DHT22_3, DHT22_4 αντίστοιχα, που συνδέονται με έναν πυκνωτή 100nF. . . . .	71
3.13	Διάγραμμα υγρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις του αισθητήρα DHT22_1 που συνδέεται με το board,(πυκνωτή 100nF και αντίσταση 10Kohm) αναπαριστώνται με μαύρο χρώμα, ενώ με κόκκινο, πράσινο και κίτρινο παρουσιάζονται οι μετρήσεις των αισθητήρων DHT22_2, DHT22_3, DHT22_4 αντίστοιχα, που συνδέονται με έναν πυκνωτή 100nF. . . . .	72
3.14	Διάγραμμα θερμοκρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις των αισθητήρων που συνδέονται με πυκνωτή 100nF και αντίσταση 10Kohm αναπαριστώνται με μαύρο και κόκκινο χρώμα και είναι οι DHT22_1 και DHT22_2 αντίστοιχα, ενώ με πράσινο και κίτρινο είναι οι μετρήσεις των αισθητήρων που συνδέονται με έναν πυκνωτή 100nF, DHT22_3 και DHT22_4. . . . .	73



3.15 Διάγραμμα υγρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις των αισθητήρων που συνδέονται με πυκνωτή 100nF και αντίσταση 10Kohm αναπαριστώνται με μαύρο και κόκκινο χρώμα και είναι οι DHT22_1 και DHT22_2 αντίστοιχα, ενώ με πράσινο και κίτρινο είναι οι μετρήσεις των αισθητήρων που συνδέονται με έναν πυκνωτή 100nF, DHT22_3 και DHT22_4. . . . .	74
3.16 Αισθητήρας DHT11 & καλώδιο σύνδεσης . . . . .	75
3.17 Διάγραμμα θερμοκρασιών - χρόνου DHT11. Με μαύρο, κόκκινο, πράσινο και κίτρινο χρώμα σημειώνονται οι τιμές των αισθητήρων DHT11_1, DHT11_2, DHT11_3 και DHT11_4 αντίστοιχα. . . . .	76
3.18 Διάγραμμα υγρασιών - χρόνου DHT11. Με μαύρο, κόκκινο, πράσινο και κίτρινο χρώμα σημειώνονται οι τιμές των αισθητήρων DHT11_1, DHT11_2, DHT11_3 και DHT11_4 αντίστοιχα. . . . .	76
3.19 Πληροφορίες του παράθυρου OPC Server για Arduino . . . . .	77
3.20 Παραμετροποίηση του παράθυρου OPC Server για Arduino . . . . .	78
3.21 Ρυθμίσεις δικτύου για την επίτευξη σύνδεσης με OPC Server για Arduino . . . . .	80
3.22 Επιλογή και σύνδεση OPC Server για Arduino στο παράθυρο του DataSpy . . . . .	82
3.23 Πληροφορίες για αναμονή σύνδεσης στο παράθυρο του OPC Server για Arduino . . . . .	83
3.24 Εμφάνιση των items του OPC Server για Arduino στο παράθυρο του OPC DataSpy . . . . .	83
3.25 Πληροφορίες για την μεταφορά των items στο παράθυρο του OPC Server για Arduino . . . . .	84
3.26 Παράθυρο Manager Properties που χρησιμοποιείται για την εισαγωγή του OPC DA Client . . . . .	85
3.27 Παράθυρο Server Selection που χρησιμοποιείται για την επιλογή του OPC Server για Arduino . . . . .	86
3.28 Παράθυρο OPC Configurator που χρησιμοποιείται για την δημιουργία καινούριου group δεδομένων . . . . .	87
3.29 Η επιλεγμένη ένδειξη "Connected" στο παράθυρο OPC Configurator καταδεικνύει την επιτυχή σύνδεση του OPC client με τον OPC Server για Arduino . . . . .	88
3.30 Παράθυρο Περιβαλλοντικών Παραμέτρων . . . . .	92
3.31 Διάγραμμα Θερμοκρασίας συναρτήσει του χρόνου του πρώτου αισθητήρα DHT22 . . . . .	92



# Κεφάλαιο 1

## ΑΝΑΒΑΘΜΙΣΗ ΤΟΥ RD51 Slow Control System

### 1.1 WinCC\_OA

#### 1.1.1 Εισαγωγή

Ο όρος SCADA (Supervisory Control and Data Acquisition) περιγράφει μια κατηγορία συστημάτων βιομηχανικού αυτομάτου ελέγχου και τηλεμετρίας. Το χαρακτηριστικό των συστημάτων SCADA είναι ότι αποτελούνται από τοπικούς ελεγκτές, που ελέγχουν επί μέρους στοιχεία και μονάδες μιας εγκατάστασης, συνδεδεμένους σε ένα κεντρικό Master Station (Κύριο Σταθμό Εργασίας). Ο κεντρικός σταθμός εργασίας μπορεί κατόπιν να επικοινωνεί τα δεδομένα που συλλέγει από την εγκατάσταση σε ένα πλήθος από σταθμούς εργασίας σε τοπικό LAN ή και να μεταδίδει τα δεδομένα της εγκατάστασης σε μακρινά σημεία μέσω κάποιου συστήματος τηλεπικοινωνίας, πχ μέσω του ενσύρματου τηλεφωνικού δικτύου ή μέσω κάποιου ασύρματου δικτύου.

Στη Φυσική Ύψηλών Ενεργειών, είναι απαραίτητη η χρήση συστημάτων SCADA. Η ανάγκη για απομακρυσμένο έλεγχο των ανιχνευτών οδήγησε στην χρήση του WinCC\_OA, ενός εργαλείου που χρησιμοποιείται για την κατασκευή συστημάτων ελέγχου. Το WinCC\_OA προσφέρει μέγιστη λειτουργικότητα και διαθέτει ένα περιβάλλον εργασίας πολύ φιλικό στους χρήστες. Αποτελεί ένα παραμετροποιήσιμο και επεκτάσιμο σύστημα, που παρέχει στους χρήστες τη δυνατότητα να συνδέσουν συσκευές με λογισμικό προκειμένου να κάνουν λήψη των δεδομένων που προέρχονται από τις συσκευές και να μπορούν να τα επεξεργαστούν περαιτέρω.

### 1.1.2 Βασικά εργαλεία WinCC\_OA

Το WinCC\_OA προκειμένου να δώσει στους χρήστες τη δυνατότητα να ελέγξουν και να παραμετροποιήσουν συσκευές και λογισμικό, τους παρέχει μια πληθώρα εργαλείων, τα σημαντικότερα από τα οποία παραθέτονται παρακάτω.

- **Βάση δεδομένων (Data base)**

Το WinCC\_OA διαθέτει μια βάση δεδομένων στην οποία αποθηκεύονται σε πραγματικό χρόνο τα δεδομένα τα οποία προέρχονται από τις συσκευές. Τα δεδομένα αυτά ομαδοποιούνται σε DataPoints (DP). Τα DataPoints (DP) ανήκουν σε κάποιο DataPoint Type (DPT), το οποίο περιγράφει τη δομή της συνδεδεμένης συσκευής από την οποία προέρχεται η μέτρηση. Στο κάθε DataPoint ο χρήστης μπορεί να προσθέσει παραμέτρους τα επονομαζόμενα στοιχεία datapoint (datapoint elements) στα οποία αποθηκεύονται πληροφορίες και τιμές που προέρχονται από τη συσκευή. Όλα τα παραπάνω, DataPoint Type (DPT), DataPoints (DP), datapoint elements (dpes), δημιουργούνται και τροποποιούνται από τους χρήστες είτε στο γραφικό εργαλείο παραμετροποίησης (graphical parameterization tool (PARA)) είτε μέσω κώδικα στα scripts ελέγχου (control scripts).

- **Γραφικό περιβάλλον επεξεργασίας (Graphical EDItor (GEDI))**

Το γραφικό περιβάλλον επεξεργασίας (Graphical EDItor (GEDI)) επιτρέπει στους χρήστες να δημιουργούν τις δικές τους "διεπαφές χρήστη" (panels). Ακόμη παρέχει μια πληθώρα γραφικών στοιχείων όπως κουμπιά, πίνακες, πεδία κειμένου κ.τ.λ τα οποία μπορούν να τοποθετηθούν σε οποιοδήποτε σημείο των panels και να παραμετροποιηθούν μέσω συγκεκριμένων συναρτήσεων του κώδικα στα scripts ελέγχου.

- **Scripts ελέγχου (Control scripts)**

Οι χρήστες του WinCC\_OA έχουν τη δυνατότητα να δημιουργήσουν Scripts ελέγχου (Control scripts) τα οποία μπορούν να χρησιμοποιηθούν είτε μέσα σε ένα panel είτε αυτόνομα. Ο κώδικας των scripts ελέγχου είναι γραμμένος σε μια γλώσσα προγραμματισμού η οποία χρησιμοποιεί τους βασικούς κανόνες της γλώσσας C σε συνδυασμό με πρόσθετες εντολές και επεκτάσεις. Μια μεγάλη βιβλιοθήκη λειτουργιών είναι διαθέσιμη έτσι ώστε μέσω αυτής, οι χρήστες να χειρίζονται τα δεδομένα, να δημιουργούν γραφικές παραστάσεις και να έχουν πρόσβαση σε αρχεία.

- **Σύνδεση με συσκευές**

Μια από τις σημαντικότερες λειτουργίες του WinCC\_OA είναι ότι συνδέεται με συσκευές. Η σύνδεση και η επικοινωνία αυτή επιτυγχάνονται όταν οι τιμές των δεδομένων που προέρχονται από τις συσκευές αποθηκεύονται στη βάση δεδομένων του WinCC\_OA. Οι τρόποι επικοινωνίας ποικίλουν ανάλογα με τη συσκευή η οποία πρόκειται να συνδεθεί. Το WinCC\_OA παρέχει πολλαπλούς οδηγούς (drivers) μέσω των οποίων γίνεται η διασύνδεση με διαφορετικούς τύπους συσκευών ή με διαφορετικά πρωτόκολλα επικοινωνίας (οι πιο διαδεδομένοι είναι οι OPC και DIM). Επίσης στην περίπτωση που ο οδηγός (driver) που πρόκειται να χρησιμοποιηθεί δεν υποστηρίζεται από το WinCC\_OA απευθείας, λόγω της εκτεταμένης χρήσης του από την κοινότητα του CERN πολλοί κατασκευαστές συσκευών παρέχουν διακομιστές OPC (OPC servers) μαζί με τα προϊόντα τους. Ένας διακομιστής OPC αποτελεί ένα αφηρημένο επίπεδο μεταξύ της πραγματικής συσκευής και του χρήστη. Υλοποιεί κάθε απαιτητική διαμόρφωση, προκειμένου να αρχικοποιηθεί η επικοινωνία με τη συσκευή και παρέχει τα δεδομένα, τα οποία είναι διαθέσιμα από τη συσκευή, καθώς και τις εντολές τις οποίες είναι σε θέση να δεχτεί η συσκευή με τρόπο εύκολα προσβάσιμο.

### 1.1.3 JCOP Framework

Στο WinCC\_OA γίνεται χρήση του JCOP Framework (Joint COntrols Project). Το JCOP Framework δημιουργήθηκε από κοινού από τις ομάδες που συμμετέχουν στα τέσσερα πειράματα LHC με τη συνεισφορά των πειραματικών προγραμμάτων και της ομάδας μηχανικών και φυσικών στο CERN. Το JCOP Framework παρέχει στους χρήστες του τις κατευθυντήριες γραμμές και τα εξαρτήματα, τα οποία μπορεί να είναι συσκευές ή εργαλεία, προκειμένου να είναι σε θέση να δημιουργήσουν εφαρμογές ελέγχου με συνεκτικό τρόπο.

Το JCOP Framework παρέχει πλήρη εξαρτήματα (components), τα οποία χρησιμοποιούνται σαν εξοπλισμός, όπως scripts ελέγχου (Control scripts), scripts βιβλιοθηκών (script libraries) και όλα τα panels που είναι απαραίτητα προκειμένου να γίνει παραμετροποίηση και χειρισμός των συσκευών. Επιπλέον το JCOP Framework παρέχει στους χρήστες εργαλεία προκειμένου να ενσωματώσουν τις δικές τους συσκευές καθώς και μερικά reference panels και scripts βιβλιοθηκών που μπορούν να χρησιμοποιηθούν προκειμένου να οικοδομηθεί μια νέα συσκευή. Τέλος υπάρχουν διάφορα panels και scripts που παρέχουν λειτουργίες για άλλες εφαρμογές ή χρησιμοποιούνται ως πρότυπα για τη δημιουργία νέων panels και scripts.

Κατά τη διάρκεια των Test Beams του RD51 υπήρχε ανάγκη για απομακρυσμένο έλεγχο των υψηλών τάσεων των ανιχνευτών. Κατά τη διάρκεια των τελευταίων πειραμάτων όλο και περισσότεροι συμμετέχοντες δοκίμασαν τους ανιχνευτές τους με τη βοήθεια της υποδομής του RD51. Αυτό δημιούργησε την ανάγκη για ένα πλήρες σύστημα το οποίο θα ελέγχει και παρακολουθεί την υψηλή τάση των υπό δοκιμή ανιχνευτών, αλλά και των ανιχνευτών που χρησιμοποιούνται για τα τηλεσκόπια δέσμης. Για το Micromegas τηλεσκόπιο μόνο, χρειάζονται 9 κανάλια υψηλής τάσης.

## 1.2 Εφαρμογή συστήματος ελέγχου RD51 Slow Control System στο PVSS

### 1.2.1 SY2527 Multichannel Power Supply Mainframe System

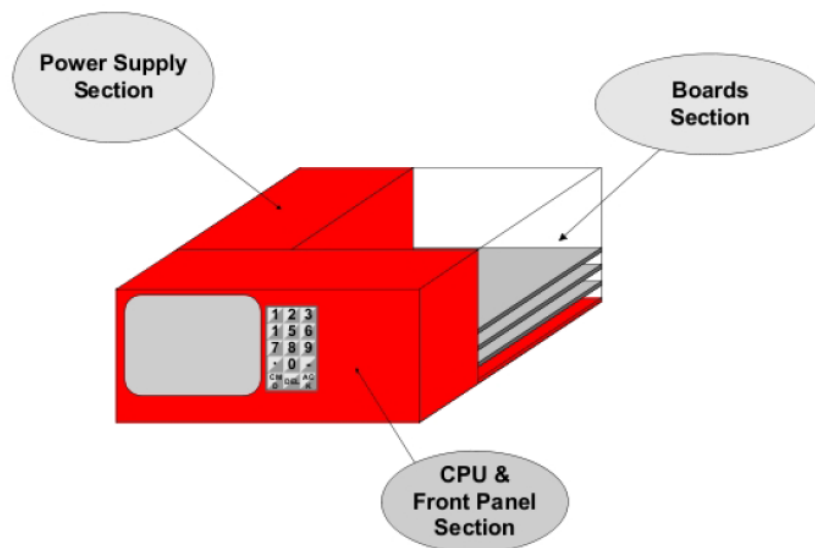


Σχήμα 1.1: SY2527 Multichannel Power Supply Mainframe

Προκειμένου να εξυπηρετηθούν οι παραπάνω ανάγκες, το RD51 αποφάσισε να κάνει χρήση ενός CAEN mainframe. Καταλληλότερο κρίθηκε το SY-2527. Το μηχάνημα υψηλής τάσης SY2527 της CAEN αποτελεί ένα πολυκαναλικό σύστημα τροφοδοσίας. Αυτό το σύστημα δίνει στους χρήστες τη δυνατότητα να "φιλοξενήσουν" σε ένα mainframe ένα ευρύ φάσμα από boards με διαφορετικές λειτουργίες, όπως για παράδειγμα boards χαμηλής και υψηλής τάσης, boards που χρησιμοποιούνται για την παρακολούθηση της υγρασίας ή της θερμοκρασίας και κλαδικούς ελεγκτές (branch controllers). Οι τελευταίοι χρησιμοποιούνται για τον απομακρυσμένο έλεγχο άλλων συσκευών παραγωγής ενέργειας και διανομής ρεύματος.

Το mainframe SY2527 της CAEN αποτελείται από τρία βασικά τμήματα:

- Το τμήμα των boards (Boards Section) το οποίο αποτελείται από 6 υποδοχές που μπορούν να "φιλοξενήσουν" boards, διανομείς (distributors) και κλαδικούς ελεγκτές (branch controllers)
- Το τμήμα παρόχής τάσης του mainframe
- Το τμήμα CPU and Front Panel Section το οποίο περιλαμβάνει όλες τις παροχές διασύνδεσης.



Σχήμα 1.2: Τμήματα του μηχανήματος SY2527 της CAEN

Ένα απλό crate μπορεί να "φιλοξενήσει" μέχρι 6 boards καναλιών, τα οποία μπορούν να επιλεγούν από ένα μεγάλο εύρος plug-in boards. Όλοι οι τύποι των boards μπορούν να τοποθετηθούν στο ίδιο crate προκειμένου να εξυπηρετήσουν τις ανάγκες των χρηστών. Οι κάρτες που χρησιμοποιήθηκαν για τις ανάγκες του RD51 Slow Control System και τοποθετήθηκαν στο SY2527 mainframe είναι οι A1821n και A1526n.

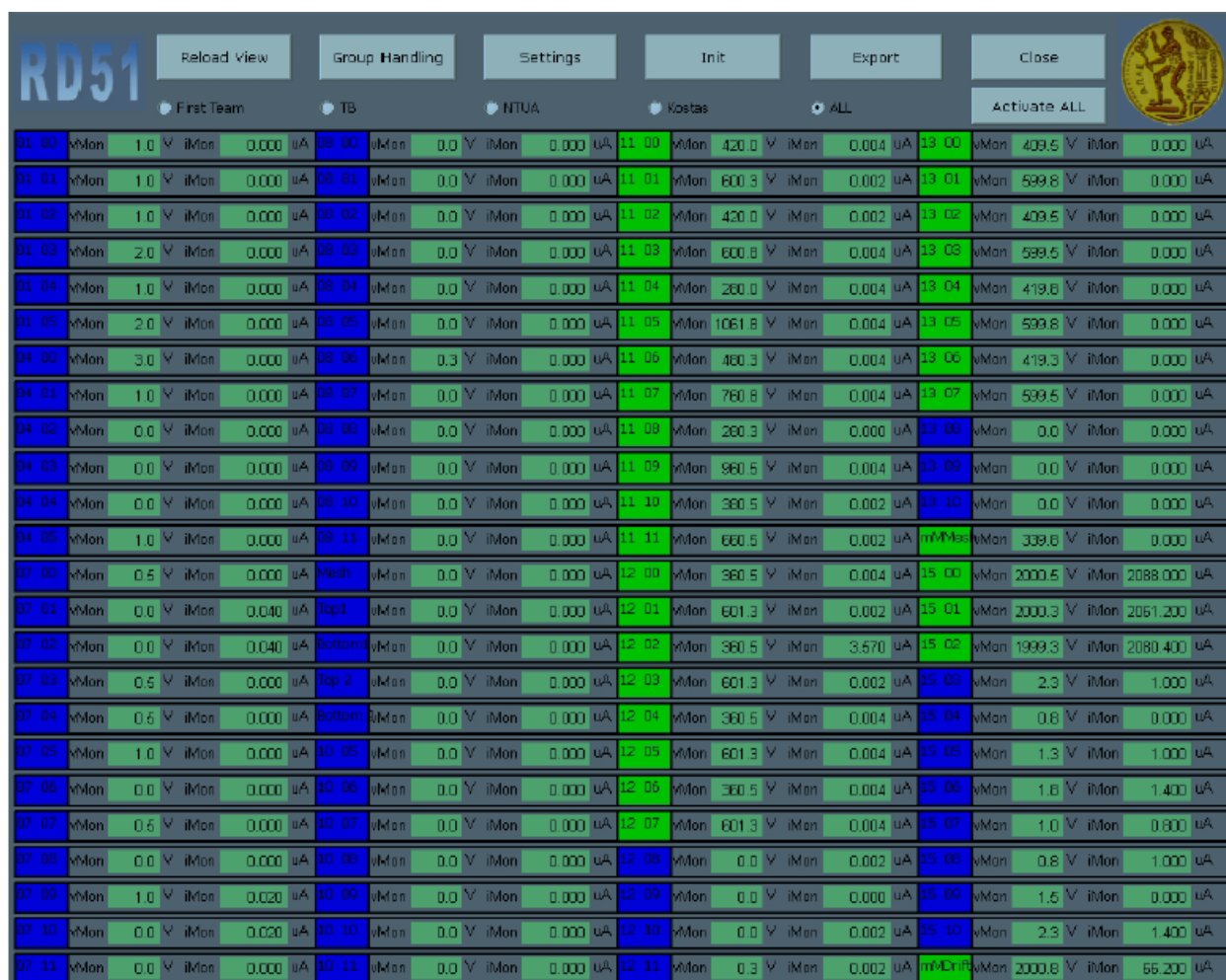
Τέλος για να ελέγχονται σωστά οι υψηλές τάσεις των ανιχνευτών βασικός στόχος είναι η συσκευή παροχής υψηλής τάσης της CAEN να βρίσκεται όσο πιο κοντά γίνεται στον ανιχνευτή.



## 1.2.2 Περιγραφή του συστήματος ελέγχου Slow Control System

Για τον έλεγχο και την χρήση του SY2527 mainframe έχει αναπτυχθεί μια εφαρμογή που βασίζεται στο PVSS. Το PVSS είναι η προηγούμενη έκδοση του WinCC\_OA και παρέχει σχεδόν τις ίδιες λειτουργίες. Το Slow Control System [1] αποτελείται από τα παρακάτω παράθυρα.

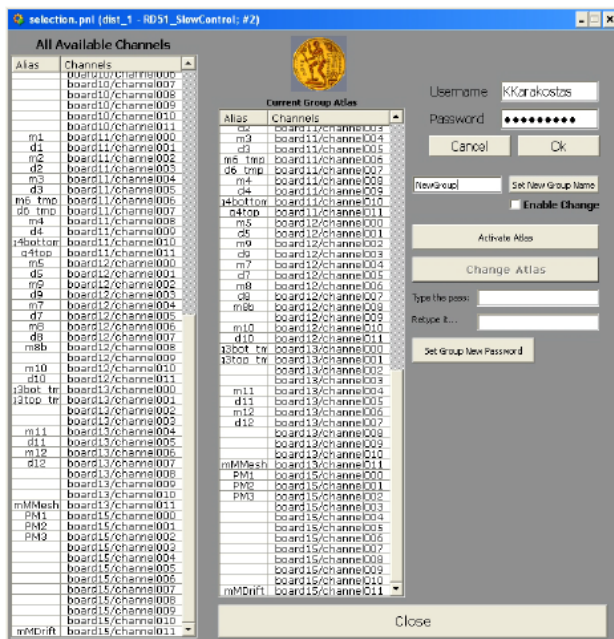
### Κύριο Παράθυρο



01_00	vMon	1.0	V	iMon	0.000	uA	02_00	vMon	0.0	V	iMon	0.000	uA	03_00	vMon	420.0	V	iMon	0.004	uA	04_00	vMon	409.5	V	iMon	0.000	uA
01_01	vMon	1.0	V	iMon	0.000	uA	02_01	vMon	0.0	V	iMon	0.000	uA	03_01	vMon	600.3	V	iMon	0.002	uA	04_01	vMon	599.8	V	iMon	0.000	uA
01_02	vMon	1.0	V	iMon	0.000	uA	02_02	vMon	0.0	V	iMon	0.000	uA	03_02	vMon	420.0	V	iMon	0.002	uA	04_02	vMon	409.5	V	iMon	0.000	uA
01_03	vMon	2.0	V	iMon	0.000	uA	02_03	vMon	0.0	V	iMon	0.000	uA	03_03	vMon	600.6	V	iMon	0.004	uA	04_03	vMon	599.8	V	iMon	0.000	uA
01_04	vMon	1.0	V	iMon	0.000	uA	02_04	vMon	0.0	V	iMon	0.000	uA	03_04	vMon	260.0	V	iMon	0.004	uA	04_04	vMon	419.6	V	iMon	0.000	uA
01_05	vMon	2.0	V	iMon	0.000	uA	02_05	vMon	0.0	V	iMon	0.000	uA	03_05	vMon	1061.8	V	iMon	0.004	uA	04_05	vMon	599.8	V	iMon	0.000	uA
01_06	vMon	3.0	V	iMon	0.000	uA	02_06	vMon	0.3	V	iMon	0.000	uA	03_06	vMon	400.3	V	iMon	0.004	uA	04_06	vMon	419.3	V	iMon	0.000	uA
01_07	vMon	1.0	V	iMon	0.000	uA	02_07	vMon	0.0	V	iMon	0.000	uA	03_07	vMon	760.8	V	iMon	0.004	uA	04_07	vMon	599.5	V	iMon	0.000	uA
01_08	vMon	0.0	V	iMon	0.000	uA	02_08	vMon	0.0	V	iMon	0.000	uA	03_08	vMon	260.3	V	iMon	0.000	uA	04_08	vMon	0.0	V	iMon	0.000	uA
01_09	vMon	0.0	V	iMon	0.000	uA	02_09	vMon	0.0	V	iMon	0.000	uA	03_09	vMon	960.5	V	iMon	0.004	uA	04_09	vMon	0.0	V	iMon	0.000	uA
01_10	vMon	0.0	V	iMon	0.000	uA	02_10	vMon	0.0	V	iMon	0.000	uA	03_10	vMon	360.5	V	iMon	0.002	uA	04_10	vMon	0.0	V	iMon	0.000	uA
01_11	vMon	1.0	V	iMon	0.000	uA	02_11	vMon	0.0	V	iMon	0.000	uA	03_11	vMon	660.5	V	iMon	0.002	uA	04_11	vMon	339.6	V	iMon	0.000	uA
02_00	vMon	0.5	V	iMon	0.000	uA	03_00	vMon	0.0	V	iMon	0.000	uA	04_00	vMon	960.5	V	iMon	0.004	uA	05_00	vMon	2000.5	V	iMon	2088.000	uA
02_01	vMon	0.0	V	iMon	0.040	uA	03_01	vMon	0.0	V	iMon	0.000	uA	04_01	vMon	601.3	V	iMon	0.002	uA	05_01	vMon	2000.3	V	iMon	2061.200	uA
02_02	vMon	0.0	V	iMon	0.040	uA	03_02	vMon	0.0	V	iMon	0.000	uA	04_02	vMon	360.5	V	iMon	3.570	uA	05_02	vMon	1999.3	V	iMon	2080.400	uA
02_03	vMon	0.5	V	iMon	0.000	uA	03_03	vMon	0.0	V	iMon	0.000	uA	04_03	vMon	601.3	V	iMon	0.002	uA	05_03	vMon	2.3	V	iMon	1.000	uA
02_04	vMon	0.5	V	iMon	0.000	uA	03_04	vMon	0.0	V	iMon	0.000	uA	04_04	vMon	360.5	V	iMon	0.004	uA	05_04	vMon	0.6	V	iMon	0.000	uA
02_05	vMon	1.0	V	iMon	0.000	uA	03_05	vMon	0.0	V	iMon	0.000	uA	04_05	vMon	601.3	V	iMon	0.004	uA	05_05	vMon	1.3	V	iMon	1.000	uA
02_06	vMon	0.0	V	iMon	0.000	uA	03_06	vMon	0.0	V	iMon	0.000	uA	04_06	vMon	360.5	V	iMon	0.004	uA	05_06	vMon	1.6	V	iMon	1.400	uA
02_07	vMon	0.5	V	iMon	0.000	uA	03_07	vMon	0.0	V	iMon	0.000	uA	04_07	vMon	601.3	V	iMon	0.004	uA	05_07	vMon	1.0	V	iMon	0.800	uA
02_08	vMon	0.0	V	iMon	0.000	uA	03_08	vMon	0.0	V	iMon	0.000	uA	04_08	vMon	0.0	V	iMon	0.002	uA	05_08	vMon	0.8	V	iMon	1.000	uA
02_09	vMon	1.0	V	iMon	0.020	uA	03_09	vMon	0.0	V	iMon	0.000	uA	04_09	vMon	0.0	V	iMon	0.000	uA	05_09	vMon	1.5	V	iMon	0.000	uA
02_10	vMon	0.0	V	iMon	0.020	uA	03_10	vMon	0.0	V	iMon	0.000	uA	04_10	vMon	0.0	V	iMon	0.002	uA	05_10	vMon	2.3	V	iMon	1.400	uA
02_11	vMon	0.0	V	iMon	0.000	uA	03_11	vMon	0.0	V	iMon	0.000	uA	04_11	vMon	0.3	V	iMon	0.002	uA	05_11	vMon	2000.8	V	iMon	65.200	uA

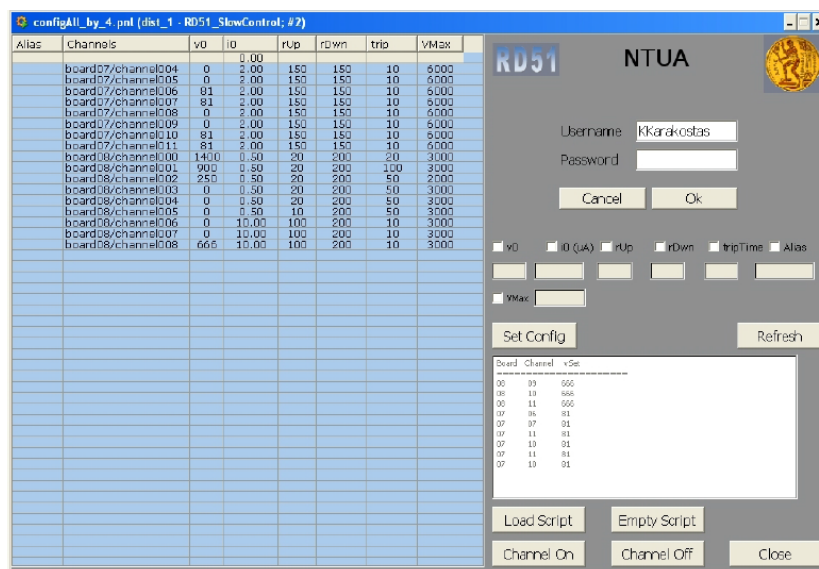
Σχήμα 1.3: Κύριο Παράθυρο Slow Control System

## Παράθυρο Ομαδοποίησης



Σχήμα 1.4: Παράθυρο Ομαδοποίησης Slow Control System

## Παράθυρο Ρυθμίσεων



Σχήμα 1.5: Παράθυρο Ρυθμίσεων Slow Control System

## Export Παράθυρο

The screenshot shows a software window titled "exportAll.pnl". On the left is a table with two columns: "Alias" and "Channels". The table contains the following data:

Alias	Channels
BNLdri	board03/channel000
BNLtop	board03/channel001
BNLbot	board03/channel002
BNLmes	board03/channel003
Mesh	board03/channel004
Drift	board03/channel005
LargeH	board03/channel006
LargeV	board03/channel007
Cathod	board03/channel008
Sac 6	board03/channel009
Sac 5	board03/channel010
	board03/channel011
	board04/channel000
	board04/channel001
	board04/channel002
	board04/channel003
test 1	board04/channel004
test 2	board04/channel005

On the right side of the window, there are two time selection sections:

**Select Start time.**

Day: 20, Month: 11, Year: 2009, Hour: 12, Minute: 4. An "Apply" button is next to these fields.

**Select End time.**

Day: 20, Month: 11, Year: 2009, Hour: 12, Minute: 4. An "Apply" button is next to these fields.

Below the time sections is a "Folder Name" label and a text input field containing "folder\_name".

At the bottom right, there are four buttons: "Export iMon vMon vSet", "Export iMon", "Export vMon", and "Export vSet". A "Close" button is located at the very bottom center.

Σχήμα 1.6: Export Παράθυρο Slow Control System

### 1.3 Μετάβαση του RD51 Slow Control System από PVSS σε HVLab στο WinCC\_OA

Μετά την κυκλοφορία της νέας βελτιωμένης έκδοσης του WinCC\_OA, το οποίο αντικατέστησε την προηγούμενη έκδοση, το PVSS, προέκυψε η ανάγκη αναβάθμισης όλων των συστημάτων ελέγχου που είχαν κατασκευαστεί και σχεδιαστεί στο PVSS, προκειμένου να είναι συμβατά με το νέο εργαλείο κατασκευής συστημάτων αυτομάτου ελέγχου το WinCC\_OA.

Για το σκοπό αυτό ακολουθήθηκε μια διαδικασία η οποία είχε ως αρχή την εγκατάσταση του WinCC\_OA και άλλων εργαλείων τα οποία χρησιμοποιήθηκαν προκειμένου να επιτευχθεί η επικοινωνία μεταξύ του WinCC\_OA και του μηχανήματος υψηλής τάσης της CAEN. Στη συνέχεια δημιουργήθηκε ένα καινούριο HVLab σύστημα στο οποίο μεταφέρθηκαν όλες οι λειτουργίες του προηγούμενου συστήματος ελέγχου Slow Control System και έγιναν ορισμένες αλλαγές που ήταν απαραίτητες προκειμένου να λειτουργήσει σωστά στο WinCC\_OA. Ουσιαστικά το σύστημα ελέγχου HVLab αποτελεί μια αναβαθμισμένη έκδοση του Slow Control System στο WinCC\_OA. Η διαδικασία αυτή ολοκληρώθηκε με την ανάπτυξη του HVLab μέσω της προσθήκης σε αυτό ορισμένων νέων λειτουργιών και χαρακτηριστικών.

#### 1.3.1 Εγκατάσταση WinCC\_OA σε Windows 8.1

Η σωστή εγκατάσταση του WinCC\_OA είναι απαραίτητη έτσι ώστε να διασφαλιστεί η μετέπειτα σωστή λειτουργία του προγράμματος. Πριν γίνει η εγκατάσταση, ο χρήστης πρέπει να ορίσει στον Πίνακα Ελέγχου των Windows ως γλώσσα εμφάνισης τα Αγγλικά προκειμένου το WinCC\_OA να εγκατασταθεί στα Αγγλικά, διαφορετικά η εγκατάσταση θα γίνει στα γερμανικά. Να σημειωθεί εδώ ότι δεν είναι απαραίτητο η γλώσσα του λειτουργικού συστήματος να είναι τα Αγγλικά, αρκεί να γίνει η επιλογή τους στον Πίνακα Ελέγχου.

Προκειμένου να γίνει σωστά η εγκατάσταση του WinCC\_OA ώστε να είναι αργότερα σε θέση να συνδεθεί μέσω του διακομιστή της CAEN με το μηχανήμα υψηλής τάσης είναι απαραίτητη η εγκατάσταση στον υπολογιστή ενός πακέτου προγραμμάτων. Αυτό το πακέτο διαθέτει προγράμματα (OPC Core Components) και οι δυο εκδόσεις των οποίων, και για x64 λειτουργικό σύστημα και για x86 λειτουργικό σύστημα, πρέπει να εγκατασταθούν στον υπολογιστή. Αυτό το πακέτο είναι άκρως απαραίτητο. Αν ο χρήστης παραβλέψει την συγκεκριμένη εγκατάσταση και

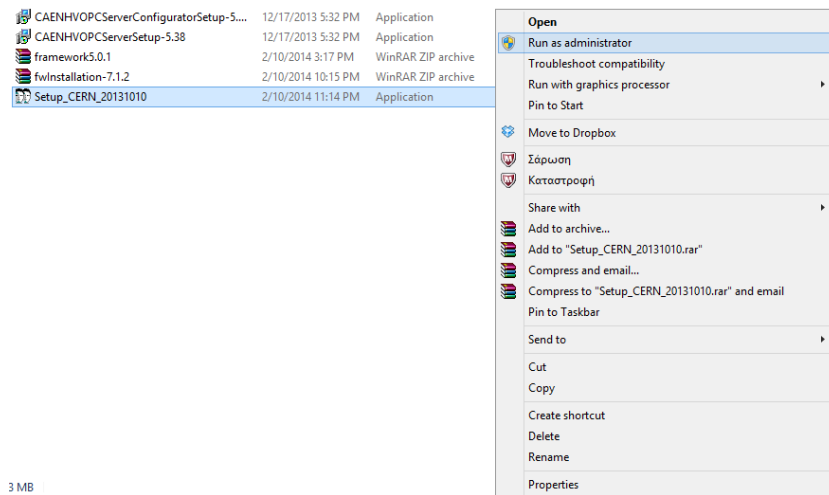
προχωρήσει παρακάτω, όταν θα έρθει η στιγμή να συνδέσει το project του WinCC\_OA με το μηχάνημα υψηλής τάσης μέσω του διακομιστή της CAEN, θα εμφανιστούν στο Log Viewer του WinCC τα παρακάτω σφάλματα:

- **IMPL, WARNING, 54, Unexpected state, OPCServer, CheckSrvInterfaces, Server does not support IOPCBrowseAddressSpace -> no browse info (Class not registered (0x80040154))**
- **IMPL, SEVERE, 54, Unexpected state, OPCServer, CheckSrvInterfaces, Server does not support IOPCServer ->server cannot be used (Class not registered (0x80040154))**

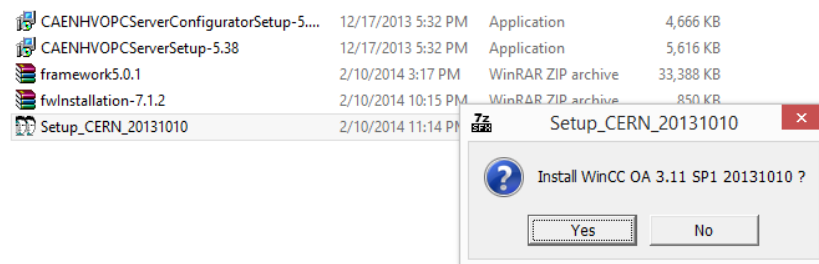
Προς αποφυγή αυτών των σφαλμάτων γίνεται εγκατάσταση του πακέτου των προγραμμάτων. Σε όλα τα προγράμματα η εγκατάσταση πρέπει να γίνει μέσω της επιλογής Run as Administrator έτσι ώστε να διασφαλιστεί η σωστή εγκατάσταση και μετέπειτα λειτουργία.

Μετά το πέρας της ολοκλήρωσης των παραπάνω ρυθμίσεων και εγκαταστάσεων ξεκινά η εγκατάσταση του WinCC OA. Η έκδοση του WinCC που χρησιμοποιήθηκε είναι η WinCC OA 3.11-SP1 και διατίθεται δωρεάν από το CERN. Είναι απαραίτητο η εγκατάσταση να γίνει μέσω της επιλογής Run as Administrator. Η επιλογή αυτή (Run as Administrator) πρέπει να χρησιμοποιείται πάντα για την εκκίνηση του WinCC\_OA.

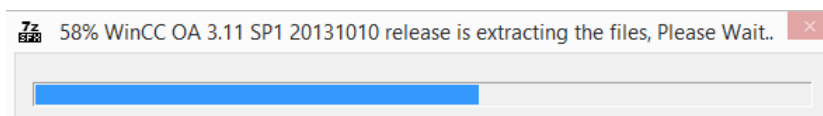
Στις παρακάτω εικόνες παρουσιάζονται τα στάδια της εγκατάστασης.



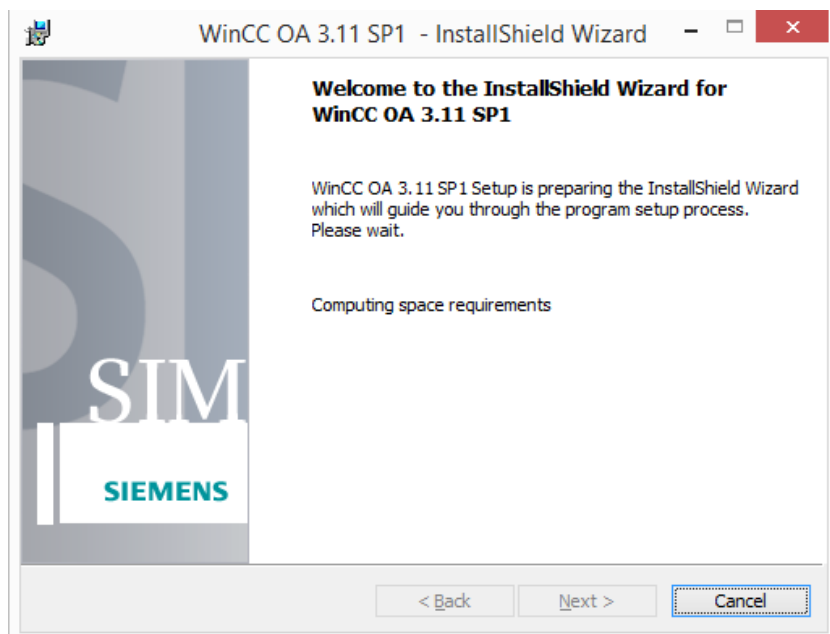
Σχήμα 1.7: Έναρξη εγκατάστασης WinCC\_OA μέσω της επιλογής Run as Administrator



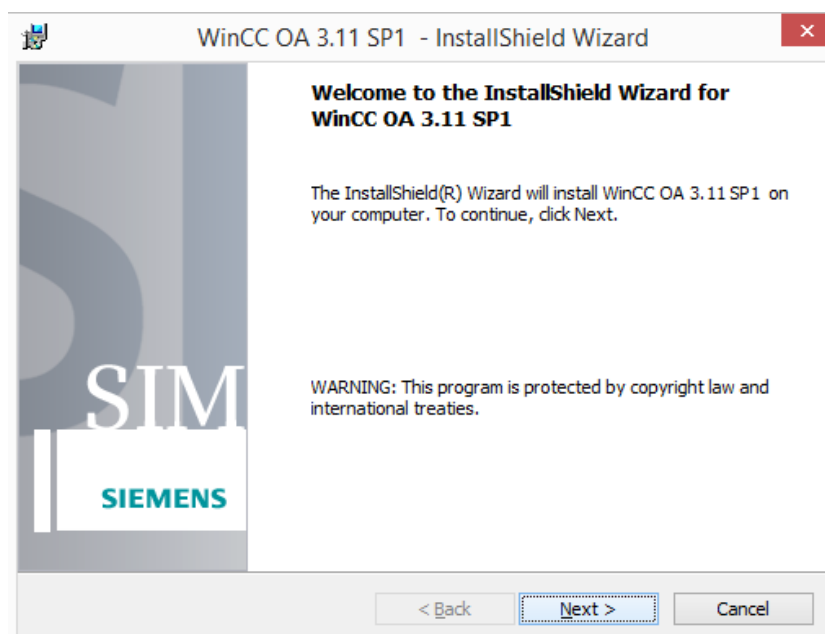
Σχήμα 1.8: Επιβεβαίωση εγκατάστασης



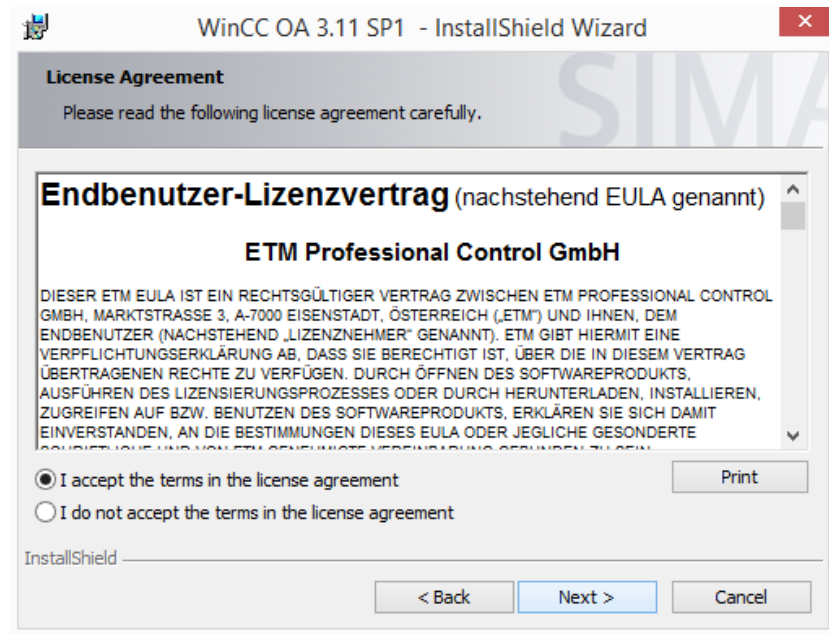
Σχήμα 1.9: Προετοιμασία εγκατάστασης



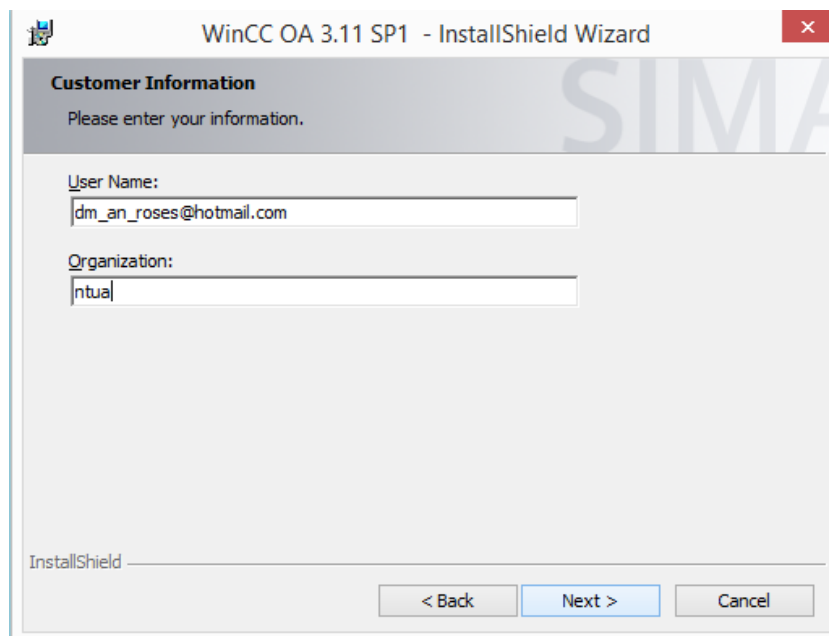
Σχήμα 1.10: Παράθυρο ενημέρωσης για προετοιμασία εγκατάστασης WinCC\_OA



Σχήμα 1.11: Παράθυρο εκκίνησης εγκατάστασης WinCC\_OA

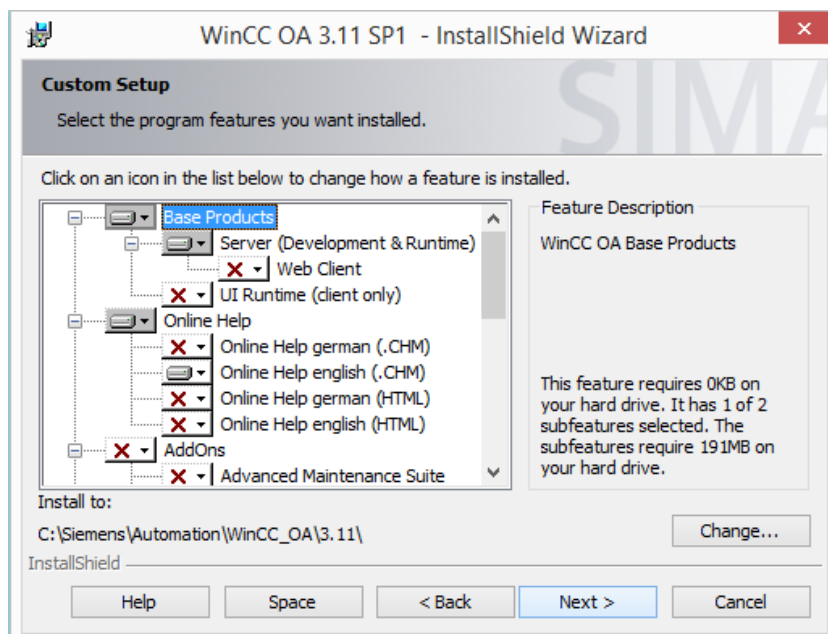


Σχήμα 1.12: Επιλογή αποδοχής όρων χρήσης WinCC\_OA

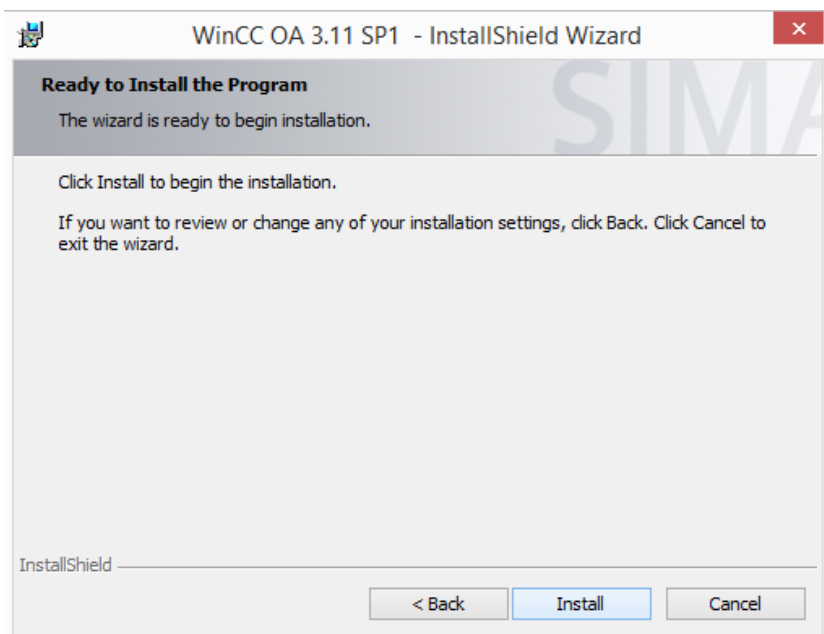


Σχήμα 1.13: Συμπλήρωση στοιχείων χρήστη του WinCC\_OA

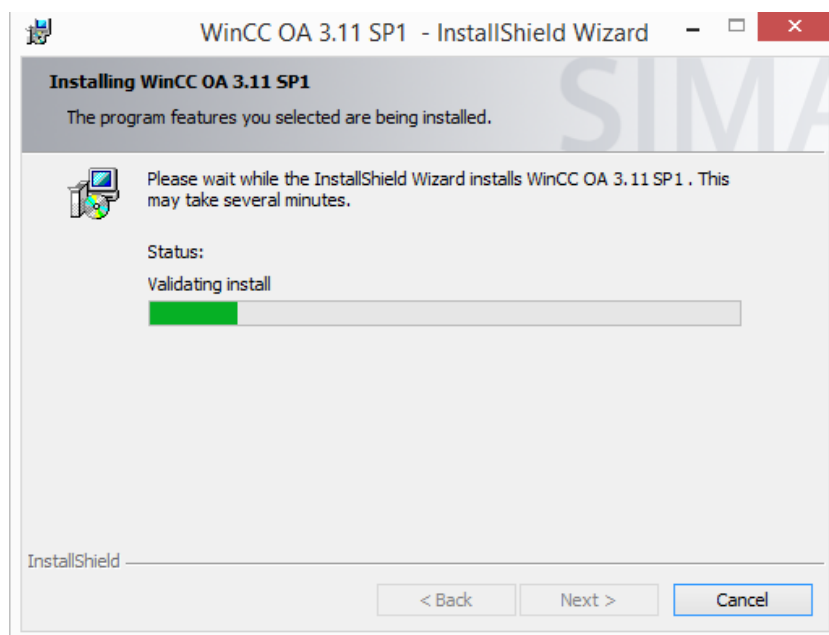




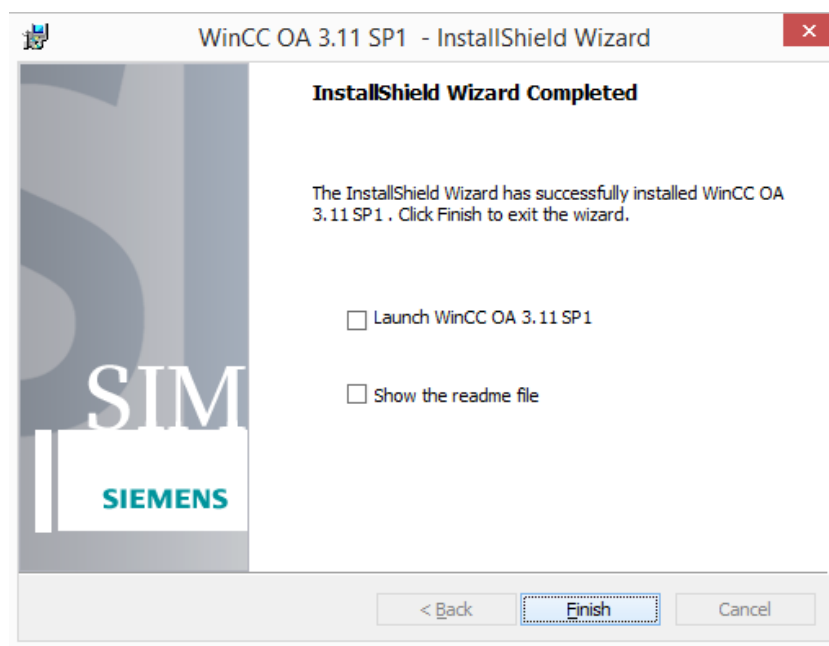
Σχήμα 1.14: Επιλογή χαρακτηριστικών του WinCC\_OA προς εγκατάσταση



Σχήμα 1.15: Εγκατάσταση WinCC\_OA

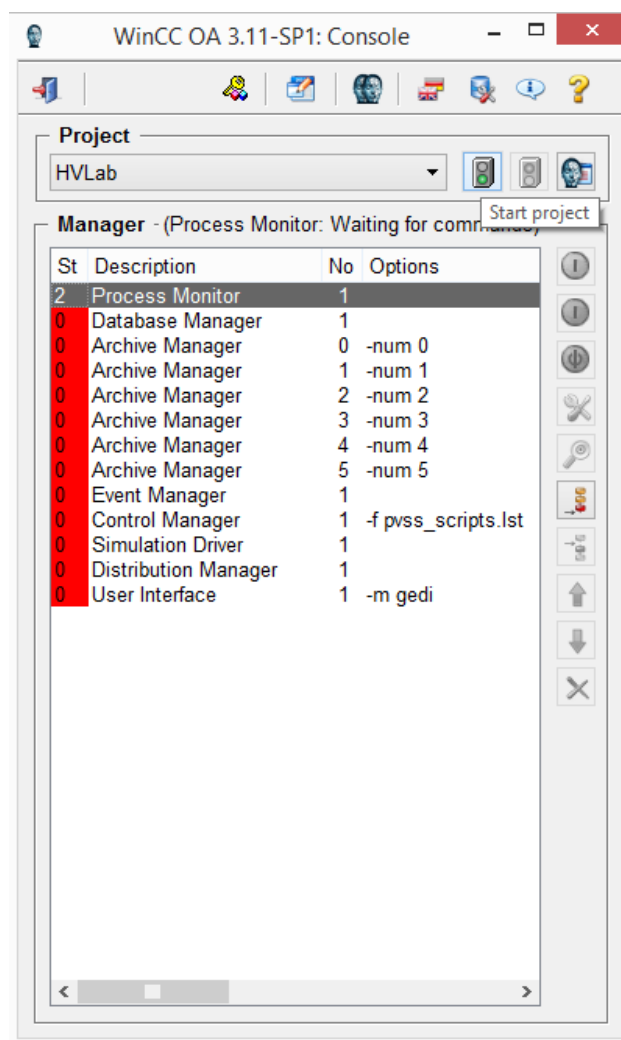


Σχήμα 1.16: Ενημέρωση κατάστασης εγκατάστασης WinCC\_OA



Σχήμα 1.17: Ολοκλήρωση εγκατάστασης WinCC\_OA

Για να χρησιμοποιούν οι χρήστες το WinCC\_OA είναι απαραίτητο να διαθέτουν μια άδεια χρήσης, διαφορετικά η λειτουργία του WinCC OA θα τερματίζεται μετά από μικρό χρονικό διάστημα. Η άδεια αυτή αποθηκεύεται σε ένα αρχείο .txt, συνήθως με το όνομα shield.txt, και αποθηκεύεται σε κάποια θέση του σκληρού δίσκου. Με αυτή την άδεια το WinCC\_OA είναι σε θέση λειτουργεί κανονικά. Μόλις περατωθεί η εγκατάσταση του WinCC OA, γίνεται η εκκίνησή του και δημιουργείται ένα καινούριο project με το όνομα HVLab.



Σχήμα 1.18: Project HVLab στην κονσόλα του WinCC\_OA

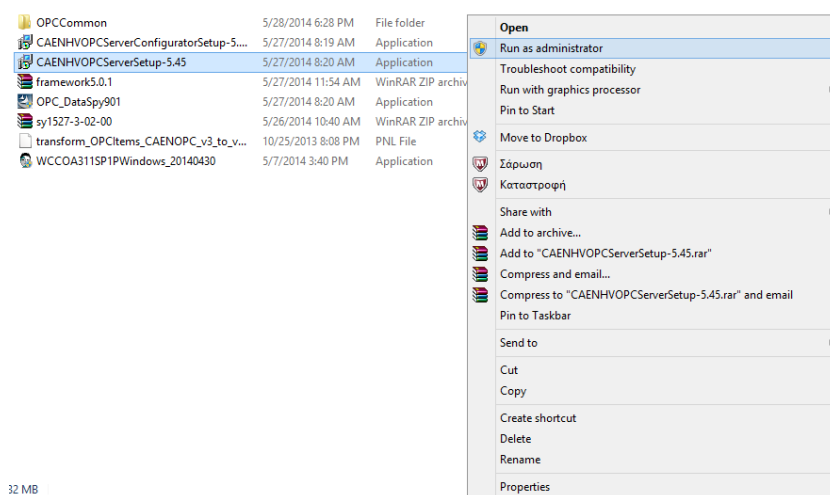
### 1.3.2 Εγκατάσταση διακομιστή της CAEN σε Windows 8.1

Για να μπορέσει το WinCC OA να συνδεθεί με το μηχάνημα υψηλής τάσης της CAEN είναι απαραίτητη η ύπαρξη ενός opc διακομιστή (server) ο οποίος μέσω του πρωτοκόλλου επικοινωνίας TCP/IP θα εξασφαλίζει την επικοινωνία του μηχανήματος υψηλής τάσης με το σύστημα αυτομάτου ελέγχου HVLab που θα αναπτυχθεί στο WinCC OA. Οι κατασκευαστές του μηχανήματος υψηλής τάσης της CAEN έχουν δημιουργήσει έναν τέτοιο διακομιστή τον οποίο παρέχουν στους χρήστες μαζί με το μηχάνημα υψηλής τάσης. Εκτός από τον διακομιστή διατίθεται από την CAEN και ένα λογισμικό, ο διαμορφωτής (Configurator), με το οποίο ο χρήστης έχει τη δυνατότητα να παραμετροποιήσει κατάλληλα τον διακομιστή ώστε να συνδεθεί με το WinCC OA. Στο HVLab οι εκδόσεις του διακομιστή (Server) και του διαμορφωτή (Configurator) που χρησιμοποιήθηκαν είναι

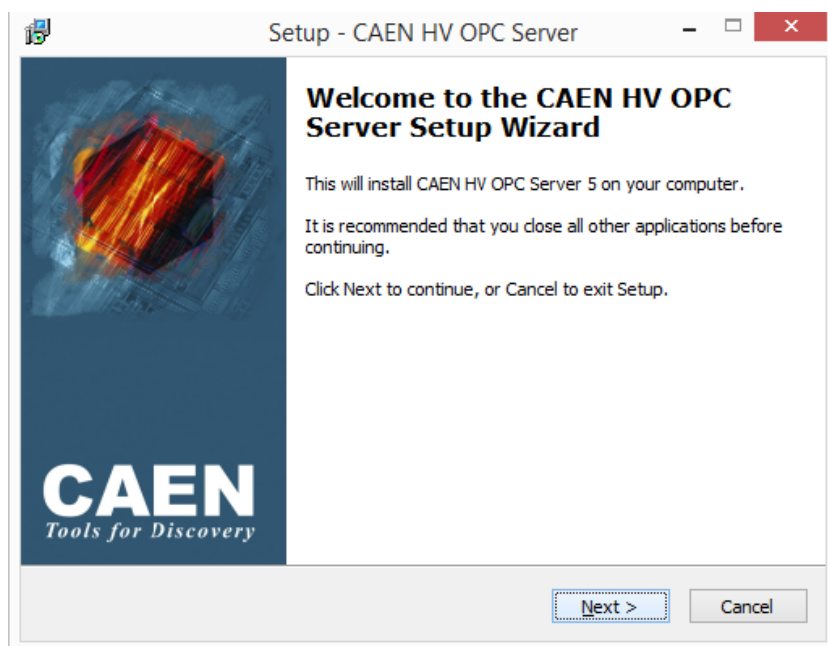
ο CAENHVOPCServer 5.45 και

ο CAENHVOPCServerConfigurator 5.32 αντίστοιχα.

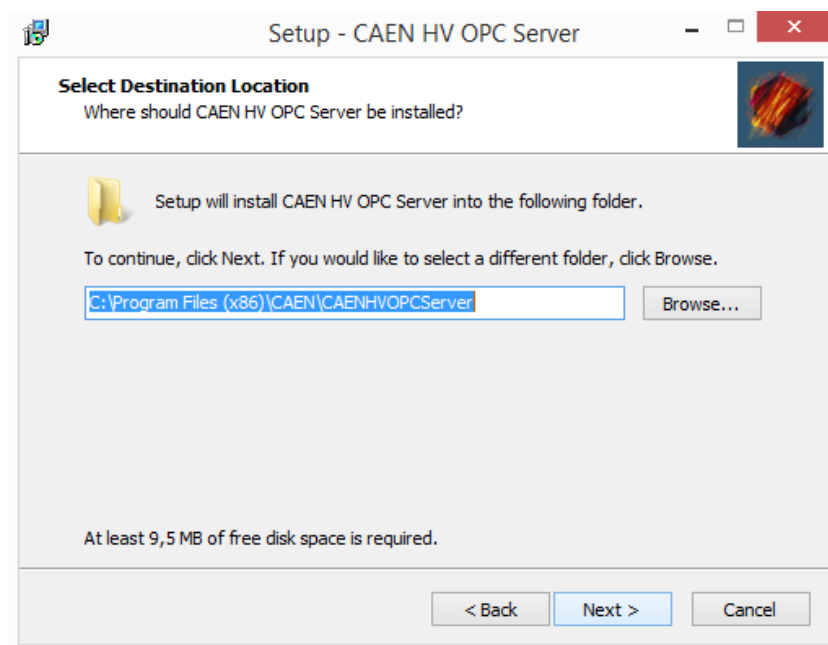
Η εγκατάσταση και εδώ του server και του configurator αλλά και η κάθε φορά εκκίνησή τους γίνεται με την επιλογή Run as Administrator. Η διαδικασία εγκατάστασής τους, ξεκινώντας από τον server, φαίνεται παρακάτω.



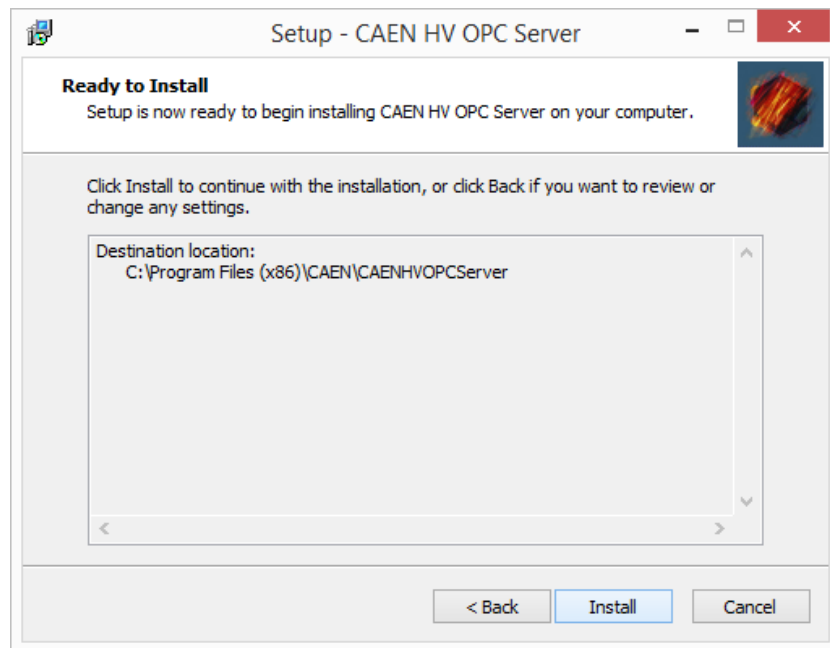
Σχήμα 1.19: Έναρξη εγκατάστασης CAENHVOPCServer 5.45 μέσω της επιλογής Run as Administrator



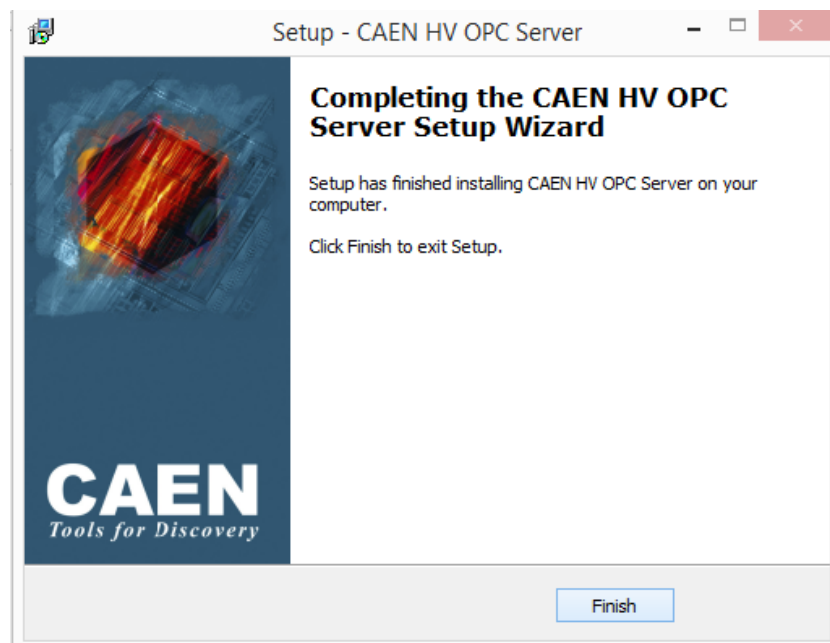
Σχήμα 1.20: Παράθυρο εκκίνησης εγκατάστασης CAENHVOPCServer 5.45



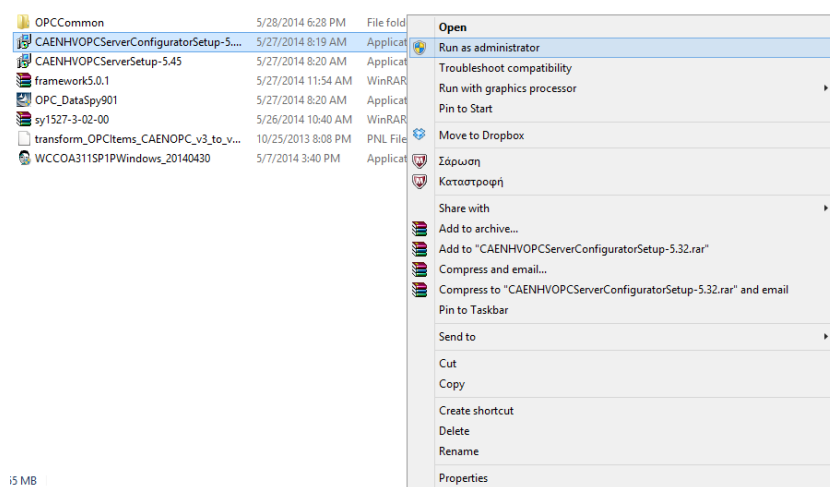
Σχήμα 1.21: Επιλογή θέσης εγκατάστασης CAENHVOPCServer 5.45



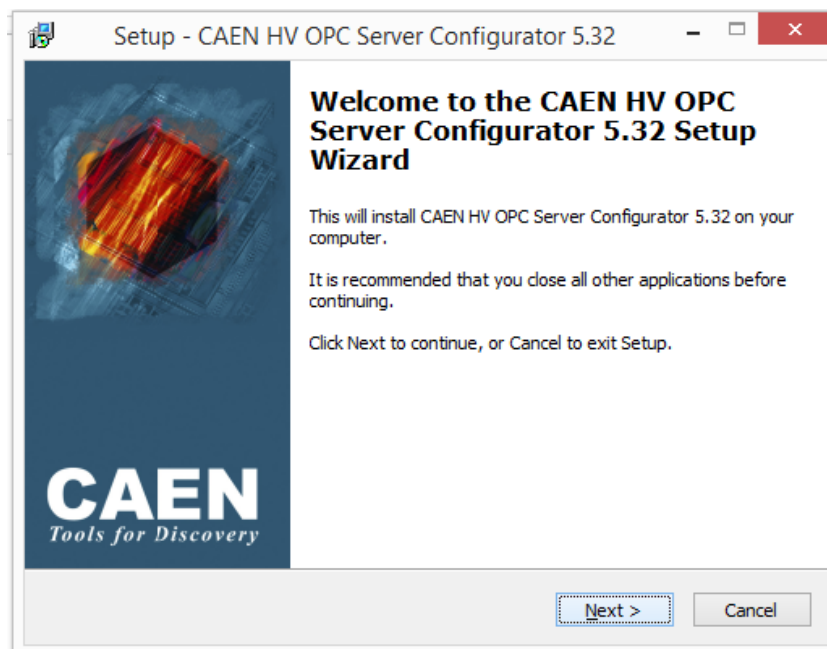
Σχήμα 1.22: Εγκατάσταση CAENHVOPCServer 5.45



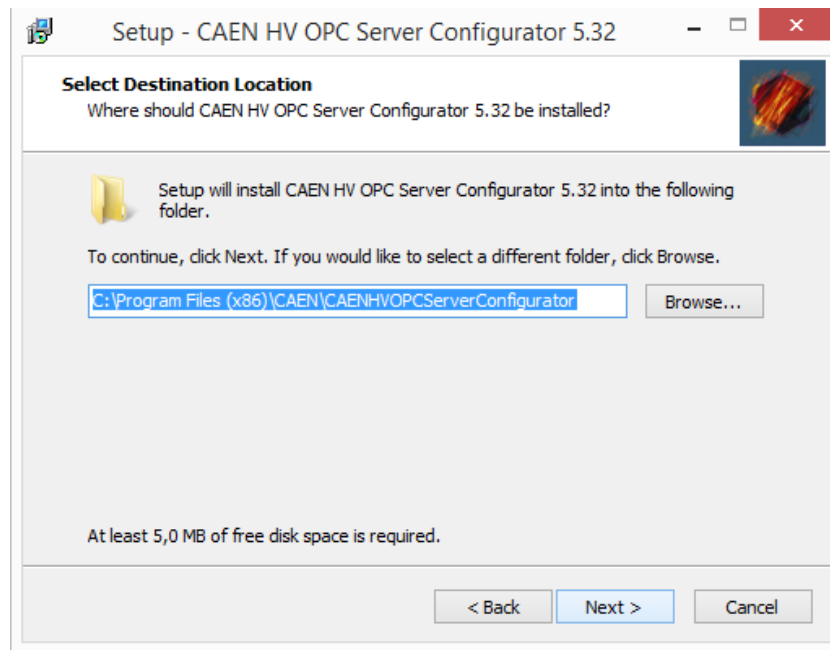
Σχήμα 1.23: Ολοκλήρωση εγκατάστασης CAENHVOPCServer 5.45



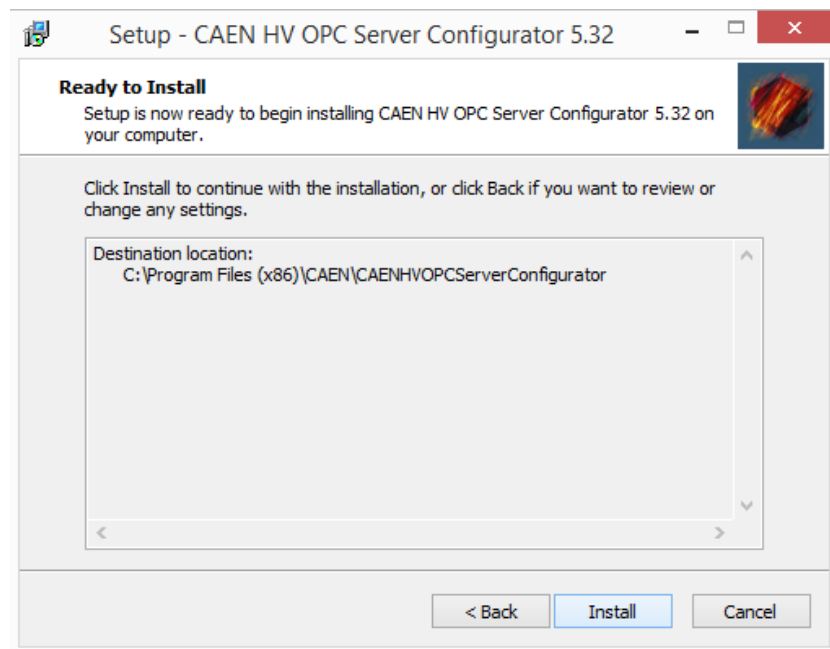
Σχήμα 1.24: Έναρξη εγκατάστασης CAENHVOPCServerConfigurator 5.32 μέσω της επιλογής Run as Administrator



Σχήμα 1.25: Παράθυρο εκκίνησης εγκατάστασης CAENHVOPCServer Configurator 5.32

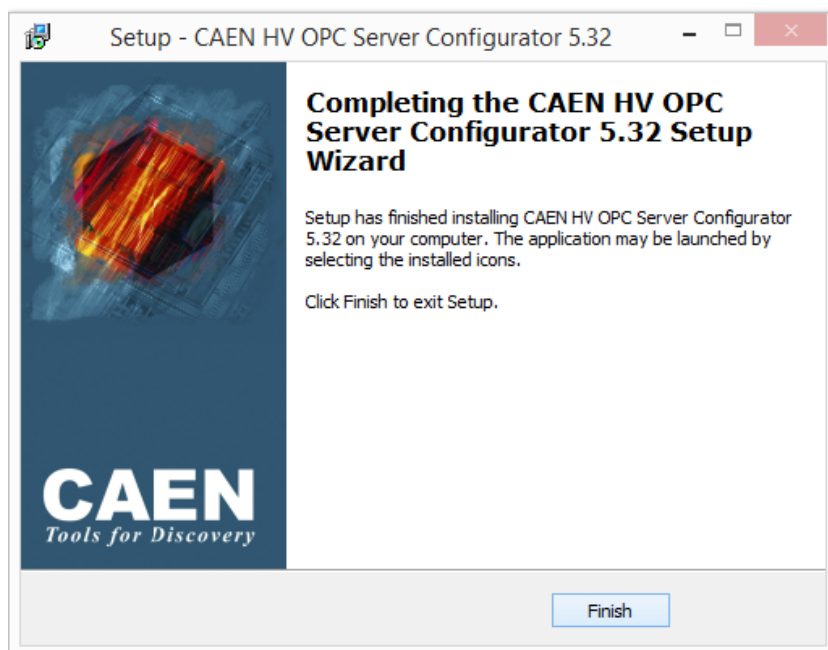


Σχήμα 1.26: Επιλογή θέσης εγκατάστασης CAENHVOPCServerConfigurator 5.32



Σχήμα 1.27: Εγκατάσταση CAENHVOPCServerConfigurator 5.32

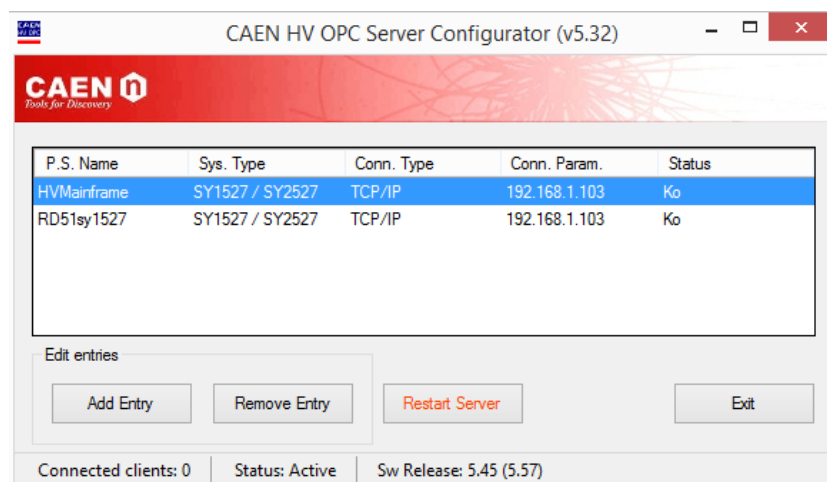




Σχήμα 1.28: Ολοκλήρωση εγκατάστασης CAENHVOPCServerConfigurator 5.32

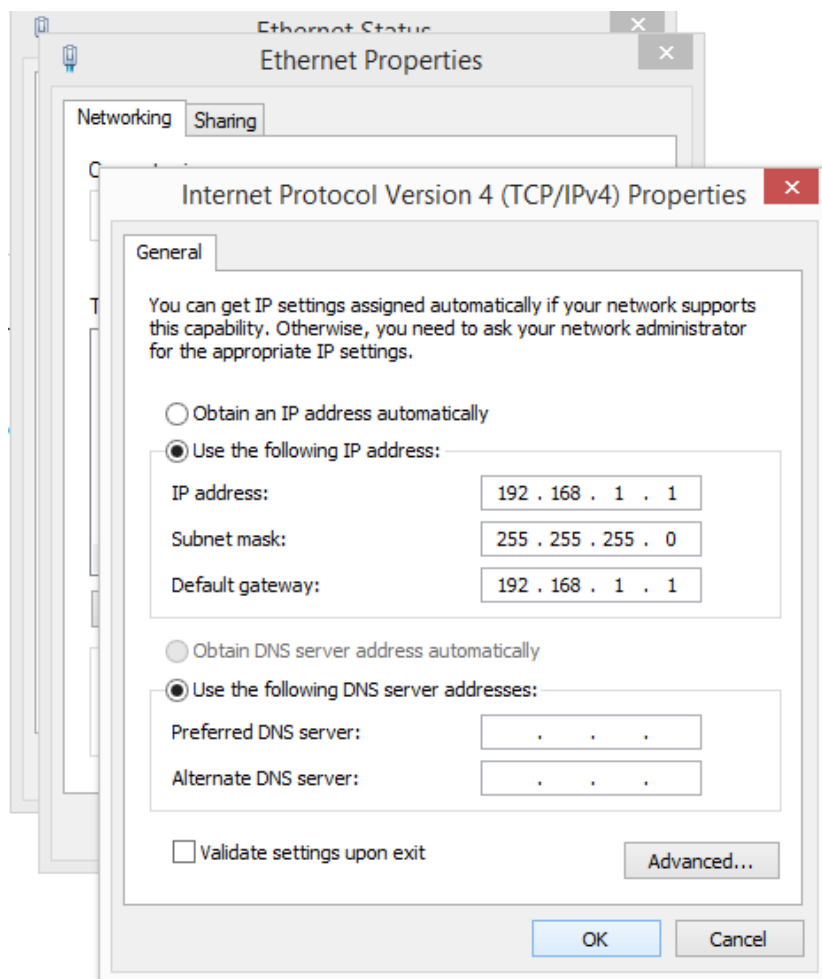
Μόλις ολοκληρωθεί η εγκατάσταση του server και του configurator ο χρήστης συνδέει το μηχάνημα υψηλής τάσης με τον υπολογιστή μέσω ethernet καλωδίου.

Στη συνέχεια εκκινεί τον server και μετά τον configurator. Μόλις το παράθυρο του CAENHVOPCServerConfigurator ανοίξει ο χρήστης τον παραμετροποιεί. Αρχικά προσθέτει μια είσοδο (Add Entry) και ορίζει τα χαρακτηριστικά της. Ορίζεται το όνομα της συσκευής, εδώ ως HVMainframe, ο τύπος της, εδώ ο τύπος της συσκευής είναι SY1527/SY2527, ο τρόπος επικοινωνίας του υπολογιστή με το μηχάνημα (Conn. Type), εδώ όπως αναφέρθηκε ο η επικοινωνία γίνεται μέσω του πρωτοκόλλου TCP/IP, και τέλος η παράμετρος επικοινωνίας (Conn.Param.) που εδώ έχει οριστεί ως 192.168.1.103. Όταν ολοκληρωθεί η παραμετροποίηση του διακοσμιστή, στο πεδίο Status του Configurator εμφανίζεται η κατάσταση σύνδεσης. Όπως φαίνεται στην παρακάτω εικόνα η κατάσταση σύνδεσης εμφανίζεται ως "Ko", γεγονός που σημαίνει ότι δεν έχει επιτευχθεί σύνδεση.



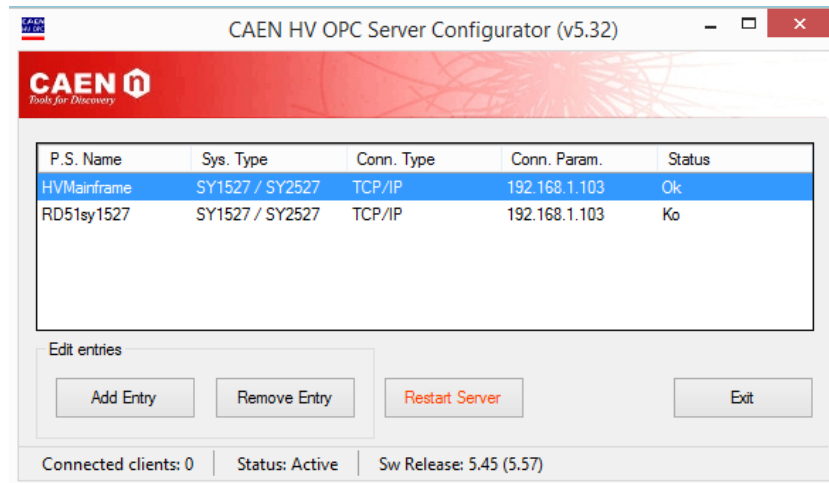
Σχήμα 1.29: Ένδειξη ανεπιτυχούς σύνδεσης στο παράθυρο ενημέρωσης του CAEN Server Configurator

Αυτό είναι αναμενόμενο από τη στιγμή που δεν έχει έχουν ορισθεί οι ρυθμίσεις ethernet δικτύου του υπολογιστή. Ως IP Address ο χρήστης πρέπει να δηλώσει την Gateway που έχει το μηχάνημα υψηλής τάσης. Την τιμή της Gateway του μηχανήματος ο χρήστης μπορεί να την αναζητήσει στην οθόνη του μηχανήματος υψηλής τάσης. Εδώ η τιμή της είναι 192.168.1.1 και με βάση αυτή ορίζονται οι παρακάτω ρυθμίσεις.



Σχήμα 1.30: Ρυθμίσεις Δικτύου

Από τη στιγμή που οι ρυθμίσεις δικτύου έχουν παραμετροποιηθεί σωστά ο χρήστης επανεκκινεί τον server (Restart Server) και στο πεδίο Status του Configurator εμφανίζεται ως κατάσταση σύνδεσης το "Ok" δηλώνοντας έτσι ότι η σύνδεση του υπολογιστή με το μηχάνημα υψηλής τάσης της CAEN είναι επιτυχής.



Σχήμα 1.31: Ένδειξη επιτυχούς σύνδεσης στο παράθυρο ενημέρωσης του CAEN Server Configurator

Ο CAENHVOPCServer είναι έτοιμος να συνδεθεί με έναν πελάτη (Client) που σε αυτήν την περίπτωση είναι ένας από τους Managers στο project HVLab του WinCC OA.

### 1.3.3 Αναβάθμιση του Slow Control System σε HVLab στο WinCC\_OA

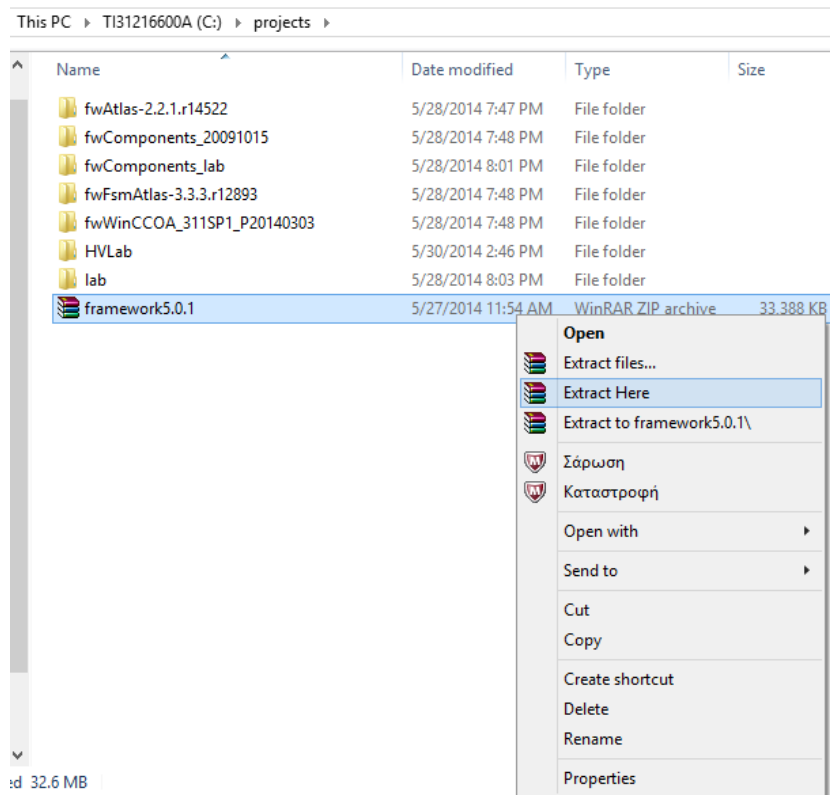
Πριν ξεκινήσει η μεταφορά των λειτουργιών (panels και pictures) του project Slow Control System του PVSS στο project HVLab του WinCC OA, είναι απαραίτητο να γίνει εγκατάσταση του JCOP Framework στο project HVLab. Το JCOP Framework θα εξοπλίσει το project HVLab με μια σειρά από δυνατότητες μέσω των οποίων το HVLab θα είναι σε θέση να συνδεθεί με το μηχάνημα υψηλής τάσης της CAEN και με τα κανάλια της κάθε κάρτας. Παράλληλα θα δώσει στους χρήστες του project HVLab τη δυνατότητα να δημιουργούν γραφικές παραστάσεις αλλά και άλλες επιπλέον εφαρμογές οι οποίες παρέχονται από το JCOP Framework μέσω έτοιμων scripts ελέγχου (Control scripts) και scripts βιβλιοθηκών (script libraries).

Για να γίνει η εγκατάσταση ο χρήστης πρέπει να κατεβάσει από τη σελίδα του CERN στο διαδίκτυο το JCOP Framework. Πρόκειται για δυο πακέτα αρχείων, το **fwInstallation- 7.1.2** και το **framework5.0.1**. Επίσης προκειμένου να λειτουργήσουν οι εφαρμογές του Slow Control System είναι απαραίτητη η χρήση παλαιότερων εκδόσεων του JCOP Framework.

Συγκεκριμένα οι εκδόσεις που χρησιμοποιούνται είναι οι παρακάτω:  
**fwAtlas-2.2.1.r14522**,  
**fwComponents\_20091015**,  
**fwFsmAtlas-3.3.3.r12893**.

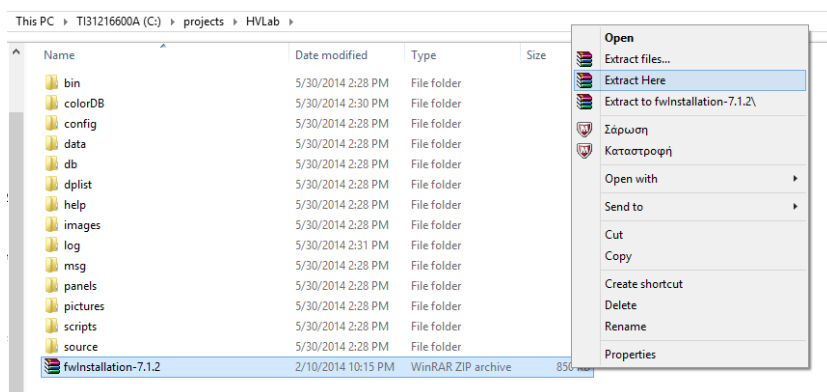
Αυτές οι παλαιότερες εκδόσεις του JCOP Framework τοποθετούνται στον φάκελο projects, όπου αποθηκεύονται όλα τα projects του WinCC\_OA.

Στις παρακάτω εικόνες παρουσιάζονται αναλυτικά τα στάδια εγκατάστασης του JCOP Framework στο project HVLab του WinCC\_OA. Ο χρήστης μόλις "κατεβάσει" το συμπιεσμένο αρχείο framework5.0.1 πρέπει να το τοποθετήσει στον φάκελο "projects", και κάνει αποσυμπίεση του αρχείου εκεί.



Σχήμα 1.32: Αποσυμπίεση αρχείου framework5.0.1 στη σωστή θέση

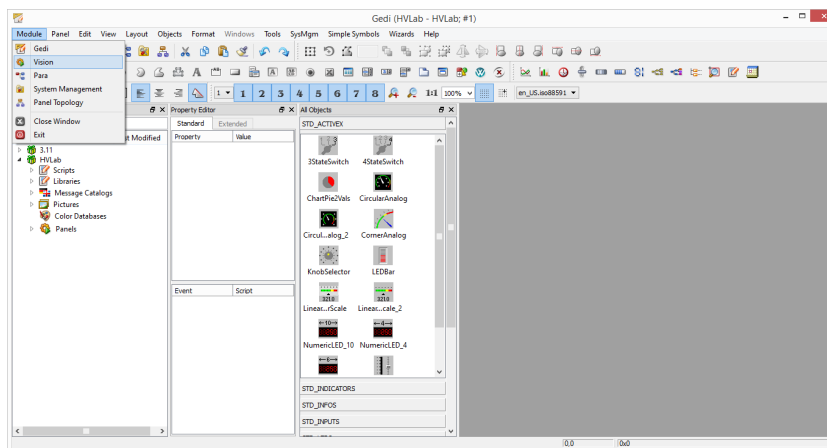
Το συμπιεσμένο αρχείο fwInstallation- 7.1.2 τοποθετείται μέσα στο φάκελο του project HVLab και αποσυμπιέζεται εκεί.



Σχήμα 1.33: Αποσυμπίεση αρχείου fwInstallation- 7.1.2 στη σωστή θέση

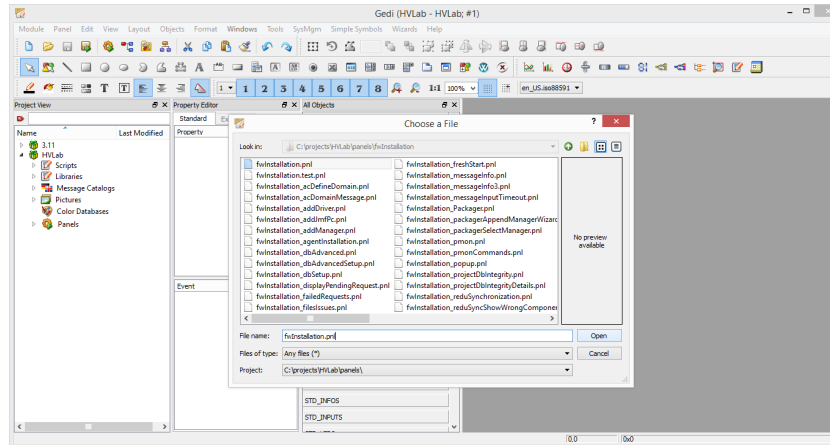
Όταν ολοκληρωθεί αυτή η διαδικασία ο χρήστης ανοίγει το project HVLab στο WinCC OA και εκτελεί τα παρακάτω βήματα.

Την πρώτη φορά που ο χρήστης θα κάνει εγκατάσταση κάποιων Components του JCOP Framework πρέπει να το κάνει από την επιλογή Module -> Vision.



Σχήμα 1.34: Επιλογή Vision στο παράθυρο Gedi

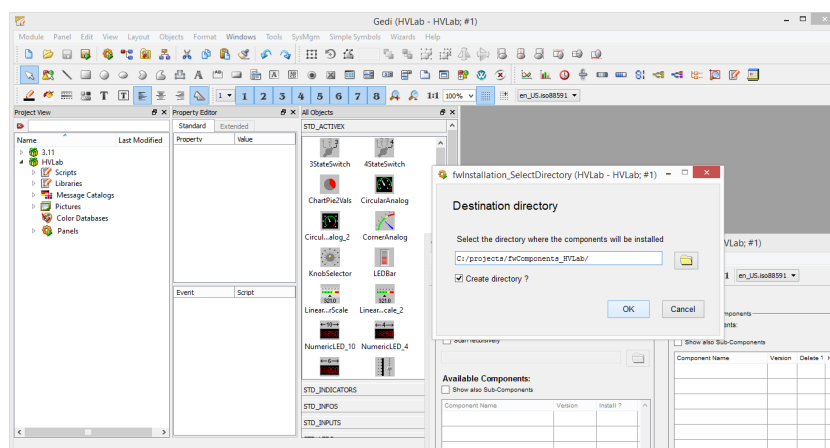
Στη συνέχεια επιλέγει το fwInstallation.pnl.



Σχήμα 1.35: Επιλογή του fwInstallation.pnl

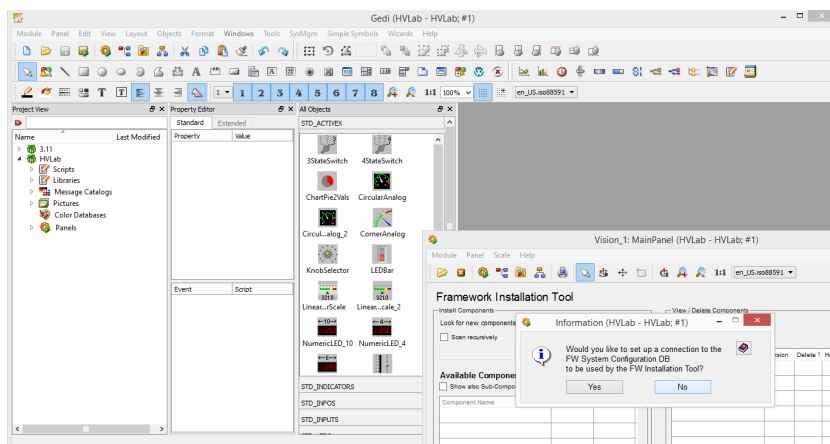
Μετά την επιλογή του fwInstallation.pnl ανοίγει ένα παράθυρο στο οποίο πραγματοποιείται η επιλογή και η δημιουργία του φακέλου στον οποίο θα αποθηκεύονται αυτόματα οι Components που ο χρήστης κάθε φορά εγκαθιστά. Εδώ η ονομασία που δώθηκε στο φάκελο είναι:

**fwComponents\_HVLab.**



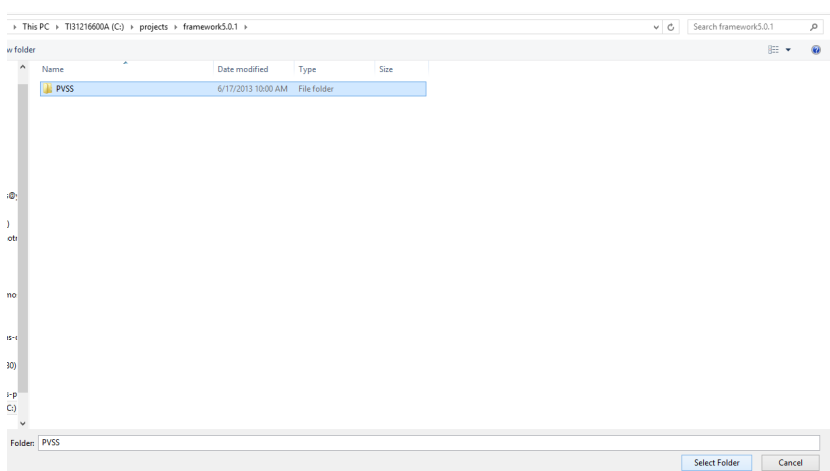
Σχήμα 1.36: Δημιουργία φακέλου για την τοποθέτηση των Components του HVLab





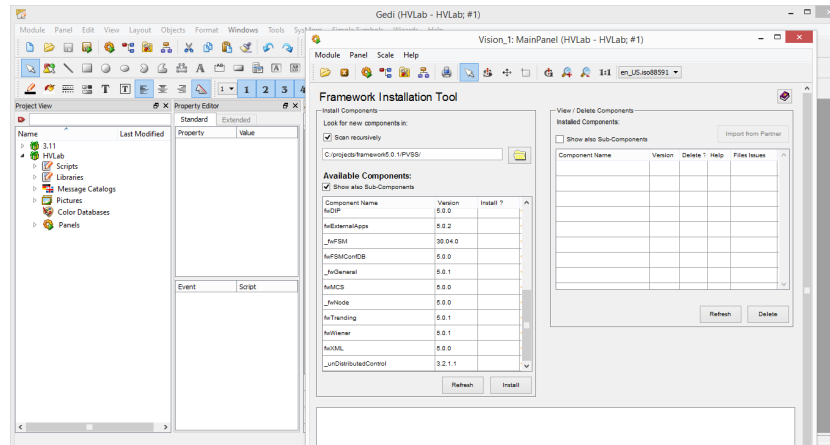
Σχήμα 1.37: Παράθυρο επιλογής σύνδεσης με το FW System Configuration DB

Στο παράθυρο του Vision, στην περιοχή αριστερά Framework Installation Tool ο χρήστης "κλικάρει" στο εικονίδιο του φακέλου και εμφανίζεται ένα παράθυρο στο οποίο ο χρήστης επιλέγει τον φάκελο στον οποίο βρίσκονται οι Components που θέλει να εγκαταστήσει.



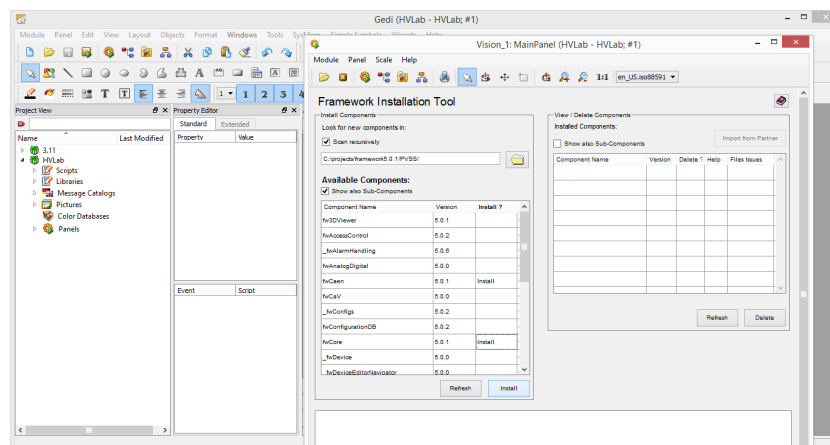
Σχήμα 1.38: Επιλογή σωστού φακέλου στον οποίο θα αναζητηθούν τα απαραίτητα Components

Έχοντας επιλεγμένα το πεδίο Scan recursively και το πεδίο Show also Sub-Components, εμφανίζονται στο πεδίο Available Components όλα τα διαθέσιμα Components του φακέλου που επιλέχθηκε προηγουμένως.



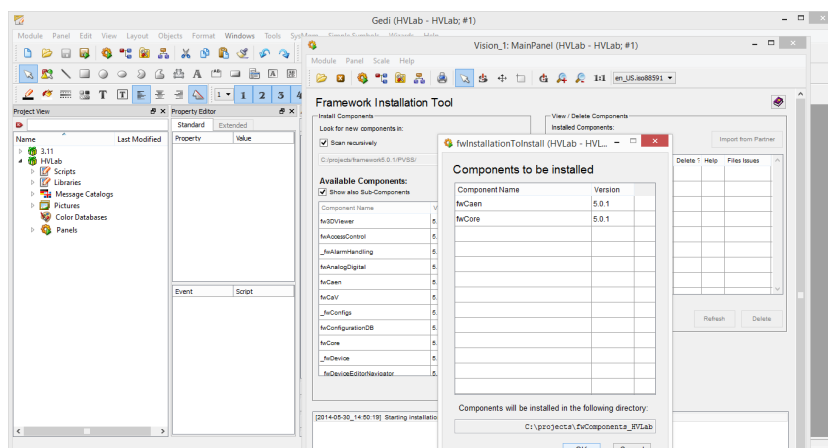
Σχήμα 1.39: Παράθυρο Vision στο αριστερό μέρος του οποίου εμφανίζονται τα διαθέσιμα Components

Ο χρήστης επιλέγει τα Components που θέλει να εγκαταστήσει και κάνει κλικ πάνω στο κενό πεδίο που βρίσκεται στην τελευταία στήλη. Αμέσως εμφανίζεται η λέξη Install στο κενό πεδίο και πατώντας ο χρήστης το κουμπί Install ξεκινά την εγκατάσταση. Στην παρακάτω εικόνα έχει γίνει επιλογή εγκατάστασης μόνο δυο Components, των **fwCaen 5.01** και **fwCore 5.01**.



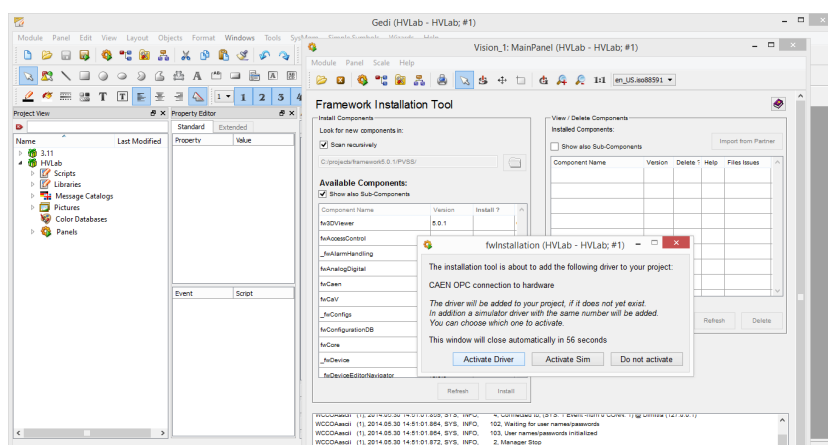
Σχήμα 1.40: Επιλογή Components προς εγκατάσταση

Μόλις ο χρήστης πατήσει το κουμπί Install εμφανίζεται ένα νέο παράθυρο επιβεβαίωσης της εγκατάστασης των Components που επιλέχθηκαν. Ο χρήστης επιλέγει OK.



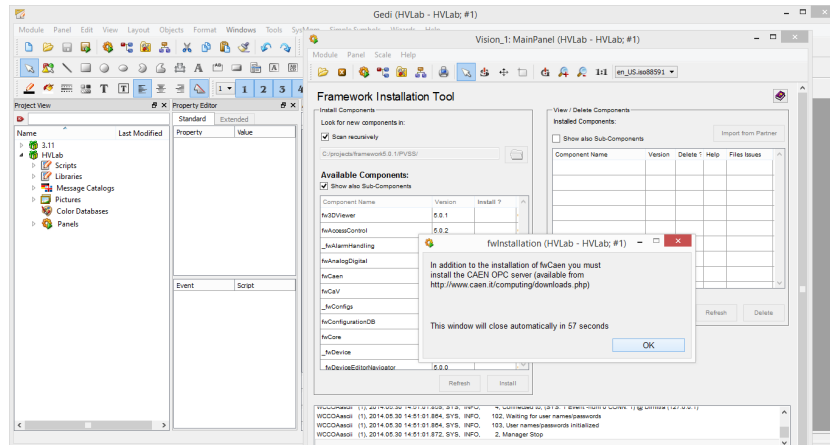
Σχήμα 1.41: Εγκατάσταση επιλεγμένων Components

Εμφανίζεται τότε ένα παράθυρο που ενημερώνει τον χρήστη ότι στο project θα προστεθεί ένας Driver ο οποίος θα εξασφαλίζει την CAEN OPC επικοινωνία του project με το μηχανήμα υψηλής τάσης της CAEN. Επιπλέον θα γίνει εγκατάσταση ενός Driver προσωμοίωσης (Simulation Driver). Ο χρήστης έχει τη δυνατότητα να επιλέξει ποιον από τους δυο Drivers θέλει να ενεργοποιήσει, κάνοντας κλικ στο αντίστοιχο κουμπί.



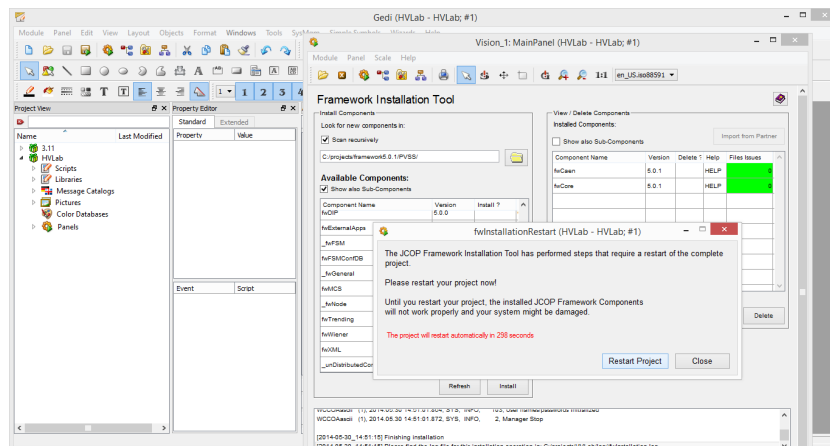
Σχήμα 1.42: Παράθυρο ενεργοποίησης ή μη ενός Driver

Στη συνέχεια εμφανίζεται ακόμη ένα παράθυρο που ενημερώνει το χρήστη ότι είναι απαραίτητη η εγκατάσταση του OPC Server της CAEN. Εδώ συγκεκριμένα, η εγκατάσταση του OPC Server της CAEN, έχει ήδη γίνει προηγουμένως όπως αναφέρθηκε παραπάνω.



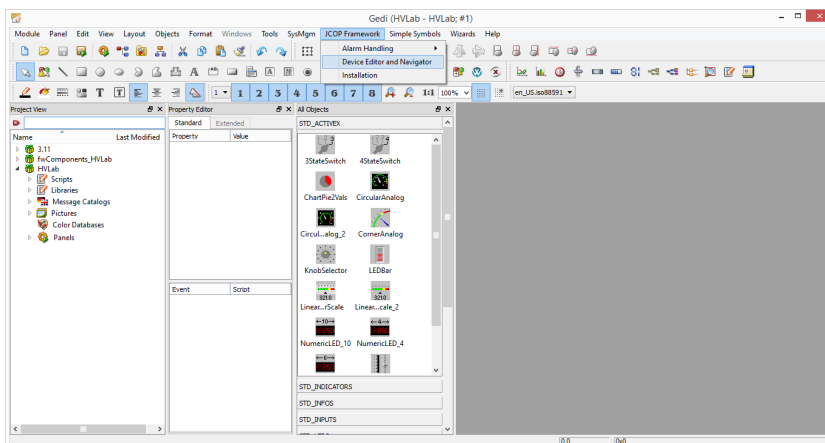
Σχήμα 1.43: Παράθυρο ειδοποίησης για την εγκατάσταση του OPC Server της CAEN

Το επόμενο παράθυρο που εμφανίζεται ενημερώνει το χρήστη ότι θα γίνει επανεκκίνηση του project προκειμένου να ολοκληρωθεί η εγκατάσταση.



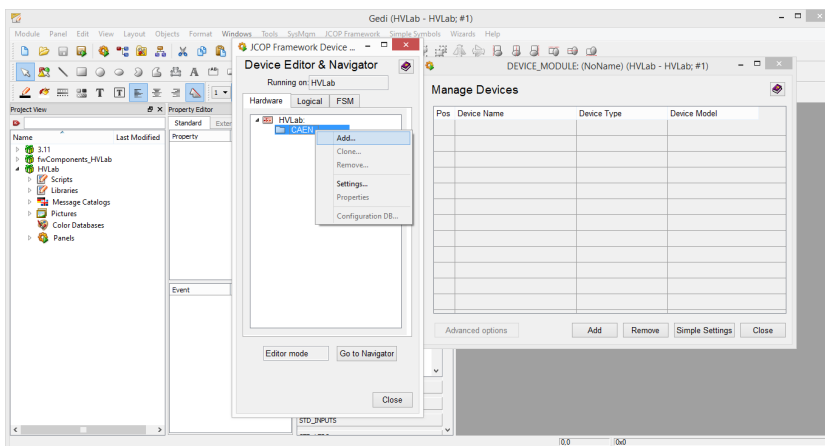
Σχήμα 1.44: Παράθυρο ενημέρωσης για την επανεκκίνηση του project

Μόλις ολοκληρωθεί η επανεκκίνηση του project και ανοίξει ο Gedi, παρατηρούμε ότι υπάρχει ένα επιπλέον κουμπί στη γραμμή εργαλείων με το όνομα JCOF Framework. Από εδώ και στο εξής ο χρήστης θα χρησιμοποιεί αυτό το κουμπί προκειμένου να προσθέσει και άλλους Components στο project μέσω της επιλογής Installation. Υπάρχουν όμως και άλλες επιλογές και το επόμενο στάδιο είναι η χρήση της επιλογής Device Editor and Navigator.



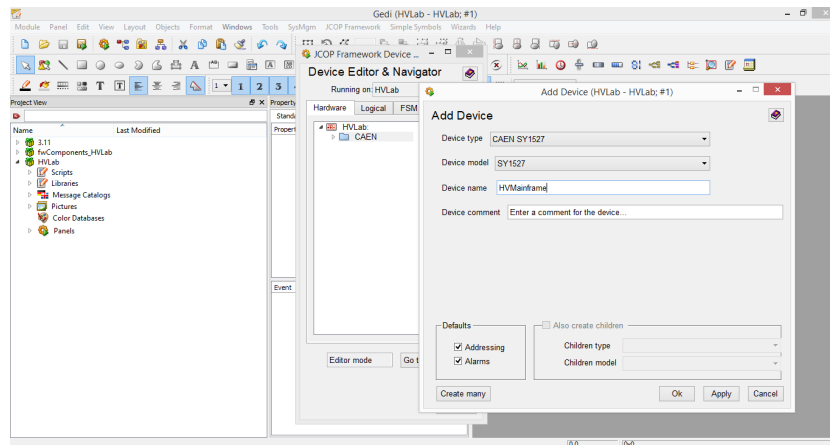
Σχήμα 1.45: Παρουσία JCOF Framework στην γραμμή εργαλείων του Gedi

Όταν ανοίξει το παράθυρο του Device Editor and Navigator, ο χρήστης επιλέγει να βρίσκεται σε Editor Mode προκειμένου να δηλώσει στην CAEN την συσκευή με την οποία θα συνδεθεί το project. Για το σκοπό αυτό επιλέγει Add.



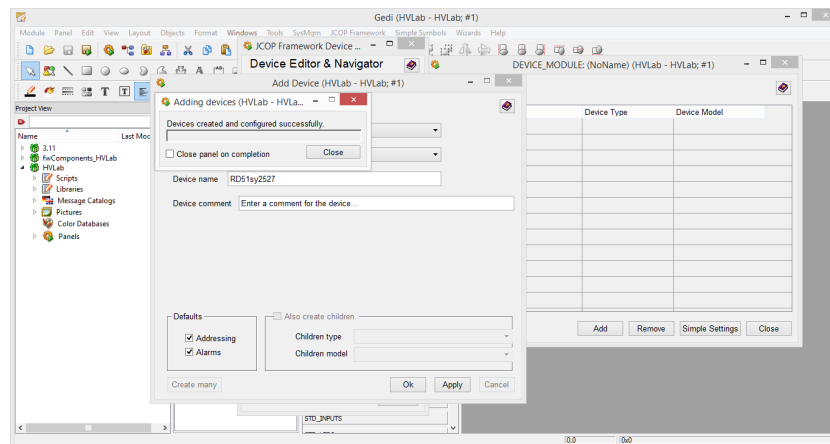
Σχήμα 1.46: Προσθήκη συσκευής στο παράθυρο Device Editor and Navigator (DEN)

Με την επιλογή Add, ανοίγει το παράθυρο Add Device, στο οποίο ορίζονται τα χαρακτηριστικά της συσκευής η οποία πρόκειται να συνδεθεί. Συγκεκριμένα ορίζεται ο τύπος της συσκευής, το μοντέλο και το όνομά της. Σημαντικό είναι το όνομα της συσκευής που θα δηλωθεί εδώ να είναι ίδιο με αυτό που έχει οριστεί στον διαμοφωτή του διακομιστή (Server Configurator). Έτσι ορίζονται τα παρακάτω.



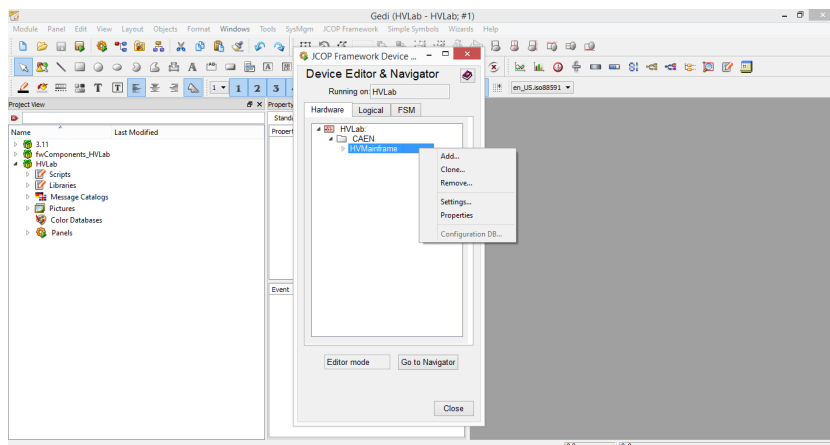
Σχήμα 1.47: Δήλωση χαρακτηριστικών της συσκευής

Στη συνέχεια εμφανίζεται ένα παράθυρο που ενημερώνει το χρήστη ότι η δήλωση της συσκευής έγινε με επιτυχία.



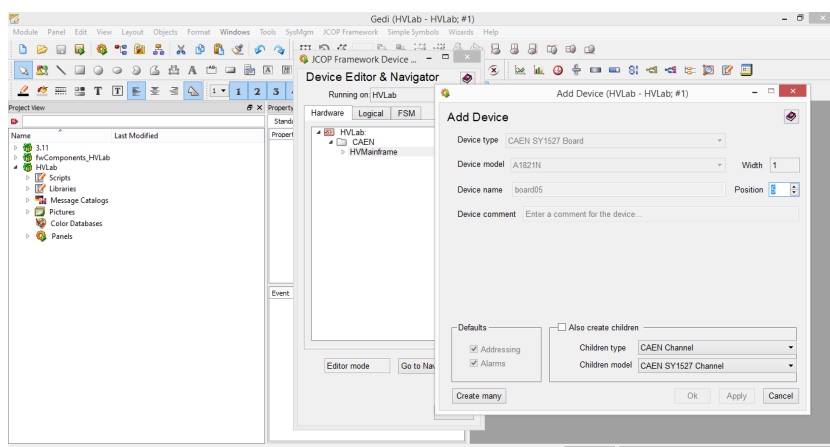
Σχήμα 1.48: Παράθυρο ενημέρωσης για την επιτυχή δήλωση της συσκευής

Με τον ίδιο τρόπο γίνεται η δήλωση στη συσκευή των καρτών που έχουν τοποθετηθεί σε αυτήν.



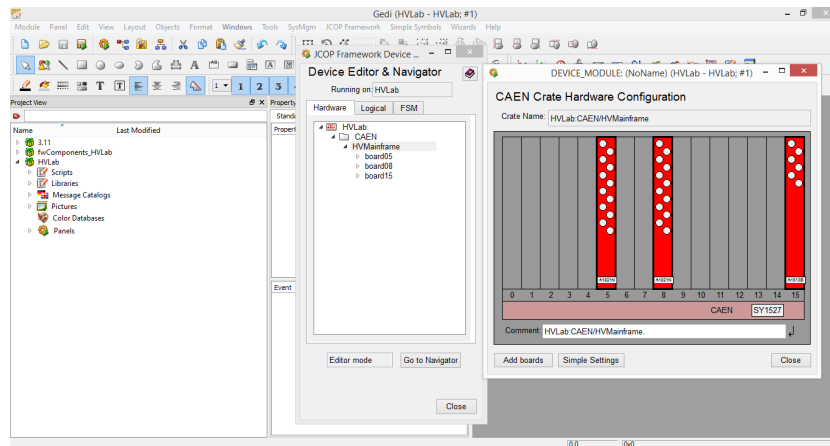
Σχήμα 1.49: Προσθήκη κάρτας στο παράθυρο Device Editor and Navigator (DEN)

Ο χρήστης και σε αυτήν την περίπτωση δηλώνει τον τύπο της κάρτας, το μοντέλο και το όνομά της. Επιπλέον όμως πρέπει να ορισθούν η θέση πάνω στο μηχανήμα στην οποία έχει τοποθετηθεί η κάρτα αλλά και να κλικαριστεί η επιλογή Also create children προκειμένου να οριστούν και τα κανάλια της κάρτας. Στην εικόνα παρακάτω ορίζεται η κάρτα **A1821N**, ενώ με τον ίδιο τρόπο ορίζονται και οι άλλες δυο κάρτες, **A1821N** στη θέση 8 (board08) και **A1315B** στη θέση 15 (board15) που χρησιμοποιήθηκαν.



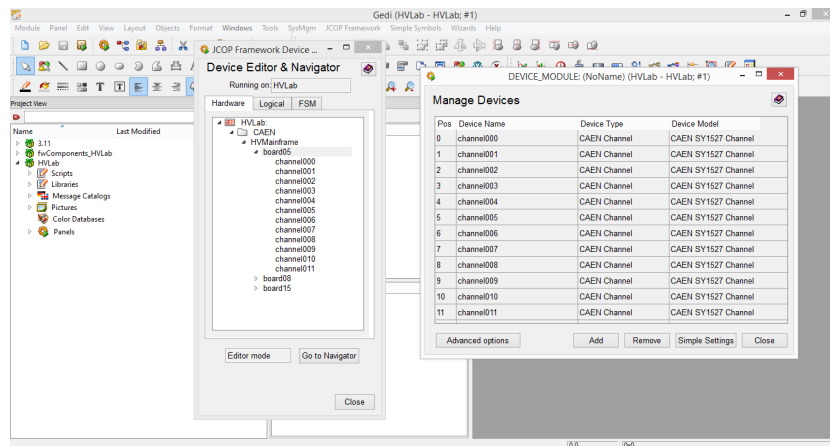
Σχήμα 1.50: Δήλωση χαρακτηριστικών της κάρτας

Στην εικόνα φαίνεται η επιτυχής δήλωση της κάρτας A1821N και των υπολοίπων καρτών.



Σχήμα 1.51: Παράθυρο ενημέρωσης για την επιτυχή δήλωση της κάρτας

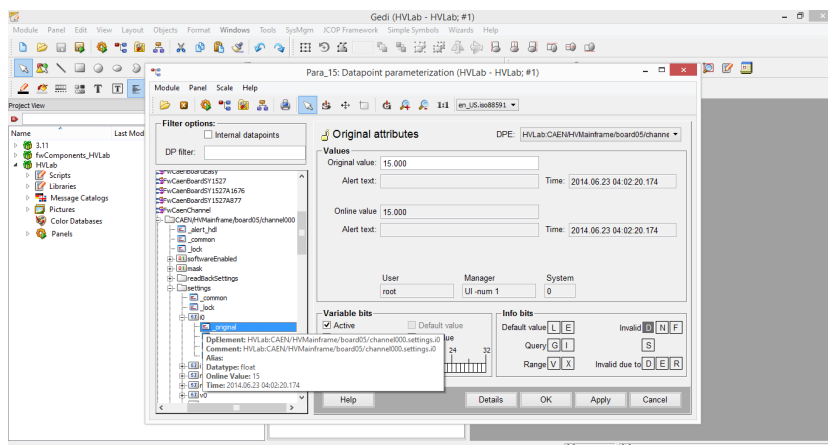
Εκτός από την κάρτα είναι φανερό ότι ορίστικαν σωστά και τα κανάλια που αυτή διαθέτει.



Σχήμα 1.52: Παράθυρο ενημέρωσης για την επιτυχή δήλωση των καναλιών της κάρτας



Ενδεικτικά μετά το πέρας της διαδικασίας αν ο χρήστης ανοίξει τον para είναι σε θέση να δει τα datapoints που αντιστοιχούν στο κανάλι0 (channel000), της κάρτας που βρίσκεται στη θέση 5 (board05) στο μηχάνημα HVMainframe της CAEN.



Σχήμα 1.53: Παράθυρο εμφάνισης των datapoints ( para: Datapoint Parametrization)

Να σημειωθεί εδώ ότι εκτός από τους **fwCaen 5.01** και **fwCore 5.01** Components που εισήχθησαν προηγουμένως, για να μπορέσει το HVLab να λειτουργεί σωστά είναι απαραίτητο να γίνει προσθήκη και των παρακάτω Components:

**fwTrending 5.0.1**

**fwXML 5.0.0**

**fwAccessControl 3.2.7**

**fwAtlas 2.2.1**

**fwFsmAtlas 3.3.3**

Η προσθήκη τους γίνεται με τον τρόπο που περιγράφηκε προηγουμένως.

Κατά την εγκατάσταση του fwAtlas θα εμφανιστεί μήνυμα ότι μερικά στοιχεία του είναι κατεστραμμένα. Αυτό οφείλεται στο γεγονός ότι θα έπρεπε να έχει γίνει προηγουμένως εγκατάσταση του fwAtlasInstallation 1.0.0, το οποίο όμως δεν μπορεί να εγκατασταθεί σωστά. Παρ' όλα αυτά το fwAtlas θα λειτουργεί, ακόμη και αν έχει ορισμένα κατεστραμμένα στοιχεία. Επιπλέον το αρχείο fwfsmAtlasWidget.ctl που βρίσκεται στο φάκελο

fwComponents\_20091015-> scripts-> libs τοποθετείται στο φάκελο  
fwComponents\_HVLab-> scripts-> libs.

Το επόμενο βήμα προκειμένου το HVLab να έχει τις λειτουργίες του Slow Control System είναι μεταφορά των panels και των εικόνων στο φάκελο των panels και στο φάκελο των pictures αντίστοιχα του HVLab.

Αρχικά όσον αφορά τις εικόνες αντικαταστάθηκαν με καινούριες προκειμένου το project HVLab να χρησιμοποιηθεί στα AIDA testbeams. Η τοποθέτηση των εικόνων στο φάκελο pictures του HVLab δεν είναι αρκετή για να εμφανιστούν οι εικόνες στο HVLab. Πιο συγκεκριμένα πρέπει ο χρήστης μέσω της ρύθμισης Fill να επιλέξει την εικόνα που θέλει από τις διαθέσιμες στο φάκελο των pictures του HVLab και να την τοποθετήσει στο κενό πεδίο του panel. Αυτή η διαδικασία ακολουθήθηκε και για το logo της AIDA και για τον πυρφόρο του πολυτεχνίου.

Όσον αφορά τα panels του Slow Control System που τοποθετήθηκαν στο φάκελο των panels του HVLab για να λειτουργήσουν σωστά έπρεπε να υποστούν ορισμένες αλλαγές στο κώδικα. Συγκεκριμένα όλα τα σημεία του κώδικα στα οποία χρησιμοποιούνται τα datapoints που αντιστοιχούν στα κανάλια του μηχανήματος υψηλής τάσης της CAEN άλλαξαν από

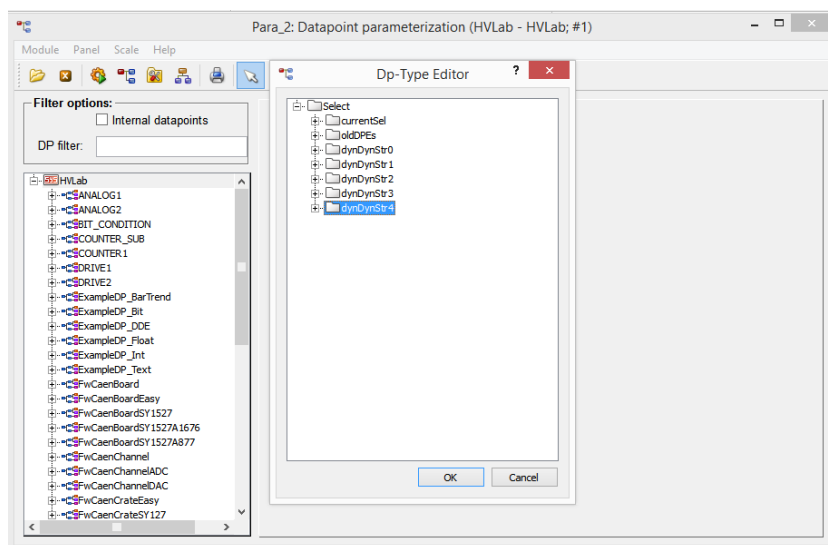
**dist\_1:CAEN/RD51sy2527/boardXX/channelXXX**

σε

**HVLab:CAEN/HVMainframe/boardXX/channelXXX.**

Έγιναν ακόμη μερικές αλλαγές στους κώδικες προκειμένου να λειτουργεί το HVLab σωστά. Σε όλους τους πίνακες του project HVLab στους οποίους αναγραφόταν το boardXX στο οποίο αντιστοιχεί κάθε κανάλι, από τη λέξη boardXX έλειπε το "b" και εμφανιζόταν ως "oardXX". Μετά από αλλαγή του κώδικα σε όλα τα σημεία που υπήρχε αυτό το πρόβλημα, το boardXX εμφανιζόταν κανονικά στους πίνακες του project.

Εκτός από τα datapoints που αντιστοιχούν στα κανάλια του μηχανήματος υψηλής τάσης της CAEN και δημιουργήθηκαν αυτόματα μέσω του JCOP Framework υπήρχαν και άλλα datapoints στο project RD51sy1527 Slow Control System, τα οποία στο HVLab χρειάστηκε να δημιουργηθούν από την αρχή. Η δημιουργία από την αρχή καινούριων datapoint elements γίνεται στον para στον Dp-Type Editor.



Σχήμα 1.54: Παράθυρο Dp-Type Editor για τη δημιουργία datapoints

Ο χρήστης έχει τη δυνατότητα να φτιάξει σε ένα Datapoint Type όσα datapoint elements επιθυμεί και να ορίσει τον τύπο (Datatype) του καθενός. Τα datapoints που δημιουργήθηκαν είναι τα παρακάτω:

- *Datapoint Type: AIDA\_DCS*
  - HVLab:user.username (*Datatype* string)
  - HVLab:user.pass (*Datatype* string)
  - HVLab:user.dpeInt (*Datatype* int)
  - HVLab:user.time\_start (*Datatype* time)
  - HVLab:user.time\_end (*Datatype* time)
  - HVLab:user.pass0 (*Datatype* string)
  - HVLab:user.pass1 (*Datatype* string)
  - HVLab:user.pass2 (*Datatype* string)
  - HVLab:user.pass3 (*Datatype* string)
  - HVLab:user.pass4 (*Datatype* string)
  - HVLab:user.curActiveGroup (*Datatype* int)
- *Datapoint Type: Select*
  - HVLab:selChannels.currentSel (*Datatype* int)
  - HVLab:selChannels.oldDPEs (*Datatype* dyn\_string)

**HVLab:selChannels.dynDynStr0** (*Datatype* dyn\_string)

**HVLab:selChannels.dynDynStr1** (*Datatype* dyn\_string)

**HVLab:selChannels.dynDynStr2** (*Datatype* dyn\_string)

**HVLab:selChannels.dynDynStr3** (*Datatype* dyn\_string)

**HVLab:selChannels.dynDynStr4** (*Datatype* dyn\_string)

Εκτός από τη δημιουργία των datapoints, ορισμένα από αυτά αρχικοποιήθηκαν. Συγκεκριμένα η Original value στα παρακάτω datapoints ορίστηκε:

**HVLab:user.pass0** *Original value:* **ntua0**

**HVLab:user.pass1** *Original value:* **ntua1**

**HVLab:user.pass2** *Original value:* **ntua2**

**HVLab:user.pass3** *Original value:* **ntua3**

**HVLab:user.pass4** *Original value:* **ntua4**

Δεδομένου ότι στο project RD51sy1527 τα παραπάνω datapoints έχουν δημιουργηθεί στο σύστημα dist\_1 μέσα στον κώδικα των panels καλούνται ως dist\_1:selChannels.currentSel. Για το λόγο αυτό στα σημεία του κώδικα που καλούνται τα παραπάνω datapoints, το όνομα του συστήματος αλλάζει σε HVLab και ενδεικτικά για το παράδειγμα του datapoint που αναφέρθηκε προηγουμένως γίνεται HVLab:selChannels.currentSel.

Μόλις ολοκληρωθεί η παραπάνω διαδικασία το project HVLab είναι σε θέση να επικοινωνεί με το μηχάνημα υψηλής τάσης της CAEN και να επιτελεί όλες τις λειτουργίες, που το project RD51 επιτελούσε στο PVSS, στο WinCC\_OA.

### 1.3.4 Ανάπτυξη του HVLab στο WinCC\_OA

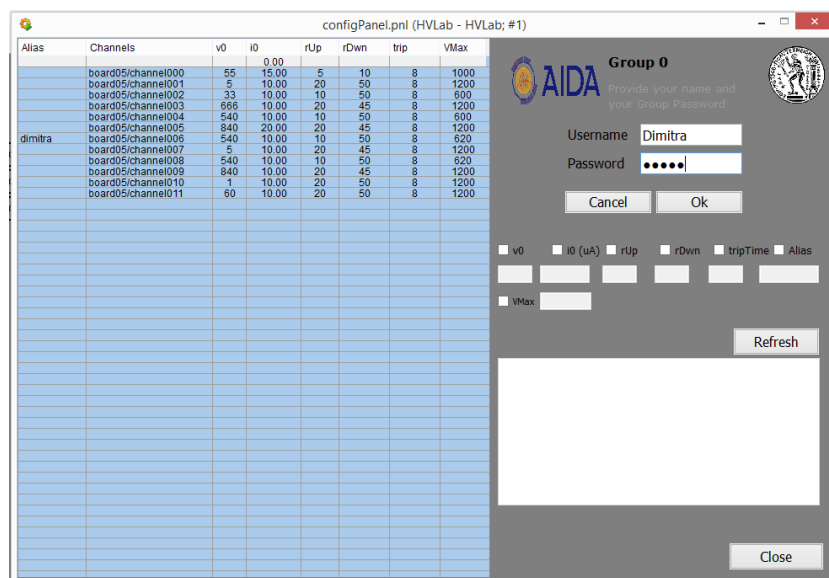
Αφού ολοκληρώθηκε η αναβάθμιση του Slow Control System από PVSS σε HVLab στο WinCC\_OA προστέθηκαν στο HVLab ορισμένες νέες λειτουργίες. Συγκεκριμένα στο παράθυρο ρυθμίσεων προστέθηκαν τρία επιπλέον κουμπιά, το κουμπί **Upload data**, το κουμπί **Load Script** και το κουμπί **Save Script**.

- **Save Script** Με αυτό το κουμπί ο χρήστης έχει τη δυνατότητα να αποθηκεύσει σε ένα αρχείο .txt, που αποθηκεύεται στο σκληρό δίσκο, όλα τα στοιχεία του κάθε καναλιού που αναγράφονται στον πίνακα του παραθύρου ρυθμίσεων. Η μορφή του αρχείου .txt που αποθηκεύεται είναι συγκεκριμένη.
- **Load Script** Το κουμπί Load Script επιτρέπει στο χρήστη να επιλέξει ένα αρχείο .txt από τον υπολογιστή του και να εμφανίσει το περιεχόμενό του στο λευκό πλαίσιο του παραθύρου ρυθμίσεων. Είναι απαραίτητο το περιεχόμενο του αρχείου .txt που θα επιλεγεί από το χρήστη να έχει συγκεκριμένη μορφή.
- **Upload data** Πατώντας αυτό το κουμπί ο χρήστης έχει τη δυνατότητα να "περάσει" τις τιμές που αναγράφονται στο λευκό πλαίσιο του παραθύρου ρυθμίσεων, στη βάση δεδομένων του project HVLab. Ουσιαστικά με αυτό το κουμπί οι τιμές των datapoints στο para αντικαθίστανται από αυτές που εμφανίζονται στο πλαίσιο με σκοπό την επικοινωνία του πλαισίου με το μηχάνημα υψηλής τάσης της CAEN, δεδομένου του ότι υπάρχει επικοινωνία μεταξύ para και του μηχανήματος υψηλής τάσης.

Η τελική μορφή του project HVLab παρουσιάζεται στις παρακάτω εικόνες.



Σχήμα 1.55: Main Panel (Κύριο Παράθυρο)



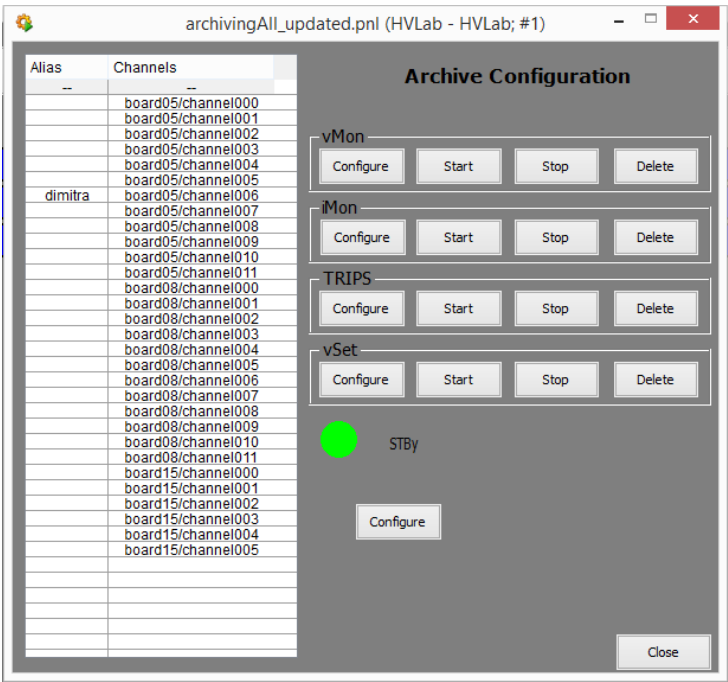
Σχήμα 1.56: Configuration Panel (Παράθυρο Παραμετροποίησης)

Alias	Channels	v0	i0	rUp	rDwn	trip	VMax
	board05/channel000	55	15.00	5	10	8	1000
	board05/channel001	5	10.00	20	50	8	1200
	board05/channel002	33	10.00	10	50	8	800
	board05/channel003	666	10.00	20	45	8	1200
	board05/channel004	540	10.00	10	50	8	600
	board05/channel005	840	20.00	20	45	8	1200
dimitra	board05/channel006	540	10.00	10	50	8	620
	board05/channel007	5	10.00	20	45	8	1200
	board05/channel008	540	10.00	10	50	8	620
	board05/channel009	840	10.00	20	45	8	1200
	board05/channel010	1	10.00	20	50	8	1200
	board05/channel011	60	10.00	20	50	8	1200

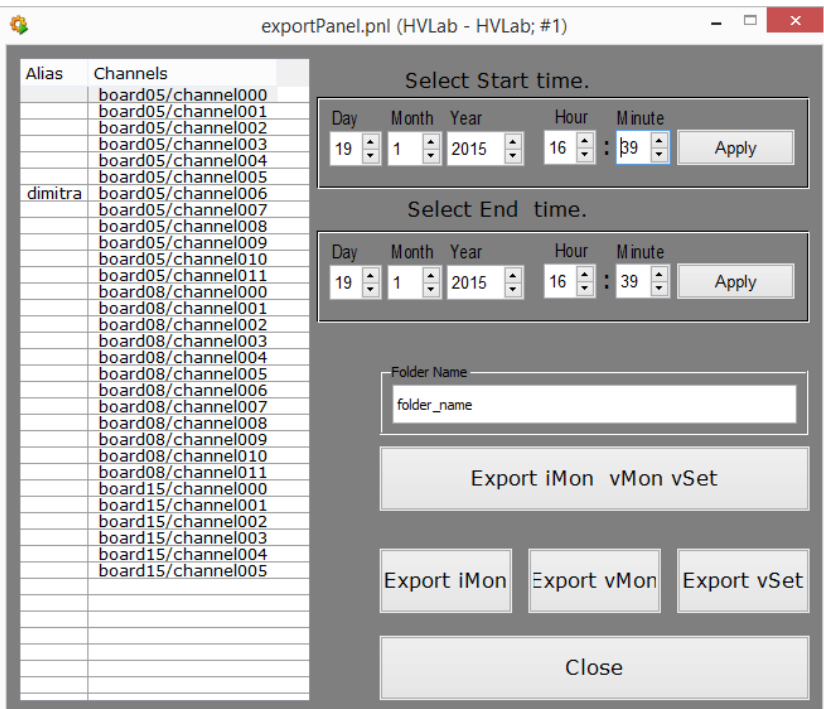
Σχήμα 1.57: Configuration Panel (Παράθυρο Παραμετροποίησης ολοκληρωμένο)

Alias	Channels
	board05/channel000
	board05/channel001
	board05/channel002
	board05/channel003
	board05/channel004
dimitra	board05/channel005
	board05/channel006
	board05/channel007
	board05/channel008
	board05/channel009
	board05/channel010
	board05/channel011
	board08/channel000
	board08/channel001
	board08/channel002
	board08/channel003
	board08/channel004
	board08/channel005
	board08/channel006
	board08/channel007
	board08/channel008
	board08/channel009
	board08/channel010
	board08/channel011
	board15/channel000
	board15/channel001
	board15/channel002
	board15/channel003
	board15/channel004
	board15/channel005

Σχήμα 1.58: Selection Panel (Παράθυρο Επιλογής)



Σχήμα 1.59: Archiving All Panel



Σχήμα 1.60: Export Panel



## Κεφάλαιο 2

# ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΙΚΗ ΠΛΑΤΦΟΡΜΑ ARDUINO



Σχήμα 2.1: Πλακέτα Arduino Mega με Ethernet Shield

Το Arduino είναι ένα εργαλείο για την ανάπτυξη υπολογιστών, που είναι σε θέση να αισθανθούν και να ελέγξουν τον φυσικό κόσμο περισσότερο απ' ό,τι οι κοινοί υπολογιστές γραφείου. Είναι μια ανοιχτή υπολογιστική πλατφόρμα βασισμένη σε μια απλή μικροελεντική πλακέτα και σε ένα αναπτυγμένο περιβάλλον εγγραφής λογισμικού. Το Arduino μπορεί να χρησιμοποιηθεί για τη λήψη δεδομένων από αισθητήρες, καθώς και για τον έλεγχο φώτων και κινητήρων. Κάθε εφαρμογή του Arduino μπορεί είτε να λειτουργήσει αυτόνομα είτε να επικοινωνεί με λογισμικό που τρέχει στον υπολογιστή. Η πλακέτα μπορεί να συναρμολογηθεί από το χρήστη ή να αγοραστεί έτοιμη. Το περιβάλλον εργασίας Arduino IDE διατηρείται δωρεάν από τη ιστοσελίδα του Arduino στο internet.

Υπάρχουν πολλοί μικροελενκτές και πλατφόρμες, Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboards, και πολλοί άλ-

λοι που προσφέρουν παρόμοιες λειτουργίες. Το Arduino όμως προσφέρει μερικά σημαντικά πλεονεκτήματα στους χρήστες του σε σχέση με άλλα συστήματα.

1. Είναι σχετικά φθηνοί αν συγκριθούν με άλλες μικροελεκτρικές πλακέτες που κυκλοφορούν.
2. Είναι συμβατοί με όλα τα λειτουργικά συστήματα (Windows, Linux, Mac) σε αντίθεση με τους παραπάνω που οι περισσότεροι είναι συμβατοί μόνο με Windows.
3. Το προγραμματιστικό περιβάλλον είναι απλό δίνοντας την δυνατότητα σε αρχάριους προγραμματιστές να δουλεύουν με ευκολία ενώ ταυτόχρονα οι πιο έμπειροι επωφελούνται από αυτό.
4. Το ελεύθερο λογισμικό του Arduino προσφέρει επίσης τη δυνατότητα περαιτέρω ανάπτυξης του από εξειδικευμένους προγραμματιστές. Η γλώσσα του Arduino μπορεί να διευρυνθεί κάνοντας χρήση βιβλιοθηκών που είναι γραμμένες σε C++ ενώ παράλληλα δίνεται η δυνατότητα κατανόησης της προγραμματιστικής γλώσσας AVR C στην οποία είναι βασισμένη η γλώσσα του Arduino.
5. Τέλος, έμπειροι σχεδιαστές κυκλωμάτων μπορούν να παρέμβουν στις πλακέτες, να τις επεκτείνουν, ακόμα και να τις βελτιώσουν.

## 2.1 Περιβάλλον εργασίας Arduino

Το περιβάλλον εργασίας του Arduino [2] περιλαμβάνει έναν κειμενογράφο για την εγγραφή κώδικα, μια περιοχή εμφάνισης μηνυμάτων, μια κονσόλα κειμένου, μια γραμμή εργαλείων με κουμπιά που υποστηρίζουν κοινές λειτουργίες και μια σειρά από μενού. Συνδέεται με το Arduino στο οποίο πέρνάει τον κώδικα του προγράμματος και επικοινωνεί με αυτό.

### 2.1.1 Δημιουργία "σκίτσων"

Οι κώδικες που δημιουργούνται με το Arduino ονομάζονται "σκίτσα". Αυτά τα "σκίτσα" γράφονται στον κειμενογράφο και σώζονται με την επέκταση ".ino".

Στην περιοχή εμφάνισης μηνυμάτων εμφανίζονται πληροφορίες κάθε φορά που εκτελείται μια ενέργεια, κατά την αποθήκευση του "σκίτσου" καθώς και κατά την φόρτωσή του στην πλακέτα του Arduino. Επιπλέον εμφανίζονται λάθη που τυχόν υπάρχουν στον κώδικα και λόγοι για τους οποίους μπορεί να μην έγινε σωστά η φόρτωση του "σκίτσου" στο Arduino.

Στην κονσόλα καταγράφεται κείμενο που προέρχεται από την εκτέλεση του προγράμματος καθώς επίσης λεπτομερή λάθη και άλλες πληροφορίες. Στην κάτω δεξιά γωνία του παραθύρου αναγράφεται το εκάστοτε είδος του Arduino και η σειριακή θύρα στην οποία είναι συνδεδεμένο.

Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και τη φόρτωση των προγραμμάτων στο Arduino, τη δημιουργία, την έναρξη και την αποθήκευση "σκίτσων" καθώς και την έναρξη της σειριακής οθόνης.

- *Verify* : Ελέγχει για λάθη στο κώδικα
- *Upload* : Ελέγχει τον κώδικα και τον φορτώνει στο Arduino I/O board.
- *New* : Δημιουργεί καινούριο "σκίτσο".
- *Open* : Ανοίγει την λίστα από τα "σκίτσα" και επιτρέπει την έναρξη του επιλεγμένου "σκίτσου".
- *Save* : Αποθηκεύει το "σκίτσο".

- *Serial Monitor* : Ανοίγει τη σειριακή οθόνη.

Επιπλέον εντολές υπάρχουν στα πέντε μενού:

- File
- Edit
- Sketch
- Tools
- Help

### 2.1.2 Sketchbook

Το περιβάλλον του Arduino χρησιμοποιεί τον τρόπο ταξινόμησης ενός τετραδίου (Sketchbook). Τα προγράμματα και τα "σκίτσα" αποθηκεύονται σε ένα συγκεκριμένο φάκελο (Sketchbook) ο οποίος δημιουργείται αυτόματα την πρώτη φορά που εκτελείται το λογισμικό του Arduino.

### 2.1.3 Tabs, Multiple Files, and Compilation

Επιτρέπουν τη διαχείριση των "σκίτσων" με περισσότερα από ένα αρχεία. Αυτά τα αρχεία μπορεί να είναι γραμμένα σε κώδικα Arduino, σε C ή C++ γλώσσα προγραμματισμού (.c extension, .cpp extension), ή header files (.h).

### 2.1.4 Φόρτωση κώδικα στο Arduino (Uploading)

Πριν τη φόρτωση του προγράμματος στην πλακέτα του Arduino, είναι απαραίτητο να γίνει η σωστή επιλογή της πλακέτας και της θύρας στην οποία είναι συνδεδεμένη. Σε Mac λογισμικό οι σειριακές θύρες μπορεί να είναι: /dev/tty.usbmodem241 (για Arduino Uno, Mega2560 ή Leonardo) ή /dev/tty.usbserial-1B1 (για Arduino Duemilanove ή παλαιότερη USB πλακέτα), ή /dev/tty.USA19QW1b1P1.1. Στα Windows, είναι πιθανότατα οι COM1 ή COM2 (για σειριακές πλακέτες) ή COM4, COM5, COM7 (για USB πλακέτες). Στα Linux, είναι /dev/ttyUSB0, /dev/ttyUSB1 ή παρόμοιες.

Από τη στιγμή που έχει γίνει η σωστή επιλογή πλακέτας και θύρας μπορεί να γίνει η φόρτωση του "σκίτσου" στην πλακέτα του Arduino.

Στις καινούριες πλακέτες γίνεται αυτόματη επαναφορά ενώ στις παλαιότερες είναι απαραίτητη η επαναφορά μέσω ενός κουμπιού (reset button) πριν γίνει η φόρτωση στην πλακέτα. Επιπροσθέτως στις περισσότερες πλακέτες κατά την διάρκεια της φόρτωσης ανάβουν συγκεκριμένα λαμπάκια του Arduino (RT, TX leds). Στο περιβάλλον εργασίας του Arduino εμφανίζονται μηνύματα για την πορεία και ολοκλήρωση της φόρτωσης καθώς και λάθη που τυχόν υπάρχουν.

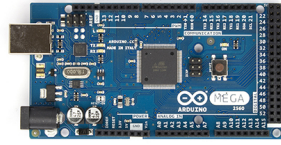
### 2.1.5 Βιβλιοθήκες (Libraries)

Οι βιβλιοθήκες παρέχουν επιπλέον λειτουργίες στα "σκίτσα", όπως για παράδειγμα την επικοινωνία με συσκευές και την διαχείριση των δεδομένων. Μέσα στις βιβλιοθήκες καθορίζεται ο τρόπος με τον οποίο η εκάστοτε συσκευή λαμβάνει και στέλνει σήματα από και προς το Arduino. Οι βιβλιοθήκες αυτές καλούνται στην αρχή του κάθε "σκίτσου" και με βάση τις λειτουργίες που αυτές παρέχουν ο χρήστης είναι σε θέση να τις χρησιμοποιήσει κατάλληλα στον υπόλοιπο κώδικα του "σκίτσου". Προκειμένου να χρησιμοποιηθούν αυτές οι βιβλιοθήκες πρέπει να βρίσκονται στο περιβάλλον του Arduino στο σωστό φάκελο. Πολλές βιβλιοθήκες βρίσκονται στο διαδίκτυο άλλες υπάρχουν ήδη στο περιβάλλον του Arduino, ενώ άλλες μπορούν να γραφτούν και από το χρήστη.

### 2.1.6 Σειριακή οθόνη (Serial Monitor)

Στη σειριακή οθόνη καταγράφονται τα δεδομένα που η πλακέτα του Arduino στέλνει στο περιβάλλον εργασίας. Για να σταλούν τώρα δεδομένα στην πλακέτα, εισάγεται κείμενο στην σειριακή οθόνη και μέσω του κουμπιού "send" φορτώνονται στο Arduino. Για να επιτευχθεί η επικοινωνία μεταξύ πλακέτας και σειριακής οθόνης είναι απαραίτητο να ορισθεί στην σειριακή οθόνη το "baud rate" έτσι ώστε να είναι ίδιο με αυτό που είχε οριστεί στο "σκίτσο" (Serial.begin()).

## 2.2 Πλακέτα Arduino Mega 2560



Σχήμα 2.2: Μπροστινή όψη Arduino Mega



Σχήμα 2.3: Πίσω όψη Arduino Mega

Το Arduino Mega 2560 είναι μια μικροελεγκτική πλακέτα βασισμένη στην πλακέτα ATmega2560. Διαθέτει 54 ψηφιακές εισόδους/εξόδους από τις οποίες οι 15 μπορούν να χρησιμοποιηθούν σαν έξοδοι διαμόρφωσης πλάτους παλμού Pulse-width modulation (PWM), 16 αναλογικές εισόδους, 4 σειριακές θύρες Universal Asynchronous Receiver/Transmitter (UARTs), έναν κρυσταλλικό ταλαντωτή 16 MHz (crystal oscillator), μια USB θύρα επικοινωνίας, μια υποδοχή ρεύματος, μια επαφή In-circuit serial programming (ICSP) και ένα κουμπί επαναφοράς. Περιλαμβάνει ό,τι είναι απαραίτητο για να υποστηρίξει τον μικροελεγκτή.

### 2.2.1 Παροχή Ενέργειας (Power)

Στο Arduino Mega μπορεί να παρασχεθεί ενέργεια μέσω της σύνδεσής του με υπολογιστή μέσω USB καλωδίου ή με εξωτερική πηγή ενέργειας. Η παροχή ενέργειας μέσω εξωτερικής πηγής μπορεί να προέρχεται είτε από μετασχηματιστή AC σε DC είτε από μπαταρία. Ο μετασχηματιστής συνδέεται με το Arduino μέσω καλωδίου που καταλήγει στην υποδοχή ρεύματος της πλακέτας. Η πλακέτα μπορεί να λειτουργήσει με εξωτερική παροχή ενέργειας από 6 έως 20 volts. Αν η παροχή ενέργειας είναι μικρότερη από 7V, η επαφή της πλακέτας που παρέχει 5V, μπορεί να δίνει λιγότερα και η πλακέτα να είναι γενικότερα ασταθής. Αν τώρα η παροχή ενέργειας είναι μεγαλύτερη από 12V, ο

ρυθμιστής τάσης της πλακέτας μπορεί να υπερθερμανθεί και να καταστρέψει την ίδια την πλακέτα. Το συνιστώμενο εύρος παροχής τάσης είναι 7 έως 12 volts.

Οι επαφές παροχής τάσης (headers) της πλακέτας είναι οι παρακάτω:

**VIN:** Είναι η τάση εισόδου στην πλακέτα του Arduino όταν γίνεται χρήση εξωτερικής πηγής τάσης. Η επαφή αυτή μπορεί να χρησιμοποιηθεί για παροχή τάσης.

**5V:** Αυτή η επαφή εξόδου παρέχει 5V που προέρχονται από το ρυθμιστή τάσης. Στην πλακέτα του Arduino μπορεί να παρασχεθεί ενέργεια είτε από DC υποδοχή ρεύματος (7 - 12V) είτε από USB σύνδεση (5V) είτε μέσω της επαφής VIN της πλακέτας (7 - 12V). Η παροχή τάσης στην πλακέτα μέσω των επαφών 5V ή 3.3V παρακάμπτει τον ρυθμιστή τάσης και μπορεί να καταστρέψει την πλακέτα, για το λόγο αυτό δεν συνίσταται.

**3V3:** Η επαφή αυτή παρέχει 3.3 volt τα οποία προέρχονται από το ρυθμιστή τάσης της πλακέτας. Η μέγιστη τιμή ρεύματος είναι τα 50mA.

**GND:** Αποτελούν τις γειωμένες επαφές.

**IOREF:** Αυτή η επαφή της πλακέτας του Arduino παρέχει την τάση αναφοράς με την οποία λειτουργεί η πλακέτα. Μία πολύ σωστά ρυθμισμένη έξτρα πλακέτα μπορεί να διαβάσει την τάση που παρέχει η συγκεκριμένη επαφή και να επιλέξει την πηγή ενέργειας ή να ενεργοποιήσει τους ρυθμιστές τάσης στις εξόδους των 5 και 3.3 V.

### 2.2.2 Μνήμη (Memory)

Το Arduino Mega 2560 διαθέτει 256KB μνήμης για την αποθήκευση κώδικα, 8 KB μνήμης Static random-access memory (SRAM) και 4 KB μνήμης Electrically Erasable Programmable Read-Only Memory (EEPROM).

### 2.2.3 Είσοδος και Έξοδος (Input and Output)

Κάθε μια από τις 54 ψηφιακές επαφές του Arduino Mega μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος μέσω των εντολών pinMode(), digitalWrite(), και digitalRead(). Λειτουργούν όλες στα 5 volts. Κάθε επαφή μπορεί να δεχτεί ή να παρέχει ένα μέγιστο των 40mA και διαθέτει μια εσωτερική αντίσταση 20-50 kOhms (προεπιλεγμένα αποσυνδεδεμένη). Επίσης κάποιες επαφές διαθέτουν μερικές επιπλέον λειτουργίες:

**Serial: 0 (RX) and 1 (TX), Serial 1: 19 (RX) and 18 (TX), Serial 2: 17 (RX) and 16 (TX), Serial 3: 15 (RX) and 14 (TX):** Χρησιμοποιούνται για να δέχονται (RX) και να μεταφέρουν (TX) Transistor–transistor logic (TTL) σειριακά δεδομένα. Οι επαφές 0 και 1 είναι επίσης συνδεδεμένες με τις αντίστοιχες επαφές του ATmega16U2 USB-to-TTL Serial chip.

**External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2):** Αυτές οι επαφές μπορούν να ρυθμιστούν έτσι ώστε να προκαλούν διακοπή στις χαμηλές τιμές, ανοδικά ή καθοδικά άκρα, ή αλλαγή σε κάποια τιμή. Γίνεται χρήση της συνάρτησης `attachInterrupt()`.

**PWM: 2 to 13 and 44 to 46:** Αυτές οι επαφές προσφέρουν 8-bit εξόδους διαμόρφωσης πλάτους παλμού (PWM) μέσω της συνάρτησης `analogWrite()`.

**SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS):** Οι συγκεκριμένες επαφές υποστηρίζουν Serial Peripheral Interface (SPI) επικοινωνία μέσω της χρήσης SPI βιβλιοθηκών. Οι επαφές SPI are also broken out on the στην ICSP επαφή η οποία είναι συμβατή με τα Arduino Uno, Duemilanove και Diecimila.

**LED: 13:** Πάνω στην πλακέτα του Arduino Mega υπάρχει ενσωματωμένο λαμπάκι (LED) συνδεδεμένο με τη ψηφιακή επαφή 13. Όταν στην επαφή αυτή η τιμή είναι υψηλή το LED είναι αναμμένο, ενώ όταν η τιμή είναι χαμηλή το LED σβήνει.

**TWI: 20 (SDA) and 21 (SCL):** Οι επαφές αυτές υποστηρίζουν Two Wire Interface (TWI) επικοινωνία χρησιμοποιώντας τη βιβλιοθήκη Wire. Σημειώνεται εδώ ότι οι επαφές TWI δεν βρίσκονται στο ίδιο σημείο στην πλακέτα του Mega σε σχέση με το σημείο που βρίσκονται στις Duemilanove ή Diecimila πλακέτες.

Το Arduino Mega2560 διαθέτει 16 αναλογικές εισόδους καθεμιά από τις οποίες παρέχει 10 bits ανάλυση (1024 διαφορετικές τιμές). Από προεπιλογή μετρούν από γείωση έως 5 volts, υπάρχει όμως η δυνατότητα αλλαγής του μεγίστου και του ελαχίστου του εύρους χρησιμοποιώντας την AREF επαφή και την `analogReference()` συνάρτηση.

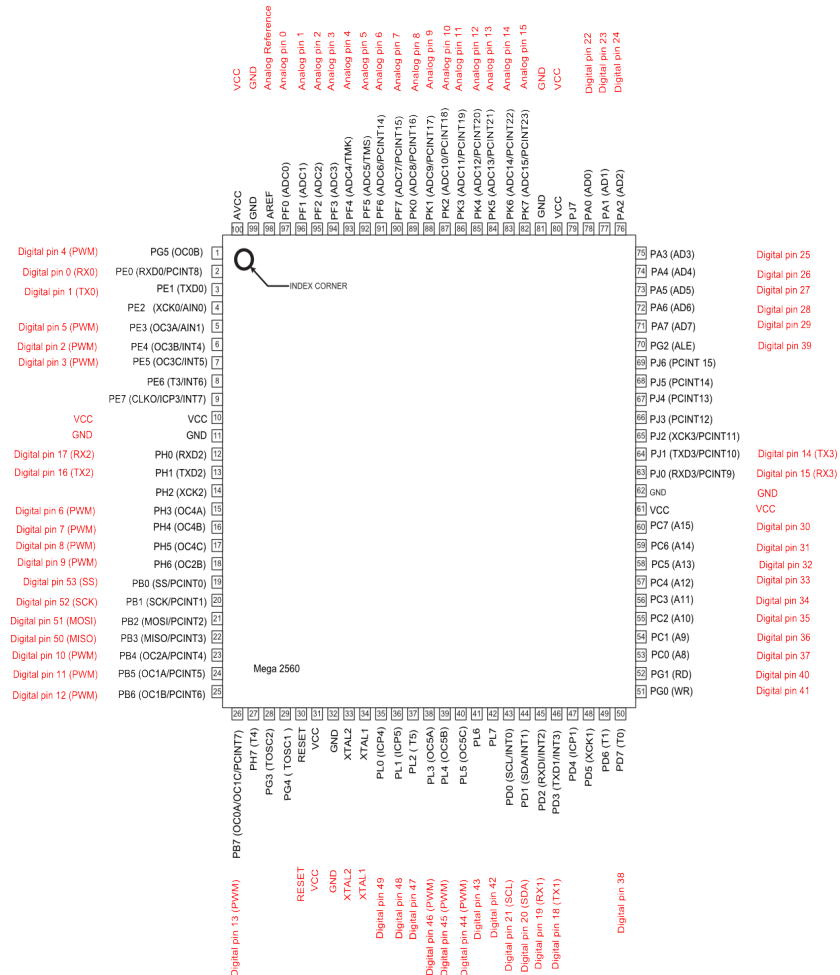
Υπάρχουν πολλές ακόμη επαφές πάνω στην πλακέτα:

**AREF:** Επαφή για την τάση αναφοράς των αναλογικών εισόδων. Όπως προαναφέρθηκε χρησιμοποιείται μαζί με την συνάρτηση `AnalogReference()`.

**Reset:** Κατεβάζει τη γραμμή χαμηλά ώστε να γίνει η επαννεκίνηση του μικροελεγκτή. Τυπικά χρησιμοποιείται για να προσθέσει ένα reset κουμπί σε έξτρα πλακέτες οι οποίες μπλοκάρουν τη λειτουργία του ήδη υπάρχοντος reset κουμπιού στην πλακέτα.

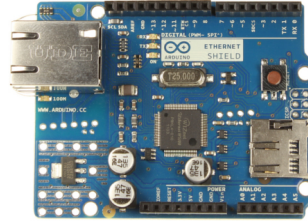


## 2.2.4 ATmega2560-Arduino Pin Mapping

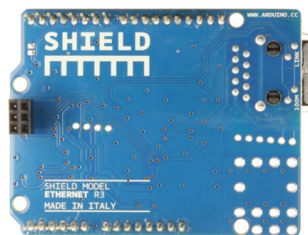


Σχήμα 2.4: Διάγραμμα επαφών Arduino Mega

## 2.3 Πλακέτα Arduino Ethernet Shield



Σχήμα 2.5: Μπροστινή Όψη του Ethernet Shield



Σχήμα 2.6: Πίσω Όψη του Ethernet Shield

Το Arduino Ethernet Shield παρέχει τη δυνατότητα σύνδεσης της πλακέτας του Arduino στο διαδίκτυο μέσα σε λίγα λεπτά. Αρχικά γίνεται σύνδεση του Ethernet Shield με την πλακέτα του Arduino Mega και στη συνέχεια συνδέεται στο διαδίκτυο μέσω ενός RJ45 καλωδίου. Όπως συμβαίνει πάντα με το Arduino, τόσο οι πλακέτες όσο και το λογισμικό, είναι στη διάθεση κάθε χρήστη ανά πάσα στιγμή. Αυτό σημαίνει ότι όλοι οι χρήστες του Arduino έχουν την δυνατότητα να το μελετήσουν και να μάθουν πως έχει σχεδιαστεί, γεγονός που αποτελεί εναρκτήριο σημείο για τον σχεδιασμό περαιτέρω κυκλωμάτων. Για να λειτουργήσει το Ethernet Shield είναι απαραίτητη μια πλακέτα Arduino, τάση 5V (η οποία παρέχεται από την πλακέτα), Ethernet Controller: W5100 with internal 16K buffer, ταχύτητα σύνδεσης στο διαδίκτυο 10/100Mb και σύνδεση με την πλακέτα του Arduino στην SPI θύρα.

### 2.3.1 Περιγραφή Ethernet Shield

Όπως προαναφέρθηκε το Arduino Ethernet Shield επιτρέπει στην πλακέτα του Arduino να συνδεθεί στο διαδίκτυο. Είναι βασισμένο στο Wiznet W5100 ethernet chip. Το Wiznet W5100 παρέχει μια στοίβα δικτύου (network (IP) stack) που υποστηρίζει τα πρωτόκολλα επικοινωνίας TCP (Transmission Control Protocol - Πρωτόκολλο Ελέγχου Μεταφοράς) και UDP (User Datagram Protocol). Υποστηρίζει μέχρι και τέσσερις ταυτόχρονες συνδέσεις υποδοχής. Για να επιτευχθεί η σύνδεση στο διαδίκτυο είναι απαραίτητη η χρήση της βιβλιοθήκης Ethernet library στον κώδικα του "σκίτσου". Το ethernet shield συνδέεται με τη πλακέτα του Arduino μέσω μεγάλων συρμάτων επαφών που εισέρχονται στις υποδοχές της πλακέτας. Με αυτό τον τρόπο οι επαφές του ethernet shield είναι ελεύθερες να συνδεθούν ακόμη και με επιπλέον ethernet shield ακριβώς απο πάνω.

Το Ethernet Shield έχει μια πρότυπη RJ-45 επικοινωνία, με ενσωματωμένο μετασχηματιστή γραμμής και με την παροχή ενέργειας μέσω Ethernet ενεργοποιημένη.

Πάνω στο Ethernet Shield υπάρχει ενσωματωμένη υποδοχή κάρτας μνήμης (micro-SD card), η οποία μπορεί να χρησιμοποιηθεί για την αποθήκευση αρχείων. Είναι συμβατή με το Arduino Uno και Mega (με χρήση πάντα της Ethernet library). Η ενσωματωμένη συσκευή ανάγνωσης καρτών microSD είναι προσβάσιμη μέσω της βιβλιοθήκης SD.

Το Ethernet Shield περιλαμβάνει επίσης έναν ελεγκτή επαναφοράς (reset controller), ώστε να εξασφαλίζεται ότι στο W5100 Ethernet module έχει γίνει σωστά η επαναφορά κατά την εκκίνηση. Προηγούμενες εκδόσεις του Ethernet Shield δεν ήταν συμβατές με το Arduino Mega και για το λόγο αυτό ήταν απαραίτητη η με το χέρι επαναφορά μετά την εκκίνηση.

Το Arduino επικοινωνεί με το W5100 Ethernet module και με την SD κάρτα χρησιμοποιώντας την επικοινωνία SPI. Αυτή επιτυγχάνεται μέσω των ψηφιακών επαφών 10, 11, 12 και 13 στο Arduino Uno και των επαφών 50, 51 και 52 στο Arduino Mega. Και για τις δυο πλακέτες Uno και Mega, η επαφή 10 χρησιμοποιείται για την επιλογή του W5100, ενώ η επαφή 4 για την SD κάρτα. Για το λόγο αυτό οι συγκεκριμένες επαφές δεν μπορούν να χρησιμοποιηθούν σαν είσοδοι ή έξοδοι. Στο Arduino Mega η επαφή 53 δεν μπορεί να χρησιμοποιηθεί για την επιλογή του W5100 ή της SD κάρτας αλλά πρέπει να δηλωθεί σαν έξοδος, διαφορετικά η διεπαφή SPI δεν θα λειτουργήσει.

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί εδώ το γεγονός ότι επειδή το W5100 και η SD κάρτα μοιράζονται την SPI επικοινωνία

νία, μόνο το ένα από τα δύο μπορεί να είναι ενεργό κάθε φορά. Αυτό πρέπει να λαμβάνεται υπ' όψη στις βιβλιοθήκες στην περίπτωση που χρησιμοποιούνται στο κώδικα το W5100 και η SD κάρτα ταυτόχρονα. Αν δεν γίνεται χρήση του ενός εκ των δυο (κάρτας ή W5100) αυτό που δεν χρησιμοποιείται πρέπει να καταργηθεί. Για την κάρτα η επαφή 4 ορίζεται σαν έξοδος υψηλής απόδοσης, ενώ για το W5100 η ψηφιακή επαφή 10 είναι αυτή που ορίζεται σαν έξοδος υψηλής απόδοσης.

Το Ethernet Shield παρέχει υποδοχή RJ45 ethernet καλωδίου ενώ το κουμπί επαναφοράς (reset button) το οποίο διαθέτει επαναφέρει τόσο την πλακέτα του Arduino όσο και το W5100.

Επιπλέον το Ethernet Shield περιλαμβάνει μια σειρά από ενημερωτικά λαμπάκια (LEDs):

**PWR:** Καταδεικνύει ότι η πλακέτα του Arduino και το Ethernet Shield τροφοδοτούνται.

**LINK:** Δείχνει την παρουσία μιας σύνδεσης δικτύου και αναβοσβήνει όταν το Ethernet Shield μεταδίδει ή λαμβάνει δεδομένα.

**FULLD:** Υποδεικνύει ότι η σύνδεση δικτύου είναι πλήρως αμφίδρομη (full duplex).

**100M:** Φανερώνει την παρουσία μιας σύνδεσης δικτύου 100 Mb/s.

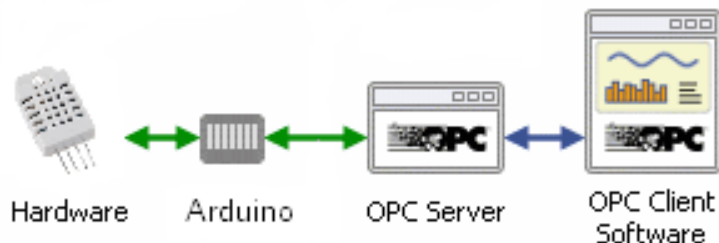
**RX:** Αναβοσβήνει όταν το Ethernet Shield λαμβάνει δεδομένα.

**TX:** Αναβοσβήνει όταν το Ethernet Shield στέλνει δεδομένα.

**COLL:** Αναβοσβήνει όταν ανιχνεύονται συγκρούσεις δικτύου.

## 2.4 OPC Server για Arduino

### 2.4.1 Λογισμικό διασύνδεσης OPC



Σχήμα 2.7: Διάγραμμα OPC Επικοινωνίας

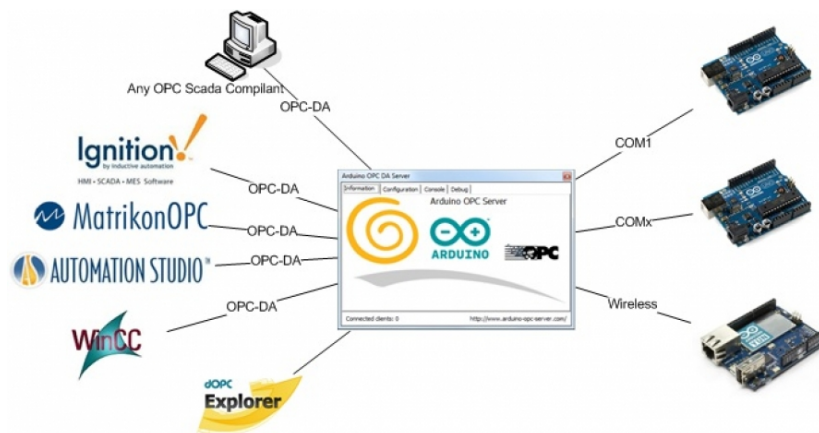
Ως OPC ορίζεται το λογισμικό διασύνδεσης που επιτρέπει σε προγράμματα των Windows να επικοινωνούν με διάφορες συσκευές. Το ακρωνύμιο "OPC" προέρχεται από το "OLE (Object Linking and Embedding) for Process Control". Η τεχνολογία OLE είναι βασισμένη στο μοντέλο COM (Component Object Model) των Windows, και είναι και αυτή COM. Μέσα σε ένα δίκτυο, το λογισμικό OPC βασίζεται στην τεχνολογία DCOM (Distributed COM), η οποία δεν είχε σχεδιαστεί για εφαρμογές πραγματικού χρόνου και γι' αυτό συχνά παραμερίζεται δίνοντας τη θέση της στο OPC tunnelling.

Το λογισμικό OPC υλοποιείται σε ζεύγη διακομιστή / πελάτη (OPC Server / OPC Client). Ο διακομιστής OPC (OPC server) είναι ένα πρόγραμμα λογισμικού που μετατρέπει το πρωτόκολλο επικοινωνίας μιας συσκευής, που χρησιμοποιείται από ένα PLC (Programmable Logic Controller: είναι ένας μικρός υπολογιστής που ελέγχει μία ή περισσότερες συσκευές), στο πρωτόκολλο OPC. Ο πελάτης OPC (OPC Client) μπορεί να είναι οποιοδήποτε πρόγραμμα το οποίο χρειάζεται να συνδεθεί με μια συσκευή. Επιπλέον ο πελάτης OPC χρησιμοποιεί τον διακομιστή OPC προκειμένου να κάνει λήψη δεδομένων ή αποστολή εντολών σε συσκευές.

Η μεγάλη αξία του λογισμικού OPC έγκυται στο γεγονός ότι αποτελεί ένα ανοιχτό πρότυπο, πράγμα που συνεπάγεται σε χαμηλότερο κόστος για τους κατασκευαστές και μεγαλύτερο εύρος επιλογών για τους χρήστες. Όσον αφορά τους κατασκευαστές, το μόνο που χρειάζεται να κάνουν είναι να παρέχουν μαζί με την συσκευή έναν μοναδικό

OPC διακομιστή ο οποίος να μπορεί να επικοινωνήσει με οποιονδήποτε OPC πελάτη. Οι προμηθευτές λογισμικού συμπεριλαμβάνουν στα προϊόντα τους τη χρήση ή την δημιουργία OPC πελατών κάνοντας τα συμβατά με χιλιάδες συσκευές. Στην πλευρά των χρηστών τώρα, τα πράγματα είναι ευκολότερα αφού μπορούν να επιλέξουν οποιοδήποτε λογισμικό OPC πελάτη χρειάζονται, όντας σίγουροι ότι αυτό θα συνδεθεί με τη διαθέσιμη OPC συσκευή και αντίστροφα.

### 2.4.2 Διακομιστής OPC για Arduino



Σχήμα 2.8: OPC Server για Arduino

Ο OPC Server for Arduino [3] αποτελεί έναν OPC διακομιστή ο οποίος διατίθεται δωρεάν, είναι εύκολος στη χρήση και στην παραμετροποίηση και παρέχει τη δυνατότητα σύνδεσης μιας πλακέτας Arduino με οποιοδήποτε (OPC Scada Compliant). Μέσω του OPC Server for Arduino δίνεται η δυνατότητα στο χρήστη να λάβει και να στείλει πληροφορίες από και προς το λογισμικό SCADA HMI και να δημιουργήσει περαιτέρω εφαρμογές χωρίς κόστος.

### 2.4.3 Χαρακτηριστικά

1. Χρήση πολλαπλών πλακετών Arduino. Με τη χρήση μόνο ενός OPC Server for Arduino μπορεί να γίνει λήψη δεδομένων απο διαφορετικές πλακέτες Arduino όπως Arduinos YUN, UNO, MEGA κ.τ.λ.
2. Ταυτόχρονη χρήση διαφορετικών σειριακών θυρών.
3. Μεγάλη ταχύτητα και ισχύ.

4. Συμβατός με JSON (JavaScript Object Notation) όταν γίνεται χρήση του Arduino YUN. Μπορεί να χρησιμοποιηθεί οποιοδήποτε πρόγραμμα περιήγησης στο διαδίκτυο προκειμένου να γίνει λήψη δεδομένων από το Arduino.

Για να λειτουργήσει ο OPC Server for Arduino σε συνδιασμό με την πλακέτα του Arduino είναι απαραίτητη η χρήση της βιβλιοθήκης OPC.h στον κώδικα του "σχίτσου". Στην συνέχεια πρέπει να οριστούν τα αντικείμενα "items" και να γίνει σύνδεσή τους με τις εκάστοτε μετρήσεις που κάθε αντικείμενο θα δημοσιεύει σε πραγματικό χρόνο.



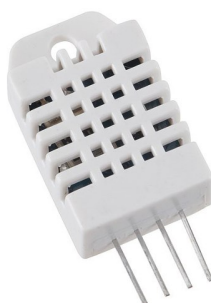


## Κεφάλαιο 3

# ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΜΕΤΡΗΣΗ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΑΙ ΥΓΡΑΣΙΑΣ

### 3.1 Αισθητήρες θερμοκρασίας και υγρασίας, DHT22 - DHT11

#### 3.1.1 Αισθητήρες DHT22



Σχήμα 3.1: Αισθητήρας θερμοκρασίας και υγρασίας DHT22

Οι αισθητήρες DHT22 [4], οι οποίοι ονομάζονται και AM2303, αποτελούν όργανα για τη μέτρηση της θερμοκρασίας και της σχετικής υγρασίας. Παρέχουν βαθμονομημένο ψηφιακό σήμα και εξαιρετική μακροπρόθεσμη σταθερότητα. Η δυνατότητα μεταφοράς του σήματος σε μεγάλες αποστάσεις (20m) καθώς και η χαμηλή κατανάλωση ενέργειας

είναι μερικά από τα σημαντικά πλεονεκτήματα που διαθέτουν. Η παροχή τάσης που απαιτούν κυμαίνεται μεταξύ 3.3-6V DC. Σε ορισμένες περιπτώσεις που τα δεδομένα τα οποία προκύπτουν στην έξοδο έχουν κάποια απόκλιση από τις κανονικές συνθήκες, ή παρουσιάζουν μεγάλες διακυμάνσεις, συνίσταται η τοποθέτηση ενός πυκνωτή 100nF ανάμεσα στην είσοδο και στην γείωση, προκειμένου να φιλτραριστεί το σήμα.

### Τεχνικά χαρακτηριστικά

**Μοντέλο:** DHT22

**Παροχή Ενέργειας:** 3.3-6V DC

**Σήμα Εξόδου:** Ψηφιακό σήμα εξόδου

**Αισθητήρια Στοιχεία:** Πολυμερικός πυκνωτής υγρασίας και DS18B20 για την ανίχνευση της θερμοκρασίας

**Έυρος Μέτρησης:** Υγρασία: 0-100%RH, Θερμοκρασία: -40-125Celsius

**Ακρίβεια:** Υγρασία:  $\pm 2\%RH$  (Max  $\pm 5\%RH$ ), Θερμοκρασία:  $\pm 0.2Celsius$

**Ευαισθησία:** Υγρασία: 0.1%RH, Θερμοκρασία: 0.1Celsius

**Επαναληψιμότητα:** Υγρασία:  $\pm 1\%RH$ , Θερμοκρασία:  $\pm 0.2Celsius$

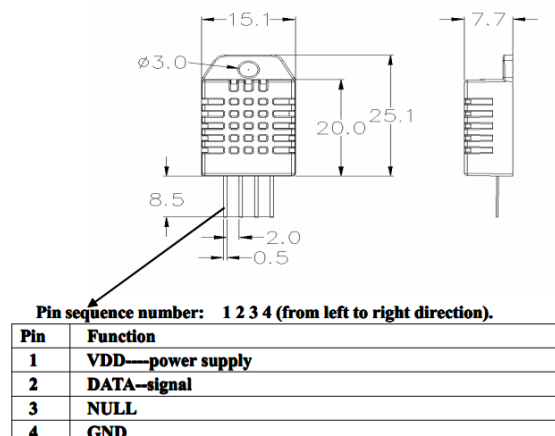
**Υστέρηση Υγρασίας:**  $\pm 0.3\%RH$

**Μακροπρόθεσμη Σταθερότητα:**  $\pm 0.5\%RH/year$

**Αισθητήρια Διάρκεια:** Μέσος όρος: 2s

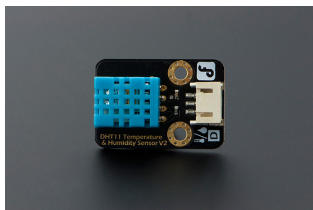
**Εναλλαξιμότητα:** Πλήρης

### Διαστάσεις



Σχήμα 3.2: Διαστάσεις αισθητήρα DHT22

### 3.1.2 Αισθητήρες DHT11



Σχήμα 3.3: Αισθητήρας θερμοκρασίας και υγρασίας DHT11

Οι αισθητήριες DHT11 [5] ανήκουν στην ίδια οικογένεια αισθητήρων με τους DHT22 και αποτελούν και αυτοί όργανα για τη μέτρηση της θερμοκρασίας και της απόλυτης υγρασίας. Η έξοδός τους δίνει ψηφιακό σήμα το οποίο έχει βραχυπρόθεσμη σταθερότητα. Όπως και οι DHT22 μπορούν να μεταφέρουν το σήμα σε μεγάλη απόσταση και καταναλώνουν ελάχιστη ενέργεια. Παρουσιάζουν όμως μερικά μειονεκτήματα αν συγκριθούν με τους DHT22, όπως είναι η μικρότερη ακρίβεια μέτρησης, το μικρότερο εύρος και η μικρότερη σταθερότητα. Η παροχή τάσης κυμαίνεται μεταξύ 3-5.5V DC. Όπως και στους DHT22 μπορεί να χρειαστεί να τοποθετηθεί ένας πυκνωτής 100nF ανάμεσα στην είσοδο και στην γείωση.

#### Τεχνικά χαρακτηριστικά

**Μοντέλο:** DHT11

**Παροχή Ενέργειας:** 3.3-5.5V DC

**Σήμα Εξόδου:** Ψηφιακό σήμα εξόδου

**Αισθητήρια Στοιχεία:** Πολυμερική αντίσταση

**Έυρος Μέτρησης:** Υγρασία: 20-90%RH, Θερμοκρασία: 0-50Celsius

**Ακρίβεια:** Υγρασία:  $\pm 4\%RH$  (Max  $\pm 5\%RH$ ), Θερμοκρασία:  $\pm 2.0Celsius$

**Ευαισθησία:** Υγρασία: 1%RH, Θερμοκρασία: 0.1Celsius

**Επαναληψιμότητα:** Υγρασία:  $\pm 1\%RH$ , Θερμοκρασία:  $\pm 1Celsius$

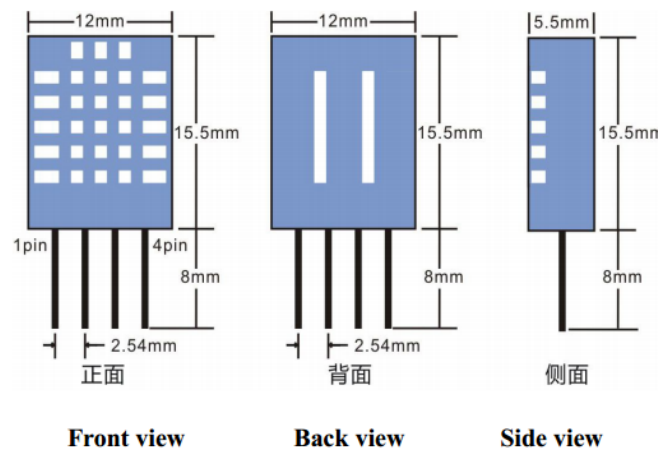
**Υστέρηση Υγρασίας:**  $\pm 1\%RH$

**Μακροπρόθεσμη Σταθερότητα:**  $\pm 0.5\%RH/year$

**Αισθητήρια Διάρκεια:** Μέσος όρος: 2s

**Εναλλαξιμότητα:** Πλήρης

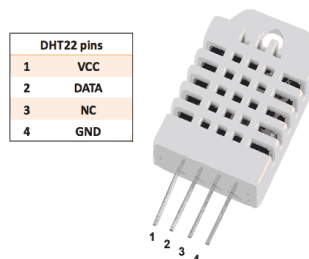
### Διαστάσεις



Σχήμα 3.4: Διαστάσεις αισθητήρα DHT11

## 3.2 Σύνδεση πλακέτας Arduino με αισθητήρες

### 3.2.1 Σύνδεση Arduino με αισθητήρες DHT22



Σχήμα 3.5: Κεφαλές αισθητήρα DHT22

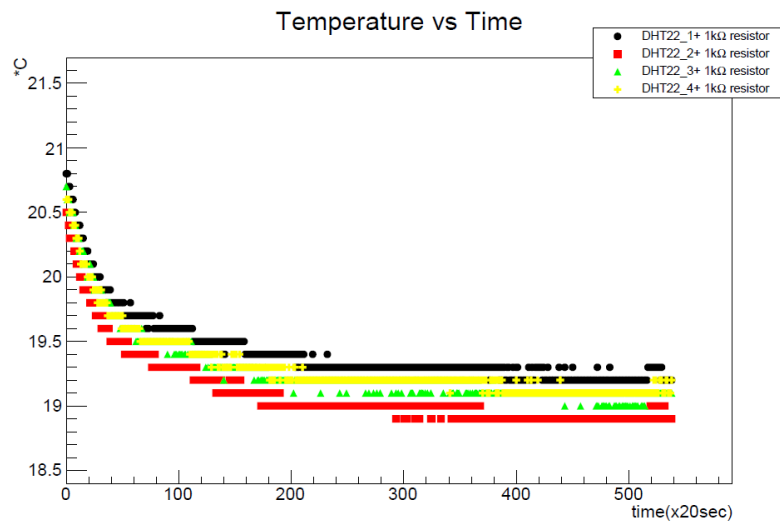
Για να δουλέψει ο αισθητήρας DHT22 και να γίνει λήψη μετρήσεων, πρέπει να συνδεθεί σωστά στο Ethernet Shield το οποίο με τη σειρά του έχει συνδεθεί με την πλακέτα του Arduino Mega. Για το λόγο αυτό η σύνδεση από τα αριστερά προς τα δεξιά έγινε ως εξής: Η πρώτη επαφή συνδέθηκε με τη επαφή 5V από το Ethernet Shield, η οποία παρέχει τάση στον αισθητήρα. Η δεύτερη επαφή του αισθητήρα συνδέθηκε με την ψηφιακή έξοδο 2 του Ethernet Shield, την οποία μέσω του κώδικα που γράφτηκε στο "σκίτσο" ο χρήστης διαβάζει και παίρνει την τιμή της θερμοκρασίας και της σχετικής υγρασίας. Η τρίτη επαφή δεν συνδέεται

με το Ethernet Shield. Η τέταρτη επαφή συνδέεται με τη γείωση που παρέχει το Ethernet Shield.

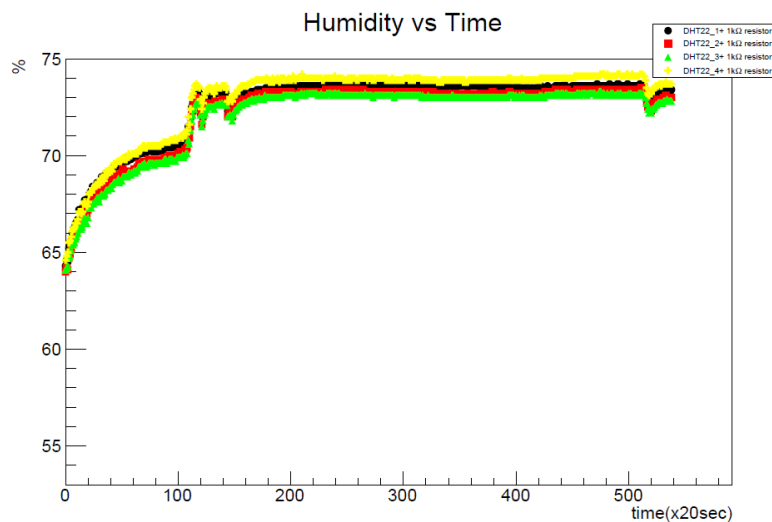
Αφού έγινε η συνδεσμολογία, γραφτηκε ο κώδικας στο "σκίτσο", το οποίο στη συνέχεια φορτώθηκε στην πλακέτα του Arduino. Ο αρχικός κώδικας ο οποίος γράφτηκε "διάβαζει" τις τιμές της θερμοκρασίας και της σχετικής υγρασίας ανά 20 second, τις τυπώνει στην σειριακή οθόνη και τις γράφει σε ένα αρχείο .txt το οποίο αποθηκεύεται στην SD κάρτα, η οποία τοποθετήθηκε και αυτή στην υποδοχή του Ethernet Shield. Επειδή οι τιμές που προέκυψαν αρχικά είχαν σχετικά μεγάλες διακυμάνσεις χρησιμοποιήθηκαν διαφορετικές αντιστάσεις και ένας πυκνωτής ως φίλτρα, έτσι ώστε οι διακυμάνσεις να περιοριστούν και το σήμα να σταθεροποιηθεί και να του αφαιρεθεί τυχόν θόρυβος. Για το λόγο αυτό έγιναν πέντε πειραματικές μετρήσεις. Σε κάθε μέτρηση χρησιμοποιήθηκαν τέσσερις αισθητήρες DHT22 το φίλτρο των οποίων σε κάθε μέτρηση άλλαζε. Για κάθε αισθητήρα DHT22 ίσχυσαν τα παρακάτω.

Στην πρώτη μέτρηση χρησιμοποιήθηκε μια αντίσταση 1kΩ το ένα άκρο της οποίας συνδέθηκε σε 5V παροχή τάσης και το άλλο στην επαφή 2 δηλαδή στην επαφή εξόδου του αισθητήρα. Οι υπόλοιπες επαφές του αισθητήρα συνδέθηκαν όπως αναφέρθηκε προηγουμένως. Στην δεύτερη μέτρηση η αντίσταση άλλαξε σε 4.7 KΩ και στην τρίτη σε 10 KΩ.

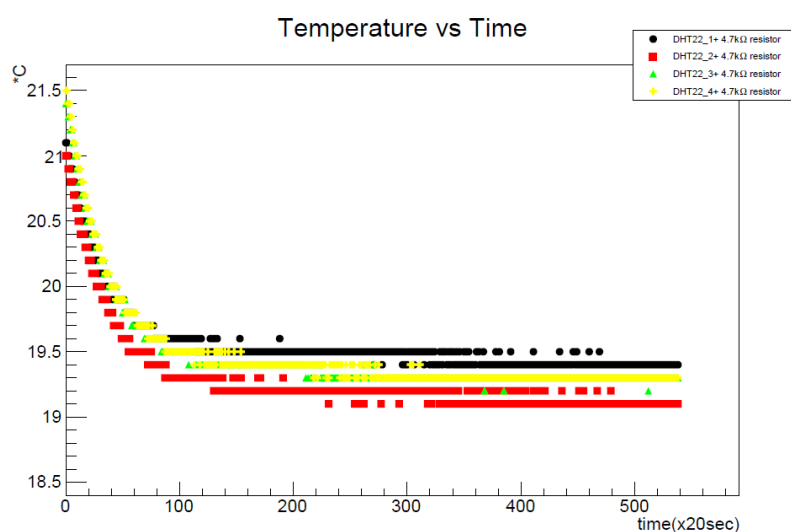
Μέσω του κώδικα του "σκίτσου" οι τιμές της θερμοκρασίας και της σχετικής υγρασίας κάθε αισθητήρα καταγράφηκαν ανά 20 second για τρεις ώρες σε αρχείο .txt απο το οποίο στην συνέχεια μέσω του Root προέκυψαν οι παρακάτω γραφικές παραστάσεις.



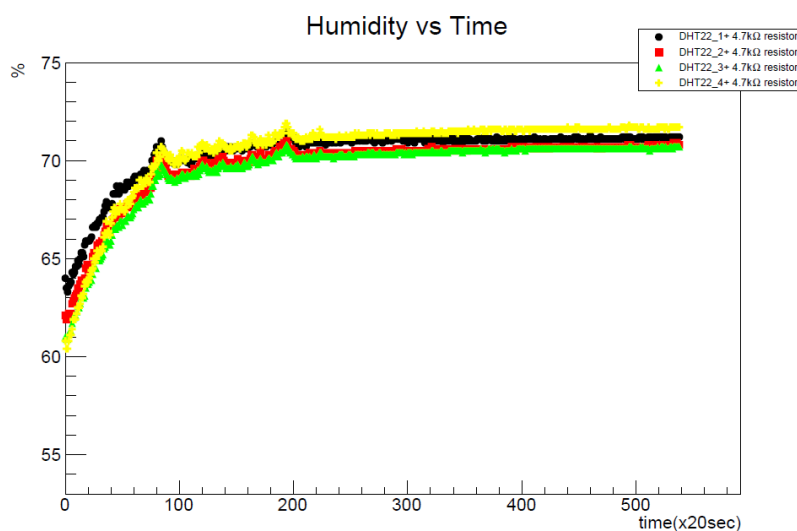
Σχήμα 3.6: Διάγραμμα θερμοκρασιών - χρόνου DHT22 με 1kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22\_1 + resistor, με κόκκινο του DHT22\_2 + resistor, με πράσινο του DHT22\_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22\_4 + resistor αισθητήρα.



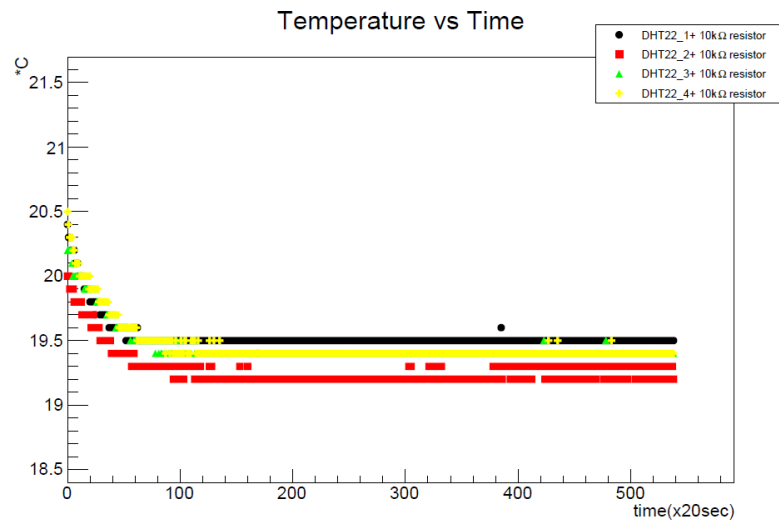
Σχήμα 3.7: Διάγραμμα υγρασιών - χρόνου DHT22 με 1kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22\_1 + resistor, με κόκκινο του DHT22\_2 + resistor, με πράσινο του DHT22\_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22\_4 + resistor αισθητήρα.



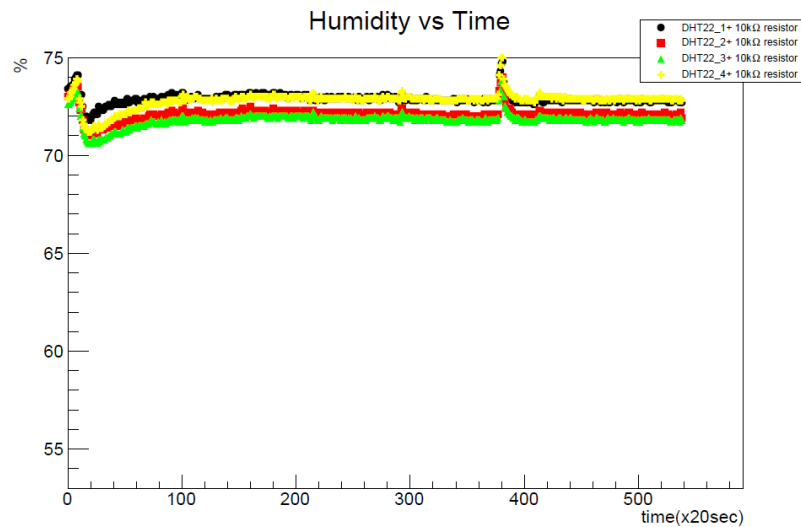
Σχήμα 3.8: Διάγραμμα θερμοκρασιών - χρόνου DHT22 με 4.7kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22\_1 + resistor, με κόκκινο του DHT22\_2 + resistor, με πράσινο του DHT22\_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22\_4 + resistor.



Σχήμα 3.9: Διάγραμμα υγρασιών - χρόνου DHT22 με 4.7kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22\_1 + resistor, με κόκκινο του DHT22\_2 + resistor, με πράσινο του DHT22\_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22\_4 + resistor.



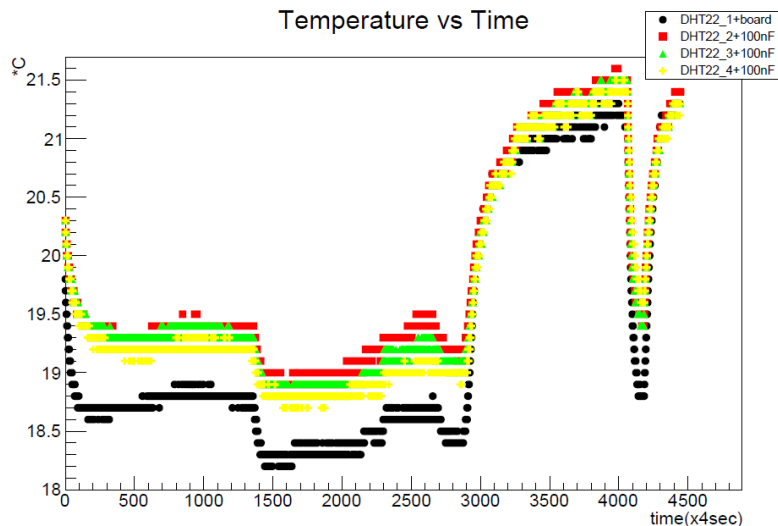
Σχήμα 3.10: Διάγραμμα θερμοκρασιών - χρόνου DHT22 με 10kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22\_1 + resistor, με κόκκινο του DHT22\_2 + resistor, με πράσινο του DHT22\_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22\_4 + resistor.



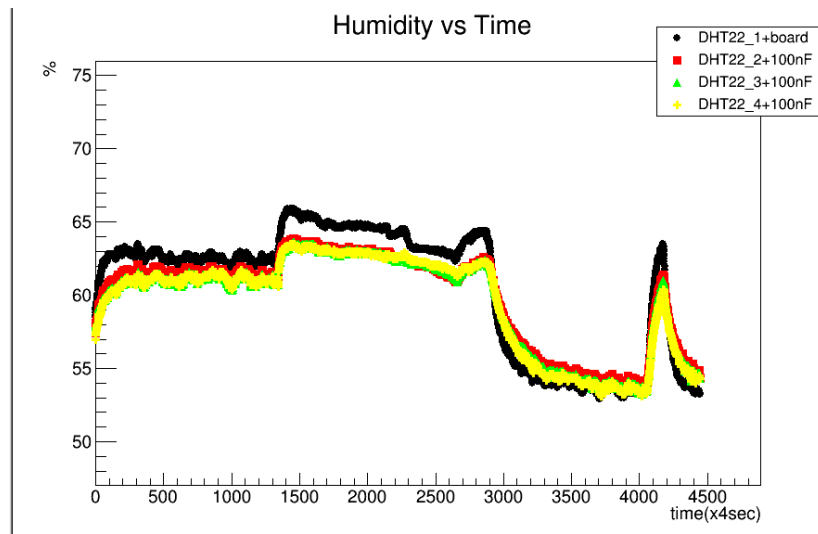
Σχήμα 3.11: Διάγραμμα υγρασιών - χρόνου DHT22 με 10kohm αντίσταση. Με μαύρο χρώμα σημειώνονται οι μετρήσεις του DHT22\_1 + resistor, με κόκκινο του DHT22\_2 + resistor, με πράσινο του DHT22\_3 + resistor και τέλος με κίτρινο οι μετρήσεις του DHT22\_4 + resistor.



Στην τέταρτη μέτρηση που διήρκησε περισσότερο από τις άλλες, στους τρεις εκ των τεσσάρων αισθητήρων χρησιμοποιήθηκε ένας πυκνωτής 100nF ο οποίος συνδέθηκε μεταξύ της επαφής 1 και 4, δηλαδή μεταξύ της επαφής εισόδου και της γείωσης. Ενώ στον έναν χρησιμοποιήθηκε ένα φίλτρο που αποτελείται από μια αντίσταση 10kΩ μεταξύ της επαφής 1 και 2 και έναν πυκνωτή 100nF μεταξύ της επαφής 1 και 4. Επιπλέον σε αυτή τη μέτρηση οι τιμές καταγράφονταν ανα 4 sec. Να σημειωθεί εδώ ότι το φίλτρο RC το οποίο χρησιμοποιήθηκε είναι ενσωματωμένο πάνω σε μία μικρή πλακέτα και για το λόγο αυτό αναφέρεται παρακάτω ως board.

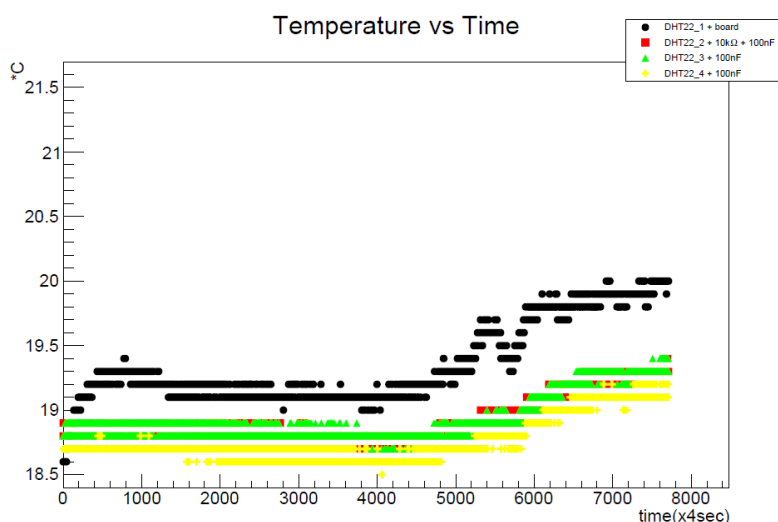


Σχήμα 3.12: Διάγραμμα θερμοκρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις του αισθητήρα DHT22\_1 που συνδέεται με το board, (πυκνωτή 100nF και αντίσταση 10Kohm) αναπαριστώνται με μαύρο χρώμα, ενώ με κόκκινο, πράσινο και κίτρινο παρουσιάζονται οι μετρήσεις των αισθητήρων DHT22\_2, DHT22\_3, DHT22\_4 αντίστοιχα, που συνδέονται με έναν πυκνωτή 100nF.

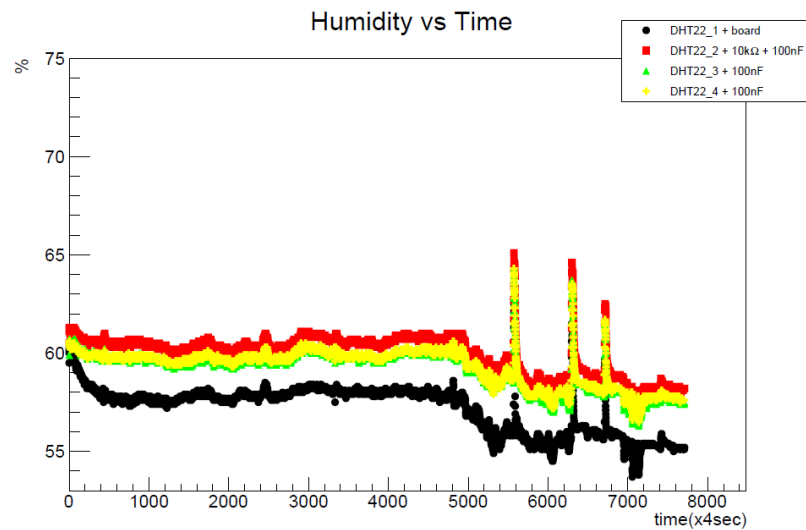


Σχήμα 3.13: Διάγραμμα υγρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις του αισθητήρα DHT22\_1 που συνδέεται με το board, (πυκνωτή 100nF και αντίσταση 10Kohm) αναπαριστώνται με μαύρο χρώμα, ενώ με κόκκινο, πράσινο και κίτρινο παρουσιάζονται οι μετρήσεις των αισθητήρων DHT22\_2, DHT22\_3, DHT22\_4 αντίστοιχα, που συνδέονται με έναν πυκνωτή 100nF.

Στην πέμπτη και τελευταία μέτρηση, τα φίλτρα που χρησιμοποιήθηκαν είναι τα ίδια με αυτά της προηγούμενης μέτρησης. Οι διαφορές αυτής της μέτρησης από την προηγούμενη είναι ότι τώρα στους δυο αισθητήρες έχει χρησιμοποιηθεί το φίλτρο RC και στους άλλους δυο ως φίλτρο χρησιμοποιείται ο πυκνωτής. Η βασική διαφορά όμως είναι ότι σε αυτήν τη μέτρηση η καταγραφή των τιμών των αισθητήρων ξεκίνησε 15 λεπτά μετά τη σύνδεση τους, διάστημα στο οποίο δεν επηρεάστηκαν από εξωγενείς παράγοντες. Επιπλέον και σε αυτή τη τελευταία μέτρηση οι τιμές καταγράφονταν ανα 4 sec.



Σχήμα 3.14: Διάγραμμα θερμοκρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις των αισθητήρων που συνδέονται με πυκνωτή 100nF και αντίσταση 10Kohm αναπαριστώνται με μαύρο και κόκκινο χρώμα και είναι οι DHT22\_1 και DHT22\_2 αντίστοιχα, ενώ με πράσινο και κίτρινο είναι οι μετρήσεις των αισθητήρων που συνδέονται με έναν πυκνωτή 100nF, DHT22\_3 και DHT22\_4.



Σχήμα 3.15: Διάγραμμα υγρασιών - χρόνου DHT22 με διάφορα φίλτρα. Οι μετρήσεις των αισθητήρων που συνδέονται με πυκνωτή 100nF και αντίσταση 10Kohm αναπαριστώνται με μαύρο και κόκκινο χρώμα και είναι οι DHT22\_1 και DHT22\_2 αντίστοιχα, ενώ με πράσινο και κίτρινο είναι οι μετρήσεις των αισθητήρων που συνδέονται με έναν πυκνωτή 100nF, DHT22\_3 και DHT22\_4.

Από τις γραφικές παραστάσεις προέκυψε ότι καλύτερες μετρήσεις ήταν αυτές στις οποίες χρησιμοποιήθηκε σαν φίλτρο ο πυκνωτής 100nF αφού οι διακυμάνσεις περιορίστηκαν σημαντικά και υπήρξε συμφωνία των πραγματικών και των πειραματικών δεδομένων. Επίσης στο τελευταίο διάγραμμα θερμοκρασιών, οι θερμοκρασίες είναι σταθερές από την αρχή της μέτρησης και δεν παρουσιάζουν την πτώση που παρατηρείται στα προηγούμενα διαγράμματα. Αυτό οφείλεται στο γεγονός ότι οι αισθητήρες DHT22 απαιτούν ένα χρονικό διάστημα μέχρι να αποκτήσουν τη θερμοκρασία περιβάλλοντος, εάν προηγουμένως είχαν επηρεαστεί από άλλους εξωγενείς παράγοντες.

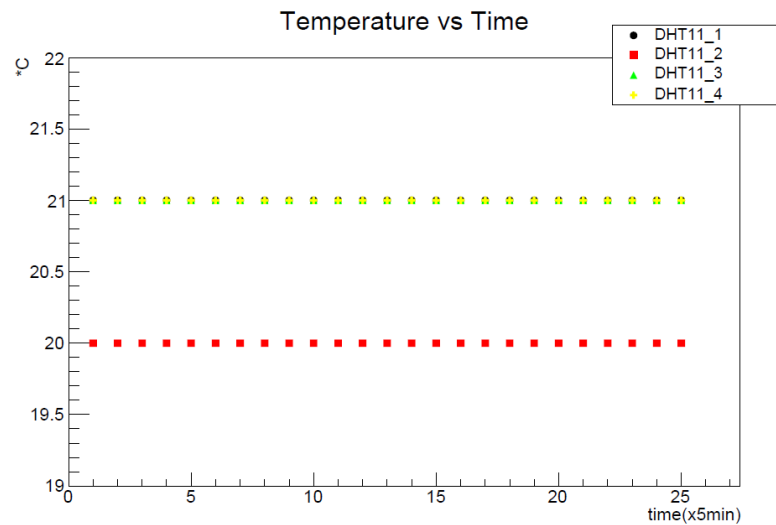
### 3.2.2 Σύνδεση Arduino με αισθητήρες DHT11



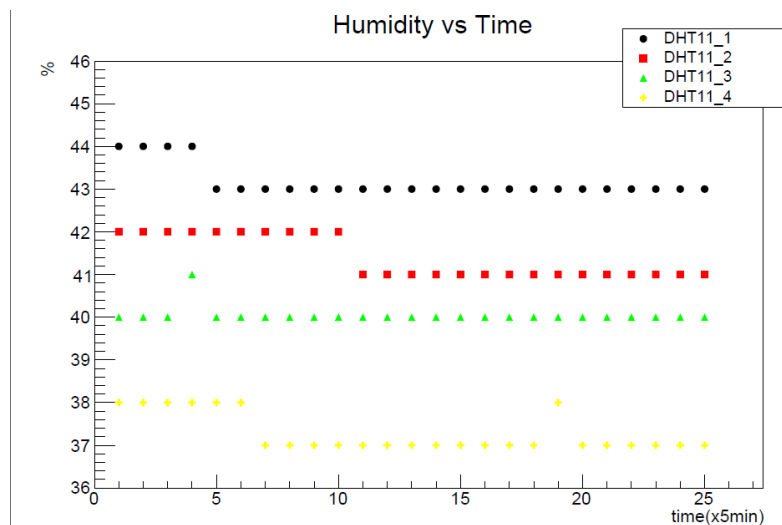
Σχήμα 3.16: Αισθητήρας DHT11 & καλώδιο σύνδεσης

Η σύνδεση του αισθητήρα DHT11 με το Ethernet Shield και κατ' επέκταση με το Arduino Mega γίνεται με το παρακάτω τρόπο προκειμένου ο DHT11 να λειτουργήσει και να "δώσει" τις τιμές της θερμοκρασίας και της σχετικής υγρασίας. Η πρώτη επαφή του αισθητήρα DHT11 (κόκκινο καλώδιο) συνδέεται με τη επαφή 5V του Ethernet Shield και είναι η είσοδος του αισθητήρα. Η δεύτερη επαφή του DHT11 (πράσινο καλώδιο) συνδέεται με μια από τις ψηφιακές επαφές του Ethernet Shield και αποτελεί την έξοδο του αισθητήρα η οποία δίνει το σήμα εξόδου. Η τρίτη επαφή του αισθητήρα δεν συνδέεται στο Ethernet Shield και στη συγκεκριμένη διάταξη του DHT11 δεν μπορεί και να συνδεθεί αφού δεν υπάρχει καλώδιο που να καταλήγει στην τρίτη επαφή. Η τέταρτη επαφή του αισθητήρα (μαύρο καλώδιο) συνδέεται με τη γείωση του Ethernet Shield.

Μετά από τη σύνδεση τεσσάρων αισθητήρων DHT11 στο Arduino έγινε η φόρτωση του "σκίτσου" στο Arduino. Το "σκίτσο" αυτό "διαβάζει" τις τιμές της θερμοκρασίας και της σχετικής υγρασίας ανά 5 minutes, τις τυπώνει στη σειριακή οθόνη και τις γράφει σε αρχείο .txt στην SD κάρτα. Σε κάθε αισθητήρα DHT11, το ένα άκρο της αντίστασης τοποθετήθηκε στην έξοδο του αισθητήρα (πράσινο καλώδιο) ενώ το άλλο σε τάση 5V. Από το .txt αρχείο των μετρήσεων προέκυψαν τα παρακάτω διάγραμματα.



Σχήμα 3.17: Διάγραμμα θερμοκρασιών - χρόνου DHT11. Με μαύρο, κόκκινο, πράσινο και κίτρινο χρώμα σημειώνονται οι τιμές των αισθητήρων DHT11\_1, DHT11\_2, DHT11\_3 και DHT11\_4 αντίστοιχα.



Σχήμα 3.18: Διάγραμμα υγρασιών - χρόνου DHT11. Με μαύρο, κόκκινο, πράσινο και κίτρινο χρώμα σημειώνονται οι τιμές των αισθητήρων DHT11\_1, DHT11\_2, DHT11\_3 και DHT11\_4 αντίστοιχα.

Όπως είναι φανερό από τα διαγράμματα η θερμοκρασία που δίνουν οι αισθητήρες DHT11 είναι σχετικά σταθερή, η υγρασία όμως δεν σταθεροποιήθηκε και συνέχιζε να παρουσιάζει σημαντικές διακυμάνσεις. Για το λόγο αυτό, αλλά και για το γεγονός ότι δεν προσφέρουν μεγάλη ακρίβεια σε σχέση με τους DHT22, δεν χρησιμοποιήθηκαν περαιτέρω.

### 3.3 Επικοινωνία αισθητήρων DHT22 & Arduino με WinCC\_OA μέσω OPC Server

#### 3.3.1 Εγκατάσταση OPC Server για Arduino

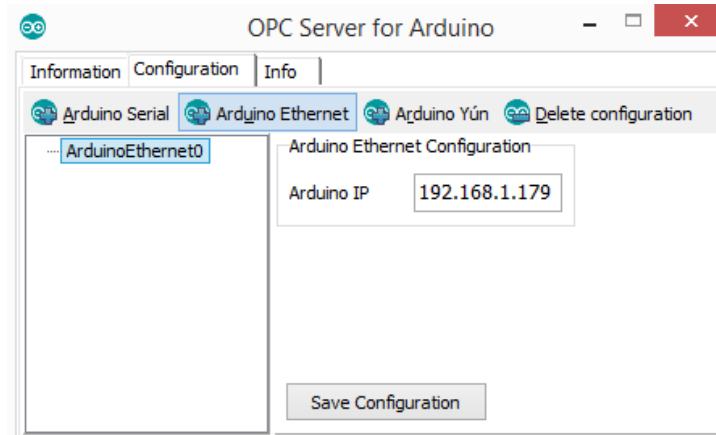


Σχήμα 3.19: Πληροφορίες του παράθυρου OPC Server για Arduino

Για να επιτευχθεί επικοινωνία μεταξύ του WinCC και της πλακέτας του Arduino Mega και για να μεταφερθούν οι τιμές της θερμοκρασίας και στην σχετική υγρασίας που μετρούν οι αισθητήρες DHT22 στο WinCC είναι απαραίτητη η χρήση του OPC διακομιστή για Arduino. Είναι επίσης απαραίτητη η χρήση καλωδίου Ethernet RJ45 το οποίο ενώνει το Ethernet Shield και την πλακέτα του Arduino Mega με τον υπολογιστή.

Αρχικά γίνεται εγκατάσταση στα Windows της τελευταίας έκδοσης OPC Server για Arduino (version 1.6). Η εγκατάσταση πρέπει να γίνει από το χρήστη μέσω της επιλογής Run as administrator έτσι ώστε να διασφαλιστεί η σωστή εγκατάσταση αλλά και η μετέπειτα ορθή λειτουργία του διακομιστή. Είναι απαραίτητο να σημειωθεί εδώ ότι πάντα η εκκίνηση του OPC Server για Arduino πρέπει να γίνεται μέσω της επιλογής Run as administrator, διαφορετικά θα υπάρχει πρόβλημα στην επικοινωνία του με το Arduino και με το WinCC.

### 3.3.2 Παραμετροποίηση OPC Server για Arduino



Σχήμα 3.20: Παραμετροποίηση του παράθυρου OPC Server για Arduino

Το επόμενο βήμα μετά την εγκατάσταση του διακομιστή OPC είναι η σωστή παραμετροποίησή του, έτσι ώστε να μπορεί να επικοινωνεί μέσω Ethernet με το Arduino. Η παραμετροποίηση αυτή πραγματοποιείται στην καρτέλα "Configuration" του περιβάλλοντος εργασίας του OPC Server. Η συγκεκριμένη καρτέλα δίνει στο χρήστη τη δυνατότητα να επιλέξει το είδος της επικοινωνίας του διακομιστή με το Arduino. Παρέχει τις επιλογές:

- Arduino Serial
- Arduino Ethernet
- Arduino Yun

Κάθε μια από τις παραπάνω επιλογές απαιτεί σωστό καθορισμό παραμέτρων, μέσω των οποίων ο διακομιστής θα συνδέεται με το Arduino. Έτσι για παράδειγμα στην Arduino Serial επικοινωνία πρέπει να καθοριστεί η σειριακή θύρα (serial port) στην οποία είναι συνδεδεμένο το Arduino καθώς και το Baud rate που έχει οριστεί στο "σκίτσο".

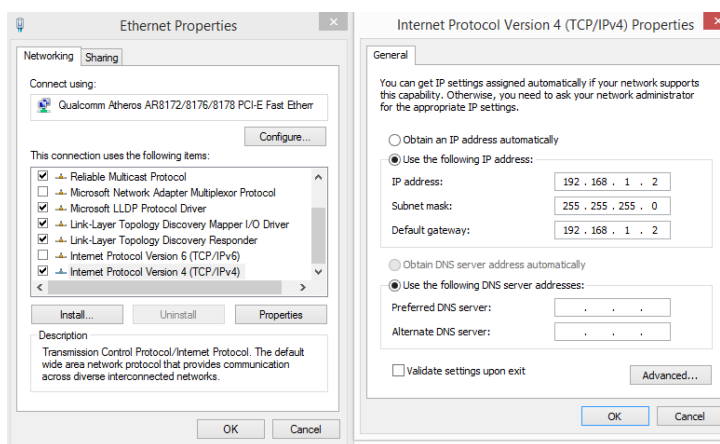
Στην περίπτωση που επιλεγεί ο τρόπος επικοινωνίας Arduino Ethernet οι παράμετροι που πρέπει να οριστούν είναι διαφορετικοί. Ο τρόπος επικοινωνίας που χρησιμοποιήθηκε είναι μέσω Ethernet, για το λόγο αυτό ορίστηκε στο πεδίο "Arduino IP" η διεύθυνση IP του Arduino. Η διεύθυνση αυτή του Arduino έχει οριστεί έτσι ώστε να είναι ίδια με αυτή που ο χρήστης επέλεξε να δώσει στο Arduino μέσα στον κώδικα του "σκίτσου" που θα φορτωθεί στην πλακέτα. Η παραμετροποίηση ολοκληρώνεται με την αποθήκευση των ρυθμίσεων. Τέλος ο χρήστης μέσω της



επιλογής "Delete Configuration" έχει τη δυνατότητα να διαγράφει την παραμετροποίηση.

### 3.4 Σύνδεση Arduino με OPC Server και Iconics OPC DataSpy

Από τη στιγμή που ο OPC server έχει εγκατασταθεί και παραμετροποιηθεί σωστά, η πλακέτα του Arduino με το Ethernet Shield και τους αισθητήρες συνδέεται στον υπολογιστή μέσω USB καλωδίου αλλά και μέσω ethernet καλωδίου RJ45. Στη συνέχεια πρέπει να φορτωθεί στο Arduino το "σκίτσο" στον κώδικα του οποίου θα υπάρχουν οι εντολές έτσι ώστε να μπορέσει το Arduino Mega και το Ethernet Shield να συνδεθούν με τον OPC Server για Arduino και να μεταφερθούν μέσω αυτού, οι τιμές της θερμοκρασίας και της σχετικής υγρασίας στον OPC DataSpy ο οποίος λειτουργεί σαν OPC πελάτης (OPC client). Για το λόγο αυτό ορίζονται στο "σκίτσο" δεδομένα που είναι απαραίτητα για να επιτευχθεί η Ethernet σύνδεση. Ο χρήστης ορίζει την IP Address του Arduino καθώς και τη Gateway όπως αυτός θέλει. Η IP Address χρησιμοποιείται στην παραμετροποίηση του OPC server, όπως προαναφέρθηκε, ενώ η Gateway η οποία ορίστηκε στον κώδικα, δηλώνεται στις ρυθμίσεις Ethernet TCP/IPV4 του πίνακα ελέγχου των Windows, ως IP Address. Συγκεκριμένα στον κώδικα του "σκίτσου" η Gateway ορίστηκε 192.168.1.2 και έτσι οι Ethernet ρυθμίσεις είναι όπως φαίνονται στο παρακάτω σχήμα.



Σχήμα 3.21: Ρυθμίσεις δικτύου για την επίτευξη σύνδεσης με OPC Server για Arduino

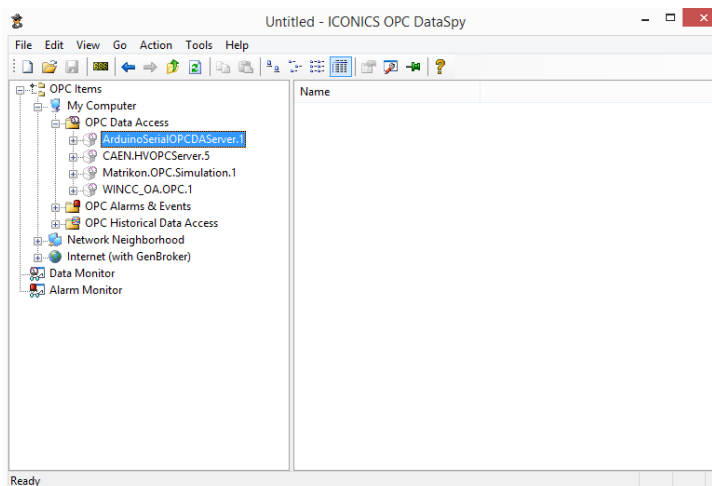
Εκτός από τις ρυθμίσεις δικτύου στον κώδικα του "σκίτσου" είναι απαραίτητη η χρήση της βιβλιοθήκης OPC.h. Προκειμένου η βιβλιοθήκη OPC.h να χρησιμοποιηθεί και να λειτουργήσει σωστά είναι απαραίτητο να οριστούν τα αντικείμενα "items" και να γίνει αντιστοίχιση σε αυτά

των δεδομένων που θα μεταφέρουν. Για το σκοπό αυτό ορίστηκαν 8 αντικείμενα ( 4 θερμοκρασίας και 4 υγρασίας αφού υπάρχουν τέσσερις αισθητήρες DHT22) τα οποία μέσω συγκεκριμένων συναρτήσεων, μιας διαφορετικής για το καθένα, μεταφέρουν τις τιμές που προέρχονται από τους αισθητήρες.

Στον κώδικα γίνεται επίσης χρήση της βιβλιοθήκης DHT.h η οποία είναι απαραίτητη έτσι ώστε μέσω εντολών όπως `.readHumidity()` και `.readTemperature()` να προκύπτουν οι τιμές της θερμοκρασίας και της σχετικής υγρασίας. Ο κώδικας φορτώνεται στο Arduino και αμέσως επιτυγχάνεται σύνδεση μεταξύ του OPC server και της πλακέτας. Αυτό που πρέπει να γίνει στη συνέχεια είναι η σύνδεση του OPC server με έναν OPC πελάτη (OPC client).

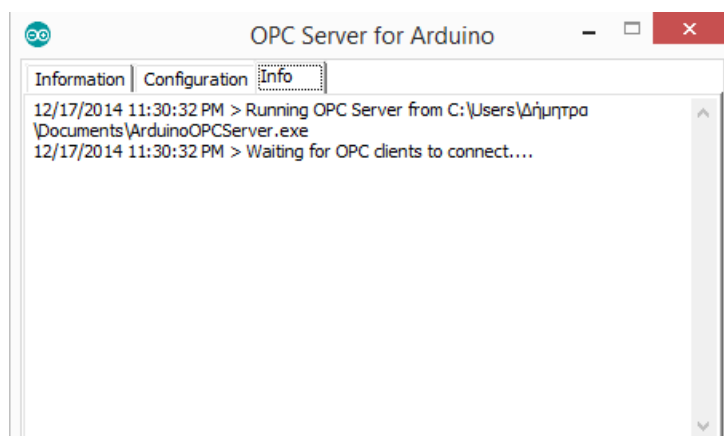
Στην αρχή γίνεται χρήση του ICONICS OPC DataSpy ο οποίος λειτουργεί σαν OPC πελάτης. Ο ICONICS OPC DataSpy είναι ένα εργαλείο λογισμικού που προσφέρεται από την εταιρία ICONICS. Ο OPC DataSpy™ αποτελεί μια ισχυρή εφαρμογή που δίνει στο χρήστη τη δυνατότητα πρόσβασης σε δεδομένα, ανάλυσης αλλά και ελέγχου αυτών των δεδομένων ενώ παράλληλα παρέχει πληροφορίες για την ποιότητα και την τις επιδόσεις τους.

Όλες οι παραπάνω λειτουργίες έχουν δημιουργηθεί κάτω από τα πρότυπα της εταιρίας OPC (OPC Foundation). Ο OPC DataSpy επιτρέπει τη σύνδεση με έναν διακομιστή OPC (OPC Server) και την εμφάνιση των τιμών που αυτός μεταφέρει. Είναι ένα εργαλείο το οποίο παρέχει έναν γρήγορο τρόπο στους χρήστες να ελέγχουν και να ταυτοποιούν τη σύνδεση του διακομιστή με συσκευές μέσω των OPC ετικετών "OPC tags". Για να ελεγχθεί η σύνδεση του OPC Server για Arduino με την πλακέτα και τους αισθητήρες γίνεται εκκίνηση του OPC DataSpy. Όταν ανοίξει το παράθυρο του OPC DataSpy στο αριστερό τμήμα, στην επιλογή OPC Data Access υπάρχει μια λίστα από τους διαθέσιμους διακομιστές που υπάρχουν στον υπολογιστή. Ο χρήστης επιλέγει τον διακομιστή στον οποίο θέλει να συνδέσει τον Data Spy και μέσω της επιλογής "Connect" ξεκινά την σύνδεση.



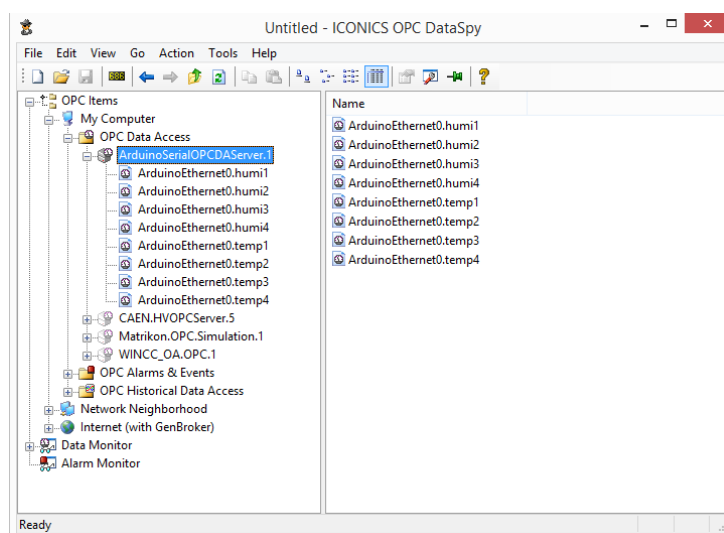
Σχήμα 3.22: Επιλογή και σύνδεση OPC Server για Arduino στο παράθυρο του DataSpy

Ο OPC Server για Arduino είναι ανοιχτός και περιμένει τον OPC DataSpy να συνδεθεί.



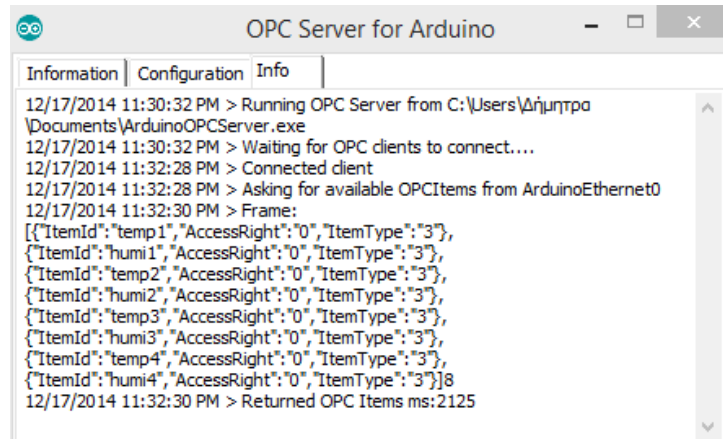
Σχήμα 3.23: Πληροφορίες για αναμονή σύνδεσης στο παράθυρο του OPC Server για Arduino

Μόλις γίνει η επιλογή του OPC Server για Arduino και επιτευχθεί η σύνδεση, εμφανίζονται στο παράθυρο του OPC DataSpy τα αντικείμενα "items" που ο χρήστης έχει ορίσει στο "σκίτσο" και τα οποία μεταφέρουν τις τιμές της θερμοκρασίας και της σχετικής υγρασίας από τους αισθητήρες σαν ετικέτες "tags". Ο χρήστης συνδέει τα αντικείμενα "items" με τις ετικέτες "tags" και με αυτόν τον τρόπο εμφανίζονται μέσω της επιλογής "Connect" οι τιμές.



Σχήμα 3.24: Εμφάνιση των items του OPC Server για Arduino στο παράθυρο του OPC DataSpy

Ταυτόχρονα στο παράθυρο του OPC Server για Arduino εμφανίζονται πληροφορίες για την κατάσταση σύνδεσης αλλά και για τα αντικείμενα "items" που μεταφέρουν τις τιμές που δίνουν οι αισθητήρες.



Σχήμα 3.25: Πληροφορίες για την μεταφορά των items στο παράθυρο του OPC Server για Arduino

Με αυτόν τον τρόπο ο χρήστης μπορεί να ελέγξει την κατάσταση σύνδεσης και να την επαληθεύσει, εφόσον έχει γίνει σωστά. Διασφαλίζεται έτσι ότι η σύνδεση μεταξύ OPC Server για Arduino και της πλακέτας του Arduino με τους αισθητήρες λειτουργεί και ο χρήστης είναι σε θέση να λαμβάνει τιμές.

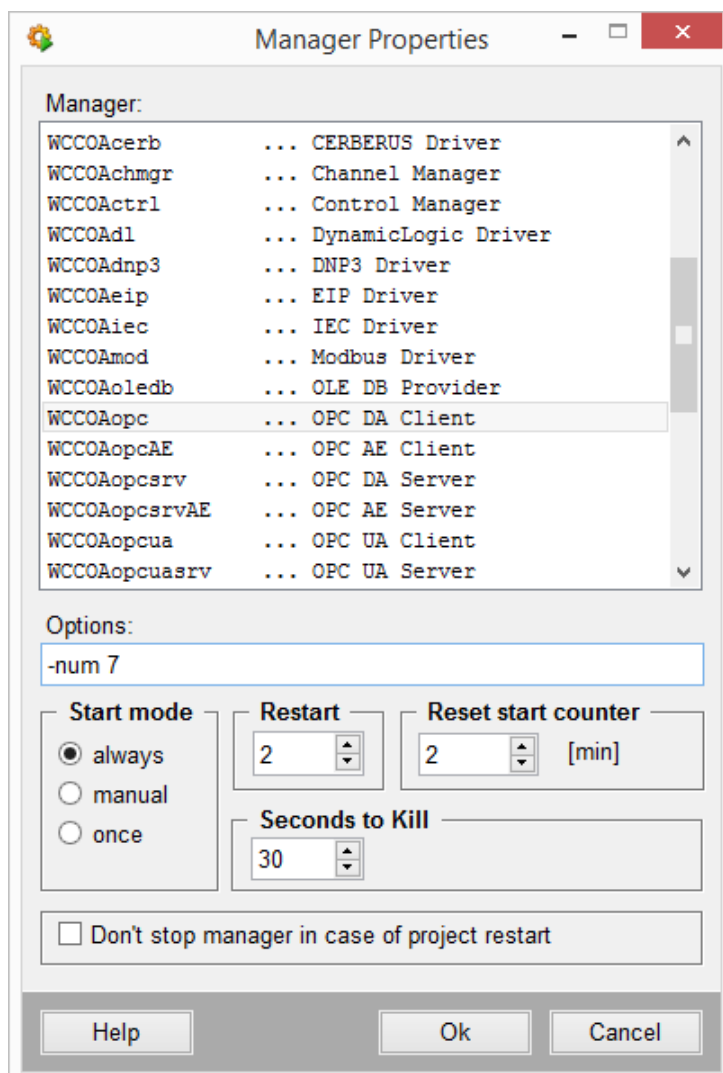
### 3.5 Σύνδεση Arduino με OPC Server & WinCC\_OA

Απο τη στιγμή που έχει ελεγχθεί και επαληθευτεί η σύνδεση της πλακέτας του Arduino και των αισθητήρων με τον OPC Server για Arduino γίνεται σύνδεση του OPC Server με το WinCC\_OA.

#### 3.5.1 Δημιουργία OPC πελάτη (OPC client) στο WinCC

Για να πραγματοποιηθεί η σύνδεση είναι απαραίτητη η παρουσία ενός OPC πελάτη (OPC client). Το WinCC\_OA δίνει τη δυνατότητα στους χρήστες να δημιουργήσουν έναν ή περισσότερους OPC πελάτες (OPC clients). Ο OPC client που θα δημιουργηθεί είναι σε θέση να συνδεθεί με οποιονδήποτε OPC διακομιστή μέσω μιας συγκεκριμένης διαδικασίας. Το πρώτο βήμα για τη δημιουργία ενός καινούριου OPC πελάτη

σε ένα project του WinCC είναι ο ορισμός του στον φάκελο παραμετροποίησης (config file) του project. Για το σκοπό αυτό γράφεται μια σειρά συγκεκριμένων εντολών σε αυτό το φάκελο. Αυτές είναι οι παρακάτω:



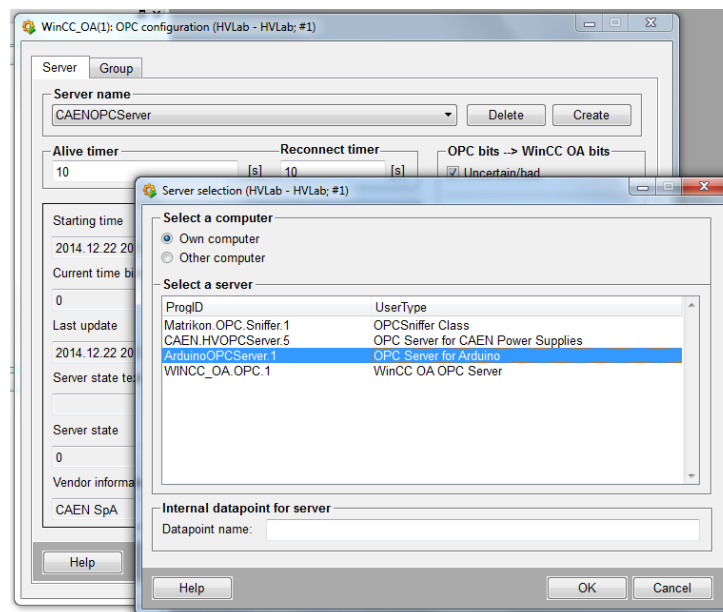
Σχήμα 3.26: Παράθυρο Manager Properties που χρησιμοποιείται για την εισαγωγή του OPC DA Client

Στη συνέχεια στην κονσόλα του WinCC μέσω της επιλογής "Insert Manager" ο χρήστης διαλέγει τον τύπο του Manager που θέλει να προσθέσει στο project. Εδώ ως πελάτης επιλέγεται ένας OPC DA Client και στο παράθυρο των ιδιοτήτων του (Manager Properties) καθορίζονται συγκεκριμένα χαρακτηριστικά του, όπως για παράδειγμα ο αριθμός που ορίζει ποιος κατά σειρά manager είναι (στο συγκεκριμένο project είναι ο

έβδομος "-num 7"). Με αυτόν τον τρόπο δημιουργείται ο OPC πελάτης και εμφανίζεται στην κονσόλα του WinCC.

### 3.5.2 Σύνδεση OPC DA Client με OPC Server για Arduino

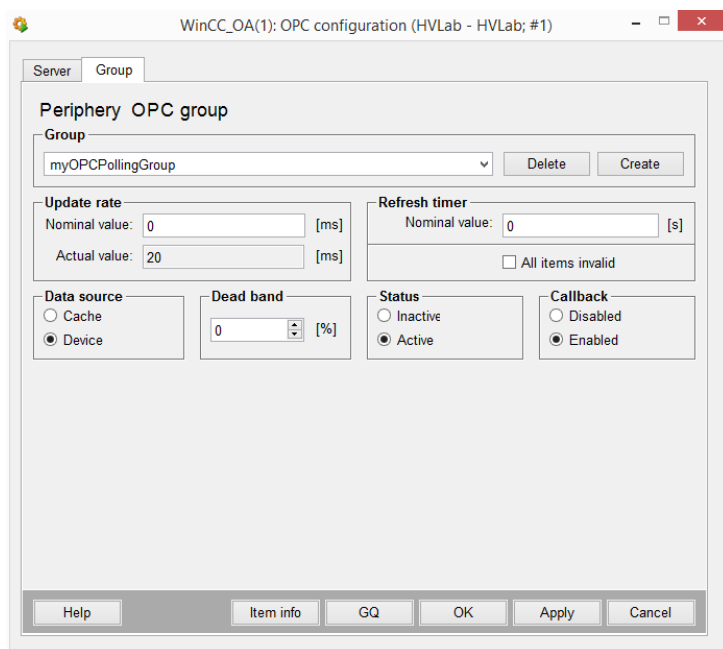
Αφού ολοκληρωθεί η διαδικασία δημιουργίας του OPC πελάτη, ακολουθείται μια διαδικασία σύνδεσης του OPC Server για Arduino με τον OPC client. Για να γίνει αυτό επιλέγεται ο OPC Server για Arduino στο παράθυρο παραμετροποίησης OPC (OPC Configuration) που βρίσκεται στο System Management-> OPC Driver -> OPC Driver.



Σχήμα 3.27: Παράθυρο Server Selection που χρησιμοποιείται για την επιλογή του OPC Server για Arduino

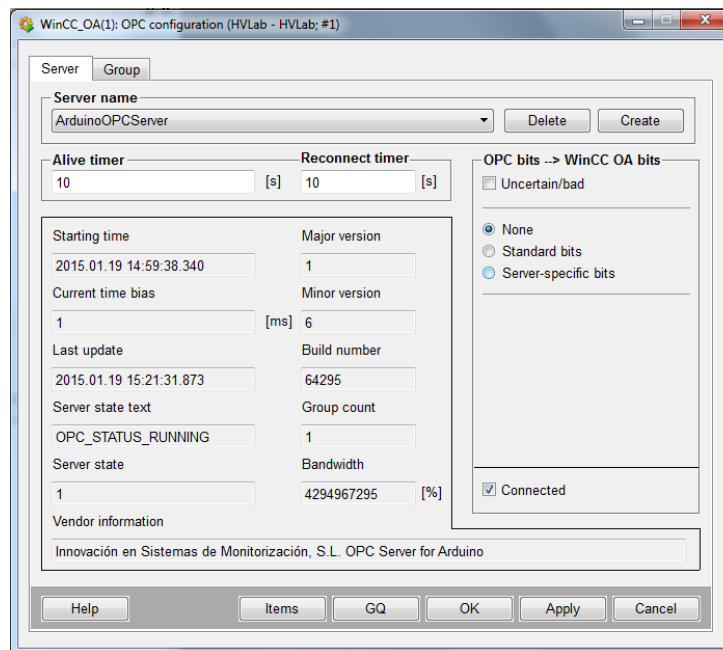


Στη συνέχεια δημιουργείται καινούριο group δεδομένων "myOPCPollingGroup" όπως φαίνεται στην εικόνα:



Σχήμα 3.28: Παράθυρο OPC Configurator που χρησιμοποιείται για την δημιουργία καινούριου group δεδομένων

Μόλις ολοκληρωθεί αυτή η διαδικασία είναι απαραίτητη η επανεκκίνηση του OPC client. Μετά από την επανεκκίνηση η σύνδεση έχει πραγματοποιηθεί, γεγονός που φαίνεται και από την κατάσταση σύνδεσης στο παράθυρο παραμετροποίησης OPC.



Σχήμα 3.29: Η επιλεγμένη ένδειξη "Connected" στο παράθυρο OPC Configurator καταδεικνύει την επιτυχή σύνδεση του OPC client με τον OPC Server για Arduino

### 3.5.3 Δημιουργία datapoints

Μετά την επίτευξη της σύνδεσης του διακομιστή OPC με τον πελάτη OPCDA client, δημιουργούνται στον παρα τα datapoints τα οποία θα συνδεθούν στην συνέχεια με τα αντικείμενα "items" που μεταφέρουν τις τιμές των αισθητήρων. Για το σκοπό αυτό δημιουργήθηκαν 8 datapoints (όσα και τα αντικείμενα "items").

Αυτά είναι:

- temperature1
- humidity1
- temperature2
- humidity2

- temperature3
- humidity3
- temperature4
- humidity4

### Periphery Address

Για να γίνει σύνδεση αυτών των datapoints με τα αντίστοιχα αντικείμενα "items" του OPC server ο χρήστης πατάει σε κάθε ένα από αυτά δεξί κλικ -> insert config -> periphery address. Τώρα το καθένα από αυτά τα datapoints μπορεί να συνδεθεί με το αντίστοιχο αντικείμενο που προσφέρει ο OPC server. Πατώντας στο "+" αριστερά από το κάθε datapoint ο χρήστης επιλέγει τη ρύθμιση διευθυνσιοδότησης. Στο δεξί μέρος του παραθύρου ανοίγει ένα παράθυρο στο οποίο γίνεται επιλογή του Driver Type -> OPCCLIENT για το κάθε datapoint. Στη συνέχεια πατώντας στο κουμπί Parameterize εμφανίζεται ένα νέο παράθυρο στο οποίο ορίζονται τα στοιχεία του server και του group που δημιουργήθηκε προηγουμένως. Εκεί επιλέγεται ο διακομιστής που χρησιμοποιείται και συγκεκριμένα ο OPC Server για Arduino καθώς και το myOPCPollingGroup. Στο συγκεκριμένο παράθυρο γίνεται και η αντιστοίχιση του κάθε datapoint στο αντικείμενο "item" του OPC Server.

Η ίδια διαδικασία ακολουθείται και για τα 8 datapoints που δημιουργήθηκαν προηγουμένως επιτυγχάνοντας έτσι την επικοινωνία τους με την πλακέτα του Arduino και με τους αισθητήρες. Μια επικοινωνία, βασικός συντελεστής της οποίας είναι η σύνδεση των datapoints με τα αντικείμενα (items) του OPC Server για Arduino τα οποία μεταφέρουν τις τιμές της θερμοκρασίας και της σχετικής υγρασίας που οι αισθητήρες DHT22 μετρούν.

### Archive

Εκτός από τη δυνατότητα της διευθυνσιοδότησης του κάθε datapoint, ο χρήστης έχει τη δυνατότητα να προσθέσει και άλλες λειτουργίες στο κάθε datapoint μέσω των οποίων μπορεί να το παραμετροποιήσει περαιτέρω. Μία από αυτές η οποία χρησιμοποιήθηκε, είναι αυτή της αρχειοθέτησης (archive) η οποία διαθέτει επιμέρους λειτουργίες όπως αυτή της εξομάλυνσης (smoothing).

Η προσθήκη του archive σε κάθε datapoint πραγματοποιήθηκε με την ίδια διαδικασία κατά την οποία έγινε και η εισαγωγή της Periphery

Address. Πιο συγκεκριμένα ο χρήστης πατάει σε κάθε ένα από τα 8 datapoints δεξί κλικ -> insert config -> Archive. Στη συνέχεια επιλέγοντας το Archive ανοίγει στο δεξί μέρος του παραθύρου ένα νέο παράθυρο στο οποίο ο χρήστης κλικάρει την επιλογή "Store original value" και διαλέγει από την λίστα των διαθέσιμων επιλογών τη "ValueArchive\_0000". Μόλις η επιλογή "Store original value" επιλεγεί, εμφανίζεται ακόμη μία η οποία πάλι επιλέγεται και είναι αυτή της εξομάλυνσης, η "Smoothing". Στη συνέχεια ο χρήστης διαλέγει τον τρόπο με τον οποίο γίνεται το "Smoothing" αλλά και το χρόνο ανά τον οποίο αυτό επαναλαμβάνεται. Συγκεκριμένα επιλέχθηκε ο τρόπος "Value-AND time-dependent smoothing" να γίνεται ανά 20 second. Η συγκεκριμένη λειτουργία είναι πολύ σημαντική αφού δίνει στο χρήστη τη δυνατότητα να αρχειοθετεί τις τιμές του κάθε datapoint ανά συγκεκριμένο χρόνο (εδώ ανά 20 second) και να μπορεί να τις επεξεργαστεί αργότερα.

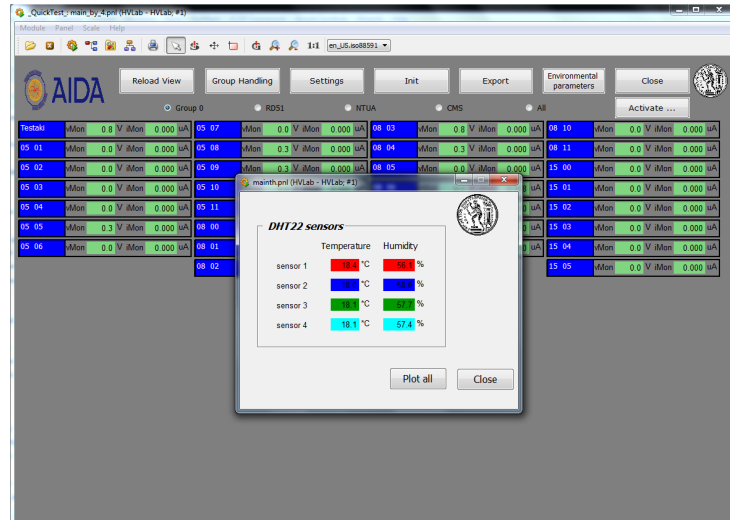
### 3.6 Παράθυρο περιβαλλοντικών παραμέτρων στο project HVLab

Όταν η ανάπτυξη του συστήματος επικοινωνίας μεταξύ WinCC, Arduino και αισθητήρων μέσω του OPC Server για Arduino έχει ολοκληρωθεί και ελεγχθεί η σωστή λειτουργία του, είναι απαραίτητη η δημιουργία ενός παραθύρου στο οποίο θα απεικονίζονται οι τιμές από τους αισθητήρες και θα επεξεργάζονται.

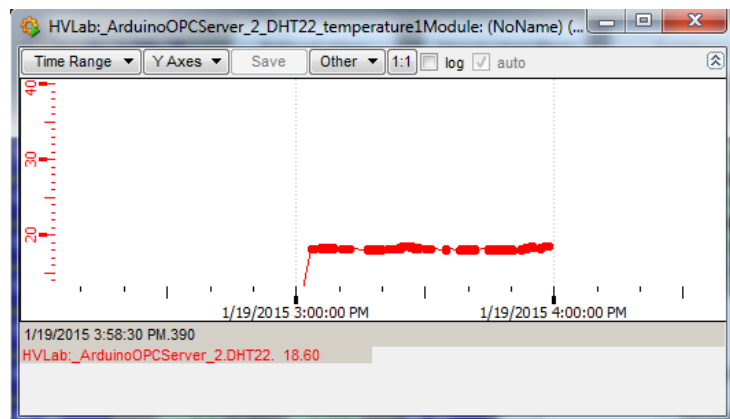
Για το σκοπό αυτό στο project HVLab προστέθηκε ένα κουμπί με την ονομασία "Environmental parameters". Όταν ο χρήστης πατήσει το συγκεκριμένο κουμπί εμφανίζεται ένα παράθυρο στο οποίο αναγράφονται οι τιμές της θερμοκρασίας και της σχετικής υγρασίας καθενός από του τέσσερις αισθητήρες DHT22. Οι τιμές αυτές αλλάζουν κάθε φορά που ο αισθητήρας "δώσει" διαφορετική μέτρηση. Ταυτόχρονα όλες οι τιμές των αισθητήρων αρχειοθετούνται και έτσι ο χρήστης όταν πατήσει δεξί κλικ πάνω στο πεδίο που αναγράφεται η τιμή της κάθε μέτρησης εμφανίζεται ένα παράθυρο στο οποίο αναπαρίστανται γραφικά οι τιμές συναρτήσεως του χρόνου. Αν παραδείγματος χάριν ο χρήστης πατήσει δεξί κλικ πάνω στο πεδίο που αναγράφεται η τιμή της θερμοκρασίας του πρώτου αισθητήρα, θα εμφανιστεί ένα παράθυρο που θα δίνει τη γραφική παράσταση της θερμοκρασίας του πρώτου αισθητήρα συναρτήσεως του χρόνου από τη στιγμή που ξεκίνησε η μέτρηση.

Στο παράθυρο αυτό υπάρχει ακόμη ένα κουμπί το "Plots" με το οποίο ο χρήστης μπορεί να κάνει δυο πράγματα. Αρχικά κάνοντας δεξί κλικ πάνω σε αυτό εμφανίζεται ένα παράθυρο στο οποίο υπάρχουν τέσσερις γραφικές παραστάσεις. Σε κάθε μία από αυτές αναπαρίσταται η εξέλιξη της θερμοκρασίας κάθε αισθητήρα συναρτήσεως του χρόνου. Κάνοντας τώρα αριστερό κλικ πάνω στο κουμπί "Plots" εμφανίζεται ένα άλλο παράθυρο στο οποίο υπάρχουν οι τέσσερις γραφικές παραστάσεις της σχετικής υγρασίας συναρτήσεως του χρόνου των τεσσάρων αισθητήρων. Κάθε γραφική παράσταση δείχνει τη χρονική εξέλιξη της σχετικής υγρασίας κάθε αισθητήρα DHT22. Προκειμένου να γίνεται διαχωρισμός των γραφικών παραστάσεων για κάθε αισθητήρα, οι γραμμή και τα σημεία που καταγράφονται, σύμφωνα με τις τιμές που μετράει ο κάθε αισθητήρας, έχουν διαφορετικό χρώμα. Έτσι οι γραφικές παραστάσεις (θερμοκρασίας και σχετικής υγρασίας) και τα σημεία του πρώτου αισθητήρα έχουν κόκκινο χρώμα, του δεύτερου μπλέ, του τρίτου πράσινο και του τέταρτου βεραμάν. Αντίστοιχα χρώματα έχουν και τα πεδία στα οποία αναγράφονται οι τιμές του κάθε αισθητήρα. Τέλος υπάρχει το κουμπί "Close" με το οποίο ο χρήστης κάνοντας αριστερό κλικ μπορεί

να κλείσει το παράθυρο.



Σχήμα 3.30: Παράθυρο Περιβαλλοντικών Παραμέτρων



Σχήμα 3.31: Διάγραμμα Θερμοκρασίας συναρτήσει του χρόνου του πρώτου αισθητήρα DHT22

# Συμπεράσματα

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε μετάβαση του συστήματος αυτομάτου ελέγχου Slow Control System από το PVSS στο WinCC\_OA. Το προηγούμενο project μετονομάστηκε σε HVLab και μεταφέρθηκαν σε αυτό όλες οι λειτουργίες του Slow Control System. Ταυτόχρονα πραγματοποιήθηκαν αλλαγές στους κώδικες των panels ώστε το καινούργιο HVLab να λειτουργεί σωστά στο WinCC\_OA. Επιπλέον προστέθηκαν στο project πρόσθετες λειτουργίες έτσι ώστε να χρησιμοποιηθεί στα AIDA test beams.

Στο δεύτερο μέρος της εργασίας πραγματοποιήθηκε ανάπτυξη ενός συστήματος μέτρησης θερμοκρασίας και υγρασίας το οποίο βασίζεται στη μικροελεγκτική πλακέτα Arduino και στο WinCC\_OA. Για το σκοπό αυτό χρησιμοποιήθηκαν οι αισθητήρες Θερμοκρασίας και υγρασίας DHT22 και DHT11 σε συνδυασμό με το Arduino Mega και το Arduino Ethernet Shield για την εξαγωγή πειραματικών δεδομένων. Προέκυψε ότι οι DHT22 υπερέχουν των DHT11 καθώς έχουν μεγαλύτερη ακρίβεια και για το λόγο αυτό, μόνο αυτοί χρησιμοποιήθηκαν στη συνέχεια. Το επόμενο στάδιο ήταν η επίτευξη ethernet επικοινωνίας μεταξύ Arduino και WinCC\_OA. Αυτό έγινε με τη χρήση ενός OPC Server για Arduino ο οποίος συνέβαλε στη μεταφορά των τιμών της θερμοκρασίας και της υγρασίας του κάθε αισθητήρα στο WinCC\_OA για περαιτέρω επεξεργασία. Μόλις η επικοινωνία ολοκληρώθηκε δημιουργήθηκε στο project HVLab ένα επιπλέον παράθυρο Περιβαλλοντικών Παραμέτρων στο οποίο εμφανίζονται οι τιμές όλων των αισθητήρων ενώ παράλληλα υπάρχει η δυνατότητα γραφικής αναπαράστασης των τιμών αυτών συναρτήσει του χρόνου για κάθε αισθητήρα ξεχωριστά αλλά και όλων ταυτόχρονα. Η ανάπτυξη του συστήματος αυτού είναι ιδιαίτερα σημαντική καθώς επιτρέπει τον έλεγχο των περιβαλλοντικών παραμέτρων των ανιχνευτών κατά τη διάρκεια των tests beams.





# Βιβλιογραφία

- [1] Μεταπτυχιακή Διπλωματική Εργασία Κωνσταντίνου Καρακώστα, Τηλεσκόπιο micromegas για τα RD51 test beam
- [2] <http://arduino.cc/en/Guide/HomePage>
- [3] <http://www.st4makers.com/>
- [4] <http://www.adafruit.com/>
- [5] <http://www.dfrobot.com/>
- [6] Διπλωματική Εργασία Παράσχου Μοσχοβάκου, Ανάπτυξη 3D Detector Control System για το New Small Wheel του πειράματος ATLAS.
- [7] <http://www.caen.it/>
- [8] <http://root.cern.ch/drupal/>
- [9] John Boxall, *Arduino Workshop*
- [10] Παναγιώτης Παπάζογλου, Σπύρος-Πολυχρόνης Λιωνής *Ανάπτυξη εφαρμογών με το Arduino*
- [11] Brian Evans *Beginning Arduino Programming* 2011.
- [12] <https://github.com/>
- [13] Emily Gertz, Patrick Di Justo *Environmental Monitoring with Arduino Building Simple Devices to Collect Data About the World Around Us* 2012
- [14] <http://wikis.web.cern.ch/wikis/display/EN/WinCC-OA+Service>