SIMATIC MANAGER

Προγραμματισμός του PLC.

1. Γενικά

Μια προσεκτική ματιά σε μια εγκατάσταση που θέλουμε να αυτοματοποιήσουμε , μας δείχνει ότι αυτή αποτελείται από επιμέρους τμήματα τα οποία είναι συνδεδεμένα μεταξύ τους με κάποια λογική.

Η πρώτη δουλειά μας είναι λοιπόν να χωρίσουμε την διαδικασία παραγωγής σε επιμέρους εργασίες :

Hardware:

- Αριθμό και τύπο εισόδων και εξόδων
- Αριθμό και τύπο μονάδων
- Αριθμό των απαιτούμενων rack
- Χωρητικότητα και τύπο CPU
- HMI συστήματα
- Συστήματα δικτύωσης

Software:

- Δομή προγράμματος
- Διαχείριση δεδομένων για τη διαδικασία αυτοματισμού
- Δεδομένα διαμόρφωσης (Configuration)
- Δεδομένα επικοινωνίας
- Τεκμηρίωση προγράμματος και project

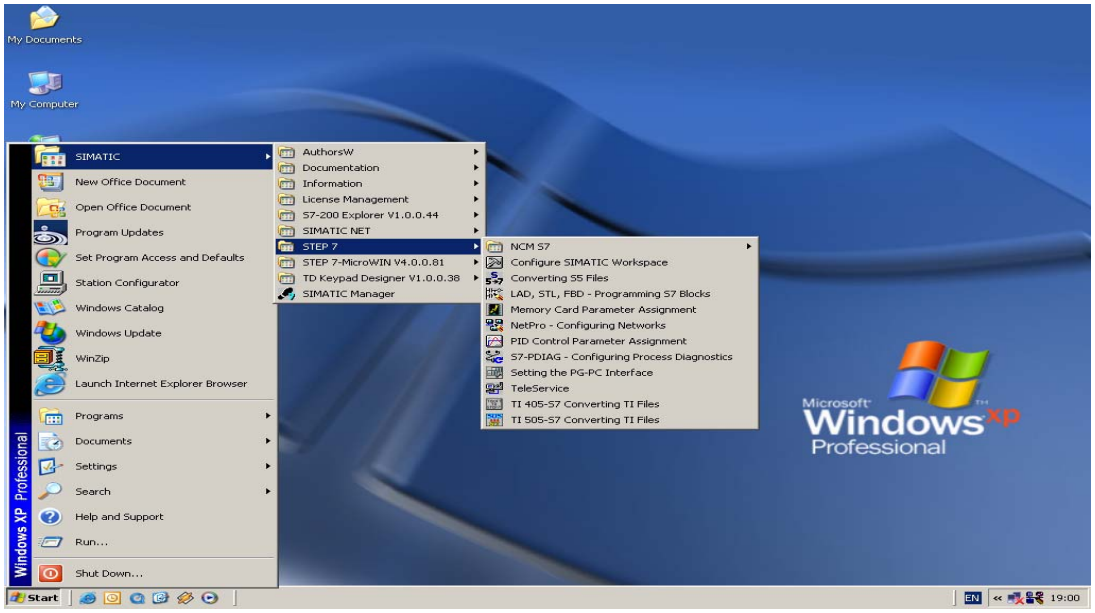
Στο Simatic S7 όλες οι απαιτήσεις σε Hardware και Software μιας διαδικασίας αυτοματισμού διαχειρίζονται μέσα από ένα project.

Ένα project περιλαμβάνει το απαραίτητο Hardware (με τη διαμόρφωση του υλικού), το δίκτυο (με τη διαμόρφωση του), όλα τα προγράμματα καθώς και ολόκληρη τη διαχείριση δεδομένων για μια λύση αυτοματισμού.

2. Ανοίγοντας τον “Simatic Manager”

Ο Simatic Manager είναι το κύριο εργαλείο στο Step 7 αφού είναι αυτός που διαχειρίζεται τα project. Το χαρακτηριστικό του εικονίδιο υπάρχει στο Desktop των Windows και πατώντας διπλό κλικ του εκκινεί το πρόγραμμα.

Ένας άλλος τρόπος εκκίνησης είναι και από την επιλογή **Start> Simatic >Step 7** ,όπως φαίνεται στο σχήμα που ακολουθεί:



Περιγραφή των σημαντικότερων επιλογών:

NCM S7: για τη παραμετροποίηση καρτών δικτύου.

Configure SIMATIC Workspace: για διαμόρφωση των παραμέτρων του πακέτου όταν χρησιμοποιείται από περισσότερους χρήστες σε δίκτυο.

Converting S5 Files : Μετατροπή προγραμμάτων από S5 σε S7.

LAD,STL,FBD : για τη δημιουργία προγραμμάτων σε μια από τις τρεις γλώσσες προγραμματισμού-Σχέδιο επαφών (LAD), λίστα εντολών (STL) και πύλες (FBD) .

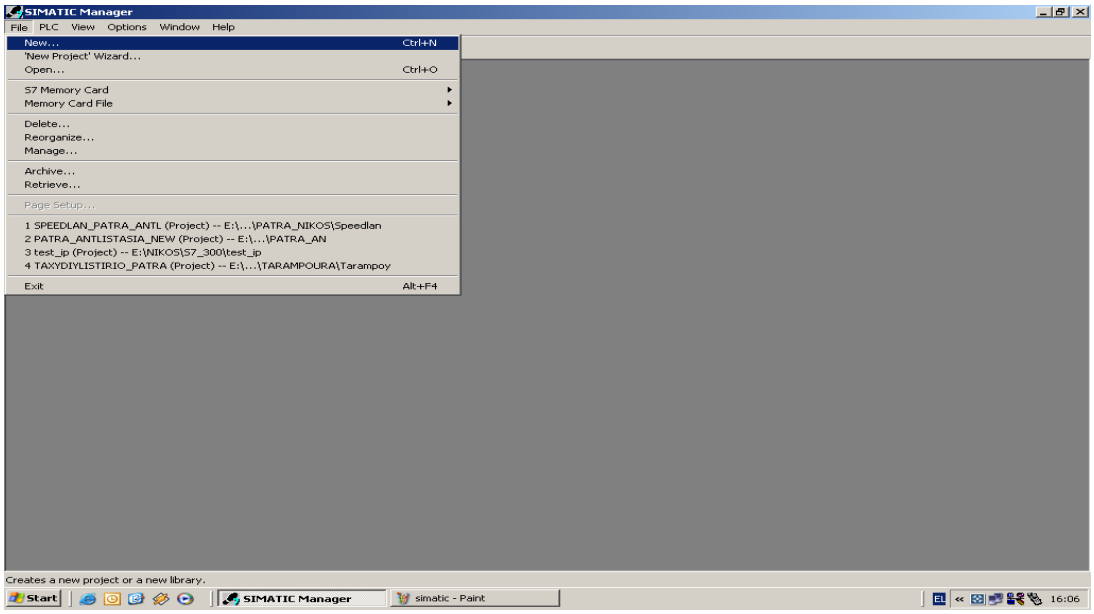
Memory Card Parameter Assignment: για αποθήκευση προγραμμάτων σε EPROM, σβήσιμο EPROM.

PID Control Parameter Assignment: για τη παραμετροποίηση των μπλοκ που υλοποιούν έλεγχο κλειστού βρόγχου.

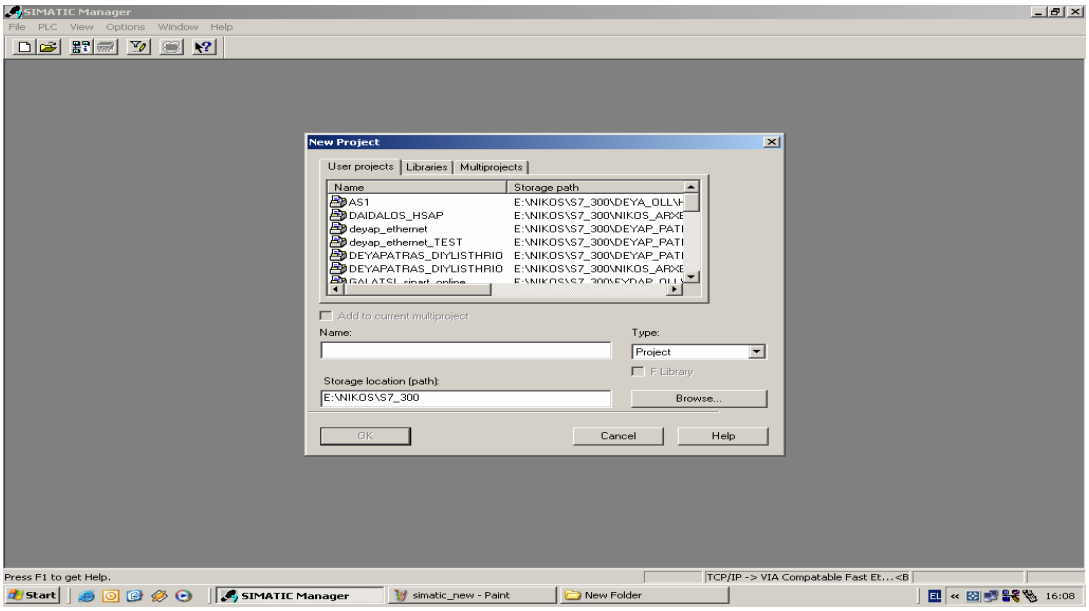
Setting the PG-PC Interface: παραμετροποίηση της κάρτας MPI (αριθμός σταθμού , ταχύτητα μετάδοσης κτλ).

3.Δημιουργία Project

Στο περιβάλλον του Simatic Manager επιλέγουμε **File > New** όπως φαίνεται στο παρακάτω σχήμα :

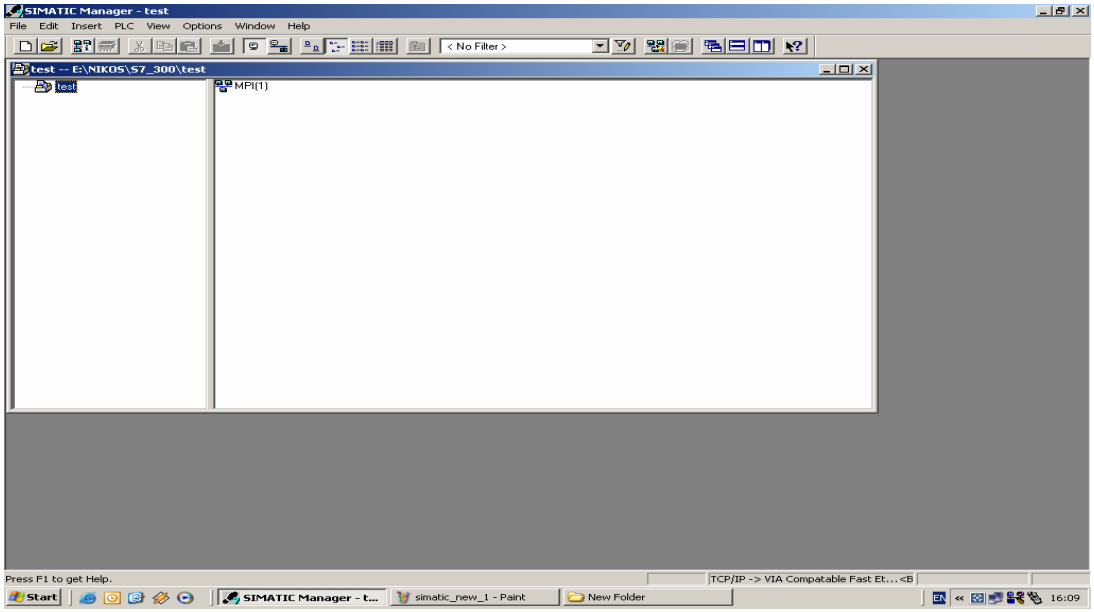


οπότε και εμφανίζεται το παρακάτω παράθυρο δημιουργίας νέας εφαρμογής στο οποίο ονομάζουμε το project και τον κατάλογο στον οποίο θα αποθηκευτεί στο σκληρό δίσκο :



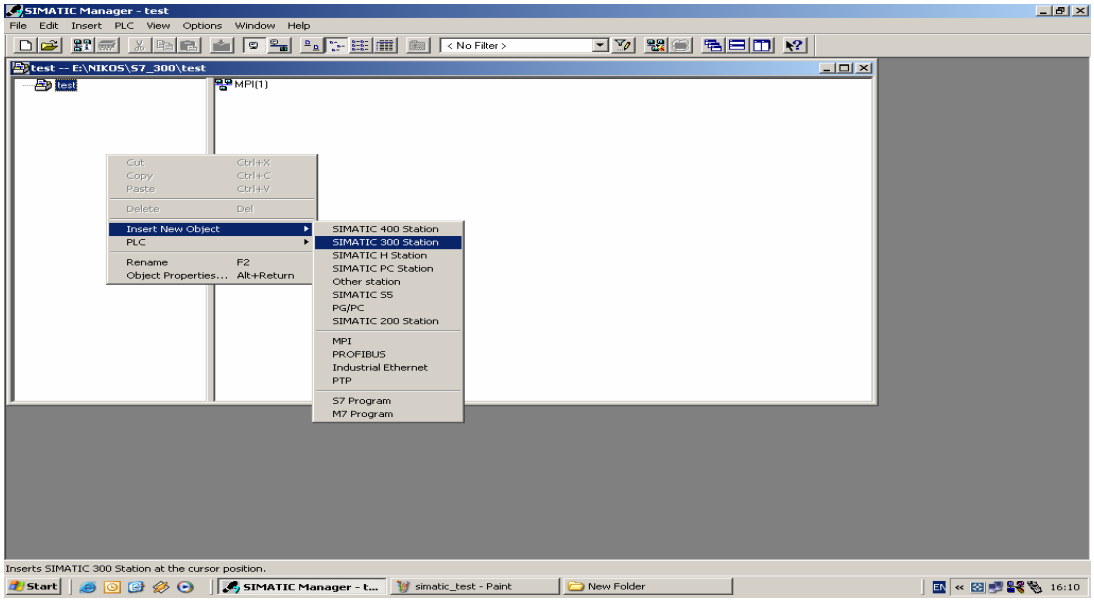
4. Εισαγωγή σταθμού

Μετά την δημιουργία του project (όπως φαίνεται στο παρακάτω σχήμα)

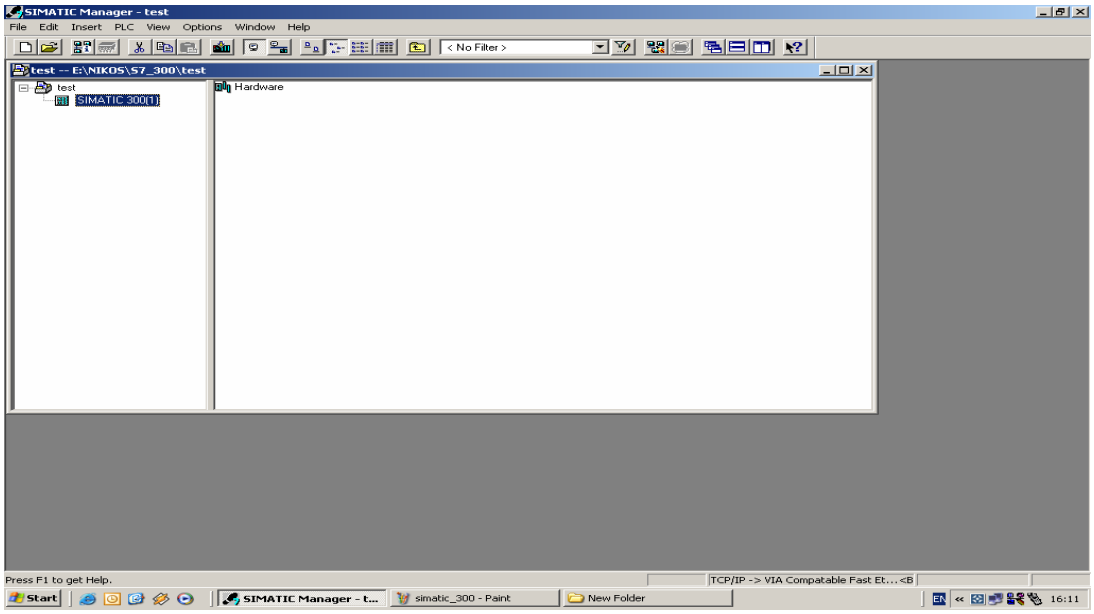


θα δηλώσουμε όλα τα PLC που υπάρχουν στην εγκατάστασή μας. Στο παράδειγμα μας θα φτιάξουμε έναν σταθμό S7-300. Αυτό γίνεται επιλέγοντας το project (κάνοντας κλικ στο όνομα του) και κατόπιν :

Insert New Object > SIMATIC 300 Station

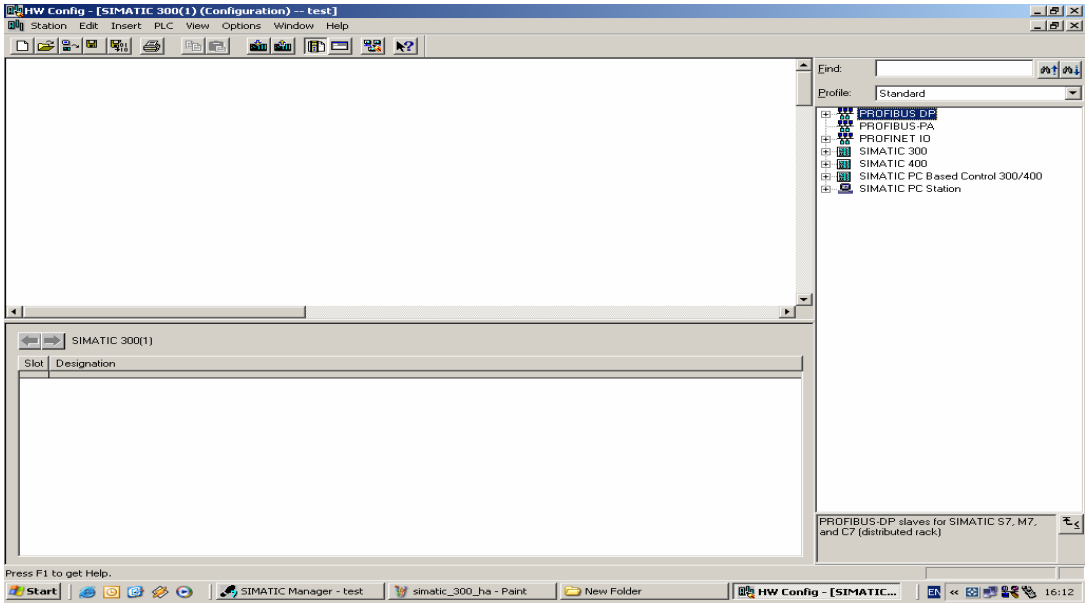


Το σύστημα μας προτείνει ένα όνομα για το σταθμό κι εμείς το αλλάζουμε (αν θέλουμε) με ένα όνομα της επιλογής μας.

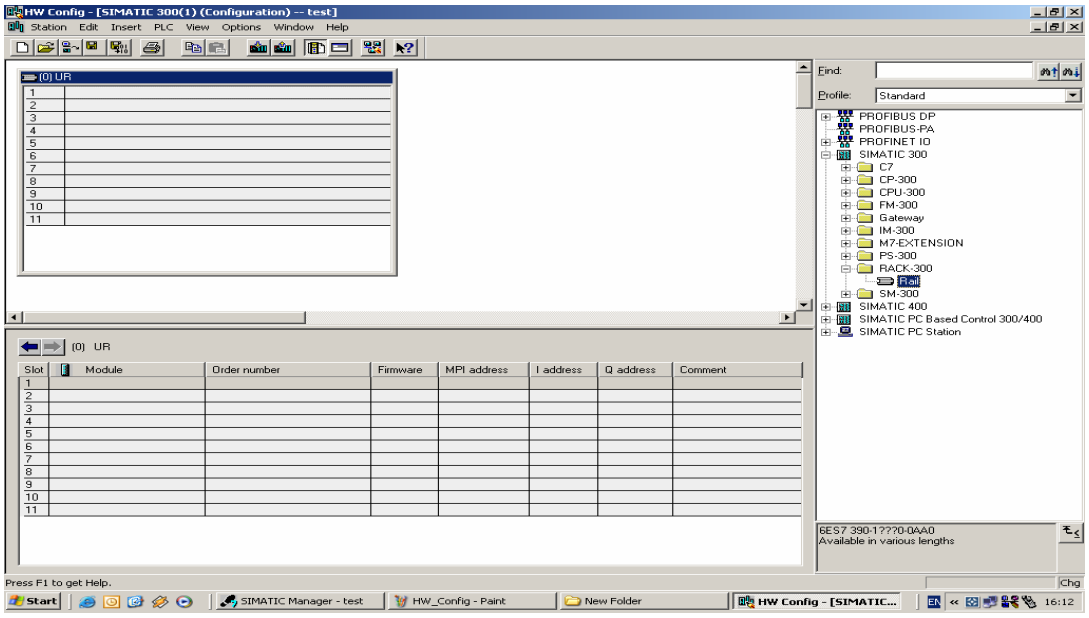


5. Ορισμός υλικού HW Config

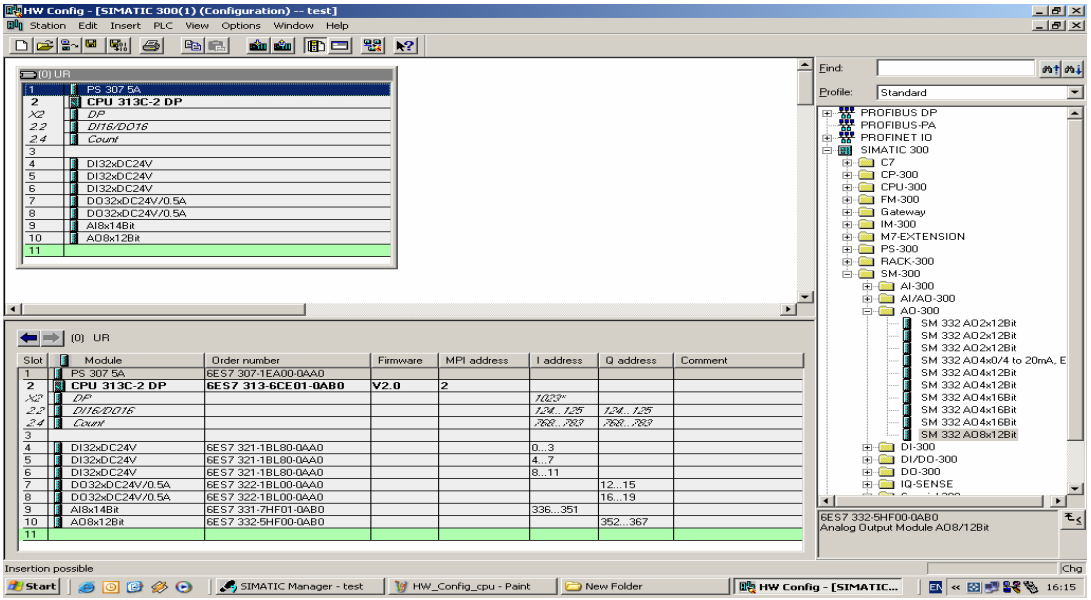
Επιλέγοντας το όνομα του σταθμού εμφανίζεται ένα εικονίδιο με την ένδειξη Hardware. Με διπλό κλικ στο εικονίδιο αυτό καλούμε το πρόγραμμα HW Configurations. Το παράθυρο που εμφανίζεται αποτελείται από 3 περιοχές – τη σύντομη περιγραφή (πάνω αριστερά) , την αναλυτική (με κωδικό και διευθύνσεις, κάτω αριστερά) και τον κατάλογο του Hardware.



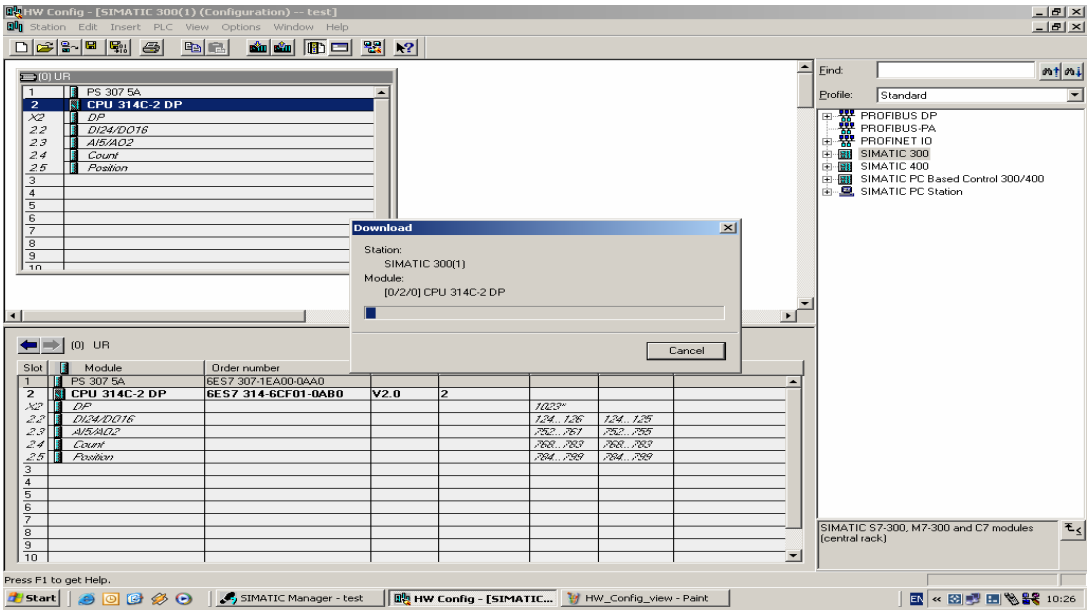
Από τον κατάλογο του Hardware επιλέγουμε πρώτα το rack οπού κατόπιν θα τοποθετήσουμε τις κάρτες (**Simatic 300 – Rack 300 – Rail**)



Οι θέσεις πάνω στο rack είναι τυποποιημένες ,δηλαδή στην θέση 1 έχουμε το τροφοδοτικό , θέση 2 τη CPU, θέση 3 τη κάρτα διασύνδεσης αν υπάρχει και άλλο rack και θέσεις από 4 έως 11 κάρτες SM, FM και CP. Στο παρακάτω σχήμα φαίνεται το hardware ενός σταθμού PLC με το τροφοδοτικό του, την CPU και τις κάρτες ψηφιακών και αναλογικών εισόδων και εξόδων.

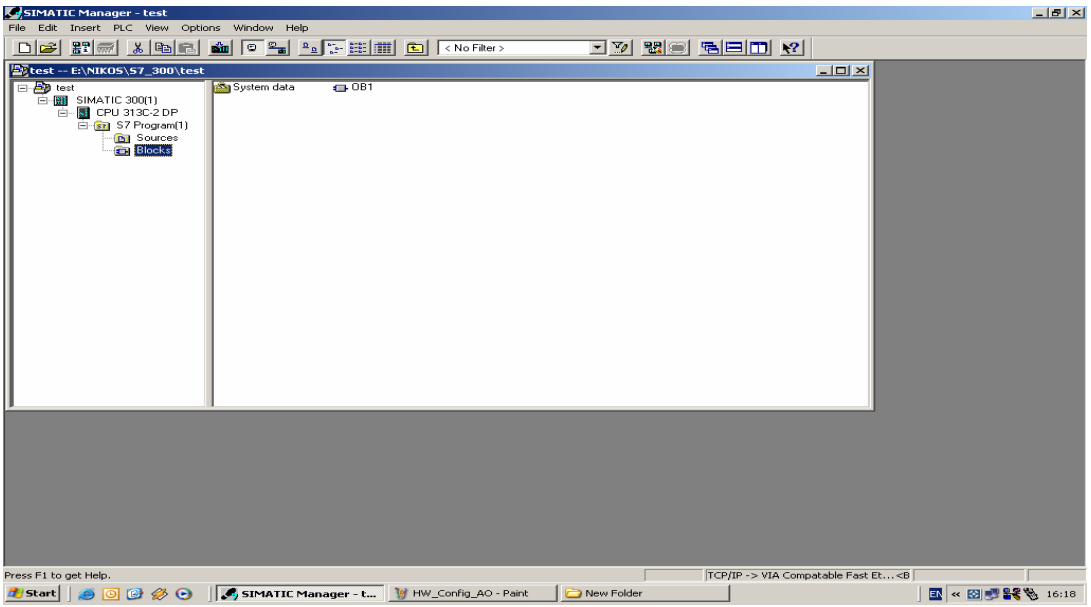


Αφού έχουμε τελείωση με την δημιουργία του Hardware , η επόμενη ενέργεια που πρέπει να κάνουμε είναι ο έλεγχος για τη σωστή τοποθέτηση και παραμετροποίηση του Hardware . Αυτό γίνεται επιλέγοντας **Station > Consistency Check**. Εάν είναι όλα εντάξει αποθηκεύουμε τα δεδομένα στο σκληρό δίσκο επιλέγοντας **Station > Save and Compile .**Επόμενη κίνηση είναι να <<κατεβάσουμε>> (Download) το Hardware στο PLC .

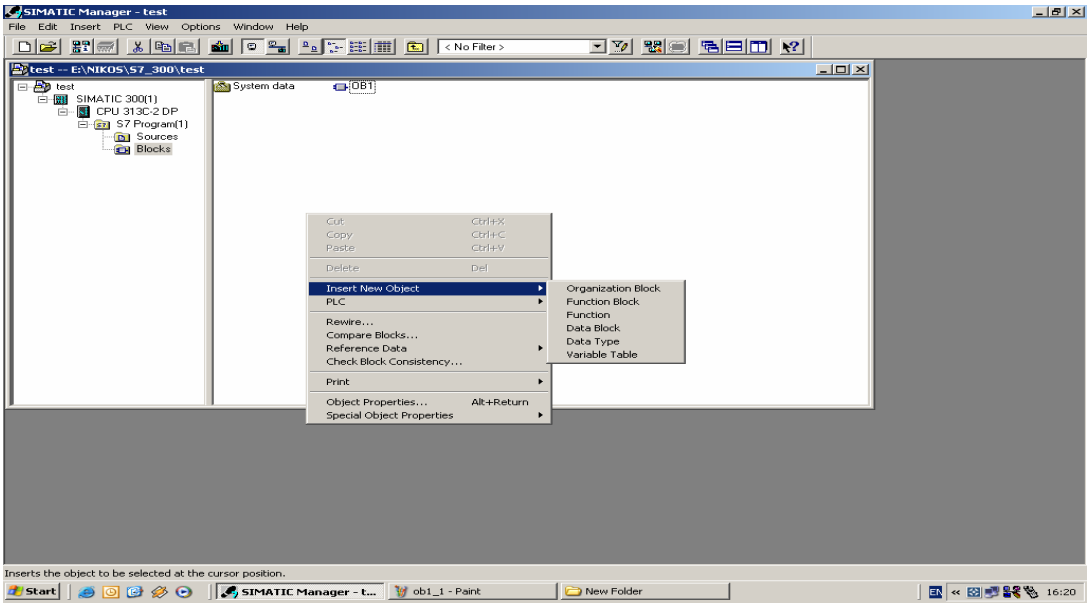


6. Δημιουργία μπλοκ στο S7

Μετά τον ορισμό του Hardware είμαστε σε θέση να γράψουμε το πρόγραμμα για την υλοποίηση του αυτοματισμού. Στο project που έχουμε δημιουργήσει επιλέγουμε το εικονίδιο Blocks όπως φαίνεται στην παρακάτω εικόνα.



Εδώ υπάρχει ήδη το OB1 που είναι ένα ειδικό μπλοκ το οποίο εκτελείται κυκλικά και είναι το μόνο που ελέγχει άμεσα ο επεξεργαστής του PLC. Εάν θέλουμε να δημιουργήσουμε και άλλα μπλοκ για την ευκολότερη επεξεργασία του προγράμματος μας επιλέγουμε Blocks δεξί κλικ με το ποντίκι **Insert New Object >** όπως φαίνεται στην παρακάτω εικόνα .



Εδώ έχουμε να επιλέξουμε μια από τις προτεινόμενες επιλογές:

- **Organizations Block (OB)**
- **Function (FC)**
- **Function Block (FB)**
- **Data Block (DB)**
- **Data Type (UDT)**
- **Variable Table VAT)**

Organization Blocks (OB). Αυτά παίζουν το ρόλο του μεσάζοντα μεταξύ του λειτουργικού συστήματος και του προγράμματος μας. Το λειτουργικό σύστημα της CPU καλεί τα OB όταν συμβεί κάποιο ειδικό γεγονός, όπως για παράδειγμα κάποιο σφάλμα στις επικοινωνίες ή κάποιο λάθος στον προγραμματισμό. Το σημαντικότερο και άκρως απαραίτητο σε κάθε πρόγραμμα είναι το OB1, αφού αυτό επεξεργάζεται κυκλικά η CPU και από δω καλούνται τα υπόλοιπα block του προγράμματος μας. Ένα άλλο σημαντικό μπλοκ είναι το OB100, το οποίο εκτελείται μόνο μια φορά, μετά την παροχή τάσης, στο σύστημα και πριν την συνεχόμενη κυκλική επεξεργασία του OB1. Στη συνέχεια θα εξηγήσουμε πως επεξεργάζεται κυκλικά η CPU τα προγράμματα της. Όταν τροφοδοτούμε το σύστημα με τάση, ο επεξεργαστής ελέγχει ένα υπάρχον δημιουργημένο OB100. Εάν υπάρχει, εκτελεί τις εντολές που υπάρχουν σ' αυτό και ξεκινά την κυκλική επεξεργασία. Στην αρχή διαβάζονται τα σήματα από τις κάρτες εισόδων και εξόδων και η κατάσταση τους αποθηκεύεται στη μνήμη

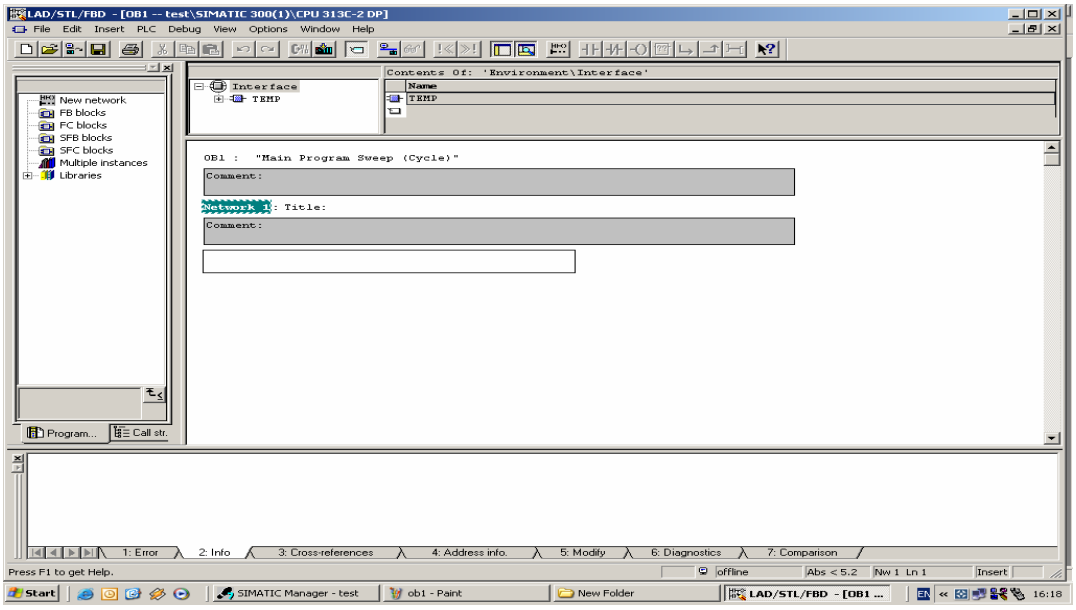
απεικόνισης εισόδων (PII). Με βάση τις τιμές στο PII εκτελείται το πρόγραμμα και παράγονται οι τιμές εξόδων, οι οποίες δεν εκτελούνται άμεσα, αλλά αποθηκεύονται προσωρινά στην μνήμη απεικόνισης εξόδων (PIQ). Όταν ολοκληρωθεί η εκτέλεση του προγράμματος, οι τιμές που βρίσκονται αποθηκεμένες στο PIQ μεταφέρονται για υλοποίηση στις κάρτες εξόδων. Μετά την ενέργεια αυτή αρχίζει και πάλι ο κύκλος σάρωσης από την αρχή .

Function Blocks (FB). Αυτό είναι ένα μπλοκ που γράφουμε κώδικα, το χαρακτηριστικό του δε είναι ότι μπορεί να έχει «μνήμη». Αυτό γίνεται με τη βοήθεια του μπλοκ δεδομένων (Instance data block). Τα FB μπορούν να παραμετροποιηθούν έχουν δε μια μεταβλητή μνήμη όπου αποθηκεύονται οι τρέχουσες τιμές τους.
Functions (FC). Αυτά είναι κατ' εξοχήν μπλοκ που γράφουμε το πρόγραμμά μας για σύνθετες και μη διαδικασίες. Μπορούν να παραμετροποιηθούν και να χρησιμοποιηθούν σε περιπτώσεις που έχουμε επαναλαμβανόμενη λογική στο πρόγραμμά μας. Μέσα από τα μπλοκ αυτά μπορούμε να «καλούμε» άλλα του ίδιου ή και διαφορετικού τύπου, για να οργανώσουμε καλύτερα το πρόγραμμα μας.

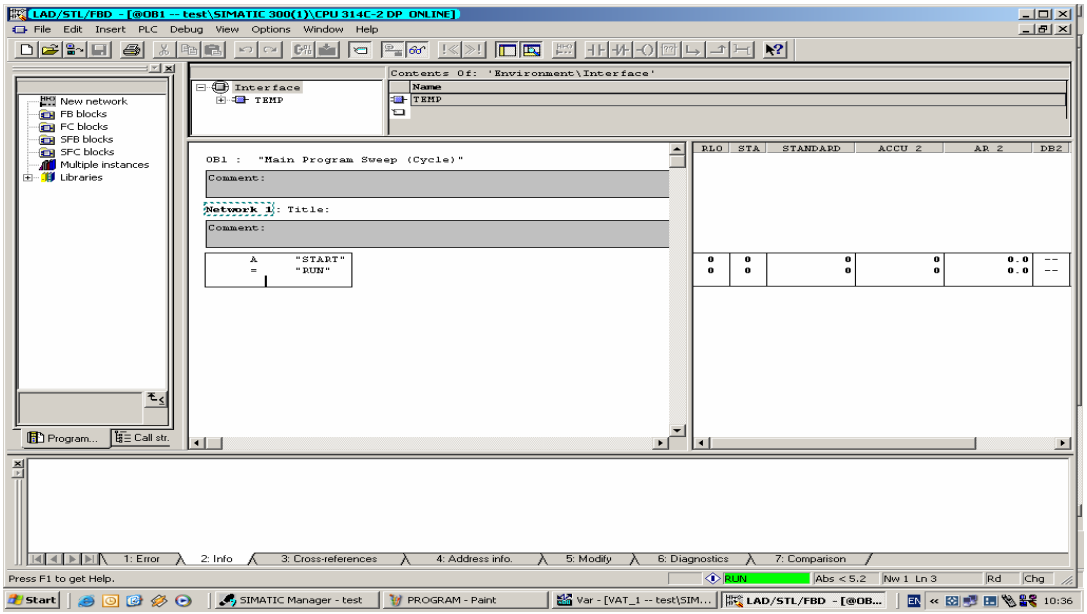
Data Blocks (DB). Αυτά τα μπλοκ είναι ένας χώρος όπου αποθηκεύονται τα δεδομένα του προγράμματος . Τα δεδομένα αυτά μπορεί να είναι για παράδειγμα μετρήσεις από αναλογικά σήματα, τιμές για set point, χρόνοι για χρονικά, περιεχόμενα απαριθμητών. Προγραμματίζοντας το DB ορίζουμε σε ποια μορφή θα σώζονται τα δεδομένα, με ποια σειρά και τι τύπου δεδομένα είναι (δυαδικά, ακέραιος αριθμός, πραγματικός αριθμός κλπ.) Τα μπλοκ δεδομένων διαφέρουν από τα άλλα μπλοκ ως προς το περιεχόμενο τους (περιέχουν μόνο αριθμούς όχι εντολές), και ως προς το τρόπο κλήσης τους στο πρόγραμμα. Τα DB μπορεί να είναι για κοινή χρήση σε όλο το πρόγραμμα (Global DB) η συνδεδεμένα με κάποιο συγκεκριμένο FB (Instance FB).

7. Δημιουργία κώδικα με χρήση του LAD/STL/FBD Editor

Η προσθήκη κώδικα στο OB1, στα FB και στις FC γίνεται κάνοντας διπλό κλικ πάνω τους, οπότε καλείται από το σύστημα η εφαρμογή LAD/STL/FBD η οποία αποτελεί έναν συντάκτη εντολών (editor).

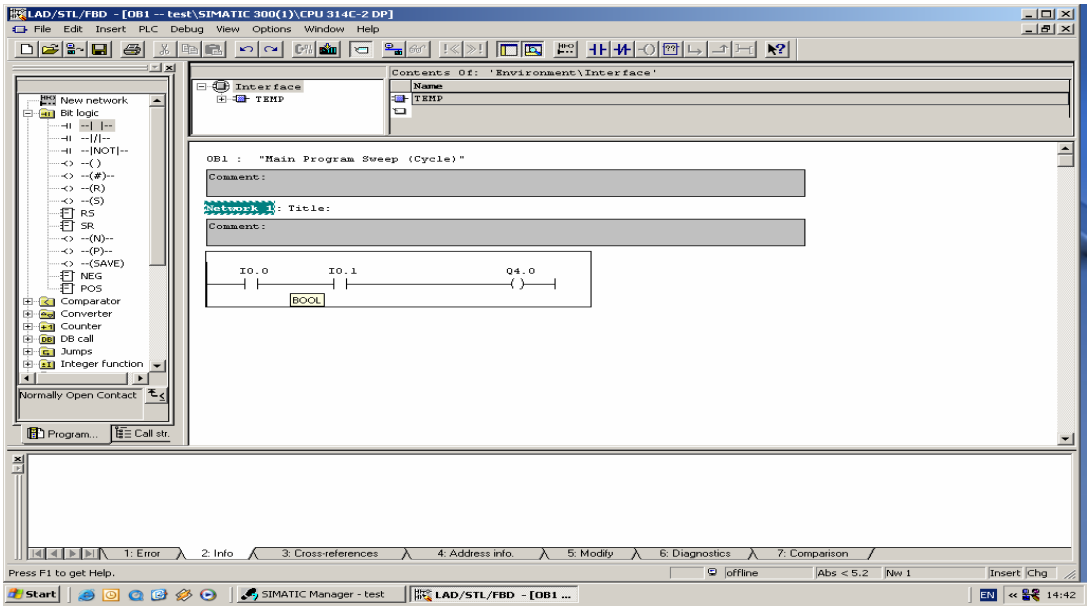


Η συγκεκριμένη εφαρμογή είναι εξαιρετικά «δυνατή», αφού εκτός των άλλων έχει τη δυνατότητα σύνδεσης και on-line παρακολούθησης του προγράμματος, όπως αυτό εκτελείται στο PLC . Αυτό γίνεται επιλέγοντας **Debug > Monitor ή επιλέγοντας το εικονίδιο με τα γυαλλάκια από το Toolbar**



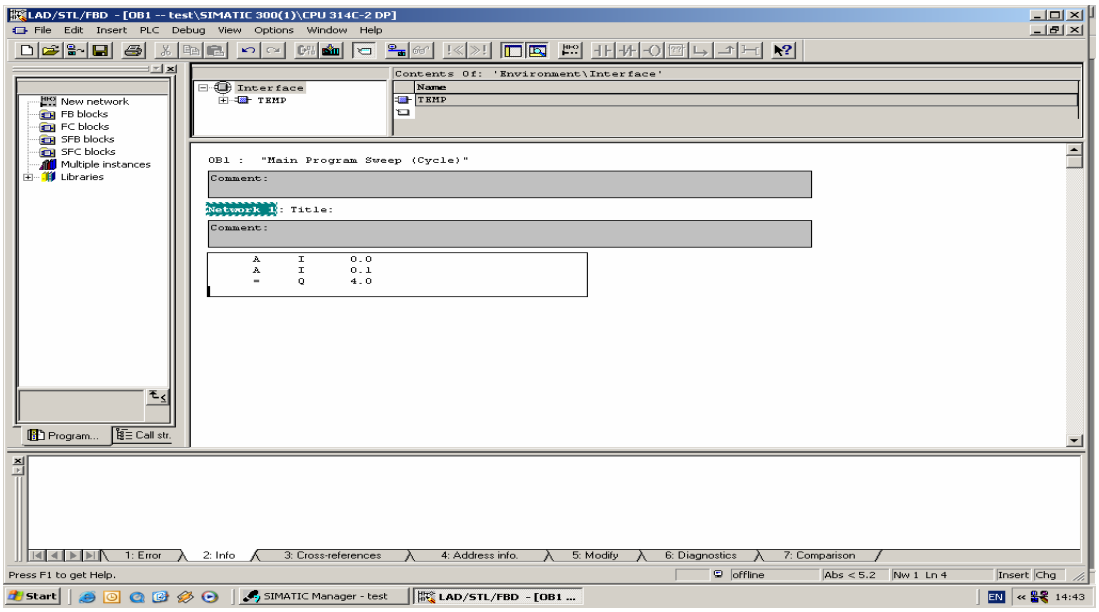
Για τη δημιουργία κώδικα υπάρχουν τρεις γλώσσες προγραμματισμού στο STEP 7 που μπορούν να χρησιμοποιηθούν σύμφωνα με τις προτιμήσεις και τη γνώση μας:

- **LAD (LADDER Logic ή γλώσσα ηλεκτρολογικών γραφικών) .**Η πρώτη γλώσσα προγραμματισμού είναι η Ladder Logic (LAD) που είναι μια γλώσσα γραφικών που χρησιμοποιεί ηλεκτρομηχανικά σύμβολα και επιτρέπει ουσιαστικά τη μεταφορά του ηλεκτρολογικού σχεδίου στο PLC. Με τη γλώσσα αυτή η εκπαίδευση των τεχνικών, που ήταν συνηθισμένοι στον κλασσικό αυτοματισμό, γινόταν εύκολα και γρήγορα, αφού δεν άλλαζε ουσιαστικά την εργασία σχεδιασμού του αυτοματισμού. Η γλώσσα LADDER χρησιμοποιεί όχι την Ευρωπαϊκή τυποποίηση στο σχεδιασμό των ηλεκτρικών επαφών, αλλά την Αμερικάνικη. Αυτό ίσως οφείλεται στο γεγονός ότι τα πρώτα PLC αναπτύχθηκαν στην Αμερική. Όμως στη συνέχεια αυτός ο σχεδιασμός "βόλεψε" και έτσι διατηρήθηκε και από τις Ευρωπαϊκές εταιρίες, με αποτέλεσμα σήμερα να είναι καθιερωμένος. Στο παρακάτω σχήμα φαίνεται το περιβάλλον της γλώσσας Ladder.

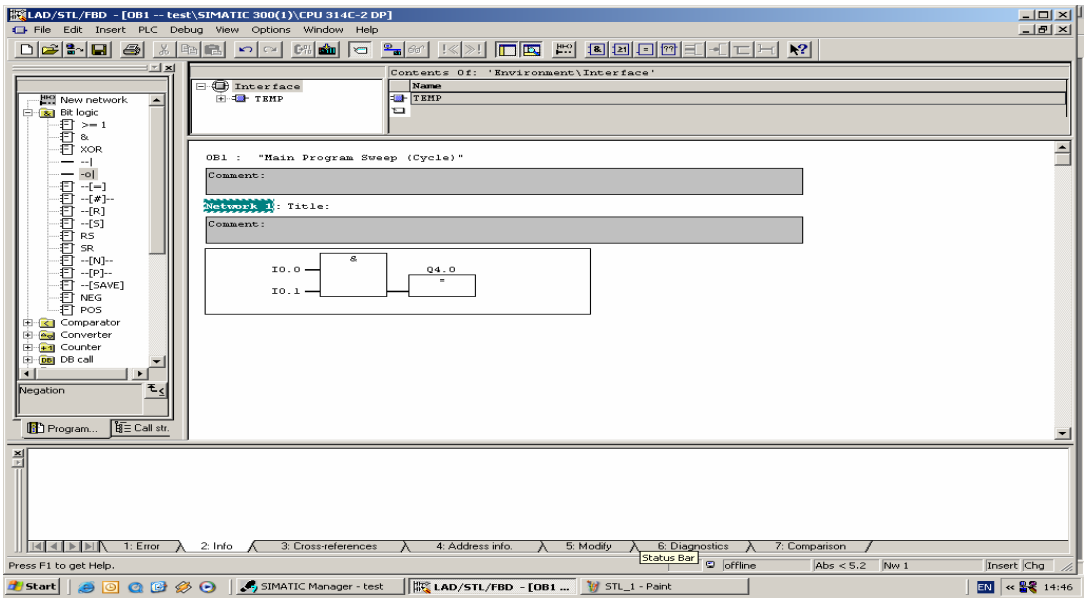


- **STL (Statement List ή γλώσσα λογικών εντολών).** Η δεύτερη γλώσσα προγραμματισμού είναι η Statement List (STL) που αναπτύχθηκε σχεδόν ταυτόχρονα με τη LADDER, αν και οι εταιρίες έδειξαν στην αρχή δισταγμό

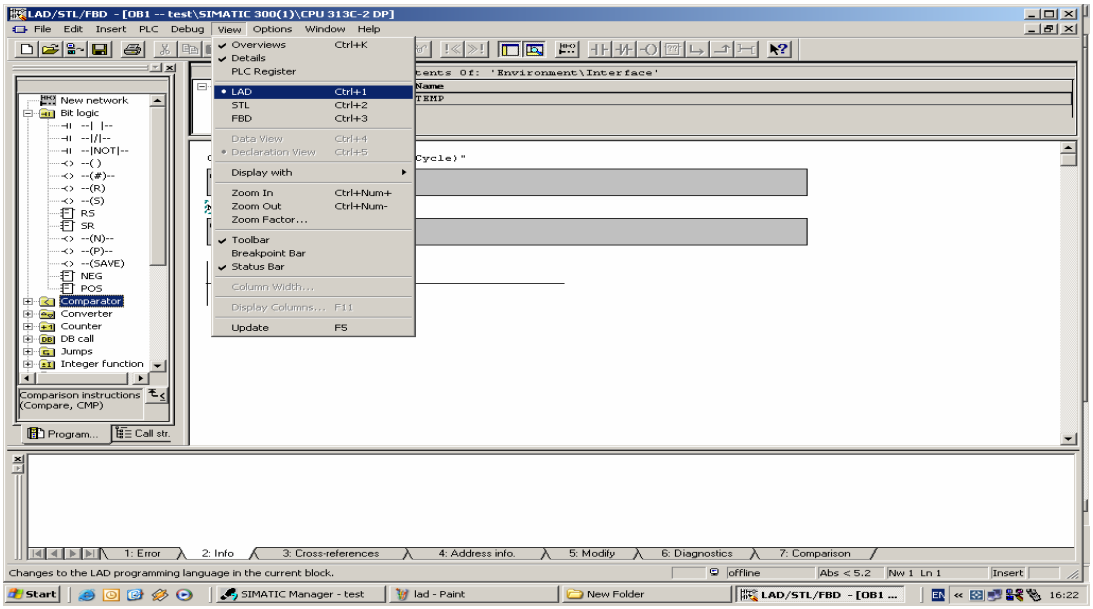
να την προωθήσουν, φοβούμενες μην τρομάξουν το τεχνικό κατεστημένο της βιομηχανίας. Η γλώσσα αυτή δημιουργεί λίστα προγράμματος με εντολές, που αντιστοιχούν στις λογικές πύλες (AND, OR NOT κτλ). Στην αρχή η γλώσσα αυτή ήταν πολύ φτωχή και περιοριζόταν μόνο στις βασικές Boolean εντολές. Στη συνέχεια οι γλώσσες αυτές αναπτύχθηκαν πολύ και συναντά κανείς σε αυτές στοιχεία από τις γλώσσες των υπολογιστών και κυρίως των γλωσσών Assembly. Ο προγραμματισμός σε αυτή τη γλώσσα απαιτεί από το χρήστη να έχει στοιχειώδεις γνώσεις προγραμματισμού. Στο παρακάτω σχήμα φαίνεται το περιβάλλον της Statement List.



- **FBD (Function Block Diagram ή λογικών γραφικών).** Η τρίτη γλώσσα είναι η Function Block Diagram η οποία χρησιμοποιεί και αυτή γραφικά, αλλά αντί του ηλεκτρολογικού σχεδίου του αυτοματισμού χρησιμοποιεί το αντίστοιχο λογικό. Η γλώσσα αυτή είναι νεότερη και δεν χρησιμοποιείται από όλες τις εταιρίες. Στο παρακάτω σχήμα φαίνεται το περιβάλλον της Function Block Diagram.



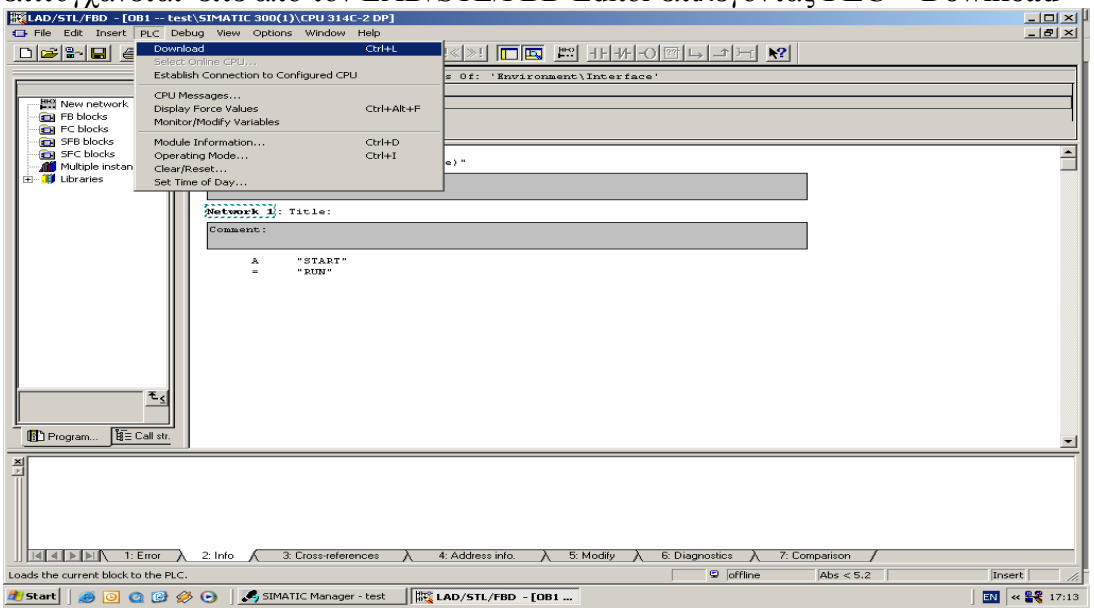
Επιλέγοντας τώρα τη γλώσσα προγραμματισμού με την οποία θέλουμε να γράψουμε των κώδικα , αυτό γίνεται επιλέγοντας **View > LAD / STL / FBD** όπως φαίνεται στην παρακάτω εικόνα:



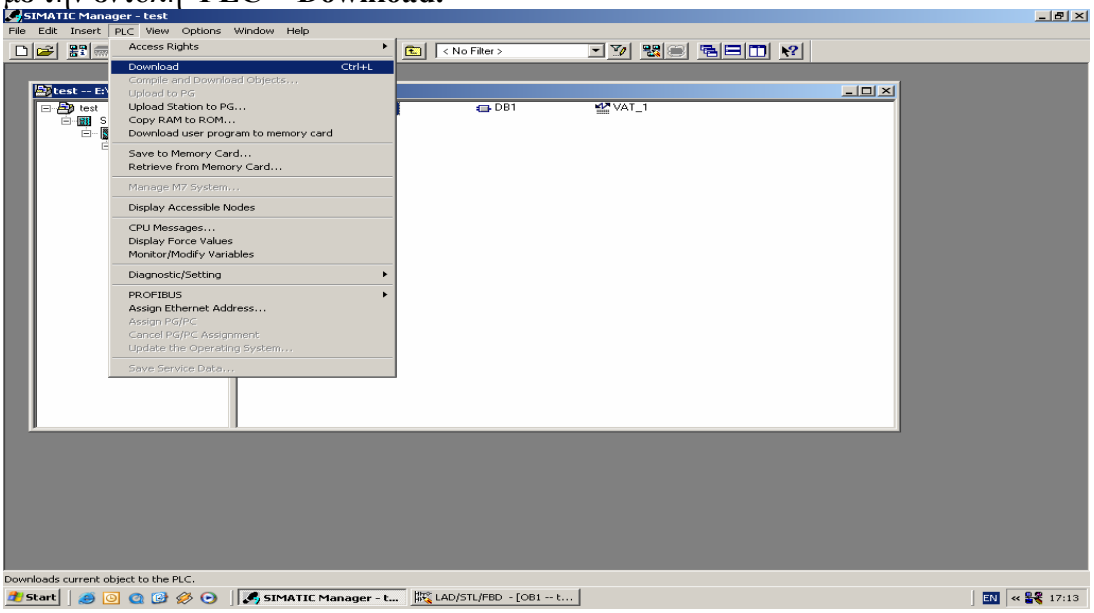
Μετά την δημιουργία του προγράμματος μας το αποθηκεύουμε στο σκληρό δίσκο επιλέγοντας **File > Save**
Αν έχουμε λάθη στη σύνταξη των εντολών του προγράμματος μας το STEP 7 δεν μας αφήνει να το αποθηκεύουμε.

8. Μεταφορά του προγράμματος στο PLC

Η δημιουργία των Blocks και ο κώδικας εντολών που έχουμε γράψει μέχρι τώρα είναι στον υπολογιστή μας , το επόμενο βήμα είναι τα μεταφέρουμε στο PLC. Αυτό επιτυγχάνεται είτε από τον LAD/STL/FBD Editor επιλέγοντας **PLC > Download**



Είτε από το Simatic Manager και εφόσον έχουμε επιλέξει τον κατάλογο Blocks ξανά με την εντολή **PLC > Download.**



Επιλέγοντας το δεύτερο τρόπο μεταφέρουμε στο PLC όλα Block που έχουμε δημιουργήσει μέσα στο project , ενώ με την πρώτη επιλογή το συγκεκριμένο Block που έχουμε ανοιχτώ.