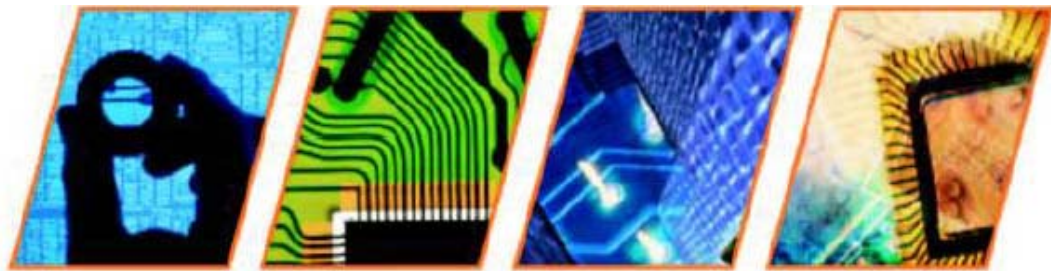


Remote monitoring of sensors and remote management of devices through TCP/IP protocol.



Student: Kotronakis Nickolaos
Professor: Dr. Barbounakis Ioannis

Abstract

Transmission Control Protocol/Internet Protocol (TCP/IP) is the protocol suite with the greatest penetration in modern networks. Although it is a connectionless mechanism without any guarantee on Quality Service, it has been adopted in most critical applications, even in real-time multimedia communications.

During the last years, there is a growing trend to minimize the TCP/IP protocol stack, in an attempt to fit it into micro-controllers. Ipsil Ipp8930 is such a micro-controller, which supports extra functionalities such as SPI, I²C, Serial apart from the classical Ethernet (UTP) interface. Ipp8930 also integrates Web Server and provides the designer-engineer with many configuration capabilities through an integrated proprietary protocol (Ipsil Control Protocol - ICP).

During this thesis, I was appointed with the task of working on the Ipp8930 micro-controller, through proper handling of digital/analog inputs/outputs in a series of remote monitoring and management cases. I had to write programming code, including JavaScript, Java applets and html. Part of my work had to do with the design and development of sensor boards dealing with luminosity measurements and motion detection.

The cases, which I developed, examined and finally carried out are indicative of the possibilities that existing technologies can achieve when they integrate TCP/IP network connectivity available everywhere nowadays.

Περιεχόμενα

1. Εισαγωγή

- 1.1 Γενικά
- 1.2 Κυρίαρχος (pervasive) υπολογισμός και κυρίαρχα (pervasive) δίκτυα
- 1.3 Γιατί το κυρίαρχο σχέδιο δικτύων χρειάζεται την προσεκτική εφαρμοσμένη μηχανική:
- 1.4 Η δικτύωση δεν είναι μόνο "Ιστοσελίδες"
- 1.5 Οι ενσωματωμένοι web-server είναι πράγματι χρήσιμοι
- 1.6 Αλληλουχία και συγχρονισμός
- 1.7 TCP/IP, Πανταχού παρόν
- 1.8 Το TCP/IP είναι μια σύνθετη, αλλά αδιαίρετη, λίστα πρωτοκόλλου
- 1.9 Το πρωτόκολλο Modbus TCP
 - 1.9.1 Το πρωτόκολλο Modbus/TCP σε σύγκριση με το πρωτόκολλο ASCII και RTU
 - 1.9.2 Τεχνικά Χαρακτηριστικά του Modbus/TCP

2. Περιγραφή πτυχιακής εργασίας και εργαλείων

- 2.1 Γενικά
- 2.2 IPμ8930
 - 2.2.1 Χαρακτηριστικά του IPμ8930
 - 2.2.2 Χαρακτηριστικά της πλακέτας IPμ8930
 - 2.2.3 Architectural Overview
- 2.3 HTML
 - 2.3.1 Μια συνοπτική ιστορία του HTML
- 2.4 Java
 - 2.4.1 Λόγοι για να χρησιμοποιήσουμε τη Java
 - 2.4.2 Πιθανά μειονεκτήματα της Java
 - 2.4.3 JavaScript
 - 2.4.4 Τι είναι ένα Applet;
 - 2.4.5 Πώς γράφουμε ένα Applet.
 - 2.4.5.1 Ένα απλό Applet

3. Διεκπεραίωση εργασίας

- 3.1 Γενικά
- 3.2 Webholes
 - 3.2.1 Κώδικας ιστοσελίδας με ενσωματωμένα τα webholes
- 3.3 ICP (IPμ8930 Control Protocol)
 - 3.3.1 Δημιουργία πακέτων ICP για έλεγχο καναλιών
 - 3.3.2 Δημιουργία πακέτων ICP για έλεγχο σειριακής
 - 3.3.3 Compile and archive
 - 3.3.3.1 Σημειώσεις εφαρμογής
 - 3.3.3.2 Χρήσιμες οδηγίες για τη δημιουργία και τη χρησιμοποίηση των αρχείων JAR σε ιστοσελίδες
- 3.4 Ipsil Config Utility (ICU)
 - 3.4.1 Configuring the I/O Channels
 - 3.4.2 Διαμόρφωση του Serial Interface
 - 3.4.3 Διαμόρφωση αισθητήρων
 - 3.4.4 Διαμόρφωση συναγερμών
- 3.5 Κατασκευή αισθητηρίων
- 3.6 Υλοποίηση ενδεικτικών εφαρμογών

Αναφορές Παράρτημα Α

ΚΕΦΑΛΑΙΟ 1^ο

“ΕΙΣΑΓΩΓΗ”

Περίληψη

Με την αυξανόμενη πανταχού παρουσία του διαδικτύου, ένας πρωτοφανής αριθμός συσκευών είναι άμεσα συνδεδεμένος με το δίκτυο. Μερικοί το βλέπουν αυτό ως συναρπαστική ευκαιρία, που δίνει τη δυνατότητα για πρωτοφανή πρόσβαση και έλεγχο στις γεωγραφικά διασκορπισμένες περιοχές, επιτρέποντας με αυτόν τον τρόπο τεράστιο εκσυγχρονισμό των λειτουργιών ανεφοδιασμών αλυσίδων-καταστημάτων, βελτίωση των διαδικασιών εξυπηρέτησης πελατών και πρωτοπορία στις βιομηχανικές αρχιτεκτονικές συστημάτων ελέγχου. Αυτές οι δυνατότητες μας οδηγούν σε σημαντικά ενισχυμένες υπηρεσίες, σε πιο ικανοποιητική εμπειρία για τον χρήστη, και σε μεγάλες μειώσεις δαπανών λειτουργίας και συντήρησης που όλοι επιθυμούμε. Εντούτοις, αναλαμβάνοντας την τεχνική μελέτη των διανεμημένων ενσωματωμένων (embedded) εφαρμογών, και σχεδιάζοντας συσκευές με διαδικτυακή συνδεσιμότητα (internet enable), δημιουργούμε αποτελεσματικά δίκτυα με πρωτοφανή πολυπλοκότητα, ταυτόχρονα, με δύο σημαντικούς τρόπους. Αφενός, ο πιο προφανής είναι ότι η κλίμακα του δικτύου, ο μεγάλος αριθμός κόμβων του, όταν κάθε συσκευή – ακόμη και ένας λαμπτήρας – θεωρείται δικτυακά συνδέσιμος, δεν έχει προηγούμενο στην ιστορία της τεχνολογικά εφαρμοσμένης μηχανικής. Αφετέρου, εάν δεν δοθεί επαρκής προσοχή σε μερικούς από τους κρυμμένους κινδύνους της τεχνικής μελέτης τέτοιων διαδικτύων, η προκύπτουσα πολυπλοκότητα ή οι δυσμενείς περιπλοκές μπορούν να οδηγήσουν σε ασυμβατότητες και σχεδιαστικές ανεπάρκειες, το κόστος των οποίων μπορεί να υπερβεί ακόμα και τη μέγιστη δυνητική αξία της αρχικής δικτυωμένης εφαρμογής.

1.1 Γενικά

Ζούμε στην εποχή του πανταχού παρόντος διαδικτύου. Το κόστος των (δια)δικτυακά συνδέσιμων συσκευών μειώνεται ραγδαία. Έχει ειπωθεί ότι το απείρως προσιτό, συνδεδεμένο με τον παγκόσμιο ιστό φυσικό περιβάλλον, ενωμένο με ένα πλήθος μικροσκοπικών κεντρικών υπολογιστών-εξυπηρετητών θα μπορούσε να σημαίνει μια ζωή εύκολης πρόσβασης στην πληροφορία. (Borriello 2000).

Τέτοια πανταχού παρούσα συνδεσιμότητα πραγματοποιείται χάρη στην ευρεία και απεριόριστη διαθεσιμότητα των ISPs και των σημείων σύνδεσης με το διαδίκτυο σε προσιτές τιμές. Επιπλέον, η εισαγωγή των συσκευών με ημιαγωγούς χαμηλού κόστους επιτρέπει την ενσωμάτωση έξυπνων υπολογιστών μέσα σε σχεδόν οποιαδήποτε συσκευή που έχει μια παροχή ηλεκτρικού ρεύματος. Συσκευές όπως το StrongARM CPU και System-on-a-chip βασισμένες στην οικογένεια επεξεργαστών ARM επιτρέπουν τις high-end συσκευές να δικτυωθούν. Συγχρόνως, οι πολύ χαμηλού κόστους συσκευές δικτύωσης όπως το Ipsil Ipru-series TCP Controller-on-a-Chip (Ipsil 2000) επιτρέπουν, ακόμη, μικρές συσκευές, μεμονωμένους διακόπτες και μονάδες αισθητήρων, της τάξεως των μερικών ευρώ ή δολαρίων να συνδέονται επικερδώς με το διαδίκτυο ή το LAN.

1.2 Κυρίαρχος υπολογισμός (pervasive computing) και κυρίαρχα (pervasive) δίκτυα

Ο κυρίαρχος (pervasive) υπολογισμός είναι μια έκφραση που περιγράφει το όνειρο της ενσωματωμένης (embedded) νοημοσύνης και του ενσωματωμένου (embedded) ελεγκτή σε πολλά αντικείμενα της καθημερινής ζωής μας. Εντούτοις, τέτοιοι ενσωματωμένοι ελεγκτές, συνήθως, δεν θα είναι μεμονωμένα κέντρα μεγάλης σοφίας και ανώτερης διάνοιας με μόνο σκοπό τον διαλογισμό πάνω σε βαθιά φιλοσοφικά προβλήματα. Αντίθετα, η λειτουργία τους θα εξαρτάται εξ

ολοκλήρου από τη δυνατότητά τους να επικοινωνήσουν με άλλες συσκευές και υπολογιστές. Αυτό δείχνει ότι ο κυρίαρχος υπολογισμός (pervasive computing) θα είναι για πάντα ένα ανεκπλήρωτο όνειρο χωρίς ένα κυρίαρχο (pervasive) δίκτυο που θα τους συνδέσει όλους.

Ένα κυρίαρχο (pervasive) δίκτυο, για να είναι αληθινά χρηστικό, πρέπει να ακολουθεί ανοικτά πρότυπα και πρέπει ελεύθερα να διασυνδέεται με το παγκόσμιο διαδίκτυο. Εάν απαιτείται από μια συγκεκριμένη εφαρμογή, είναι δυνατό οι πραγματικές διασυνδέσεις να ελέγχονται ή να συντονίζονται απομονώνοντας "ευαίσθητα" LANs από το δημόσιο δίκτυο, ή με τη χρησιμοποίηση των "τειχών προστασίας" (firewall) και των μηχανισμών πιστοποίησης χρηστών. Αλλά αυτό πρέπει να είναι μια διοικητική απόφαση και δεν πρέπει να εφαρμοστεί με υιοθέτηση των ιδιόκτητων (proprietary) ή "κλειστών" τεχνολογιών δικτύωσης.

Σε αυτήν την παρουσίαση, θα μελετήσουμε μερικά από τα ζητήματα που δείχνουν γιατί η θέσπιση ανοικτών προτύπων, όπως το TCP/IP, είναι επιθυμητή για να προταθεί παντού, και γιατί τα συστήματα με τα "νησιά" των ιδιόκτητων (proprietary) δικτύων δεν είναι επιθυμητά. Τέλος, οι χαμηλού κόστους συσκευές ημιαγωγών "TCP-On-A-Chip" επιτρέπουν τη διαδικτυακή συνδεσιμότητα ακόμα και στα πιο φθηνά κυκλώματα.

1.3 Γιατί το κυρίαρχο (pervasive) σχέδιο δικτύων χρειάζεται τον προσεκτικό, τεχνικό σχεδιασμό πελάτη:

Τα κυρίαρχα δίκτυα (pervasive networks) μπορούν να κατασκευαστούν από τα γνωστά στοιχεία μεταγωγής πακέτων για δικτύωση υπολογιστών. Εντούτοις, είναι πολύ σημαντικό να αποφευχθούν μερικές κοινές παγίδες στο σχεδιασμό-μελέτη τέτοιων δικτύων. Χωρίς ιδιαίτερη προσοχή σε μερικές από τις απλές, αλλά σημαντικές φιλοσοφικές αρχές, είναι δυνατό να σχεδιαστούν ενσωματωμένες εφαρμογές με δίκτυα που να προκαλούν περισσότερη απογοήτευση και ζημιά από προστιθέμενη αξία. Τα δίκτυα των έξυπνων συσκευών και τα δίκτυα που συνδέουν πολλές συσκευές, κάθε μια με ενσωματωμένο υπολογισμό και δικτύωση, πρέπει να σχεδιαστούν καλά.

1.4 Η δικτύωση δεν είναι μόνο "Ιστοσελίδες"

Η προσθήκη διαδικτύωσης σε μια συσκευή με έναν ενσωματωμένο ελεγκτή, δεν είναι μόνο προσθήκη ενός εξυπηρετητή ιστοσελίδων (web-server). Ενώ είναι αλήθεια ότι η προσθήκη ενός web-server σε μια συσκευή έχει τις χρήσεις της, το πεδίο της δικτύωσης των συσκευών την υπερβαίνει. Στην πραγματικότητα, για πολλές εφαρμογές το HTTP, το πρωτόκολλο που χρησιμοποιείται για την πρόσβαση σε ιστοσελίδες, ταιριάζει ιδιαίτερα καλά. Αφετέρου, υπάρχουν άλλα, πιο κατάλληλα πρωτόκολλα που μπορούν να "τρέξουν" πάνω σε TCP/IP και ίσως να εξυπηρετήσουν την εφαρμογή καλύτερα.

1.5 Οι ενσωματωμένοι web-server είναι πράγματι χρήσιμοι

Φυσικά, δεν υπάρχει καμία αμφιβολία για τη χρησιμότητα των ενσωματωμένων εξυπηρετητών ιστοσελίδων (web-servers). Μια συσκευή με έναν ενσωματωμένο web-server θα επέτρεπε τα εγχειρίδια του χρήστη, τα βοηθητικά αρχεία και τις παραμέτρους διαμόρφωσης να αποθηκευτούν στην ίδια τη συσκευή, ξεπερνώντας έτσι το πολύ κοινό πρόβλημα των χαμένων εγχειριδίων χρήσης της συσκευής. Οποιαδήποτε στιγμή κατά τη διάρκεια της ζωής της συσκευής, οποιεσδήποτε ρυθμίσεις διαμόρφωσης μπορούν να εκτελεσθούν εύκολα χρησιμοποιώντας έναν κοινό φυλλομετρητή (browser). Είναι βεβαίως πιο άνετο να γίνει αυτό με έναν φυλλομετρητή (browser) από το να χρησιμοποιείται ένα μικρό LCD και κάποια κουμπάκια, τα οποία είναι συνήθως αυτά με τα οποία είναι εξοπλισμένες οι κοινές συσκευές όσον αφορά το user interface.

1.6 Αλληλουχία και συγχρονισμός

Κατά τον έλεγχο μιας συσκευής μέσα από οποιαδήποτε διασυνδετική διάταξη (interface), είτε πρόκειται για μια φυσική διασυνδετική διάταξη (interface) μέσα από το δίκτυο, ή διαφορετικά, ολόκληρη η αλληλεπίδραση του χρήστη ή της οντότητας ελέγχου με τη συσκευή μπορεί θεωρηθεί ως διάλογος που αποτελείται από μια ακολουθία αλληλεπιδράσεων με τη συσκευή. Σε ένα τέτοιο διάλογο, η σειρά των διαδικασιών και η ακεραιότητα κάθε βήματος είναι πολύ σημαντικά. Το HTTP πρωτόκολλο εκτελεί λειτουργίες ενός αρχείου τη φορά και έτσι δεν σχεδιάστηκε αρχικά να παρέχει τους εύκολους μηχανισμούς και να εκτελεί τους εκτεταμένους διαλόγους χωρίς αδικαιολόγητο κόστος.

Αφετέρου, οι περισσότερες αλληλεπιδράσεις με έξυπνες συσκευές, δηλ., εκείνες με τους κυρίαρχους (pervasive) κόμβους υπολογισμού ενσωματωμένους σε αυτές, δεν θα ήταν "ένα προς ένα" αλληλεπίδραση που περιλαμβάνει μόνο έναν ανθρώπινο χρήστη και συσκευή. Μάλλον, ένας μεγάλος αριθμός αλληλεπιδράσεων θα ήταν αυτός του ενός ανθρώπινου χειριστή που προσπαθεί να διαμορφώσει δύο ή περισσότερες συσκευές ταυτόχρονα, έτσι ώστε να μπορούν έπειτα να είναι συνδεδεμένες η μία με την άλλη. Επόμενες επικοινωνίες από τις συσκευές θα περιελάμβαναν συσκευές που αλληλεπιδρούν άμεσα η μια με την άλλη χωρίς ανθρώπινη επίβλεψη, σε πραγματικό χρόνο. Πρωτόκολλα προσανατολισμένα προς τη συναλλαγή πρωτοκόλλων, όπως οι Remote Procedures Calls (RPC) και οι μηχανισμοί επικοινωνίας, όπως η Composable Working Memory (Shrikumar 2000), ταιριάζουν καλύτερα με τέτοιες εφαρμογές.

Σαν απλό παράδειγμα, ένα κουδούνι με δικτυακή συνδεσιμότητα και μια φωτογραφική μηχανή με δικτυακή συνδεσιμότητα μπορούν να αγοραστούν χωριστά και να συνδεθεί το ένα με το άλλο μετά από την εγκατάσταση. Τα δύο θα πρέπει να είναι προγραμματισμένα έτσι ώστε, όταν χτυπά ένας επισκέπτης το κουδούνι, να στέλνει ένα σήμα στη ψηφιακή φωτογραφική μηχανή, η οποία μεταδίδει έπειτα μια εικόνα του ατόμου που είναι στην πόρτα του σπιτιού. Το γεγονός ότι οι δικτυωμένες συσκευές θα επέτρεπαν στον ιδιοκτήτη να ελέγξει την είσοδο του σπιτιού του και να επιτρέψει στους ανθρώπους να μπουν στο σπίτι από μακριά, αποκομίζει αναμφίβολο όφελος από την πανταχού παρουσία του διαδικτύου.

Σε οποιαδήποτε τέτοια εφαρμογή με πολλές συσκευές, είναι επιτακτικό οι συσκευές να είναι σε συγχρονισμό μεταξύ τους. Κατά συνέπεια, οποιοσδήποτε ενέργειες που μια συσκευή δημιουργεί για να εκτελεστούν σε μια άλλη συσκευή δεν πρέπει να οδηγούν σε ασυνέπειες ή ασύμβατες εσωτερικές καταστάσεις των δύο συσκευών. Περαιτέρω, αυτή η προϋπόθεση υπονοεί επίσης ότι σε οποιοσδήποτε (τριών ή περισσότερων συμβαλλόμενων συσκευών) αλληλεπιδράσεις, είτε όλες οι επικοινωνίες και ενέργειες εκτελεστούν είτε καμία από αυτές, πρέπει να έχουν ως αποτέλεσμα μια δεσμευμένη δράση. Αυτό πρέπει να γίνει για να εξασφαλιστεί η συνέπεια της κατάστασης των αλληλεπιδρώντων συσκευών.

1.7 TCP/IP, Πανταχού παρόν

Το TCP σημαίνει πρωτόκολλο ελέγχου μετάδοσης (Transmission Control Protocol), η γλώσσα του διαδικτύου. Είναι το πρωτόκολλο που τα chip των υπολογιστών χρησιμοποιούν, όταν «μιλάει» το ένα στο άλλο μέσω του διαδικτύου. Σύμφωνα με την σχεδιαστική αρχή του διαδικτύου, δηλαδή την αρχή από τερματικό σε τερματικό, το TCP αποδεικνύεται ότι περιβάλλεται από ένα σύνολο πολύ σημαντικών ευθυνών. Αυτές οι ευθύνες, που παρατίθενται όλες στο οριστικό πρότυπο του διαδικτύου γνωστό ως RFC1122 (RFC1122), αφορούν το τι κάνει το TCP/IP, ένα αρκετά σύνθετο και εξελιγμένο "σωρό" πρωτοκόλλου.

1.8 Το TCP/IP είναι ένας σύνθετος, αλλά αδιαίρετος, "σωρός" πρωτοκόλλου

Δεν υπάρχει παρόμοιο πράγμα όπως έναν απλουστευμένο σωρό TCP, ή stripped down σωρό TCP.

Για τους λόγους που αναφέρθηκαν παραπάνω, κάθε σωρός TCP απαιτείται για να εκτελεί όλες τις διαδικασίες που διευκρινίζονται στα πρότυπα του TCP. Η έλλειψη συμμόρφωσης με οποιαδήποτε από τις απαιτήσεις είναι επαρκής για να απαγορεύσει έναν τέτοιο σωρό από τη σύνδεση με το διαδίκτυο, γεγονός που κάνει το TCP ένα σύνθετο αλλά ώριμο και αξιόπιστο "σωρό" πρωτοκόλλων.

Κατ' αρχάς, το TCP είναι η οντότητα που έχει την ευθύνη να προσδιορίσει πόση κυκλοφορία μπορεί να αντέξει ένα δίκτυο με ασφάλεια διεκπεραίωσης. Αντίθετα από το τηλεφωνικό δίκτυο, το διαδίκτυο δεν κάνει καμία παρακράτηση του εύρους ζώνης. Κατά τη διάρκεια της ζωής μιας και μονής σύνδεσης, οι συνθήκες συμφόρησης στο δίκτυο θα μπορούσαν να ποικίλουν ευρέως. Αυτό απαιτεί το TCP να ρυθμίζει περιοδικά τους ρυθμούς μετάδοσής του. Εάν δεν ακολουθούνται οι διαδικασίες για να αυξηθούν το ποσοστό μετάδοσης όταν είναι διαθέσιμο το εύρος ζώνης του δικτύου, τότε δεν οδηγούμαστε σε απαραίτητη κακή απόδοση. Αντιθέτως, εάν το TCP δεν υπαναχωρεί, όταν το δίκτυο υπερφορτώνεται και παρουσιάζει αρχικά σημάδια συμφόρησης, αυτό είναι ιδιαίτερα επιβλαβές, δεδομένου ότι μια τέτοια συμπεριφορά μπορεί να προκαλέσει μια καταστροφική κατάρρευση της χρήσιμης κυκλοφορίας στο διαδίκτυο. Μια τέτοια κατάρρευση από συμφόρηση κάνει μεγάλα τμήματα του διαδικτύου να "σέρνονται". Το TCP εξασφαλίζει ότι αυτό δεν θα συμβεί επειδή συνεργάζεται με άλλες ροές TCP στο διαδίκτυο και προλαμβάνει μια τέτοια περίπτωση κακής συμφοράς στην αρχή της υπαναχωρώντας όταν η συμφόρηση φαίνεται επικείμενη.

Σαν δεύτερο παράδειγμα, μια ικανή ποσότητα πολυπλοκότητας των διαδικασιών του TCP προέρχεται από τις πληροφορίες κατάστασης που διατηρούνται για να ανακτηθούν κατάλληλα τα πακέτα που χάνονται στο διαδίκτυο. Μια ιδιαίτερα επιβλαβής κατάσταση προκύπτει, εάν ένα πακέτο που θεωρήθηκε χαμένο ξαναέρχεται στην επιφάνεια στο δίκτυο και γίνεται αποδεκτό ως έγκυρο πακέτο από μια εν λειτουργία σύνδεση TCP. Το TCP έχει σύνθετες διαδικασίες που έχουν ως σκοπό να ανιχνεύσουν τέτοια πακέτα και να τα απορρίψουν με ασφάλεια ώστε να μην τα μεταφέρουν προς μια εφαρμογή, όπου μπορεί να παρερμηνευθούν [3-9].

1.9 Το πρωτόκολλο Modbus/TCP

Το Modbus είναι ένα ανοικτό πρωτόκολλο. Σε συνδυασμό με το Ethernet TCP/IP αποτελεί μια ιδανική ανοικτή λύση για τη βιομηχανική αυτοματοποίηση. Επίσης, είναι ιδανικό για την εκτέλεση εντολών ανάγνωσης/γραφής σε απλές συσκευές δικτύου TCP/IP. Το πρωτόκολλο αυτό δημιουργήθηκε το 1970 ως ένα σειριακό πρωτόκολλο για επικοινωνίες μεταξύ συσκευών εργοστασίου. Από τότε έχει εξελιχθεί στο ανοικτό MODBUS/TCP σαν μέθοδος για να στέλνει MODBUS εντολές χρησιμοποιώντας το πρωτόκολλο TCP/IP

Το πρωτόκολλο MODBUS/TCP είναι βασισμένο στο TCP/ IP και παραλλαγή του πρωτοκόλλου Modbus RTU. Καλύπτει την αποστολή του μηνύματος Modbus σε ένα περιβάλλον ενδοδικτύου ή διαδικτύου.

1.9.1 Το πρωτόκολλο Modbus/TCP σε σύγκριση με το πρωτόκολλο ASCII και RTU

Το πρωτόκολλο MODBUS/TCP χρησιμοποιεί τη δυαδική κωδικοποίηση των σφαλμάτων και του μηχανισμού ανίχνευσης λάθους (error detection mechanism) TCP/IP για την ανίχνευση των σφαλμάτων μετάδοσης.

Το πρωτόκολλο ASCII χρησιμοποιεί μια δεκαεξαδική κωδικοποίηση ASCII των δεδομένων και ένα checksum 8 bit. Τα πλαίσια (frames) των μηνυμάτων

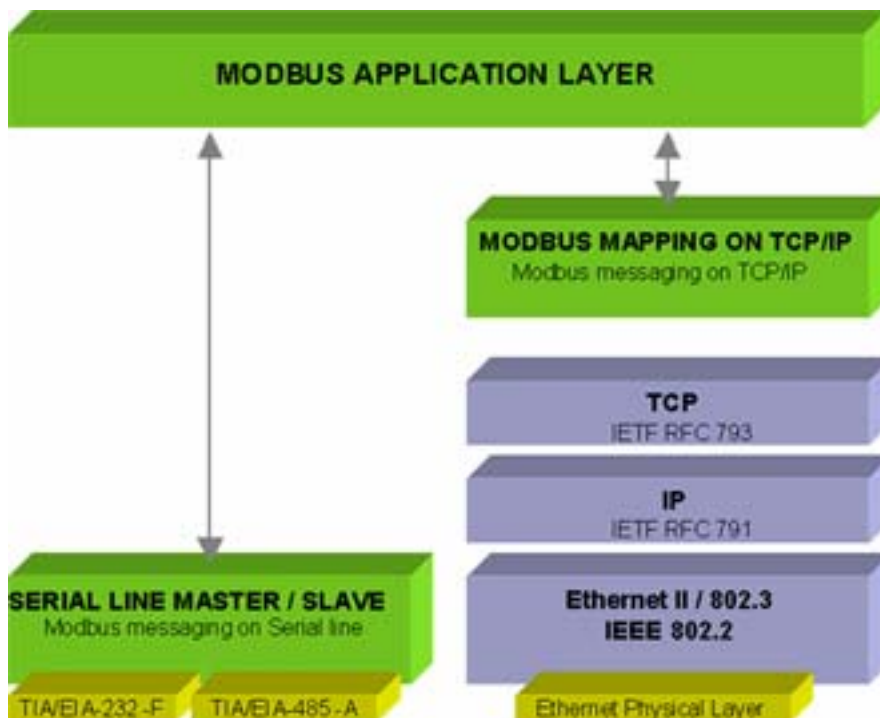
οριοθετούνται με τον χαρακτήρα ":" στην αρχή και μια ακολουθία του carriage return/linefeed στο τέλος.

Το πρωτόκολλο RTU χρησιμοποιεί τη δυαδική κωδικοποίηση των δεδομένων και έναν έλεγχο 16 bit CRC για την ανίχνευση των λαθών μετάδοσης. Τα πλαίσια (frames) των μηνυμάτων οριοθετούνται από ένα σιωπηλό διάστημα τουλάχιστον 3,5 φορές τον χαρακτήρα μετάδοσης πριν και μετά από την αποστολή του μηνύματος.

Σε αντίθεση με τα πρωτόκολλα ASCII και RTU, το MODBUS/TCP είναι ένα προσανατολισμένο πρωτόκολλο σύνδεσης (connection oriented protocol). Επιτρέπει τις ταυτόχρονες συνδέσεις στον ίδιο "σκλάβο" (slave) καθώς επίσης και τις ταυτόχρονες συνδέσεις σε πολλαπλάσιες δευτερεύουσες (slave) συσκευές.

Σε περίπτωση υπέρβασης χρονικής διάρκειας (timeout) ή αποτυχίας του TCP/IP πρωτοκόλλου, ο "κύριος" (master) θα κλείσει και θα ανοίξει πάλι τη σύνδεση και θα επανεκπέμψει έπειτα το μήνυμα.

1.9.2 Τεχνικά Χαρακτηριστικά του Modbus/TCP:



Το Modbus είναι ένα master/slave πρωτόκολλο με ημιαμφίδρομη μετάδοση.

Μπορεί να υπάρχει ένας "κύριος" (master) και μέχρι 247 συσκευές "σκλάβων" (slave) μπορούν να υπάρξουν ανά δίκτυο.

Το πρωτόκολλο καθορίζει τη διαμόρφωση και μεταφορά μηνυμάτων καθώς επίσης τα δεδομένα και τις λειτουργίες ελέγχου.

Το πρωτόκολλο δεν καθορίζει ένα φυσικό στρώμα δικτύων. Το Modbus λειτουργεί σε διαφορετικά φυσικά στρώματα δικτύων. Το σειριακό (serial) πρωτόκολλο λειτουργεί σε RS 232, RS 422 και RS 485 φυσικά δίκτυα. Το πρωτόκολλο TCP/IP λειτουργεί σε όλα τα φυσικά στρώματα δικτύων που υποστηρίζουν το TCP/IP. Αυτό συμβιβάζει το 10BASE-T και το 100BASE-T LANs καθώς επίσης και τα σειριακά στρώματα δικτύων PPP και SLIP.

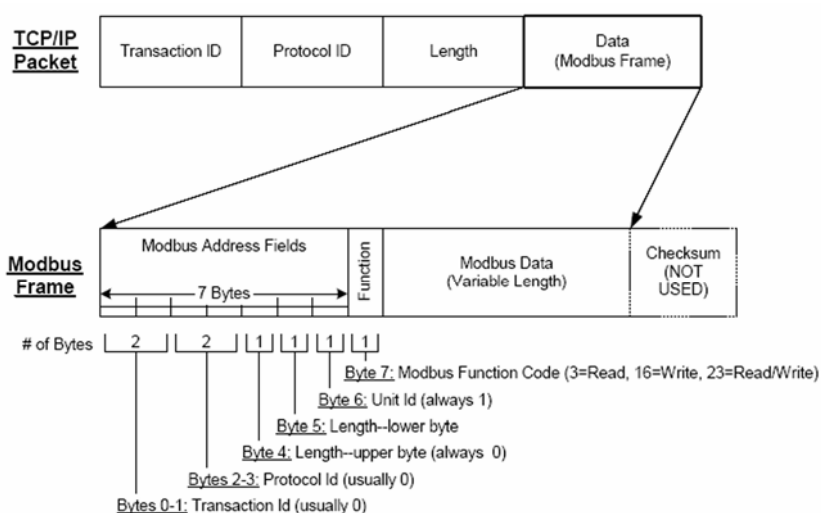
Οι εντολές Irsil/Modbus είναι μορφοποιημένες (formatted) χρησιμοποιώντας τα ανοικτά πρότυπα Modbus και τα διαμορφωμένα τμήματα δεδομένων του πακέτου TCP/IP που γράφεται στην καταχωρημένη θύρα (port) 502.

Το Modbus είναι ένα σταθερό πρωτόκολλο και ένα Modbus "συνομιλίας" αποτελείται από ένα αίτημα (request) και μια απάντηση (response). Ένα αίτημα (read, write, or read/write) από έναν χρήστη (όπως το PC) θα παράγει ένα πακέτο

απάντησης από το Ipsil IPμ8931 Board. Αυτό το πακέτο απάντησης θα περιέχει ή τα δεδομένα ή έναν κώδικα αποτελέσματος που δείχνει την επιτυχή ή ανεπιτυχή εκτέλεση της προηγούμενης εντολής Modbus.

Όπως φαίνεται στην εικόνα 1.1, το πλαίσιο Modbus είναι σπασμένο σε τρία τμήματα: τμήμα διευθύνσεων 7 byte για Modbus, ένα πεδίο με κώδικα λειτουργίας και ένα πεδίο (field) με δεδομένα μεταβλητού μήκους. Ανεξάρτητα από τη λειτουργία που εκτελείται, η μορφή (format) (όχι απαραίτητως των δεδομένων!) των πρώτων οκτώ byte, δηλαδή τα πεδία με διευθύνσεις Modbus και ο κώδικας λειτουργίας, είναι ίδια. Οι πληροφορίες που ζητούνται ή/και παρέχονται ως απάντηση συμπεριλαμβάνονται στο πεδίο δεδομένων Modbus μεταβλητού μήκους.

Η μορφή αυτού του πεδίου δεδομένων ποικίλει ανάλογα με το αίτημα λειτουργίας και ανεξάρτητα απ' το γεγονός ότι το πακέτο Modbus αντιπροσωπεύει ένα αίτημα δεδομένων ή μια απάντηση σε ένα αίτημα δεδομένων. Στο σχεδιάγραμμα για την εντολή ανάγνωσης του Modbus (ο κώδικας λειτουργίας 3) φαίνεται λεπτομερώς παρακάτω.



Εικόνα 1.1: Μορφή πλαισίου MODBUS

0	1	2	3	4	5	6	7	8	9	10	11	BYTE#
00	00	00	00	00	06	01	03	00	00	00	01	DATA
Modbus Addr.Prefix				Len	UID	FC	RegH	RegL	LenH	LenL	FIELDS	

Εικόνα 1.2 : παράδειγμα πακέτου αιτήματος MODBUS. Η εντολή αυτή διαβάζει ένα ή περισσότερους καταχωρητές της Ipsil..

0	1	2	3	4	5	6	7	8	9	10	BYTE#
00	00	00	00	00	05	01	03	02	00	34	DATA
Modbus Addr. Prefix				Len	UID	FC	Bytes	Data1	Data2	FIELDS	

Εικόνα 1.3 : παράδειγμα πακέτου απάντησης MODBUS

Μπορείτε να δείτε τα πεδία των κωδικών διευθύνσεων και λειτουργίας μέσα στο πλαίσιο Modbus παρακάτω:

Αίτημα:

Byte 7: FC: 03

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

Byte 8-9: αριθμός καταχωρητή

Byte 10-11: μετρητής λέξεων (word count) (1-125) (word=2 bytes)

Απάντηση:

Byte 7: FC: 03

Byte 8-9: μετρητής Byte απάντησης ($B=2 \times \text{word count}$)

Byte 9-(B+8): τιμές μεταβλητών

Σημείωση: το "byte 9-(B+8)" δεν είναι μαθηματικός υπολογισμός αλλά η περιοχή των bytes των δεδομένων Modbus με το 9^ο byte όπου "B" είναι ο αριθμός των bytes του Modbus

Εξαιρέσεις:

Byte 7: FC: 83 (hex)

Byte 8: κωδικός εξαίρεσης: 01 ή 02

Όλες οι εξαιρέσεις σημειώνονται προσθέτοντας το 0X80 στον κώδικα του αιτήματος. Οι εξαιρέσεις του Modbus βρίσκονται στο παράρτημα A, παράγραφος 8.3.4 [2,9]

ΚΕΦΑΛΑΙΟ 2^ο

“Περιγραφή πτυχιακής εργασίας
και εργαλείων”

2.1 Γενικά

Με την συγκεκριμένη πτυχιική εργασία επιδιώκεται ο έλεγχος και η διαχείριση περιφερειακών συσκευών και αισθητηρίων μέσω του πρωτοκόλλου TCP/IP. Σε αυτό θα μας βοηθήσουν το Developer Kit IPμ8930 της Ipsil (<http://www.ipsil.com>), ο προγραμματισμός σε java applet διαφόρων προγραμμάτων και η δημιουργία ιστοσελίδων .htm/.html τις οποίες υποστηρίζει το παραπάνω kit. Για το compile και το "τρέξιμο" των java applet χρησιμοποιήσαμε jsdk και jre αντίστοιχα. Η έκδοση με την οποία είναι συμβατό το Development kit είναι 1.4.1.07. Με τις νεότερες εκδόσεις παρουσιάζονται προβλήματα κατά την κωδικοποίηση των αρχείων με αποτέλεσμα να μην μπορούν να εκτελεστούν σωστά.

Τα αισθητήρια με τα οποία ασχοληθήκαμε είναι ένας φωτοδιακόπτης και ένας αισθητήρας κίνησης (πομπός - δέκτης). Ο φωτοδιακόπτης, του οποίου μπορεί να ρυθμιστεί η ευαισθησία του, διεγείρεται με το φως που πέφτει πάνω στην φωτοαντίσταση. Εφαρμογές αυτού είναι συστήματα συναγερμού και ασφάλειας, αυτόματο άναμμα και σβήσιμο φώτων ανάλογα με τον εξωτερικό φωτισμό κλπ. Εμείς καταφέραμε με επιτυχία το αυτόματο αλλά και το χειροκίνητο άναμμα και σβήσιμο φώτων! Το αισθητήριο της κίνησης αποτελείται από ένα πομπό ρυθμιζόμενων συχνοτήτων (0-70KHz) και ένα ανιχνευτή συχνοτήτων 38KHz. Εφαρμογές αυτού είναι συστήματα συναγερμού και ασφάλειας, έλεγχος ατόμων που περνούν από ένα σημείο κ.λ.π.

2.2 IPμ8930

Το IPμ8930 είναι ελεγκτής δικτύων γενικού σκοπού που είναι εύκολο να επιτηρηθεί από ένα κεντρικό υπολογιστή δικτύου, να ελέγξει και να επικοινωνήσει με τους μακρινούς αισθητήρες, ενεργοποιητές και σχεδόν οποιοσδήποτε συσκευές μέσω σειριακής θύρας, μέσω του πρωτοκόλλου TCP/IP. Το IPμ8930 είναι σχεδιασμένο είτε για αυτόνομη εργασία είτε για σύνδεση με ένα άλλο μικροελεγκτή.

Το IPμ8930 αποτελείται από τα παρακάτω:

- IPμ8930 Module
- IPμ8930 Developer Board
- Regulated +5V DC Power plug/transformer
- RJ-45 to DB9 adapter cable

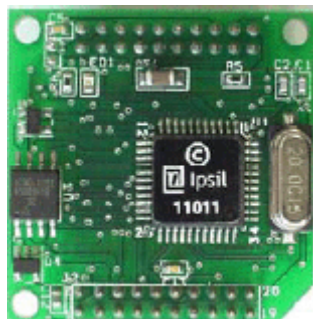
Στην καρδιά του IPμ8930 Developer Kit είναι το module IPμ8930, ένας συμπαγής TCP/IP ελεγκτής και webserver που επιτρέπει στους υπεύθυνους για την ανάπτυξη νέων προϊόντων να προσθέσουν γρήγορα την σύνδεση δικτύων. Το IPμ8930 συνδυάζει έναν ελεγκτή TCP/IP, HTTP - webserver, έναν κόμβο TCP Modbus και έναν μετατροπέα A/D σε έναν ενιαίο, μικρό (1.3x1.4"/3.3x3.4cm) daughterboard. Το IPμ8930 έχει ως σκοπό να επιτρέψει απομακρυσμένη παρακολούθηση και έλεγχο μέσω TCP/IP χωρίς το κόστος και την πολυπλοκότητα των παραδοσιακών λύσεων που απαιτούν τη γνώση σχετικά με το πώς να υποστηρίξουν και χρησιμοποιήσουν προγράμματα σε λειτουργικά συστήματα πραγματικού χρόνου. Το module IPμ8930 είναι ικανό να λειτουργεί αυτόνομα ή ως TCP/IP περιφερειακή συσκευή για ένα χωριστό MCU. Στην εικόνα 2.1 παρουσιάζεται η πάνω όψη του IPμ8930 σε πραγματικό μέγεθος.

Εκτός από τη δυνατότητα του χρήστη να αποθηκεύει και να ανακτεί ιστοσελίδες στο IPμ8930, οι υπεύθυνοι για την ανάπτυξη μπορούν να έχουν πρόσβαση στις I/O θύρες στο IPμ8930 χρησιμοποιώντας Modbus TCP, δημοφιλές πρότυπο για πρόσβαση συσκευών σε ένα δίκτυο βιομηχανικών αυτοματοποιήσεων.

Επίσης, το IPμ8930 Developer Kit περιλαμβάνει την πλακέτα IPμ8930. Το module IPμ8930 ενσωματώνεται στην πλακέτα IPμ8930. Η προς ανάπτυξη

πλακέτα (development kit) IPμ8930 είναι εφοδιασμένη με τροφοδοσία, 10BaseT network jack και ποικίλα φινιρίσματα που κάνουν εύκολο το στήσιμο, τον προγραμματισμό και το τεστάρισμα του IPμ8930.

Στη επόμενη ενότητα παρέχονται επιπλέον πληροφορίες για τα χαρακτηριστικά του IPμ8930 και της προς ανάπτυξη IPμ8930 πλακέτας.



Εικόνα 2.1: η IPμ8930 πλακέτα

2.2.1 Χαρακτηριστικά του IPμ8930

Το IPμ8930 παρέχει τα παρακάτω χαρακτηριστικά:

- Υποστηρίζει και τα δύο TCP/IP and UDP—read & write
- Περιλαμβάνει HTTP v1.0 compliant embedded webserver
- Έχει 8 αναλογικές ή ψηφιακές θύρες για παρακολούθηση / έλεγχο εξωτερικών συσκευών.
- 10-bit ADC
- Σειριακή θύρα
 - o Με ροή ελέγχου (CTS/RTS, XON/XOFF, or None)
 - o Μπορεί να χρησιμοποιηθεί για επικοινωνία με εξωτερικές σειριακές συσκευές.
- Ικανότητα να προσδιοριστούν τα webholes σε HTML σελίδες που μπορούν να «γεμίσουν» με τιμές από προσδιορίσιμες I/O θύρες.
- Δυνατότητα διαχείρισης μόνο του ή ως περιφερειακή συσκευή της εξωτερικής MCU χρησιμοποιώντας την σειριακή θύρα.
- Πλήρη διαχείριση ιδιοτήτων δικτύου και συστήματος συμπεριλαμβανομένων:
 - o Υποστήριξη για δυναμική/στατική IP διεύθυνση.
 - o Καθορισμός από τον χρήστη IP διεύθυνσης.
- Συμμόρφωση με το RFC-1122, το πρότυπο για TCP/IP host στο internet.
- Υποστηρίζει τα παρακάτω πρωτόκολλα:
 - o ARP, DHCP, ICMP (ping), Modbus, and Ipsil Control Protocol (v1.0)
- 4 megabit (512KB) μνήμη για τις ιστοσελίδες
- Time stamp functionality
- Σύστημα 500 αρχείων μεταβαλλόμενου μεγέθους

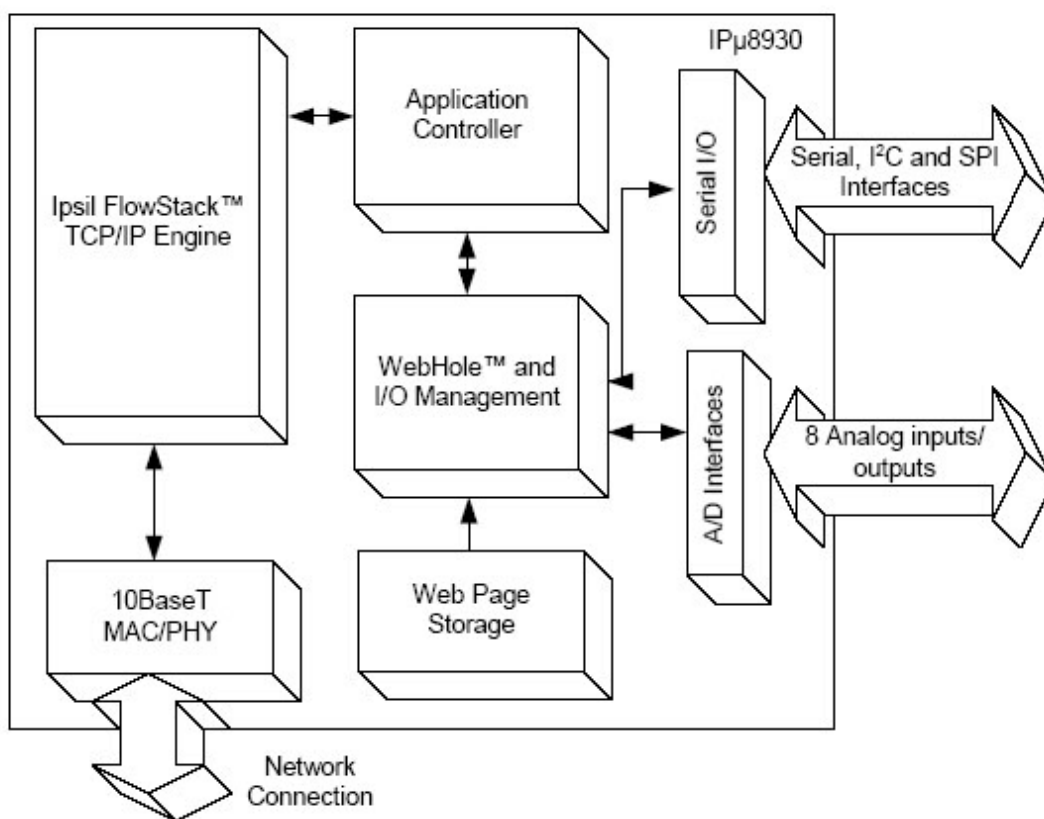
Χωρίς οποιοδήποτε πρόσθετο υλικό, ένας χρήστης μπορεί να φορτώσει τις ιστοσελίδες από το IPμ8930 board και να ελέγχει τις αναλογικές/ψηφιακές γενικές I/O θύρες στην αναπτυξιακή πλακέτα. Με πρόσθετες συσκευές που συνδέονται στην σειριακή θύρα, ένας χρήστης μπορεί να ελέγξει και να επιτηρήσει μία ευρεία ποικιλία των συσκευών.

2.2.2 Χαρακτηριστικά της πλακέτας IPμ8930

Η πλακέτα IPμ8930 είναι σχεδιασμένη να βοηθήσει γρήγορα τα προς ανάπτυξη προγράμματα χρησιμοποιώντας το IPμ8930 module. Η πλακέτα IPμ8930 περιέχει τα παρακάτω χαρακτηριστικά:

- Διεπαφή Ethernet με δύο ενιαία LED που δείχνουν την δραστηριότητα Rx/Tx
- RS-232 συνδεσιμότητα μέσω σειριακής διεπαφής RJ-45 (RJ-45 σε DB9 συμπεριλαμβανόμενο καλώδιο) με έλεγχο ροής
- 8 γενικά I/O κανάλια προσιτά μέσω της προς ανάπτυξη πλακέτας IPμ8930
- Υψηλές LEDs 5V, 3.3V, Tx, Rx, RTS, δραστηριότητα CTS
- Ενσωματωμένος I2C αισθητήρας θερμοκρασίας
- Κουμπί αναστοιχειοθέτησης
- Led ρεύματος και δραστηριότητας σύγκρουσης
- Εξωτερικός power jack

2.2.3 Architectural Overview



Εικόνα 2.2: IPμ8930 Block Diagram

2.3 HTML

Για να δημοσιεύσει τις πληροφορίες για τη γενική διανομή, κάποιος χρειάζεται μια παγκοσμίως κατανοητή γλώσσα, ένα είδος έκδοσης της μητρικής γλώσσας που όλοι οι υπολογιστές μπορούν ενδεχομένως να καταλάβουν. Η γλώσσα έκδοσης που χρησιμοποιείται από το World Wide Web είναι HTML (από τη γλώσσα σήμανσης HyperText).

Η γλώσσα HTML επιτρέπει:

- Να δημοσιευτούν σε απευθείας σύνδεση τα έγγραφα με τους τίτλους, το κείμενο, τους πίνακες, τους καταλόγους, οι φωτογραφίες, κ.λ.π.
- Να ανακτηθούν σε απευθείας σύνδεση τις πληροφορίες μέσω των υπερκείμενων συνδέσεων, με ένα κλικ ενός κουμπιού.

- Να σχεδιαστούν οι μορφές των επικοινωνιακών συναλλαγών με τις μακρινές υπηρεσίες, για τη χρήση στην έρευνα των πληροφοριών.
- Να περιληφθούν υπολογισμοί με λογιστικό φύλλο (spreadsheet), video clips, sound clips, και άλλες εφαρμογές άμεσα στα έγγρατά τους.

2.3.1 Μια συνοπτική ιστορία του HTML

Το HTML αναπτύχθηκε αρχικά από Tim Berners-Lee στο Κέντρο Πυρηνικών Μελετών και Ερευνών (CERN), και διαδόθηκε από τον φυλλομετρητή (browser) MOSAIC αναπτυγμένο σε NCSA. Κατά τη διάρκεια της δεκαετίας του '90 είχε ανθίσει με την εκρηκτική αύξηση του Ιστού. Κατά τη διάρκεια αυτής της περιόδου, το HTML έχει επεκταθεί με διάφορους τρόπους. Ο Ιστός εξαρτάται από τους συντάκτες και τους προμηθευτές ιστοσελίδων που μοιράζονται τις ίδιες συμβάσεις για το HTML. Αυτό έχει παρακινήσει την κοινή εργασία για τις προδιαγραφές για το HTML.

Το HTML 2.0 (Νοέμβριος 1995, βλέπε [RFC1866]) αναπτύχθηκε υπό την αιγίδα της ομάδας εργασίας εφαρμοσμένης μηχανικής διαδικτύου (IETF) για να κωδικοποιήσει την κοινή πρακτική (1994), την HTML + (1993) και την HTML 3.0 (1995, βλέπε [HTML3.0]) που προτάθηκαν ως οι πλουσιότερες εκδόσεις της HTML. Παρόλο που δεν είχαν λάβει τη συναίνεση στις συζητήσεις, αυτά τα σχέδια οδήγησαν στην υιοθέτηση μιας σειράς νέων χαρακτηριστικών γνωρισμάτων. Οι προσπάθειες της ομάδας εργασίας HTML της World Wide Web κοινοπραξίας να κωδικοποιήσει την κοινή πρακτική το 1996 οδήγησαν στο HTML 3.2 (Ιανουάριος 1997, βλέπε [HTML3.2]).

Οι περισσότεροι άνθρωποι συμφωνούν ότι τα έγγραφα HTML πρέπει να λειτουργούν καλά στους διαφορετικούς browsers και τις πλατφόρμες. Η επίτευξη της διαλειτουργικότητας χαμηλώνει τις δαπάνες στους ικανοποιημένους προμηθευτές δεδομένου ότι πρέπει να αναπτύξουν μόνο μια έκδοση ενός εγγράφου. Εάν δεν καταβληθεί προσπάθεια, υπάρχει πολύ μεγάλος κίνδυνος ο Ιστός να μεταβάλλεται σε έναν κόσμο ασύμβατων formats, μειώνοντας τελικά την εμπορική δυνατότητά του για όλους τους συμμετέχοντες.

Κάθε έκδοση του HTML έχει προσπαθήσει να συγκεντρώσει τη μεγαλύτερη συναίνεση μεταξύ των φορέων βιομηχανίας έτσι ώστε η επένδυση που γίνεται από τους ικανοποιημένους προμηθευτές να μην σπαταλάται και τα έγγρατά τους να μην γίνουν έτσι που να μην μπορούν να διαβαστούν σε μια μικρή χρονική περίοδο.

Το HTML έχει αναπτυχθεί με το όραμα ότι όλος ο τρόπος των συσκευών πρέπει να είναι σε θέση να χρησιμοποιήσει τις πληροφορίες για τον Ιστό: PCs με τα γραφικά περιβάλλοντος ποικίλων αναλύσεων οθόνης και χρώματος, τα κινητά τηλέφωνα, φορητές συσκευές, συσκευές για την ομιλία για την παραγωγή και την εισαγωγή, υπολογιστές με το υψηλό ή χαμηλό εύρος ζώνης, και τα λοιπά

2.4 Java

Σύμφωνα με τα λόγια του Tim Lindholm Senior Staff Engineer της JavaSoft σε άρθρο του τον Απρίλιο του 1996:

«Με την Java σαν γλώσσα επέκτασης, ένα πρόγραμμα πλοήγησης του Web θα μπορούσε να έχει απεριόριστες

δυνατότητες. Οι προγραμματιστές θα μπορούσαν να γράψουν τα applets μια φορά και αυτά μετά θα έτρεχαν σε οποιαδήποτε μηχανή, οπουδήποτε. Οι επισκέπτες σε σελίδες με Java δυνατότητες, θα μπορούσαν να χρησιμοποιήσουν το περιεχόμενο αυτών των σελίδων εφησυχασμένοι ότι τίποτα δεν θα μπορούσε να φθείρει την μηχανή τους.

Με τα applets σαν αρχικό σημείο εστίασης, η Java επέδειξε ένα νέο τρόπο για την διάθεση λογισμικού με το Internet. Αυτός ο νέος τρόπος προγραμματισμού

πηγαίνει πέρα από τα προγράμματα πλοήγησης. Πιστεύουμε ότι είναι ένας νεωτερισμός με την δυνατότητα να αλλάξει την πορεία της Πληροφορικής.»

2.4.1 Λόγοι για να χρησιμοποιήσουμε τη Java

Η Java είναι σχεδιασμένη για το Internet. Προγράμματα της Java μπορούν να τρέχουν σε προγράμματα πλοήγησης εφοδιασμένα με την κατάλληλη JVM (Java Virtual Machine). Αφού το πρόγραμμα κατέβει από το Internet εκτελείται στη μηχανή του πελάτη και όχι στο Server. Αυτά τα προγράμματα της Java είναι γνωστά σαν Java Applets.

Η Java είναι σχεδιασμένη με στόχο την εξασφάλιση ασφάλειας. Ένα Java Applet εκτελείται με απόλυτη ασφάλεια στη μηχανή του χρήστη και δεν μπορεί να κάνει τίποτα που να μην επιτρέπεται από την πολιτική ασφάλειας της JVM της μηχανής του πελάτη.

Η Java προσφέρει εκτέλεση σε όλους τους υπολογιστές. Με το επίπεδο αφαίρεσης υλικού της JVM ένα πρόγραμμα Java εκτελείται σε όλους τους υπολογιστές ανεξαρτήτως λειτουργικού Συστήματος. Αρκεί αυτές να είναι εφοδιασμένες με την κατάλληλη JVM (Java Virtual Machine)

Η Java είναι μια πλούσια γλώσσα προγραμματισμού. Η Java είναι πολύ ισχυρή αντικειμενοστραφής γλώσσα προγραμματισμού για το Internet (και όχι μόνο) και όχι μία απλή script γλώσσα για την σύνθεση συστατικών (π.χ. VBScript με ActiveX controls). Αυτό μας επιτρέπει να γράφουμε προγράμματα τα οποία είναι εύκολο να συντηρηθούν και να εξελιχθούν στο χρόνο και όχι απλά σενάρια (scripts) τα οποία παίζουν το ρόλο της κόλλας για components που έχουν γίνει με άλλες γλώσσες αλλά γενικά δεν μπορούμε να τα αλλάξουμε ή να ελέγξουμε την εξέλιξή τους.

Η Java προσφέρει την ευελιξία της δυναμικής σύνδεσης. Ένα πρόγραμμα Java εκτελείται από μία εικονική μηχανή και οι κατηγορίες (τα αρχεία .class που περιέχουν τα bytcodes) φορτώνονται από την JVM όταν εκτελείται το πρόγραμμα. Αυτό σημαίνει ότι μπορεί κανείς να αντικαταστήσει μία παλιά κατηγορία με μία νέα ακόμα και την στιγμή που εκτελείται το πρόγραμμα. Αυτό είναι ένα βασικό χαρακτηριστικό των components. Έτσι, οι κατηγορίες Java πάνε πέρα από τον αντικειμενοστραφή προγραμματισμό, στον προγραμματισμό βασισμένο σε συστατικά (component-based programming). Είναι επίσης σημαντικό ότι η Java είναι σχεδιασμένη γι' αυτό και δεν απαιτείται η προσθήκη κάποιου πολύπλοκου μοντέλου (όπως το COM) της Microsoft, πάνω από μία γλώσσα προγραμματισμού που δεν είναι σχεδιασμένη με αυτό τον τρόπο (π.χ. Visual C++).

2.4.2 Πιθανά μειονεκτήματα της Java

Τα προγράμματα πλοήγησης ενδέχεται να μην μπορούν να εκτελέσουν το Applet μας κι αυτό διότι τα προγράμματα πλοήγησης ενδέχεται να μην είναι εφοδιασμένα με την τρέχουσα έκδοση της Java.

Ο κώδικας που είναι μεταγλωττισμένος για μια συγκεκριμένη μηχανή είναι ταχύτερος από τον κώδικα που εκτελείται από ένα μεταφραστή (interpreter). Πάντως αξίζει να σημειωθεί ότι μία τεχνική που ονομάζεται Just-In Time μεταγλώττιση (JIT compilation) έχει βελτιώσει σημαντικά τους χρόνους εκτέλεσης των Java προγραμμάτων. Αν και ακόμα τα προγράμματα της Java δεν έχουν γίνει ταχύτερα από τον μεταγλωττισμένο κώδικα της C++, πολλοί είναι αυτοί που υποστηρίζουν ότι είναι αρκετά γρήγορα για να μην γίνει η ταχύτητα η αιτία να μην πετύχει η Java.

2.4.3 JavaScript

Η JavaScript είναι μια συλλογή από επεκτάσεις στην HTML γλώσσα και έχει σχεδιασθεί από την Netscape Corporation. Κυρίως χρησιμοποιείται για να ελέγχει τον browser, για παράδειγμα το άνοιγμα και κλείσιμο παραθύρων, τη διαχείριση

φορμών, τις ρυθμίσεις του browser και τέλος την εκτέλεση Java applets. Μολονότι η σύνταξη της JavaScript είναι παρόμοια με της Java, διαφέρουν σε πολλά σημεία.

Σε αντίθεση με τα "κενά ασφαλείας" της Java, τα οποία μπορούν να προκαλέσουν την αλλαγή των δεδομένων στο σκληρό δίσκο του χρήστη, τα "κενά ασφαλείας" της JavaScript έχουν να κάνουν με την παραβίαση του απόρρητου του χρήστη. Τα περισσότερα bugs που έχουν αναφερθεί, όπως η υποκλοπή της e-mail διευθύνσεως του χρήστη, η υποκλοπή των αρχείων του σκληρού δίσκου και η καταγραφή των ενεργειών του χρήστη, έχουν καλυφθεί, όμως νέα εμφανίζονται καθημερινά. Η JavaScript πρέπει να θεωρείται ανασφαλής και συνίσταται η απενεργοποίησή της. Το τελευταίο bug αναφέρθηκε τον Φεβρουάριο του 1998.

2.4.5 Τι είναι ένα Applet;

Ένα Applet είναι ένα πρόγραμμα Java το οποίο εκτελείται σε μία ιστοσελίδα. Τα βήματα που πρέπει να γίνουν για να δημιουργηθεί ένα Java Applet είναι τα ακόλουθα:

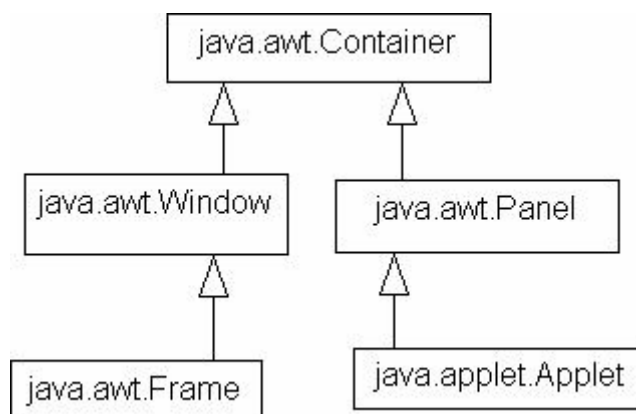
- 1) Συγγραφή του Applet ακριβώς όπως συμβαίνει με κάθε άλλο πρόγραμμα Java.
- 2) Μετάφραση του Applet για να παραχθεί το αρχείο με επέκταση class με τα bytecodes
- 3) Ενσωμάτωση του Applet σε μία ιστοσελίδα χρησιμοποιώντας (κατ' ελάχιστο) την ετικέτα <Applet> η οποία έχει την εξής (ελάχιστη) μορφή:

```
<applet code = όνομα-αρχείου.class width=πλάτος  
height=ύψος></applet>
```

Όπως γίνεται φανερό η ετικέτα applet έχει τις υποχρεωτικές φράσεις code, width και height. Το code προσδιορίζει το αρχείο που θα εκτελεστεί όταν θα φορτωθεί η ιστοσελίδα που περιέχει την ετικέτα σε ένα browser. Το width προσδιορίζει το πλάτος που θα καταλάβει το applet σε εικονοστοιχεία (pixels) στην περιοχή του browser και το height είναι το ύψος που θα καταλάβει το applet σε pixels στην περιοχή του browser. Τέλος το </applet> τερματίζει την ετικέτα Applet. Το Applet θα καταλάβει το χώρο αυτό στο σημείο που βρίσκεται στην ιστοσελίδα.

Ένα Applet εκτελείται όταν φορτώνεται η ιστοσελίδα που το περιέχει σε κάποιο browser.

Όπως γίνεται προφανές από την πιο πάνω συζήτηση, ένα Applet θα πρέπει να έχει μια οπτική παράσταση για να μπορεί να εμφανιστεί σε μια ιστοσελίδα και να καταλάβει κάποιο χώρο σε αυτή. Γι' αυτό το λόγο τα Applets είναι υποκατηγορίες της κατηγορίας java.awt.Panel. Σαν υποκατηγορία της κατηγορίας Panel ένα Applet έχει διαστάσεις που μπορούν να καθοριστούν από την μέθοδο setBounds και μπορεί να λειτουργήσει ως container άλλων παραθυρικών συστατικών (windows components). Δηλαδή σε ένα Applet μπορούμε να έχουμε παραθυρικά συστατικά όπως κουμπιά (buttons), πλαίσια ελέγχου (checkboxes) κ.λ.π. Αυτό που δεν μπορούμε να έχουμε είναι μενού και αυτό διότι τα μενού προσαρτώνται μόνο σε παράθυρα (Frames) και ένα Panel δεν είναι Frame. Αυτή η ιεραρχία κατηγοριών που περιγράφηκε πιο πάνω είναι όπως δείχνει το ακόλουθο UML μοντέλο:



Εικόνα 2.3: UML μοντέλο

Όπως φαίνεται ένα Applet παρότι επεκτείνει τη κατηγορία `java.awt.Panel` και επομένως είναι `java.awt.Container` για να μπορεί να περιέχει όλα τα παραθυρικά συστατικά που θα αποτελέσουν την γραφική του εικόνα προς το χρήστη δεν ανήκει στο πακέτο `java.awt` αλλά στο πακέτο `java.Applet`.

2.4.6 Πώς γράφουμε ένα Applet.

Ένα πρόγραμμα Java το οποίο θα χρησιμοποιηθεί σαν Applet θα πρέπει να επεκτείνει (extend) την κατηγορία `java.applet.Applet`. Έτσι, συνήθως, κάνουμε import το πακέτο `java.applet` όπως και το πακέτο `java.awt` σε κάθε Applet.

Ένα Applet έχει λοιπόν, συνήθως, την ακόλουθη γενική μορφή:

```
import java.awt.*;
import java.applet.*;

public class όνομα-τάξης-Applet extends Applet
{
}
}
```

2.4.6.1 Ένα απλό Applet

Σκοπός μας είναι να μελετήσουμε- το applet HelloWorld - το οποίο εμφάνιζε στη περιοχή του Applet το μήνυμα HelloWorld!.

```
import java.applet.Applet;
import java.awt.Graphics;

public class HelloWorldApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
}
```

Όπως βλέπουμε λοιπόν το Applet αυτό επεκτείνει την κατηγορία `java.applet.Applet`. Επίσης αντί να εισάγουμε με import όλο το πακέτο `java.awt` εισάγουμε μόνο τη κατηγορία `java.awt.Graphics` που χρησιμοποιούμε για τη μέθοδο `paint`. Η μέθοδος `paint` είναι μια μέθοδος της κατηγορίας `Container` που είναι υπερτάξη της κατηγορίας `Panel` που με τη σειρά της είναι υποκατηγορία της κατηγορίας `Applet`. Η μέθοδος αυτή καλείται αυτόματα όταν εμφανίζεται το Applet και δουλειά της είναι να εμφανίσει το `Container` (δηλαδή το Applet) και όλα τα

συστατικά του στην οθόνη. Στη συγκεκριμένη περίπτωση αν δεν υπερβαίναμε (override) τη μέθοδο paint το Applet μας θα ήταν ένα πολύ βαρετό Applet μια και δεν θα βλέπαμε απολύτως τίποτα στη περιοχή του. Βεβαίως και τώρα που την υπερβαίνουμε δεν κάνει και τίποτα το συγκλονιστικό. Το μόνο που κάνει είναι να εμφανίσει στην οθόνη τη φράση Hello World!. Ίσως να απορείτε για το τι είναι η παράμετρος g που είναι ένα αντικείμενο της τάξης java.awt.Graphics. Αυτή είναι, ουσιαστικά η περιοχή σχεδίασης του Applet και διαθέτει μεθόδους που μας επιτρέπουν να γράφουμε, να σχεδιάζουμε, να εμφανίζουμε εικόνες κ.λ.π. πάνω στην επιφάνεια του Applet όπως η drawString που καλείται και εμφανίζει το String που δίνεται σαν πρώτη παράμετρος. Το String αυτό εμφανίζεται χρησιμοποιώντας σαν την πάνω αριστερή γωνία τη θέση που προσδιορίζεται από τις συντεταγμένες x και y που δίνονται σαν δεύτερη και τρίτη παράμετρος αντίστοιχα.

Όταν, επομένως, φορτωθεί η ιστοσελίδα που θα περιέχει το πιο πάνω Applet, θα εμφανιστεί η φράση "Hello World!" στη περιοχή του Applet. Αλλά πως περιλαμβάνουμε ένα Applet σε μία ιστοσελίδα; Η απάντηση αυτή ήδη έχει δοθεί: Θα πρέπει να βάλουμε στην ιστοσελίδα την ετικέτα <Applet> προσδιορίζοντας ως code το αρχείο class του Applet.

Η HTML που απαιτείται γι' αυτό δίνεται στη συνέχεια:

```
<applet code="HelloWorldApplet.class" width=200 height=100> </applet>
```

Αν, λοιπόν, μεταφράσετε το HelloWorldApplet.java και γράψετε τη πιο πάνω γραμμή σε ένα αρχείο με επέκταση html (ή htm) και στη συνέχεια φορτώσετε το αρχείο σε ένα browser θα δείτε το applet να τρέχει και θα δείτε κάτι ανάλογο με αυτό που φαίνεται εδώ [1]:



ΚΕΦΑΛΑΙΟ 3^ο

“Διεκπεραίωση εργασίας”

3.1 Εισαγωγή

Για να διεκπεραιωθεί η πτυχιακή εργασία ασχοληθήκαμε με τα webholes, το πρωτόκολλο Ipsil Control Protocol (ICP), που μας βοηθά να στέλνουμε πακέτα δεδομένων για να ενεργοποιούμε/ απενεργοποιούμε τα I/O του μικροελεγκτή μέσω της εφαρμογής Ipsil Config Utility (ICU).

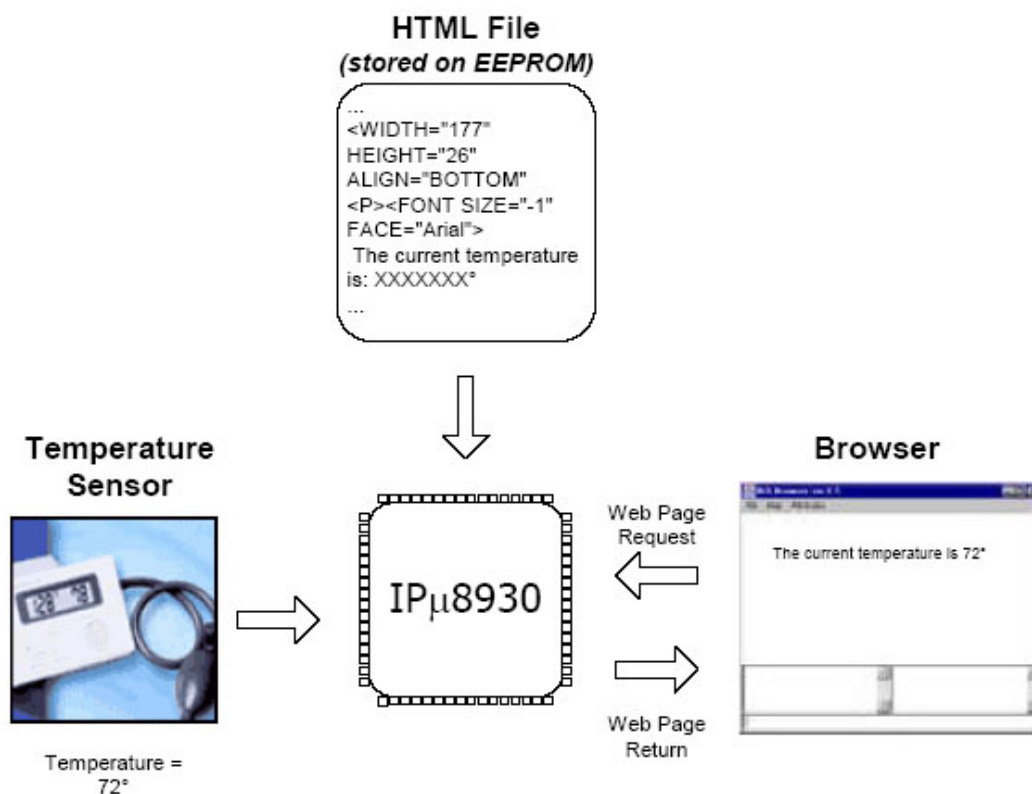
3.2 Webholes

Ένα από τα μοναδικά χαρακτηριστικά γνωρίσματα του IPμ8930 είναι η δυνατότητα να καθορίζονται placeholders σε ένα αρχείο html που είναι αποθηκευμένο στον IPμ8930, τα οποία καλούνται WebHoles. Τα WebHoles επιτρέπουν στους υπεύθυνους την ανάπτυξη και την δημιουργία δυναμικών ιστοσελίδων (.html) που ενημερώνονται με βάση την διεύθυνση μνήμης της ψηφιακής εισόδου στην οποία αποθηκεύεται σε ψηφιακή μορφή η έξοδος του συνδεδεμένου αισθητηρίου (πίνακας 3.1). Πλήρεις διευθύνσεις RAM (Internal EEPROM) στο Παράρτημα Α, σελίδα 20-25.

Αισθητήριο	Θέση (δεκαδικό σύστημα)
Εσωτερικής θερμοκρασίας (I ² C)	280
1	282
2	284
3	286
4	288
5	290
6	292
7	294
8	296

Πίνακας 3.1: διευθύνσεις μνήμης των δεδομένων του εκάστοτε αισθητηρίου

Έτσι για παράδειγμα όταν μία ιστοσελίδα ήδη αποθηκευμένη στον IPμ8930 ζητείται από κάποιον απομακρυσμένο χρήστη μέσω του web browser τότε ο IPμ8930 webServer θα εισαγάγει τις στιγμιαίες τιμές του WebHole πριν στείλει την σελίδα (εικόνα 3.1).



Εικόνα 3.1: Παροχή δεδομένων στις ιστοσελίδες

Κατά αυτόν τον τρόπο τα Webholes παρέχουν σε πραγματικό χρόνο δεδομένα στις ιστοσελίδες του web server IPμ8930.

Εκτός από τα I/O pins του IPμ8930, τα Webholes συνδέονται με τις μεταβλητές της μνήμης RAM, την εσωτερική και την εξωτερική EEPROM, τη θύρα I2C και τη θύρα SPI. Όταν "κατεβάζουμε" μία ιστοσελίδα από τον webserver του IPμ8930, η τιμή από την I/O θύρα λαμβάνεται κι ενσωματώνεται στην σελίδα html στην προκαθορισμένη θέση (πίνακας 3.1).

Ο HTTP server ενσωματώνει στις ιστοσελίδες του τα webholes με συγκεκριμένη δομή σύμφωνα με τον πίνακα 3.2 που ακολουθεί ώστε να επιτρέψει τη δυναμική αντικατάσταση των περιεχομένων των ιστοσελίδων. Αυτή η δομή χαρακτηρίζει τις θέσεις WebHoles στις συγκεκριμένες ιστοσελίδες και συνδυάζει κάθε WebHole με χαρακτηριστικά όπως τύπος δεδομένων (bytes), πλήθος δεδομένων, προέλευση δεδομένων κ.α. όταν προσκομίζεται η ιστοσελίδα.

Field	Name	Length	Contents
1	IDATAFORMAT	1	Καθορίζει πώς το στοιχείο πηγής γράφεται στον προορισμό: Οι ακόλουθοι κώδικες σχήματος ισχύουν 0 = Copy (no conversion) 1 = HEX8, 2-digit ASCII Hex, e.g., "7F" 2 = DEC8, 3-digit ASCII Decimal, e.g., "127" 3 = BIN8, 8-digit ASCII Binary, e.g., "01111111" 4 = DEC16, 5-digit ASCII Decimal up to 65535 (e.g., "32764")
2	LENGTH	1	Μήκος στις ψηφιολέξεις του WebHole. Μέγιστος από 255 ψηφιολέξεις
3	DESTH	1	Δείκτης ψηφιολέξεων (MSB) στον προορισμό Στοιχεία
4	DESTL	1	Δείκτης ψηφιολέξεων (LSB) στον προορισμό Στοιχεία
5	HOLETYPE	1	Προσδιορίζει την πηγή των δεδομένων που "πληρώνουν" το WebHole. Οι έγκυρες τιμές περιλαμβάνουν 0 = Constant (DATASRCH/DATASRCL contains data) 1 = RAM variable (see section 5.2) 2 = External EEPROM (see section 6.1) 3 = Internal EEPROM (see section 5.3) 4 = Byte Write from incoming packet
6	DATASRCH	1	MSB της θέσης των στοιχείων που "θα γεμίσουν" το WebHole. Αυτός ο τομέας χρησιμοποιείται μαζί με τους τομείς HOLETYPE και DATASRCL.
7	DATASRCL	1	LSB της θέσης των στοιχείων που "θα γεμίσουν" το WebHole. Αυτός ο τομέας χρησιμοποιείται στο conjunction με τους τομείς HOLETYPE και DATASRCH.
8	DATANXT	1	Ετικέτα τελών. Εάν αυτό είναι 0xFF, κατόπιν αυτό είναι το τέλος από αυτήν την είσοδο WebHole. Οποιαδήποτε άλλη αξία δείχνει ότι ένα άλλο webhole ακολουθεί αυτήν.

Πίνακας 3.2 Ο πίνακας της δομής των Webholes

3.2.1 Κώδικας ιστοσελίδας με ενσωματωμένα τα webholes

Με βάση τα παραπάνω αρχίσαμε την κατασκευή μιας ιστοσελίδας με ονομασία 085.htm¹, όπου εμφανίζονται δυναμικά τα περιεχόμενα των I/O pins καθώς και του αισθητήρα εσωτερικής θερμοκρασίας.

Ο κώδικας της ιστοσελίδας είναι ο παρακάτω:

```
<HTML><HEAD>
<META HTTP-EQUIV="refresh" CONTENT="5">
```

¹ Σημείωση: Όλες οι σελίδες τις οποίες θα "ανεβάσουμε" στο Development Kit πρέπει να αρχίζουν με έναν αριθμό από 031 μέχρι 512, για να μπορούν να αναγνωριστούν από το Kit και να αποθηκευτούν στην κατάλληλη θέση. Οι σελίδες 000-030 χρησιμοποιούνται από τα αρχεία συστήματος της IPSIL.

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
<meta http-equiv="content-type" content="text/html charset=windows-1253">
<TITLE>Αναλογικές Είσοδοι</TITLE>
</HEAD>
<BODY bgcolor="white" text="black" link="blue" vlink="purple" alink="red">
<p><FONT FACE="Arial"><SCRIPT LANGUAGE="Javascript">
<!--
function decodeTemp1(word0) { // εισαγωγή της τιμής του αισθητηρίου
result = word0 / 256 //διαίρεση της τιμής του αισθητηρίου με 256
return Math.round(result*10)/10 // στρογγυλοποίηση αριθμού
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech1(word1) {
result = word1 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech2(word2) {
result = word2 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech3(word3) {
result = word3 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech4(word4) {
result = word4 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech5(word5) {
result = word5 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech6(word6) {
result = word6 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech7(word7) {
result = word7 / 196
return Math.round(result*10)/10
}// -->
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech8(word8) {
result = word8 / 196
```

```
return Math.round(result*10)/10
} // -->
</SCRIPT>
</FONT></p>
<p align="center"><FONT FACE="Verdana"><b><span style="font-size:14pt">Εμφάνιση
Αναλογικών Εισόδων </span></b></FONT></p>
<p align="center"><FONT FACE="Verdana"><b><span style="font-size:14pt">του
μικροελεγκτή IPμ 8930</span></b></FONT></p>
<p><FONT FACE="Arial">Εσ. Θερμοκρασία
<SCRIPT>document.write(decodeTemp1( ))</SCRIPT> C </FONT></p>
<p><FONT FACE="Arial">Μετρητής Τάσης
<SCRIPT>document.write(decodech1( ))</SCRIPT>
V</FONT></p>
<p><FONT FACE="Arial">Μετρητής2 <SCRIPT>document.write(decodech2(
))</SCRIPT> V</FONT></p>
<p><FONT FACE="Arial">Μετρητής3 <SCRIPT>document.write(decodech3(
))</SCRIPT> V</FONT></p>
<p><FONT FACE="Arial">Μετρητής4 <SCRIPT>document.write(decodech4(
))</SCRIPT> V</FONT></p>
<p><FONT FACE="Arial">Μετρητής5 <SCRIPT>document.write(decodech5(
))</SCRIPT> V</FONT></p>
<p><FONT FACE="Arial">Μετρητής6 <SCRIPT>document.write(decodech6(
))</SCRIPT> V</FONT></p>
<p><FONT FACE="Arial">Μετρητής7 <SCRIPT>document.write(decodech7(
))</SCRIPT> V</FONT></p>
<p><FONT FACE="Arial">Μετρητής8 <SCRIPT>document.write(decodech8(
))</SCRIPT> V</FONT></p>
<p></p>
<p><FONT FACE="Arial">Η σελίδα ανανεώνεται κάθε 5
δευτερόλεπτα</FONT></p>
</BODY>
</HTML>
```

Παρατηρούμε ότι μέσα στα script `<SCRIPT>document.write(decodech2())</SCRIPT>` υπάρχει μία παρένθεση κενή.

Εκεί, με έναν τρόπο που θα εξηγήσουμε παρακάτω, ο μικροελεγκτής στέλνει τις πληροφορίες του αισθητηρίου. Για παράδειγμα, έχουμε ένα αισθητήριο φωτεινότητας με έξοδο τάσης 0-5 Volt. Ο μικροελεγκτής παίρνει σε ψηφιακή μορφή 10bit την έξοδο του αισθητηρίου και την μεταφέρει στην θέση του webhole σε δεκαδική μορφή. Μέσω των scripts που έχουμε δημιουργήσει και ενσωματώσει στην ιστοσελίδα σε γλώσσα javascript γίνεται η μετατροπή της πληροφορίας στο πεδίο τιμών του φυσικού μεγέθους της φωτεινότητας (βλ. εικόνα 3.2).

Εμφάνιση Αναλογικών Εισόδων του μικροελεγκτή IPμ 8930

Εσ. Θερμοκρασία 25 C

Μετρητής Τάσης 5 V

Μετρητής2 5 V

Μετρητής3 5 V

Μετρητής4 5 V

Μετρητής5 5 V

Μετρητής6 5 V

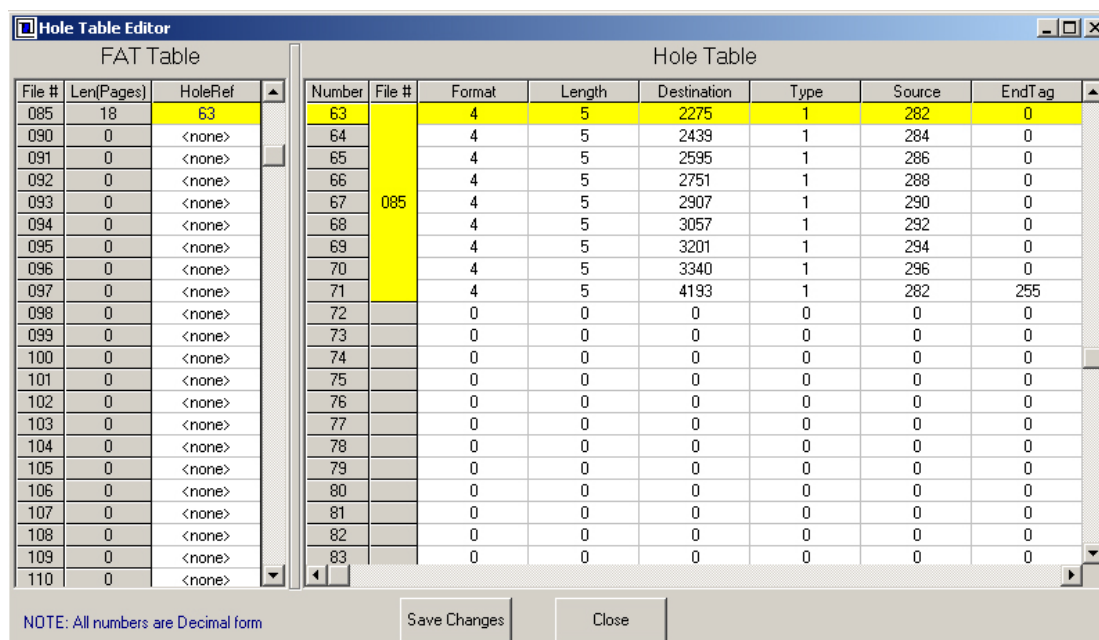
Μετρητής7 5 V

Μετρητής8 5 V

Η σελίδα ανανεώνεται κάθε 5 δευτερόλεπτα ²

Εικόνα 3.2: Ανάγνωση πληροφοριών αισθητηρίων

Στην προσπάθειά μας αυτή χρησιμοποιούμε ένα ειδικό πρόγραμμα του ICU, το Hole Table Editor (βλ. εικόνα 3.3), ώστε να δηλώσουμε σε ποιες διευθύνσεις της ιστοσελίδας (εκφρασμένες στο δεκαδικό σύστημα) θέλουμε να ενσωματωθούν οι 10bit αριθμοί που διαβάζονται από τις διευθύνσεις της μνήμης RAM (εκφρασμένες στο δεκαδικό σύστημα) και συνδέονται άμεσα με τα αισθητήρια εισόδου/εξόδου



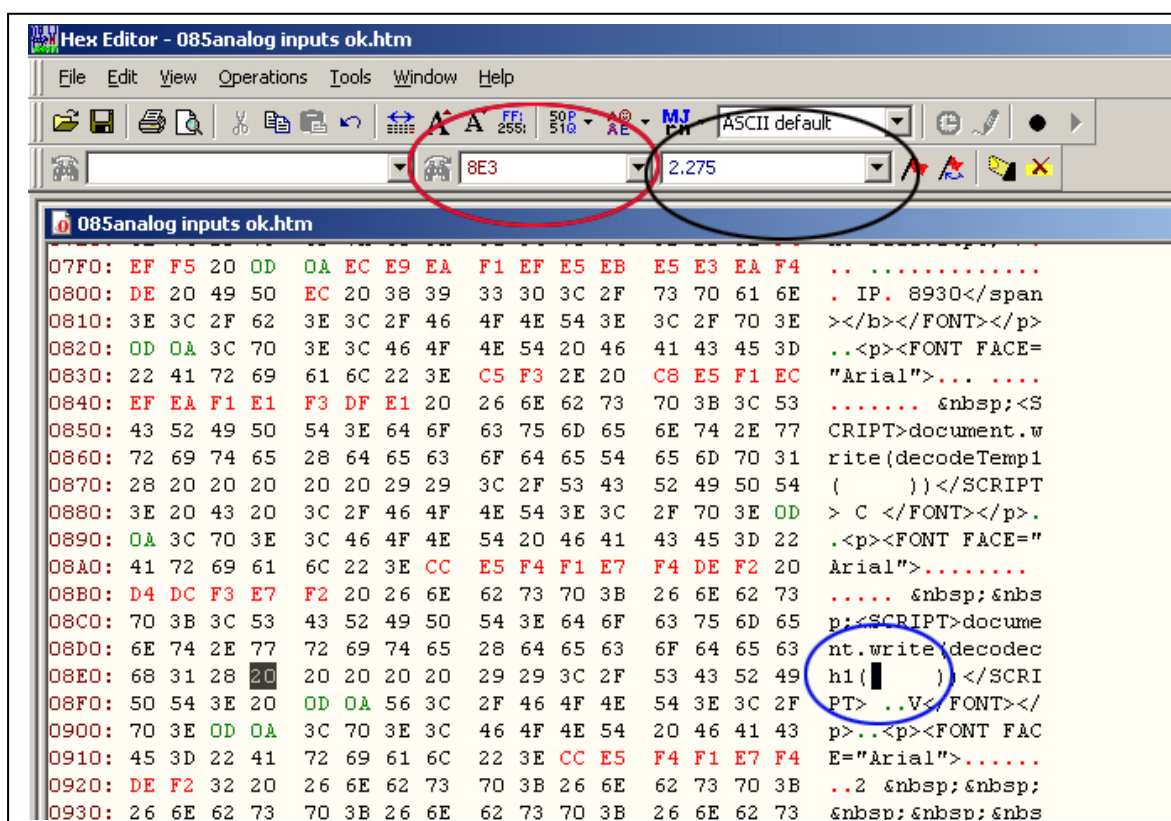
Εικόνα 3.3: Hole Table Editor

² Έχουμε δημιουργήσει ένα script για να ανανεώνεται η ιστοσελίδα και να βλέπουμε διαρκώς τις μεταβαλλόμενες τιμές των αισθητηρίων/συσκευών.

Στην στήλη File# είναι τα αριθμημένα αρχεία html, τα οποία είναι αποθηκευμένα στον μικροελεγκτή. Στη στήλη Holref ορίζουμε σε ποια θέση (Hole Number) θα κάνουμε τις απαιτούμενες επεξεργασίες. Εμείς έχουμε το αρχείο 085.htm και το βλέπουμε στην θέση #085. Συνολικά έχουμε 9 αισθητήρια, άρα θα καταλάβουμε 9 θέσεις που έχουμε ορίσει να αντιστοιχίζονται στα Web Holes 63-71. Μπορούσαμε να αρχίσουμε από οποιοδήποτε (Webholes) αρκεί να υπάρχουν 9 συνεχόμενα ελεύθερα Webholes. Για να καταλάβει ο μικροελεγκτής ότι το 9^ο Webhole είναι το τελευταίο βάζουμε τον αριθμό 255 (FF) στην στήλη end tag³. Έτσι, θα δούμε στην εικόνα 3.3 ότι στο σημείο με διαφορετικό χρώμα περιέχονται οι λεπτομέρειες που θα βλέπουμε στην ιστοσελίδα για τα αισθητήριά μας. Η στήλη source είναι η θέση της μνήμης RAM από την οποία παίρνουμε τις πληροφορίες για το εκάστοτε αισθητήριο. Η πληροφορία για το κάθε ένα από τα 9 αισθητήρια έχει μία μοναδική θέση μνήμης.

Κάθε σελίδα μπορεί να είναι κατ' ελάχιστο μέγεθος 1024 byte. Αν μια σελίδα υπερβαίνει τα 1024 byte τότε δεσμεύει και επόμενες σελίδες μνήμης (1 σελίδα μνήμης= 1024 bytes) του μικροελεγκτή.

Στη στήλη destination βάζουμε τη θέση της τρέχουσας ιστοσελίδας στην οποία θέλουμε να εμφανιστεί η πληροφορία (10bit), την οποία θα επεξεργαστούμε για να απεικονίζει πιστά το φυσικό μέγεθος του προσαρτώμενου αισθητηρίου. Για να δούμε ποια είναι η θέση την οποία θέλουμε, ανοίγουμε με έναν hex editor και βρίσκουμε την σχετική θέση του 1^{ου} byte, όπου θα καταχωρηθούν τα δεδομένα του αισθητηρίου.



Εικόνα 3.4: απεικόνιση προγράμματος Hex Editor

³ Οι αριθμοί απεικονίζονται σε δεκαδική μορφή. Ο μικροελεγκτής τους διαχειρίζεται στο δεκαεξαδικό σύστημα.

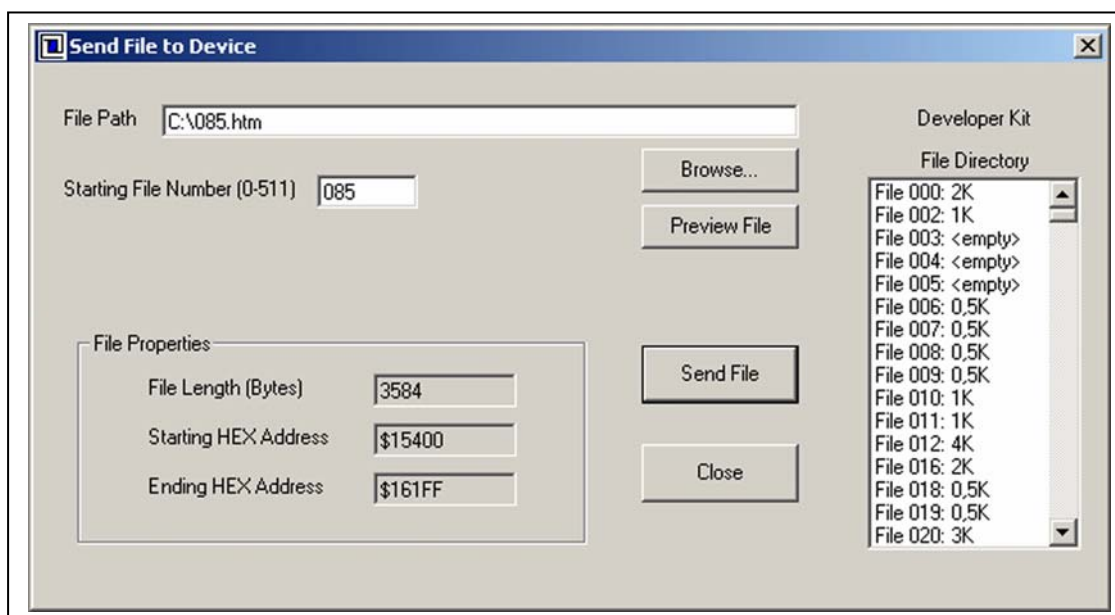
Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

Στον κάτω δεξιά κύκλο βλέπουμε το σημείο στο οποίο θέλουμε να μπει η πληροφορία. Πάνω αριστερά δείχνει σε 16δική μορφή τη θέση ενώ πάνω δεξιά την δεκαδική. Βάζοντας τον αριθμό 2275 στο destination, θα πάρουμε την μέτρηση του αισθητηρίου που έχουμε δηλώσει και που αποθηκεύεται στις 5 θέσεις που έχουν δεσμευτεί από 2275 έως 2279. Τα 10bit δεδομένα χρησιμοποιούνται ως είσοδος στα scripts που ενσωματώνουμε στον κώδικα της ιστοσελίδας ώστε να μας δώσουν ένα πραγματικό, φυσικό μέγεθος. Για παράδειγμα το παρακάτω script επεξεργάζεται τα δεδομένα του αισθητηρίου φωτεινότητας και εξάγει σε μονάδες φωτεινότητας την μέτρηση του αισθητηρίου.

```
</SCRIPT><SCRIPT LANGUAGE="Javascript">
<!--
function decodech1(word1) {
result = word1 / 196
return Math.round(result*10)/10
}// -->
```

Αυτό το κάνουμε για το καθένα από τα αισθητήρια.

Τελευταίο στάδιο της διαδικασίας είναι η μεταφορά της ιστοσελίδας στο ολοκληρωμένο Irm8930 μέσω της δυνατότητας Send file to device μέσω του Ipsil Configuration Utility.. Αυτό γίνεται με τον παρακάτω τρόπο:



Εικόνα 3.5: αποστολή αρχείων στο Development kit

Με την επιλογή browse στο αντίστοιχο παράθυρο διαλέγουμε το αρχείο το οποίο θέλουμε. Στο πεδίο Starting File Number βάζουμε σε ποια θέση μνήμης στη RAM θέλουμε να αποθηκευτεί. Επειδή οι θέσεις από 0-30 είναι δεσμευμένες από το σύστημα, επιλέγουμε μία θέση από 31 μέχρι 512, που είναι η μνήμη. Στο παράδειγμά μας έχουμε το αρχείο 085.htm το οποίο θα αποθηκευτεί στο file 085. Παρατηρούμε στο File Properties και στο πεδίο File Length ότι το μέγεθος του αρχείου είναι μεγαλύτερο από 1024 bytes. Άρα, θα καταλάβει και τις παρακάτω θέσεις. Συγκεκριμένα, θα καταλάβει $3584/1024=3,5 \Rightarrow 4$ θέσεις, δηλαδή από το 085 έως και το 088.

3.3 ICP (IPμ8930 Control Protocol)

Στη συνέχεια προσπαθήσαμε να εξετάσουμε το πρόβλημα της διαχείρισης των συσκευών από απόσταση αλλά και επικοινωνίας μέσω της σειριακής θύρας. Σε αυτό μας βοήθησε το ICP (IPμ8930 Control Protocol).

Το πρωτόκολλο ελέγχου IPμ8930 (ICP) παρέχει την προγραμματική πρόσβαση σε όλα τα I/O και στους αποθηκευμένους πόρους στο IPμ8930 board, συμπεριλαμβανομένων των γενικών I/O pins.(a.k.a., Port A), της σειριακής θύρας, IPμ8930 RAM, της εσωτερικής EEPROM, και εξωτερικής EEPROM. Σημειώνεται ότι υπάρχει πρόσβαση ανάγνωσης/γραφής στις SPI και I2C θύρες αλλά δεν θα υποστηριχθεί μέχρι IPμ8930 v1.1. Και οι δύο διαδικασίες ανάγνωσης/γραφής υποστηρίζονται από τους πόρους του IPμ8930.

Το ICP είναι ένα σταθερό πρωτόκολλο που υποστηρίζει την ανάγνωση και τη γραφή στο IPμ8930 μέσω της θύρας 8930.Υποστηρίζει μηχανισμούς TCP και UDP. Εάν το IPμ8930 λάβει ένα αίτημα μέσω UDP, θα απαντήσει στο αίτημα χρησιμοποιώντας UDP. Ομοίως, εάν ένα αίτημα TCP ληφθεί, το εξερχόμενο πακέτο θα είναι ένα πακέτο TCP.

(Σημείωση: Οι δομές δεδομένων ICP που περιγράφονται παρακάτω πρέπει να γραφτούν σαν περιεχόμενο στοιχείων ενός τυποποιημένου πακέτου TCP ή UDP.)

Ο έλεγχος ροής για το ICP είναι καθορισμένος. Ένα πρόγραμμα πρέπει, αφού στείλει ένα πακέτο, να περιμένει μια απάντηση πριν στείλει ένα άλλο πακέτο.

Η μορφή (format) των πακέτων ICP αποτελείται από διάφορα μέρη. Κατ' αρχάς, κάθε πακέτο γραφής αρχίζει με έναν κωδικό πρόσβασης, που απαιτείται, εάν έχει επιλεγεί ο ασφαλής τρόπος, ειδάλλως, αγνοείται. Ένα OP CODE και μια διεύθυνση απαιτούνται έπειτα. Τα υπόλοιπα πεδία αποτελούν πληροφορίες που διαβάζονται από ή γράφουν στο IPμ8930. Το περιεχόμενο της δομής των δεδομένων διαφέρει βασισμένο στη λειτουργία και εξαρτάται από το αν είναι ένα αίτημα (Request) ή απάντηση (Response).

Οι ακόλουθοι πίνακες 3.3-3.8 παρουσιάζουν τη δομή πακέτων ενός ICP που διαβάζει, ενός ICP που γράφει και τις αντίστοιχες απαντήσεις τους:

Byte #	Μήκος (Bytes)	Όνομα Πεδίου	Περιγραφή/Έγκυρες τιμές
0x00	5	PASSWD	Κωδικό με 5 ASCII χαρακτήρες (απαιτείται πάντα). Θα αγνοηθεί αν δεν έχει ενεργοποιηθεί ο ασφαλής τρόπος (SECMODE). Αν το SECMODE είναι απενεργοποιημένο, τότε προτείνεται αυτό το κενό να είναι με μηδενικά
0x05	1	OPCODE	2 = Ανάγνωση από εξωτερική EEPROM 4 = Ανάγνωση από εσωτερική EEPROM 7 = Ανάγνωση από θέση RAM 9 = Ανάγνωση από serial port 10 = Ανάγνωση I2C (v1.1) 12 = Ανάγνωση από SPI (v1.1)
0x06	4	ADDR	Δήλωσης θέσης για την παραπάνω λειτουργία. Οι θέσεις είναι big-endian format π.χ. MSB έρχεται πρώτο
0x0A	1	READLEN	Αριθμός των bytes για να διαβάσει. Δυαδικός αριθμός (πρέπει να είναι 1-32 για UDP πακέτα και 1-20 για TCP πακέτα)

Πίνακας 3.3: ICP Packet – Read Request

Byte #	Μήκος (Bytes)	Όνομα Πεδίου	Περιγραφή/Έγκυρες τιμές
0x00	4	ADDR	Δήλωσης θέσης για τα αιτούμενα πακέτα. Είναι η ίδια θέση που στάλθηκε για το αίτημα διαβάσματος
0x04	4	TIMESTAMP	MSB είναι πρώτο και το LSB είναι το τελευταίο. Είναι ο τρέχων χρόνος.
0x8	1-32	READDATA	Τα δεδομένα που ζητήθηκαν.

Πίνακας 3.4: ICP Packet – Read Response

Byte #	Μήκος (Bytes)	Όνομα Πεδίου	Περιγραφή/Έγκυρες τιμές
0x00	5	PASSWD	Κωδικό με 5 ASCII χαρακτήρες (απαιτείται πάντα). Θα αγνοηθεί αν δεν έχει ενεργοποιηθεί ο ασφαλής τρόπος (SECMODE). Αν το SECMODE είναι απενεργοποιημένο τότε προτείνεται αυτό το κενό να είναι με μηδενικά
0x05	1	OPCODE	3 = Εγγραφή σε εξωτερική EEPROM 5 = Εγγραφή σε εσωτερική EEPROM 6 = Εγγραφή σε θέση RAM 8 = Εγγραφή στη serial port 11 = Εγγραφή σε I2C (v1.1) 13 = Εγγραφή σε SPI (v1.1)
0x06	4	ADDR	Δήλωσης θέσης για την παραπάνω λειτουργία. Η θέσεις είναι big-endian format π.χ. MSB έρχεται πρώτο
0x0A	1	WRITEDATA	Αριθμός των bytes για να γράψει. Δυαδικός αριθμός (πρέπει να είναι 1-32 για UDP πακέτα και 1-20 για TCP πακέτα)

Πίνακας 3.5: ICP Packet – Write Request

Byte #	Μήκος (Bytes)	Όνομα Πεδίου	Περιγραφή/Έγκυρες τιμές
0x00	1	STATUS	Θα είναι ίσο με 0xFF αν είναι επιτυχής. Αλλιώς θα έχει έναν λάθος κωδικό
0x01	1	SEQNUM	Σειριακό συνεχόμενος αριθμός. Αυτό θα υπάρχει σε απάντηση του op-codes 8 ή 9. Περιέχει έναν συνεχή αριθμό από την σειριακή εντολή που μόλις στάλθηκε.

Πίνακας 3.6: ICP Packet – Write Response

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

Ο προγραμματιστής μπορεί επίσης να ζητήσει να γίνει μία επανεκκίνηση στο IPμ8930. Αυτό θα αναγκάσει το IPμ8930 να επανέλθει στις τιμές εκκίνησής του που αποθηκεύονται πρώτα στην εσωτερική EEPROM.

Byte #	Μήκος (Bytes)	Όνομα Πεδίου	Περιγραφή/Έγκυρες τιμές
0x00	5	PASSWD	Κωδικό με 5 ASCII χαρακτήρες (απαιτείται πάντα). Θα αγνοηθεί αν δεν έχει ενεργοποιηθεί ο ασφαλής τρόπος (SECMODE). Αν το SECMODE είναι απενεργοποιημένο τότε προτείνεται αυτό το κενό να είναι με μηδενικά
0x05	1	OPCODE	14 = Επανεκκίνηση συσκευής
0x06	4	ADDR	Αυτά τα bytes αγνοούνται για την εντολή επανεκκίνησης

Πίνακας 3.7: Reset Device Request

Δεν υπάρχει απάντηση για το αίτημα της επανεκκίνησης

Η ολοκληρωμένη λίστα των σωστών OPCODEs, παρουσιάζεται παρακάτω:

Serial OPCODE List
2 = Ανάγνωση από εξωτερική EEPROM
3 = Εγγραφή σε εξωτερική EEPROM
4 = Ανάγνωση από εσωτερική EEPROM
5 = Εγγραφή σε εσωτερική EEPROM
6 = Εγγραφή σε θέση RAM
7 = Ανάγνωση από θέση RAM
8 = Εγγραφή στη serial port
9 = Ανάγνωση από serial port
10 = Ανάγνωση I2C (v1.1)
11 = Εγγραφή σε I2C (v1.1)
12 = Ανάγνωση από SPI (v1.1)
13 = Εγγραφή σε SPI (v1.1)
14 = Επανεκκίνηση συσκευής
15 = Αποστολή πακέτου UDP (μόνο για το MCU mode)
16 = Έναρξη ενσωματωμένης (embedded) καταγραφής δεδομένων
17 = Τερματισμός ενσωματωμένης (embedded) καταγραφής δεδομένων

Πίνακας 3.8: Complete ICP OPCODE List

Σημείωση: Όταν δημιουργούμε πακέτα για την επικοινωνία μέσω της σειριακής στο πεδίο ADDR (πίνακας 3.3 και 3.5) το συμπληρώνουμε με μηδενικά (0X00, 0X00, 0X00,)

3.3.1 Δημιουργία πακέτων ICP για έλεγχο καναλιών

Με βάση τους πίνακες φτιάχνουμε τα παρακάτω τμήματα κώδικα java τα οποία θα ενσωματωθούν σε ολοκληρωμένη εφαρμογή στη συνέχεια:

Μέρος κώδικα για να γίνουν όλα τα κανάλια ψηφιακές έξοδοι σε κατάσταση απενεργοποίησης:

```
// Arrays
// Write RAM variable ICP command
// Configures all channels as Digital Outputs
// Operation code = 0x06, address in RAM 0x113, bytes to write
2
byte [] setAllDigitalOutput =
{ 0x00, 0x00, 0x00, 0x00, 0x00, // Password (5 bytes len)
  0x06, // "Write RAM" cmd (1 byte len)
  0x00, 0x00, 0x01, 0x13, // Start location (4 bytes len)
  (byte)0x86, // All channels Digital (1 byte len)
  0x004, // All channels as Outputs (1 byte len)
};

// Set all digital outputs to LO level
// Write RAM variable ICP command
// Operation code = 0x06, address in RAM 0x117, 0000 0000 - all
channels "1" output
// Each bit of the last byte represents each channel state
accordingly
byte [] setAllDigitalLo =
{ 0x00, 0x00, 0x00, 0x00, 0x00, // Password (5 bytes len)
  0x06, // "Write RAM" cmd(1 byte len)
  0x00, 0x00, 0x01, 0x17, // Start location (4 bytes len)
  (byte)0x005, // All channels LO(1 byte len)
};
```

Μέρος κώδικα για να γίνουν όλα τα κανάλια ψηφιακές έξοδοι σε κατάσταση ενεργοποίησης

```
// Arrays
// Write RAM variable ICP command
// Configures all channels as Digital Outputs
// Operation code = 0x06, address in RAM 0x113, bytes to write
2
byte [] setAllDigitalOutput =
{ 0x00, 0x00, 0x00, 0x00, 0x00, // Password (5 bytes len)
  0x06, // "Write RAM" cmd(1 byte len)
  0x00, 0x00, 0x01, 0x13, // Start location (4 bytes len)
  (byte)0x86, // All channels Digital (1 byte len)
  0x00, // All channels as Outputs (1 byte len)
};
```

⁴ Εντολές προσδιορισμού των καναλιών σαν εισοδοι ή έξοδοι π.χ. (byte)0x00,
// All channels as Outputs(1 byte len)

Το byte 0X00 αντιστοιχίζεται στα κανάλια ως ακολούθως: 0000 0000
3210 7654

Αν θέλουμε για παράδειγμα να έχουμε ως έξοδο το bit 4 (κανάλι 5) θα δώσουμε 0XEF
Το 0XEF σημαίνει 1111 1110
3210 7654

⁵ Εντολές ενεργοποίησης/απενεργοποίησης των καναλιών π.χ. 0x00,
// All channels LO (1 byte len)

Το byte 0X00 αντιστοιχίζεται στα κανάλια ως ακολούθως: 0000 0000
7654 3210

Αν θέλουμε για παράδειγμα να έχουμε LOW το bit 4 (κανάλι 5) θα δώσουμε 0X7F
Το 0X7F σημαίνει 1110 1111
7654 3210

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
// Set all digital outputs to HI level
// Write RAM variable ICP command
// Operation code = 0x06, address in RAM 0x117, 1111 1111 - all
channels "1" output
// Each bit of the last byte represents each channel state
accordingly
byte [] setAllDigitalHi =
{ 0x00, 0x00, 0x00, 0x00, 0x00, // Password (5 bytes len)
  0x06, // "Write RAM" cm (1 byte len)
  0x00, 0x00, 0x01, 0x17, // Start location (4 bytes len)
  (byte)0xff, // All channels HI(1 byte len)
};
```

Μέρος κώδικα για να γίνει επικοινωνία μέσω της σειριακής:

```
// Arrays
// Write serial ICP command
// Configures all channels as Digital Outputs
// Operation code = 0x08, address in RAM 0x0, bytes to write 4
byte [] setAllDigitalOutput =
{ 0x00, 0x00, 0x00, 0x00, 0x00, // Password (5 bytes len)
  0x08, // "Write serial" cmd (1 byte len)
  0x00, 0x00, 0x01, 0x13, // Start location (4 bytes len)
  (byte)0x86, // All channels Digital (1 byte len)
  (byte)0x54, (byte)0x45, (byte)0x53, (byte)0x54, //
"TEST"(4 byte len)
};
```

Ο τελικός κώδικας της εφαρμογής που θέτει όλα τα κανάλια ψηφιακά και τα ενεργοποιεί είναι ο ακόλουθος:

```
import java.applet.Applet ;
import java.io.* ;
import java.net.* ;
import java.util.* ;
import java.awt.* ;

public class setDigitalHi extends Applet
{
    // Constants
    private int IPSIL_PORT = 8930 ; // Ipsil device port number

    // Variables
    InetAddress ia = null;
    Socket sock = null; // Socket
    InputStream sIn = null; // Input data stream
    OutputStream sOut = null; // Output data stream
instance

    StringBuffer buffer = new StringBuffer();

    // Arrays
    // Write RAM variable ICP command
    // Configures all channels as Digital Outputs
    // Operation code = 0x06, address in RAM 0x113, bytes to write
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
byte [] setAllDigitalOutput =
{ 0x00, 0x00, 0x00, 0x00, 0x00,      // Password (5 bytes len)
  0x06,                               // "Write EEPROM" cmd (1 byte len)
  0x00, 0x00, 0x01, 0x13, // Start location (4 bytes len)
  (byte)0x86,                // All channels Digital (1 byte len)
  0x00,                       // All channels as Outputs (1 byte len)
};

// Set all digital outputs to HI level
// Write RAM variable ICP command
// Operation code = 0x06, address in RAM 0x117, 1111 1111 - all
channels "1" output
// Each bit of the last byte represents each channel state
accordingly
byte [] setAllDigitalHi =
{ 0x00, 0x00, 0x00, 0x00, 0x00,      // Password (5 bytes len)
  0x06,                               // "Write EEPROM" cmd (1 byte len)
  0x00, 0x00, 0x01, 0x17, // Start location (4 bytes len)
  (byte)0xff,                // All channels HI (1 byte len)
};

byte [] resp = new byte[1024];      // Network response buffer

public void init()
{
    try
    {
        //ia = InetAddress.getByName("194.177.198.209");
        //ia = InetAddress.getLocalHost();
        //addItem(ia.getHostAddress());
        ia = InetAddress.getByName( getCodeBase().getHost()
);

        // Init TCP socket. Connect to the device itself on
Ipsil port

        sock = new Socket(ia, IPSIL_PORT);
        addItem(ia.toString());

        // Create input data stream
        sIn = sock.getInputStream();

        // Create output data stream
        sOut = sock.getOutputStream();

    }

    catch (Exception e)
    {
        e.printStackTrace();
        addItem(" Init " );
    }
}

private void addItem(String newWord)
{
    System.out.println(newWord);
    buffer.append(newWord);
    repaint();
}
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
    }

    public void paint(Graphics g)
    {
        //Draw a Rectangle around the applet's display area.
        g.drawRect(0, 0, getSize().width - 1, getSize().height -
1);

        //Draw the current string inside the rectangle.
        g.drawString(buffer.toString(), 5, 15);
    }

    public void start()
    {
        try
        {
            // Send ICP command - Configure all channels as
digital outputs
            sOut.write(setAllDigitalOutput);

            // Read and ignore replay byte from the Ipsil
device
            sIn.read(resp);

            // Send ICP command - Set HI all on all of the
channels
            sOut.write(setAllDigitalHi);

            // Read and ignore replay byte from the Ipsil
device
            sIn.read (resp);

            addItem(" All Digital HI Done!");
        }

        catch (Exception e)
        {
            e.printStackTrace();
            addItem(" Start ");
        }
    }
} //end of the applet
```

Ο τελικός κώδικας της εφαρμογής που θέτει όλα τα κανάλια ψηφιακά και τα απενεργοποιεί είναι ο ακόλουθος:

```
import java.applet.Applet ;
import java.io.* ;
import java.net.* ;
import java.util.* ;
import java.awt.* ;

public class setDigitalLo extends Applet
{
    // Constants
    private int IPSIL_PORT = 8930 ;           // Ipsil device port
number
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
// Variables
InetAddress ia      = null;
Socket           sock = null;           // Socket
InputStream sIn    = null;             // Input data stream
OutputStream     sOut = null;          // Output data stream
instance

StringBuffer buffer = new StringBuffer();

// Arrays
// Write RAM variable ICP command
// Configures all channels as Digital Outputs
// Operation code = 0x06, address in RAM 0x113, bytes to write
2
byte [] setAllDigitalOutput =
{ 0x00, 0x00, 0x00, 0x00, 0x00,      // Password (5 bytes len)
  0x06,                          // "Write EEPROM" cmd (1 byte len)
  0x00, 0x00, 0x01, 0x13, // Start location (4 bytes len)
  (byte)0x86,                  // All channels Digital (1 byte len)
  0x00,                          // All channels as Outputs (1 byte len)
};

// Set all digital outputs to LO level
// Write RAM variable ICP command
// Operation code = 0x06, address in RAM 0x117, 0000 0000 - all
channels "1" output
// Each bit of the last byte represents each channel state
accordingly
byte [] setAllDigitalLo =
{ 0x00, 0x00, 0x00, 0x00, 0x00,      // Password (5 bytes len)
  0x06,                          // "Write EEPROM" cmd (1 byte len)
  0x00, 0x00, 0x01, 0x17, // Start location (4 bytes len)
  (byte)0x00,                  // All channels LO (1 byte len)
};

byte [] resp = new byte[1024];        // Network response
buffer

public void init()
{
    try
    {
        //ia = InetAddress.getByName("194.177.198.209");
        //ia = InetAddress.getLocalHost();
        //addItem(ia.getHostAddress());
        ia = InetAddress.getByName( getCodeBase().getHost()
);

        // Init TCP socket. Connect to the device itself on
Ipsil port

        sock = new Socket(ia, IPSIL_PORT);
        addItem(ia.toString());

        // Create input data stream
        sIn = sock.getInputStream();

        // Create output data stream
        sOut = sock.getOutputStream();
    }
}
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
        }

        catch (Exception e)
        {
            e.printStackTrace();
            addItem(" Init " );
        }
    }

    private void addItem(String newWord)
    {
        System.out.println(newWord);
        buffer.append(newWord);
        repaint();
    }

    public void paint(Graphics g)
    {
        //Draw a Rectangle around the applet's display area.
        g.drawRect(0, 0, getSize().width - 1, getSize().height -
1);

        //Draw the current string inside the rectangle.
        g.drawString(buffer.toString(), 5, 15);
    }

    public void start()
    {
        try
        {
            // Send ICP command - Configure all channels as
digital outputs
            sOut.write(setAllDigitalOutput);

            // Read and ignore replay byte from the Ipsil
device
            sIn.read(resp);

            // Send ICP command - Set HI all on all of the
channels
            sOut.write(setAllDigitalLo);

            // Read and ignore replay byte from the Ipsil
device
            sIn.read (resp);

            addItem(" All Digital Lo Done!");
        }

        catch (Exception e)
        {
            e.printStackTrace();
            addItem(" Start ");
        }
    }
} //end of the applet
```

Έχοντας να επιλύσουμε το πρόβλημα διαχείρισης ενός καναλιού π.χ. 6 όταν τρία κανάλια χρησιμοποιούνται μόνο ως εξόδοι (π.χ. 6,7,8) χρειάζεται να αναπτύξουμε 8 εφαρμογές που θα λαμβάνουν υπόψη τους την αρχική κατάσταση αυτών των εξόδων σύμφωνα με τον πίνακα 3.9:

0:κλειστό	1:ανοικτό		
κατάσταση/κανάλια	6	7	8
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Πίνακας 3.9: Καταστάσεις κατά τις οποίες μπορούν να βρεθούν τα κανάλια 6,7,8

Για να μπορέσουμε να εκτελέσουμε αυτές τις εφαρμογές δημιουργήσαμε το παρακάτω αρχείο java, το οποίο ισχύει για όλες τις περιπτώσεις του πίνακα 3.9 αλλάζοντας μόνο μια γραμμή (σημειώνεται με έντονο μαύρο χρώμα).

```
import java.applet.Applet;
import java.io.*;
import java.net.*;
import java.util.*;
import java.awt.*;

public class setDigitalHi extends Applet
{
    // Constants
    private int IPSIL_PORT = 8930;           // Ipsil device port
number

    // Variables
    InetAddress ia      = null;
    Socket          sock = null;           // Socket
    InputStream     sIn  = null;          // Input data stream
    OutputStream    sOut = null;          // Output data stream
instance

    StringBuffer buffer = new StringBuffer();

    // Set all digital outputs to HI level
    // Write RAM variable ICP command
    // Operation code = 0x06, address in RAM 0x117, 1111 1111 - all
channels "1" output
    // Each bit of the last byte represents each channel state
accordingly
    byte [] setAllDigitalHi =
    { 0x00, 0x00, 0x00, 0x00, 0x00,      // Password
(5 bytes len)
      0x08,                               // "Write serial" cmd   (1
byte len)
      0x00, 0x00, 0x01, 0x17,           // Start location      (4
bytes len)
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
        (byte)0x1F,                // All channels HI
    (1 byte len)
    };

    byte [] resp = new byte[1024];    // Network response buffer

    public void init()
    {
        try
        {

            //ia = InetAddress.getByName("194.177.198.209");
            //ia = InetAddress.getLocalHost();
            //addItem(ia.getHostAddress());
            ia = InetAddress.getByName( getCodeBase().getHost()

);

            // Init TCP socket. Connect to the device itself on
Ipsil port

            sock = new Socket(ia, IPSIL_PORT);
            addItem(ia.toString());

            // Create input data stream
            sIn = sock.getInputStream();

            // Create output data stream
            sOut = sock.getOutputStream();

        }

        catch (Exception e)
        {
            e.printStackTrace();
            addItem(" Init " );
        }

    }

    private void addItem(String newWord)
    {
        System.out.println(newWord);
        buffer.append(newWord);
        repaint();
    }

    public void paint(Graphics g)
    {
        //Draw a Rectangle around the applet's display area.
        g.drawRect(0, 0, getSize().width - 1, getSize().height -
1);

        //Draw the current string inside the rectangle.
        g.drawString(buffer.toString(), 5, 15);
    }

    public void start()
    {
        try
        {
```


Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
channels // Send ICP command - Set HI all on all of the
sOut.write(setAllDigitalHi);

device // Read and ignore replay byte from the Ipsil
sIn.read (resp);

addItem(" All Digital HI Done!");
}

catch (Exception e)
{
    e.printStackTrace();
    addItem(" Start ");
}

}

} //end of the applet
```

Για την κάθε μία κατάσταση του πίνακα 3.9 έχουμε τον ανάλογο hex κωδικό στην συγκεκριμένη εντολή του javascript:

Καταστάσεις	Hex κωδικός
000	1F
001	9F
010	5F
011	DF
100	3F
101	BF
110	7F
111	FF

Για να μπορέσουμε να ελέγχουμε την κατάσταση του καναλιού 6 με εύχρηστο τρόπο, δημιουργήσαμε μία πρόσθετη σελίδα η οποία μας δίνει κάθε φορά ΜΟΝΟ μία από τις 8 επιλογές λαμβάνοντας υπόψη την τρέχουσα κατάσταση των τριών εξόδων και έχοντας ως σκοπό να μην αλλάξει την κατάσταση των υπολοίπων καναλιών 7 και 8. Ο κώδικας παρουσιάζεται παρακάτω:

```
<HTML>
<SCRIPT LANGUAGE="Javascript">
<!--
function U() {return "http://" + location.hostname}
// -->
</SCRIPT>

<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE></TITLE>
</HEAD>
<BODY>

<p align="center"><b><font face="Verdana">Διαχείριση καναλιού
6</font></b><font face="Verdana"></font></p>
<p align="center">&nbsp;</p>
<p align="left"><font face="Verdana"><span style="font-size:10pt;">
```

```
<SCRIPT>
if (    >500)
    R6=1;
else
    R6=0;
if (    >500)
    R7=1;
else
    R7=0;
if (    >500)
    R8=1;
else
    R8=0;

if ((R6==0) & (R7==0) & (R8==0))
document.write("<A href = "+U( )+"/214.htm>ενεργοποίηση</A><br>")
if ((R6==0) & (R7==0) & (R8==1))
document.write("<A href = "+U( )+"/207.htm>ενεργοποίηση</A><br>")
if ((R6==0) & (R7==1) & (R8==0))
document.write("<A href = "+U( )+"/208.htm>ενεργοποίηση</A><br>")
if ((R6==0) & (R7==1) & (R8==1))
document.write("<A href = "+U( )+"/209.htm>ενεργοποίηση</A><br>")
if ((R6==1) & (R7==0) & (R8==0))
document.write("<A href = "+U( )+"/210.htm>απενεργοποίηση</A><br>")
if ((R6==1) & (R7==0) & (R8==1))
document.write("<A href = "+U( )+"/211.htm>απενεργοποίηση</A><br>")
if ((R6==1) & (R7==1) & (R8==0))
document.write("<A href = "+U( )+"/212.htm>απενεργοποίηση</A><br>")
if ((R6==1) & (R7==1) & (R8==1))
document.write("<A href = "+U( )+"/213.htm>απενεργοποίηση</A><br>")
</SCRIPT></span></font></p>
<p align="left">&nbsp;   </p>

</BODY>
</HTML>
```

3.3.2 Δημιουργία πακέτων ICP για έλεγχο σειριακής:

```
import java.applet.Applet;
import java.io.*;
import java.net.*;
import java.util.*;
import java.awt.*;

public class setDigitalHi extends Applet
{
    // Constants
    private int IPSIL_PORT = 8930;           // Ipsil device port
number

    // Variables
    InetAddress ia      = null;
    Socket        sock  = null;           // Socket
    InputStream   sIn   = null;         // Input data stream
    OutputStream  sOut  = null;         // Output data stream
instance
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
StringBuffer buffer = new StringBuffer();

// Set all digital outputs to LO level
// Write RAM variable ICP command
// Operation code = 0x06, address in RAM 0x117, 0000 0000 - all
channels "1" output
// Each bit of the last byte represents each channel state
accordinly
byte [] setAllDigitalHi =
{ 0x00, 0x00, 0x00, 0x00, 0x00, // Password      (5 bytes len)
  0x08, // "Write serial" cmd (1 byte len)
  0x00, 0x00, 0x00, 0x00, // Start location (4 bytes len)
  (byte)0x54, (byte)0X45, (byte)0X53, (byte)0X54, //
All channels LO (1 byte len)
};

byte [] resp = new byte[1024]; // Network response
buffer

public void init()
{
    try
    {
        //ia = InetAddress.getByName("10.1.1.20");
        //ia = InetAddress.getLocalHost();
        //addItem(ia.getHostAddress());
        ia = InetAddress.getByName( getCodeBase().getHost()
);
        // Init TCP socket. Connect to the device itself on
Ipsil port
        sock = new Socket(ia, IPSIL_PORT);
        addItem(ia.toString());

        // Create input data stream
        sIn = sock.getInputStream();

        // Create output data stream
        sOut = sock.getOutputStream();

    }

    catch (Exception e)
    {
        e.printStackTrace();
        addItem(" Init " );
    }

}

private void addItem(String newWord)
{
    System.out.println(newWord);
    buffer.append(newWord);
    repaint();
}

public void paint(Graphics g)
```

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

```
    {
        //Draw a Rectangle around the applet's display area.
        g.drawRect(0, 0, getSize().width - 1, getSize().height -
1);

        //Draw the current string inside the rectangle.
        g.drawString(buffer.toString(), 5, 15);
    }

    public void start()
    {
        try
        {

            // Send ICP command - Set HI all on all of the
channels
            sOut.write(setAllDigitalHi);

            // Read and ignore replay byte from the Ipsil
device
            sIn.read (resp);

            addItem(" Send bytes to serial Done!");
        }

        catch (Exception e)
        {
            e.printStackTrace();
            addItem(" Start ");
        }
    }
} //end of the applet
```

Στην επόμενη παράγραφο θα δούμε πώς θα τα μεταφράσουμε (compile) και ποια θα είναι η μορφή τους πριν να αποθηκευτούν στο IPμ8930.

3.3.2 Compile and archive

3.3.2.1 Σημειώσεις εφαρμογής

Αυτή η ενότητα παρουσιάζει πώς χρησιμοποιούνται τα java applets για να διαμορφώσουν και να ενεργοποιήσουν/απενεργοποιήσουν τα ψηφιακά κανάλια στο IPμ8930. Αυτά τα java applets χρησιμοποιούν το "socket class" για να επικοινωνήσουν με τη συσκευή της Ipsil. Συγκεκριμένα, αυτά τα java applets χρησιμοποιούν το ICP over TCP για να έχουν πρόσβαση σε όλους τους πόρους I/O και αποθήκευσης του IPμ8930.

Πρέπει να εξεταστούν διάφορα ζητήματα κατά τη χρησιμοποίηση των java applets με το IPμ8930:

- Ασφάλεια: Τα java applets μπορούν να εντοπίσουν μόνο μια σύνδεση TCP/UDP στη συσκευή, από την οποία το applet «τρέχει». Κατά συνέπεια, όλα τα java applets πρέπει πάντα να χρησιμοποιούν μία συγκεκριμένη διεύθυνση IP του IPμ8930 για όλες τις αναγνώσεις και εγγραφές.

- Χρήση αρχείων JAR αντί των java applets σε ιστοσελίδες: Το σύστημα αρχείων IPμ8930 απαιτεί το πρόθεμα όλων των ονομάτων αρχείου για να είναι ένας τριψήφιος αριθμός (π.χ., «160myfile», «170myfile»). Επιπλέον, τα ονόματα κατηγορίας για τα java applets δεν μπορούν να αρχίζουν με αριθμό. Η λύση είναι να χρησιμοποιούνται μόνο αρχεία JAR που αποθηκεύουν τα java applets. Ένα αρχείο JAR είναι ένα συμπιεσμένο αρχείο των java applets (ή οποιωνδήποτε άλλων αρχείων) και μπορεί να έχει ένα όνομα που να αρχίζει με έναν αριθμό. Το αρχείο JAR έχουν επίσης ένα επιπλέον όφελος. Καταλαμβάνουν λιγότερο χώρο στο σύστημα αρχείων IPμ8930 δεδομένου ότι είναι συμπιεσμένα. Παραδείγματος χάριν, ένα έγκυρο όνομα αρχείου είναι το «203Classes.jar» και θα χρησιμοποιηθεί εάν το αρχείο JAR θα αποθηκευτεί στην θέση (slot) 203 στο σύστημα αρχείων IPμ8930. Για να χρησιμοποιηθούν τα αρχεία JAR σε μια ιστοσελίδα, χρησιμοποιούμε την επιλογή «ARCHIVE = [jar file name]» μέσα στον κώδικα της ιστοσελίδας μας. (εικόνα 3.6)

- Γεμίζοντας τα αρχεία JAR για να είναι πολλαπλάσια των 512 byte: Το σύστημα αρχείων IPμ8930 ομαδοποιεί τα αρχεία σε πολλαπλάσια των 512 byte. Όταν ένα αρχείο java JAR μεταφέρεται σε έναν web browser, μεταφέρονται τα δεδομένα που είναι πολλαπλάσια των 512 bytes. Εντούτοις, εάν το αρχείο JAR δεν είναι ακριβές πολλαπλάσιο των 512 bytes, θα υπάρχουν μερικά trailing bytes (< 512) που στέλνονται επίσης στον web browser. Μερικοί web browsers μπορούν να εμποδίσουν αυτό το αρχείο JAR ως πιθανό ιό λόγω αυτών των trailing bytes. Η λύση είναι να δημιουργηθούν αρχεία JAR που να είναι ακριβώς πολλαπλάσια 512 bytes στο μέγεθος. Αυτό μπορεί να γίνει εύκολα με την προσθήκη ασυμπίεστου αρχείου κειμένου που περιέχει τόσο πολλά διαστήματα (spaces), όσα απαιτούνται για να γίνει το αρχείο JAR αρχείο πολλαπλάσιο των 512 bytes.

3.2.2.2 Χρήσιμες οδηγίες για τη δημιουργία και τη χρησιμοποίηση των αρχείων JAR σε ιστοσελίδες

Τα παρακάτω είναι μερικά χρήσιμα βήματα για να δημιουργήσουμε και να διαχειριστούμε τα java applets μας με το IPμ8930:

- 1) Δημιουργούμε ένα αρχείο "myclass.java" (βλ. κεφ. 3.3.1) με τον java κώδικα μας και το μεταφράζουμε σε ένα αρχείο java.class με τη χρησιμοποίηση της εντολής "javac myclass.java". Κατά συνέπεια, προκύπτει το αρχείο "myclass.class" αρχείο

- 2) Ομοίως, δημιουργούμε όσα άλλα αρχεία .class χρειαζόμαστε για το πρόγραμμά μας (π.χ., "myclass2.class", "myclass3.class")

- 3) Δημιουργούμε ένα κενό αρχείο «trailer.txt». Θα χρησιμοποιηθεί αργότερα για την δημιουργία του αρχείου JAR μεγέθους πολλαπλάσιο 512 bytes.

- 4) Χρησιμοποιούμε την εντολή "jar -cfv 203Classes.jar myclass.class myclass2.class..." για την δημιουργία του αρχείου 203Classes.jar. (Σημείωση: το πρόθεμα 203 για το αρχείο JAR αντιπροσωπεύει τη θέση (slot) αρχείου στο σύστημα αρχείων του IPμ8930 όπου το JAR τελικά αποθηκεύεται. Αυτό το πρόθεμα δύναται να αλλάζει ανάλογα με την περίπτωση.

- 5) Χρησιμοποιούμε την εντολή "jar -ufv0 203Classes.jar trailer.txt" για την προσθήκη του κενού αρχείου trailer στο java αρχείο.

- 6) Ελέγχουμε το μέγεθος του προκύπτοντος αρχείου jar. Εάν δεν είναι ακριβώς πολλαπλάσιο των 512 bytes, προσθέτουμε αριθμό X διαστημάτων στο αρχείο «trailer.txt», όπου το X είναι η διαφορά μεταξύ του τρέχοντος μεγέθους του αρχείου jar (σε bytes) και του επόμενου πολλαπλάσιου των 512 bytes. Παραδείγματος χάριν, εάν το τρέχον αρχείο JAR είναι 2040bytes, το επόμενο πολλαπλάσιο των 512 bytes είναι το 2048. Συνεπώς, θα πρέπει να προσθέσουμε 8 διαστήματα στο αρχείο trailer.txt

7) Χρησιμοποιούμε την εντολή «jar -ufv0 203Classes.jar trailer.txt» για να προσθέσουμε αυτό το αρχείο ασυμπίεστο στο java applet αρχείο. Ελέγχουμε το μέγεθος του αρχείου JAR. Πρέπει τώρα να είναι ένα ακριβές πολλαπλάσιο 512bytes.

8) Δημιουργούμε ένα απλό αρχείο HTML για την εκτέλεση του applet(s) στο αρχείο JAR. Παραδείγματος χάριν, ο παρακάτω κώδικας HTML μπορεί να χρησιμοποιηθεί για την ιστοσελίδα «200.htm»:

```
<HTML>

<SCRIPT LANGUAGE="Javascript">
<!--
function U() {return "http://" + location.hostname}
// -->
</SCRIPT>

<HEAD>
<TITLE>
Control an Ipsil device.
</TITLE>
</HEAD>
<BODY>

<SCRIPT>
document.write("All Digital LO<BR>")
document.write("<APPLET CODE = MyClass.class" + " "
+ " CODEBASE = " + U() + " "
+ " NAME = MyClass" + " "
+ " ARCHIVE = 203Classes.jar" + " "
+ " WIDTH = 500 HEIGHT =400" + " "
+ " VIEWASTEXT >")
</SCRIPT>
</BODY>
</HTML>
```

Εικόνα 3.6: κώδικας HTML για την εμφάνιση εντολών

9) Ο παραπάνω κώδικας HTML έχει την εντολή **U ()**, η οποία επιστρέφει την τρέχουσα διεύθυνση IP από τη συσκευή από την οποία η σελίδα HTML έχει φορτωθεί. Αυτή η εντολή επιτρέπει στον κώδικα HTML να γραφτεί ανεξάρτητα από την γνώση της διεύθυνσης IP IPμ8930. Αυτό είναι σημαντικό επειδή η διεύθυνση IP του IPμ8930 μπορεί πάντα να είναι αλλαγμένη από έναν χρήστη ή τον DHCP server αργότερα, χωρίς να υπάρχει πρόβλημα στις HTML σελίδες. Η διεύθυνση IP που επιστρέφεται από αυτήν την εντολή χρησιμοποιείται για να δημιουργήσει μια δυναμική σύνδεση HTML με το java applet που χρησιμοποιεί τη μέθοδο "document.write". Αυτή η σύνδεση θα αρχίσει το java applet MyClass.class από το αρχείο 203Classes.jar.

10) Φορτώνουμε το αρχείο JAR και την ιστοσελίδα HTML στο σύστημα αρχείων IPμ8930 στις επιθυμητές ελεύθερες θέσεις (slot) αρχείων (π.χ., slot 200 για το αρχείο HTML και την slot 203 για το αρχείο JAR) μέσω του ICU/Upload (βλέπε εικόνα 3.5)

11) Ανοίγουμε έναν web browser και εισάγουμε τη διεύθυνση για την ιστοσελίδα "200.htm". Παραδείγματος χάριν, εάν η διεύθυνση IP για το IPμ8930 μας είναι 192.168.0.254 και η σελίδα είναι στο slot 200, εισάγει http://192.168.0.254/200.htm σαν διεύθυνση.

Με τον παραπάνω τρόπο μεταφράζουμε τα 8 αρχεία που δημιουργήσαμε προηγουμένως και τα "ανεβάζουμε" στο kit. Επίσης δημιουργούμε 9 ιστοσελίδες

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

ακόμα. Οι 8 θα περιέχουν τα αρχεία java applets και η 9^η θα συμπεριλαμβάνει τις 8 ιστοσελίδες με τέτοιον τρόπο ώστε κάθε φορά να παρουσιάζεται η κατάλληλη. Όλες αυτές οι ιστοσελίδες χρησιμοποιούνται για ενεργοποίηση/απενεργοποίηση του καναλιού 6 χωρίς να επηρεάζονται τα άλλα 2.

Τα αρχεία που έχουμε δημιουργήσει αναλύονται παρακάτω:

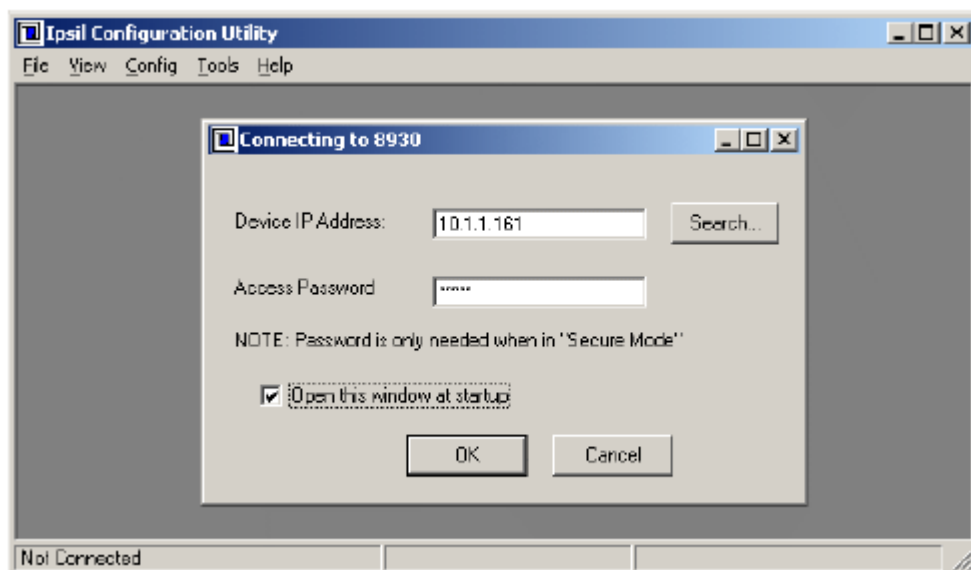
	Κεντρική σελίδα: 030.htm			
Καταστάσεις	Παρακολούθηση ιδιοτήτων Σελίδα:020.htm	Παρακολούθηση Αναλογ.εισόδων Σελίδα:153.htm	Διαχείριση Αισθ./συσκ. Σελίδα:235.htm	Αποστολή Δεδ. σειριακή Σελίδα:303.htm
1			207.htm 207setDigitalHi.java 215.jar	303.htm 303setDigitalHi.java 300.jar
2			208.htm 208setDigitalHi.java 217.jar	
3			209.htm 209setDigitalHi.java 219.jar	
4			210.htm 210setDigitalHi.java 221.jar	
5			211.htm 211setDigitalHi.java 223.jar	
6			212.htm 212setDigitalHi.java 225.jar	
7			213.htm 213setDigitalHi.java 227.jar	
8			214.htm 214setDigitalHi.java 229.jar	

Για κάθε εντολή (ενεργοποίηση ή απενεργοποίηση) χρειάζονται δύο αρχεία, 2XX.htm και 2XX.jar. Το 3ο αρχείο που αναφέρεται παραπάνω, 2XXsetDigitalHi.java, είναι ο πηγαίος κώδικας.

3.4 Ipsil Config Utility (ICU)

Το Ipsil Config Utility (ICU) χρησιμοποιείται για να διαμορφώσει το IPμ8930. Αυτή η παράγραφος εξηγεί πώς διαμορφώνει μερικές από τις πιο κοινές παραμέτρους IPμ8930 (π.χ., το user-assigned IPμ8930 name ή τον προκαθορισμό της IP διεύθυνσης).

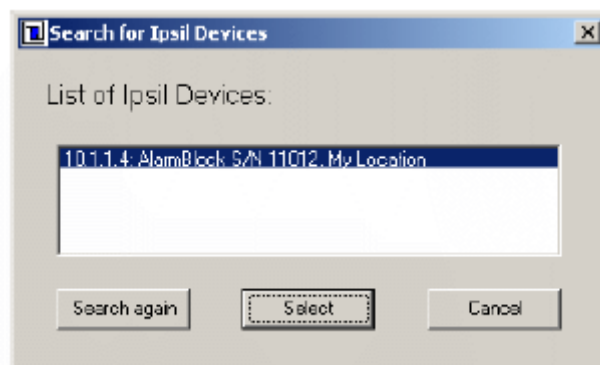
Όταν ανοίγουμε το ICU, θα παρουσιαστεί η ακόλουθη οθόνη (εικόνα 3.7). Εναλλακτικά, μπορούμε, επίσης, να έχουμε πρόσβαση σε αυτήν την οθόνη με την επιλογή file > connect menu option.



Εικόνα 3.7: σύνδεση στο Development kit μέσω του ICU

Ο προεπιλεγμένος κωδικός πρόσβασης είναι "ipsil". Εάν ξέρουμε τη διεύθυνση IP του IPμ8930 με το οποίο θέλουμε να συνδεθούμε, μπορούμε να το εισαγάγουμε στο πλαίσιο κειμένου IP διευθύνσεων συσκευών και να πατήσουμε το OK. Εντούτοις, εάν δεν ξέρουμε τη IP διεύθυνση για το IPμ8930, μπορούμε να το αναζητήσουμε, πατώντας το κουμπί search, μέσα στο ίδιο υποδίκτυο με τον υπολογιστή μας. Εάν καμία συσκευή IPμ8930 δεν βρεθεί, θα λάβουμε ένα μήνυμα λάθους.

Εάν ένα ή περισσότερα IPμ8930s βρεθούν, η ακόλουθη ανίχνευση για το παράθυρο διαλόγου συσκευών Ipsil θα παρουσιαστεί (εικόνα 3.8). Αυτό το παράθυρο διαλόγου απαριθμεί όλες τις συσκευές IPμ8930 που βρίσκονται στο ίδιο υποδίκτυο με τον υπολογιστή μας.



Εικόνα 3.8: εύρεση συσκευής Ipsil στο υποδίκτυο

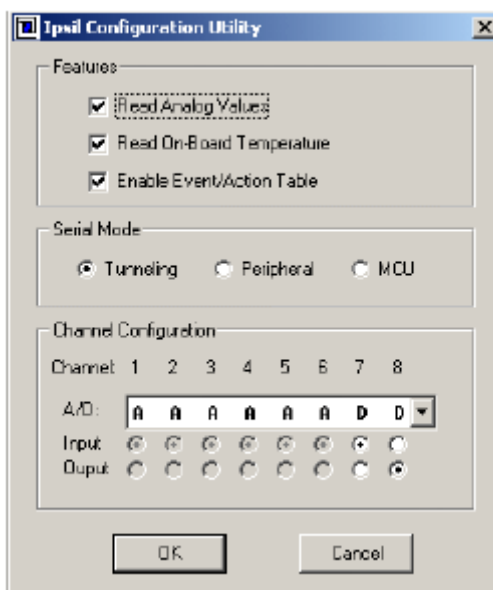
Πατάμε select για να επιλέξουμε ένα IPμ8930. Αυτό θα κλείσει την ανίχνευση για το παράθυρο συσκευών Ipsil και θα αντιγράψει την επιλεγμένη διεύθυνση IP στο Configure Connection window. Κατόπιν πατάμε OK στο Configure Connection window για να συνδεθούμε με το IPμ8930.

Το ICU θα θυμηθεί τη IP διεύθυνση του τελευταίου IPμ8930 με το οποίο συνδεθήκαμε επιτυχώς. Εάν δεν επιλέξουμε το open this window at startup, το ICU θα χρησιμοποιήσει την τελευταία IP διεύθυνση που χρησιμοποιήθηκε

3.4.1 Configuring the I/O Channels

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

Για να ενεργοποιήσουμε/απενεργοποιήσουμε τα χαρακτηριστικά των I/O και καναλιών στο IPμ8930, επιλέγουμε Config > I/O menu option. Το ακόλουθο παράθυρο διαλόγου θα εμφανιστεί (εικόνα 3.3):



Εικόνα 3.9: χαρακτηριστικά των I/O

Στον παρακάτω πίνακα περιγράφονται τα παραπάνω πεδία:

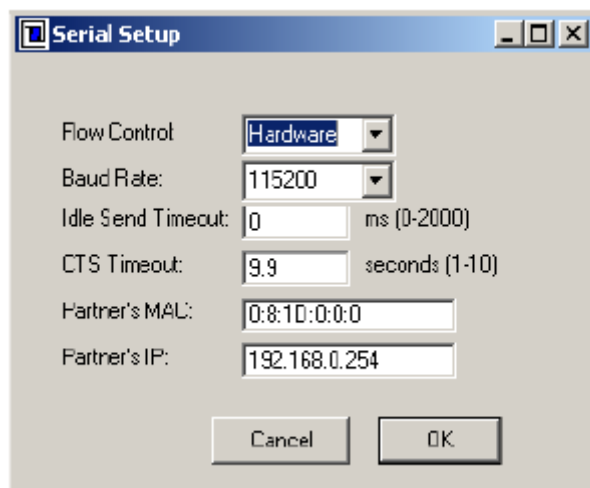
Πεδίο	Περιγραφή	Διαχειριζόμενο από τον χρήστη	Απαιτείται	Προεπιλογή
<i>Read Analog Value</i>	Καθορίζει αν είναι ενεργοποιημένη η ανάγνωση των αναλογικών τιμών	Ναι	Ναι	Enabled
<i>Read On-Board Temperature</i>	Καθορίζει αν είναι ενεργοποιημένη η ανάγνωση της θερμοκρασίας του Kit	Ναι	Ναι	Enabled
<i>Enable event/action table</i>	Καθορίζει αν είναι ενεργοποιημένο το event/action table	Ναι	Ναι	Enabled

<i>Serial mode</i>	Καθορίζει τον σειριακό τρόπο για το Ipm8930.Tunneling mode: Μπορούν να χρησιμοποιηθούν δύο Ipm8930 για να συνδέσουν δύο σειριακές συσκευές σε δίκτυο. Peripheral mode: ένα Ipm8930 μπορεί να χρησιμοποιηθεί για να λάβει/στείλει δεδομένα από μία σειριακή συσκευή χρησιμοποιώντας TCP/UDP calls. MCU mode: Για να μπορεί να επικοινωνήσει με έναν μικροελεγκτή	Ναι	Ναι	Serial Tunnel
<i>Channel Configuration</i>	Εξαρτάται από τα χαρακτηριστικά που έχουν επιλεγεί παραπάνω. Καθορίζει συνδυασμούς αναλογικών/ψηφιακών καναλιών και για το αν είναι είσοδοι/έξοδοι	Ναι	Ναι	

Πίνακας 3.10: περιγραφή των χαρακτηριστικών I/O καναλιών

3.4.2 Διαμόρφωση του Serial Interface

Για να αλλάξουμε τις ιδιότητες της σειριακής θύρας για το Ipm8930, επιλέγουμε Config > Serial menu option στο Config Utility. Το ακόλουθο παράθυρο διαλόγου θα εμφανιστεί (εικόνα 3.9):



Εικόνα 3.10: διαμόρφωση σειριακής

Παρακάτω αναλύονται όλα τα πεδία του παραθύρου (εικόνα 3.10)

Πεδίο	Περιγραφή	Διαχειριζόμενο από τον χρήστη	Απαιτείται	Προεπιλογή
<i>Flow Control</i>	Τύπος ελέγχου ροής που χρησιμοποιείται για τη σειριακή επικοινωνία. Οι επιλογές είναι Χον/Χoff, CTS/RTS and None	Ναι	Ναι	CTS/RTS
<i>Baud Rate</i>	Ο Ρυθμος της απόδοση που χρησιμοποιείται για τις σειριακές επικοινωνίες. Οι επιλογές είναι 9600 ..19200 ..28800 ..33600 ..57600 και 115200 baud	Ναι	Ναι	115200
<i>Idle Send Timeout</i>	Το μη ενεργό διάστημα διαλείμματος στο σειριακό interface μετά από ένα πακέτο IP που τοποθετεί σειριακά τα στοιχεία που έχουν δημιουργηθεί από το IPμ8930	Ναι	Ναι	0
CTS Timeout	Διάστημα διαλείμματος για CTS	Ναι	Ναι	9.9
<i>Partner's MAC</i>	Διεύθυνση MAC για το IPμ8930 που συνεργάζεται. Απαιτήσε μόνο όταν το IPμ8930 χρησιμοποιείται με τον σειριακό τρόπο	Ναι	Όχι	
<i>Partner's IP</i>	IP διεύθυνση για το IPμ8930 που συνεργάζεται. Απαιτήσε μόνο όταν το IPμ8930 χρησιμοποιείται με τον σειριακό τρόπο	Ναι	Όχι	

Πίνακας 3.11: περιγραφή των πεδίων του παραθύρου της διαμόρφωσης της σειριακής

3.4.3 Διαμόρφωση αισθητήρων

Το IPμ8930 Developer Board υποστηρίζει αισθητήρες 4-20ma, αναλογικούς ή επαφής 0-5V. Εξ ορισμού, όλα τα κανάλια είναι ρυθμισμένα για να λειτουργούν με αισθητήρες (ως είσοδοι) 4-20ma. Η αλλαγή των τύπων αισθητήρων από 4-20ma σε έναν άλλο τύπο αισθητήρων απαιτεί μια αλλαγή σε ένα εσωτερικό jumper. (Παράρτημα Α για περιγραφή της θέσης και τις τοποθετήσεις για τον τύπο jumper του αισθητήρα.)

Η βασική διαφορά μεταξύ ενός αισθητήρα 4-20ma και ενός αισθητήρα 0-5V είναι ότι το πρώτο παράγει ένα ρεύμα μεταξύ 4 και 20ma, ενώ το τελευταίο παράγει μια τάση μεταξύ 0 και 5V. Και το ρεύμα και η τάση που παράγεται από

αυτούς τους δύο τύπους αισθητήρων αντιπροσωπεύουν την αξία αισθητήρων σύμφωνα με κάποια προκαθορισμένη λειτουργία μετατροπής. Ένα παράδειγμα ενός αισθητήρα υγρασίας είναι ένας διακόπτης αναστροφής ή οι επαφές ενός ρελέ. Καμία τάση ή ρεύμα δεν εισάγεται στο IPμ8930 — αντ' αυτού, το σήμα και η γείωση του τερματικού είναι είτε κλειστή είτε ανοιχτή, ανάλογα με την κατάσταση του αισθητήρα. Οι αισθητήρες 4-20ma είναι τυπικά επικρατέστεροι στις βιομηχανίες όπου λειτουργούν με μια πηγή ενέργειας 24V και διανύουν μεγάλες αποστάσεις. Οι αισθητήρες 0-5V είναι πιο κοινοί στις μη βιομηχανικές καταστάσεις και λειτουργούν με μια 5V παροχή ηλεκτρικού ρεύματος. Οι αισθητήρες 0-5V δεν είναι καλά προσαρμοσμένοι για να διανύσουν μεγάλες αποστάσεις λόγω της εξασθένισης της τάσης και των παρεμβολών.

Το IPμ8930 υποστηρίζει μέχρι δέκα bit ακρίβειας και για 4-20ma και για τους αισθητήρες 0-5V. Το αναλογικό ρεύμα ή η τάση που παράγεται από αυτούς τους αισθητήρες μετατρέπεται από το IPμ8930 σε έναν γραμμικά 10-bit μεταξύ 0 και 1023. Αυτός ο αριθμός 10-bit αντιπροσωπεύει τον 4-20ma ή 0-5V αναλογικό αισθητήρα ο οποίος πρέπει να μετατραπεί περαιτέρω σε μια τιμή "πραγματικού φυσικού μεγέθους" σύμφωνα με τη συγκεκριμένη λειτουργία μετατροπής αισθητήρων που θα μετατρέψει την "raw" τιμή σε μια "πραγματική" όπως η θερμοκρασία. Αυτή η μετατροπή γίνεται από τις client εφαρμογές που αλληλεπιδρά με το IPμ8930 (δηλ., ιστοσελίδας HTML ή το java applet). Οι πιο χαρακτηριστικές 4-20ma λειτουργίες μετατροπής είναι γραμμικές, τετραγωνικές, αλγοριθμικές, εκθέτης και θερμοηλεκτρικά ζεύγη. Υπάρχουν μερικοί αισθητήρες που έχουν λειτουργίες μετατροπής που δεν μπορούν να περιγραφούν επαρκώς από μια απλή λειτουργία. Σε αυτές τις περιπτώσεις ο raw αριθμός 0-1023 θα πρέπει να μετατραπεί σε μια τιμή αισθητήρων που υπολογίζεται με βάση έναν πίνακα look-up από μια client εφαρμογή.

Για να διαμορφωθούν οι αισθητήρες που συνδέονται με IPμ8930, ακολουθούμε πέντε βήματα:

Βήμα 1. Ανοίγουμε το ICU και συνδεόμαστε στο IPμ8930 όπου ο νέος αισθητήρας έχει συνδεθεί με καλώδιο και επιλέγουμε Tools> Configure Sensor. Θα μας παρουσιαστεί ένα νέο παράθυρο για να διαμορφώσουμε το αισθητήριο.

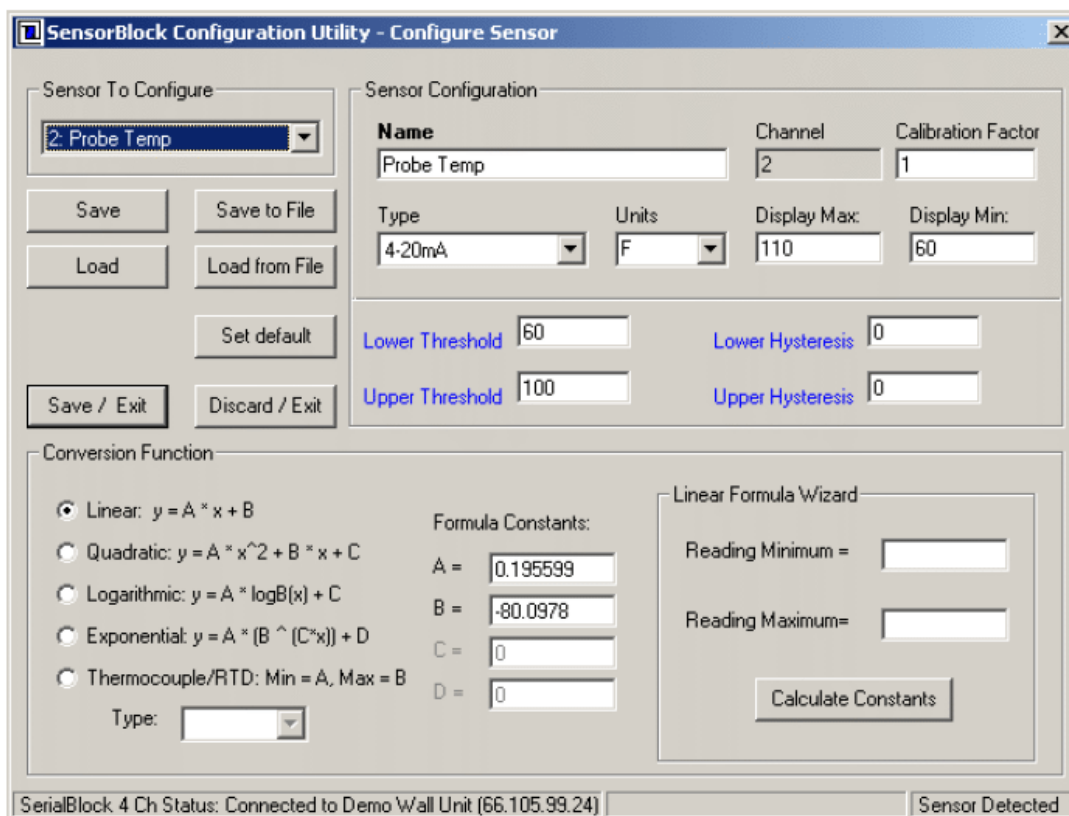
Βήμα 2. Επιλέγουμε τον αισθητήρα που είναι για διαμόρφωση από το drop-down menu βασισμένο στο κανάλι στο οποίο το νέος αισθητήριο έχει συνδεθεί με καλώδιο.

Βήμα 3. Εισάγουμε τις γενικές πληροφορίες διαμόρφωσης αισθητήρων όπως το όνομα, τύπος, κ.λπ.

Βήμα 4. Καθορίζουμε τη λειτουργία μετατροπής για τον αισθητήρα.

Βήμα 5. Αποθηκεύουμε τις πληροφορίες διαμόρφωσης στη συσκευή.

Οι ακόλουθες παράγραφοι παρέχουν πρόσθετες λεπτομέρειες για τις διαδικασίες διαμόρφωσης και μετατροπής που απαιτούνται στο παράθυρο (window) διαμόρφωση αισθητήρων (εικόνα 3.11).



Εικόνα 3.11: διαμόρφωση αισθητήριων

Το τμήμα διαμόρφωσης αισθητήρων αποτελείται από τους ακόλουθα πεδία:

Πεδίο	Περιγραφή	Διαχειριζό μενο από τον χρήστη	Απαιτεί ται	Προεπιλογή
Name	Όνομα που θέλει να δώσει ο χρήστης (32 χαρακτήρες μέγιστο) για το αισθητήριο το οποίο μπορεί να εμφανιστεί στην καταγραφή αισθητήρων στις ιστοσελίδες και στο WebLogger πρόγραμμα.	Ναι	Όχι	X αισθητήρας (όπου «x» είναι ο αριθμός καναλιού)
Channel	Το κανάλι στο οποίο ο αισθητήρας είναι συνδεδεμένος		Ναι	Το ίδιο όπως το Sensor to configure drop-down item
Sensor type	Καθορίζει τον τύπο του αισθητηρίου: 4-20ma, 0-5V, 1-5V, υγρασίας ή I2C (αισθητήρας ες. Θερμοκρασίας)	Ναι	Ναι	κενό
Units	Μονάδες για τον αισθητήρα (π.χ., m, Celsius, psi). Μια λίστα συνηθισμένων μονάδων	Ναι	Ναι	Κενό

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

	είναι διαθέσιμες μέσω του pull-down menu. Αν δεν υπάρχει η μονάδα που θέλουμε, μπορούμε να την καθορίσουμε εμείς (μέγιστο 8 χαρακτήρες) στο πεδίο.			
Calibration factor	Το calibration factor συνθέτει την έξοδο από την συνάρτηση της μετατροπής αυτού του αριθμού. Με άλλα λόγια, αν $y = f(x)$ είναι η συνάρτηση μετατροπής του αισθητήρα και CF είναι το conversion factor, μετά η "πραγματική τιμή" του αισθητήρα γράφεται και εμφανίζεται μέσω των ιστοσελίδων και WebLogger είναι $y = CF * f(x)$. Εκτός αν είμαστε σίγουροι ότι ο αισθητήρας χρειάζεται να καλιμπραριθεί, αυτός ο αριθμός θα πρέπει να παραμείνει στο 1.	Ναι	Ναι	1.0
Display min	Καθορίζει το κάτω όριο εξόδου στις ιστοσελίδες και στο WebLogger. Αυτή η μεταβλητή είναι μόνο για σκοπό παρουσίασης και δεν σχετίζεται με το λειτουργικό όριο του αισθητήρα.	Ναι	Ναι	Κενό
Display max	Καθορίζει το άνω όριο εξόδου στις ιστοσελίδες και στο WebLogger. Αυτή η μεταβλητή είναι μόνο για σκοπό παρουσίασης και δεν σχετίζεται με το λειτουργικό όριο του αισθητήρα.	Ναι	Ναι	κενό
Upper treshold	Καθορίζει το άνω όριο για τους συναγερμούς του αισθητήρα. Επίσης, καθορίζει την τιμή μέγιστης εμφάνισης στο οποίο η έξοδος παριστάνεται σε ιστοσελίδες και WebLogger προβάλλεται με ξεχωριστό χρώμα.	Ναι	Ναι	κενό
Lower treshold	Καθορίζει το κάτω όριο για τους συναγερμούς του	Ναι	Ναι	κενό

Διαχείριση περιφερειακών συσκευών - αισθητήρων μέσω πρωτοκόλλου TCP/IP

	αισθητήρα. Επίσης καθορίζει την τιμή μέγιστης εμφάνισης στο οποίο η εξόδος παριστάνεται σε ιστοσελίδες και στο WebLogger προβάλλεται με ξεχωριστό χρώμα.			
Upper hysteresis	Καθορίζει το άνω όριο hysteresis (εντελώς διαφορετικό από το άνω threshold) για τους συναγερμούς του αισθητήρα.	Ναι	Ναι	0
Lower hysteresis	Καθορίζει το κάτω όριο hysteresis (εντελώς διαφορετικό από το κάτω threshold) για τους συναγερμούς του αισθητήρα.	Ναι	Ναι	0

Πίνακας 3.12 : περιγραφή των πεδίων του παραθύρου της διαμόρφωσης των αισθητήρων

Ο χρήστης πρέπει επίσης να καθορίσει μια συνάρτηση μετατροπής για τον αισθητήρα με την επιλογή του τύπου μετατροπής και τη συμπλήρωση των κατάλληλων πεδίων:

Πεδίο	Περιγραφή	Διαχειριζόμε νο από τον χρήστη	Απαιτείται	Προεπιλογή
<i>Conversion Function</i>	Καθορίζει τον τύπο συνάρτησης μετατροπής που αντιπροσωπεύει καλύτερα τον αισθητήρα. Το IPμ8930 υποστηρίζει γραμμικό, τετραγωνικό, εκθετικό, λογάριθμο και θερμοηλεκτρικό ζεύγος συναρτήσης μετατροπής. Η πιο συνηθισμένη συνάρτηση μετατροπή για τους αισθητήρες είναι η γραμμική.	Ναι	Ναι	<i>Linear</i>
<i>Function Constants</i>	Τιμή των σταθερών για την επιλεγμένη συνάρτηση μετατροπής για να μετατρέψει τις raw τιμές (0 to 1023) σε πραγματικές τιμές (π.χ., 75F).	Ναι	Ναι	κενό
<i>Linear Formula Wizard</i>	Εάν η συνάρτηση μετατροπής αισθητήρων είναι γραμμική, ο γραμμικός τύπου wizard μπορεί να χρησιμοποιηθεί για να υπολογίσει αυτόματα τις σταθερές. Απλά εισάγουμε την ελάχιστη και	Ναι	Όχι	κενό

	μέγιστη τιμή που συνδέεται με 4ma/1V και 20ma/5V και πατάμε Calculate Constants. Το ICU θα υπολογίσει αυτόματα τις γραμμικές σταθερές τύπου.			
--	--	--	--	--

Πίνακας 3.13: Συνάρτηση μετατροπής του αισθητήρα

Αφού έχει διαμορφωθεί ο αισθητήρας, η νέα διαμόρφωση αισθητήρων πρέπει να αντιγραφεί στο IPμ8930 επιλέγοντας κουμπί Save. Οι αλλαγές δεν έχουν αποθηκευτεί στο IPμ8930 έως ότου να πατήσουμε Save. Πατώντας Cancel θα αναγκάσει το παράθυρο να κλείσει χωρίς να έχει σωθεί η ενημερωμένη διαμόρφωση αισθητήρων στο IPμ8930.

Η παραπάνω σειρά των βημάτων πρέπει να επαναληφθεί για κάθε αισθητήρα που πρέπει να διαμορφωθεί.

Το παράθυρο διαμόρφωσης αισθητήρων παρέχει, επίσης, τη δυνατότητα να διαγραφτεί ένας αισθητήρας.

Εάν επιθυμούμε να αφαιρέσουμε ένα κανάλι αισθητήρων από το IPμ8930, επιλέγουμε το κανάλι στο Sensor to Configure section, και πατάμε το κουμπί Delete Sensor. Θα εμφανιστεί ένα παράθυρο διαλόγου για να επιβεβαιώσουμε τη διαγραφή.

Εάν έχουμε και άλλα IPμ8930 με την ίδια διαμόρφωση αισθητήρων, μπορούμε να χρησιμοποιήσουμε Save to File και Load from file από τα κουμπιά αρχείων για να επιστεφύσουμε τη διαδικασία διαμόρφωσης. Αρχίζουμε με τη διαμόρφωση του πρώτου IPμ8930. Αφότου έχει διαμορφωθεί, πατάμε το κουμπί Save to File και σώζουμε τις παραμέτρους διαμόρφωσης τοπικά στο σκληρό δίσκο μας. Κατόπιν, συνδέουμε την επόμενη συσκευή IPμ8930, επιλέγουμε Config > Sensors menu option και πατάμε το κουμπί Load from file. Όταν εμφανιστεί, εντοπίζουμε το αρχείο που περιέχει τις παραμέτρους διαμόρφωσης αισθητήρων στο σκληρό δίσκο μας.

3.4.4 Διαμόρφωση συναγεργμών

Ένα από τα κύρια χαρακτηριστικά του IPμ8930 είναι η ικανότητά του να ενεργοποιήσει/απενεργοποιήσει ένα ή περισσότερα κανάλια εξόδου βασισμένη σε ένα προκαθορισμένο όριο κατωφλίου.

Αυτά τα γεγονότα/συναγεργμοί ελέγχονται μέσω ενός εσωτερικού πίνακα γεγονότων, ο οποίος μπορεί να διαμορφωθεί χρησιμοποιώντας το ICU.

Ο πίνακας γεγονότων είναι απλά ένας κατάλογος όρων που ελέγχονται διαδοχικά κάθε 8ms

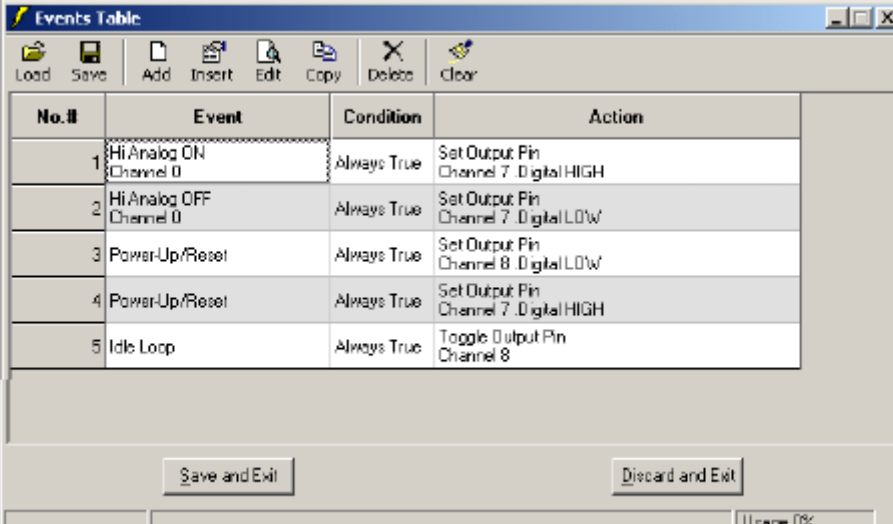
#	Event	Action
1	Event1	Turn channel 4 on
2	Event2	Turn channel 3 on
3	Event3	Turn channel 2 off
4	Event4	Turn channel 1 off
...

Η δημιουργία των συναγεργμών είναι μία διαδικασία δύο βημάτων:

Βήμα 1: Δημιουργούμε τις τιμές κατωφλίου και (προαιρετικά) μια τιμή υστέρησης για τους αισθητήρες. Ένα ανώτερο και χαμηλότερο όριο κατωφλίου μπορεί να καθοριστεί για κάθε κανάλι εισαγωγής. Η διαδικασία για τον καθορισμό ενός αισθητήρα εξηγείται στην προηγούμενη παράγραφο.

Βήμα 2: Καθορίζουμε ένα ή περισσότερα γεγονότα/ συναγεμμούς χρησιμοποιώντας το ICU.

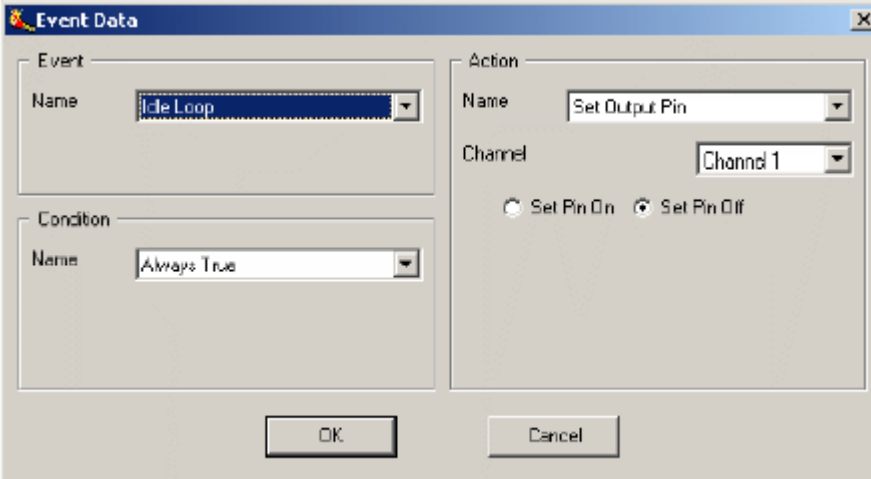
Επιλέγουμε Config > Events Editor από του ICU. Θα εμφανιστεί το παρακάτω παράθυρο:



No. #	Event	Condition	Action
1	Hi Analog ON Channel 0	Always True	Set Output Pin Channel 7 Digital HIGH
2	Hi Analog OFF Channel 0	Always True	Set Output Pin Channel 7 Digital LOW
3	Power-Up/Reset	Always True	Set Output Pin Channel 8 Digital LOW
4	Power-Up/Reset	Always True	Set Output Pin Channel 7 Digital HIGH
5	Idle Loop	Always True	Toggle Output Pin Channel 8

Εικόνα 3.12: event editor

Στον πίνακα γεγονότων, καθορίζουμε τα γεγονότα και τις αντίστοιχες ενέργειες που λαμβάνονται εάν τα γεγονότα πραγματοποιηθούν. Πατάμε το κουμπί Add. Θα εμφανιστεί το ακόλουθο παράθυρο διαλόγου και θα προσθέσουμε τα νέα γεγονότα στον πίνακα:



The dialog box 'Event Data' is divided into two main sections: 'Event' and 'Action'.
In the 'Event' section, the 'Name' dropdown is set to 'Idle Loop'.
In the 'Condition' section, the 'Name' dropdown is set to 'Always True'.
In the 'Action' section, the 'Name' dropdown is set to 'Set Output Pin', and the 'Channel' dropdown is set to 'Channel 1'.
Below the 'Action' section, there are two radio buttons: 'Set Pin On' (which is unselected) and 'Set Pin Off' (which is selected).
At the bottom of the dialog, there are 'OK' and 'Cancel' buttons.

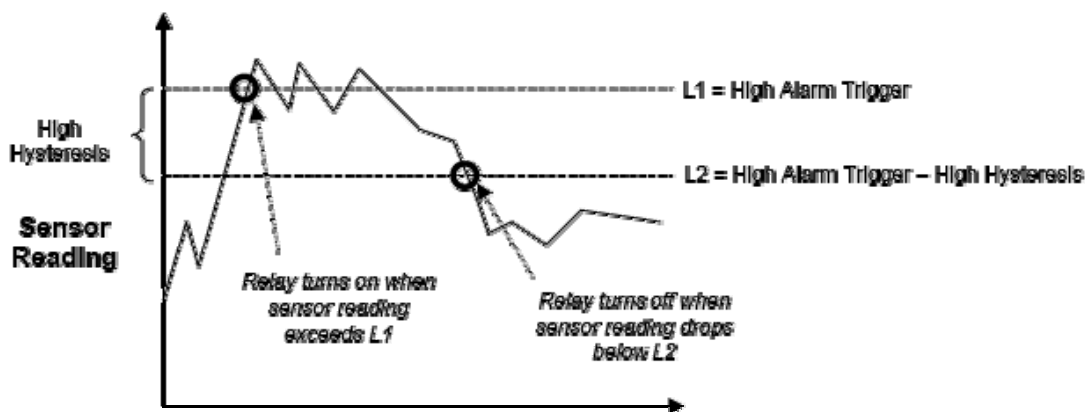
Εικόνα 3.13: παράθυρο διαλόγου για την δημιουργία γεγονότος (event)

Υπάρχει τέσσερις έγκυροι τύποι συναγεμμοί/γεγονότα στο IPμ8930:

- High Threshold On/Off (analog or digital)
- Low Threshold On/Off (analog or digital)
- Power Up
- Idle Loop

Οι υψηλοί και χαμηλοί συναγεμμοί ορίων κατωφλίου είναι συγχρονισμένοι με τα ανώτερα και χαμηλότερα όρια κατωφλίου που καθορίζονται στο παράθυρο

διαμόρφωσης αισθητήρων (προηγούμενη παράγραφος για περισσότερες πληροφορίες).



Εικόνα 3.14: παρουσίαση πιθανής υστέρησης

Με τη χρήση μιας ανώτερης ή χαμηλότερης υστέρησης, μπορούμε να αποφύγουμε έναν συναγερμό πολλές φορές σε μια μικρή χρονική περίοδο εάν τα δεδομένα από τον αισθητήρα "τρέμουν" γύρω από το επίπεδο ορίων κατωφλίου συναγερμών. Η παραπάνω εικόνα επεξηγεί τη σχέση μεταξύ μιας τιμής κατωφλίου και της υστέρησης του.

Αφού καθορίσουμε έναν τύπο συναγερμών, πρέπει έπειτα να καθορίσουμε την προκύπτουσα δράση όταν ο συναγερμός/γεγονός προκαλείται. Υπάρχουν τρεις έγκυρες ενέργειες:

- Fire Output Pin (pin must be configured as a digital)
- Toggle Output Pin (pin must be configured as a digital)
- Fire Another Event

Οι ενέργειες Fire και Toggle Output απαιτούν έναν αριθμό καναλιών. Ομοίως, το Fire Another Event απαιτεί έναν αριθμό συναγερμών ως παράμετρο.

Σημείωση, στην τρέχουσα έκδοση firmware, η συνθήκη γεγονότος είναι "πάντα αληθινή".

3.5 Κατασκευή αισθητήριων

Τα αισθητήρια τα οποία κατασκευάστηκαν είναι ένας φωτοδιακόπτης και ένας αισθητήρας κίνησης (πομπός - δέκτης). Τα υλικά που χρησιμοποιήθηκαν είναι τα παρακάτω:

Υλικά κατασκευής φωτοδιακόπτη:

- 1 trimmer 1k
- 1 led
- 1 HCF 4093
- 1 LDR
- 1 C=10nF

Υλικά κατασκευής αισθητήρα κίνησης: Πομπός

- 1 HCF 4093
- 2 C=10nF
- 1 trimmer 10k

- 1 led IR

Δέκτης

- 1 R=1,2k
- 1 R=470Ω
- 1 R=680Ω
- 1 led
- 1 TSOP 1738
- 1 C=10nF

3.6 Υλοποίηση ενδεικτικών εφαρμογών:

Τα βήματα που ακολούθησαν για την επίτευξη της εργασίας είναι τα παρακάτω:

Δημιουργήσαμε:

- 1) τα java applets (βλ. παράγραφος 3.2.1 και 3.2.2)
- 2) τις ιστοσελίδες (βλ. παράγραφος 3.2.1)
- 3) "ανεβάσαμε" τα παραπάνω στο IPμ8930 (βλ. παράγραφος 3.1)
- 4) τα κατάλληλα webholes (βλ. παράγραφος 3.1)

Συγκεκριμένα οι τιμές τις οποίες δώσαμε είναι οι παρακάτω:

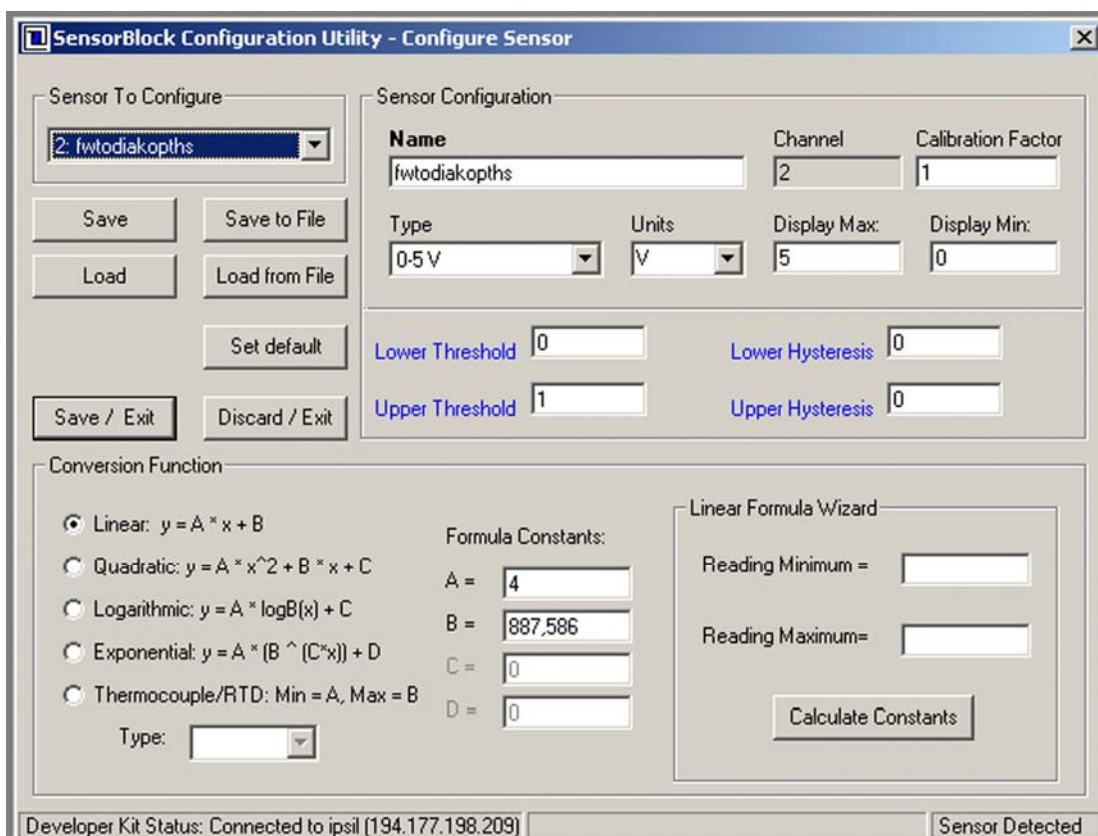
Για την σελίδα εμφάνισης των τιμών των καναλιών

Format	Length	Destination	Type	Source	EndTag
4	5	2161	1	280	0
4	5	2275	1	282	0
4	5	2439	1	284	0
4	5	2595	1	286	0
4	5	2751	1	288	0
4	5	2907	1	290	0
4	5	3057	1	292	0
4	5	3201	1	294	0
4	5	3340	1	296	255

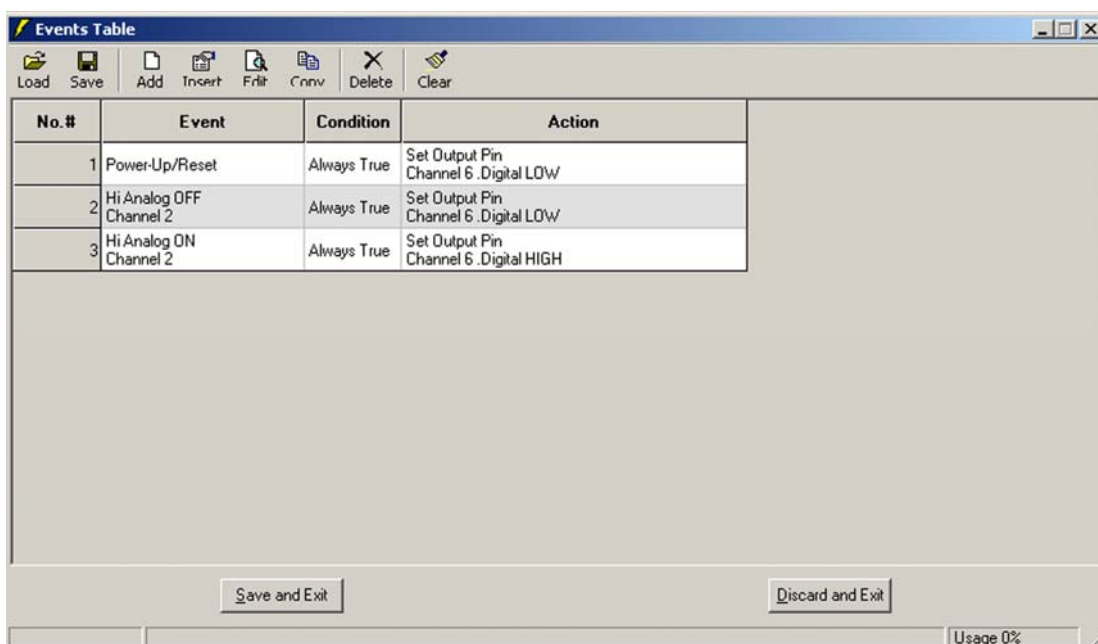
Για την σελίδα χειροκίνητης ενεργοποίησης/απενεργοποίησης καναλιού 6

Format	Length	Destination	Type	Source	EndTag
4	5	662	1	292	0
4	5	704	1	294	0
4	5	746	1	296	255

- 5) διαμορφώσαμε κατάλληλα τα αισθητήρια (βλ. παράγραφος 3.3.3)



6) διαμορφώσαμε κατάλληλα τους συναγερμούς (βλ. παράγραφος 3.3.4) και



7) συνδέσαμε τα αισθητήρια και μια λάμπα στα κανάλια εισόδου του IPμ8930.

Συνδέσαμε τη λάμπα στο κανάλι 6, ώστε να μπορούμε να την ενεργοποιούμε /απενεργοποιούμε χειροκίνητα, και στο κανάλι 1 και 2 το αισθητήρια κίνησης και το φωτοδιακόπητη αντίστοιχα.

Αναφορές

- 1) παρουσίαση της γλώσσας Java, από τον Κακαρόντζα Γιώργο.
<http://www.cs.teilar.gr/gkakaron/java/Index.html>
- 2) What You should know about Modbus
<http://www.focus-sw.com/fieldtalk/libmbusmaster/java/doc/modbus.html> - [mbustcpprotocol](#)
- 3) W3C Recommendation, Steven Pemberton et al., 26 January 2000
[XHTML 1.0: The Extensible HyperText Markup Language, A Reformulation of HTML 4 in XML 1.0](#)
- 4) **(Saltzer 1984)** J.H. Saltzer, D.P. Reed and D.D. Clark, End-to-End Arguments in Systems
- 5) Design, ACM Transactions in Computer Systems 2, 4, November, 1984, pages 277-288.
<http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.rtf>
- 6) **(Borriello 2000)** Gaetano Borriello, Roy Want, Embedded Computation Meets The World Wide Web, Communications of the ACM, May 2000, Vol 43, No 5.
- 7) **(Shrikumar 2000)** Shrikumar H, Huan Li, Krithi Ramamritham, Composable Working Memory,
- 8) An Adaptive Real Time Publisher Subscriber Mechanism, University of Massachusetts, Dec 2000.
- 9) **(Ipsil 2000)** Ipsil Inc, IPm Product Brief, December 2000

Παράρτημα Α

- 3. Architectural Overview
- 4. Hardware Overview
- 5. RAM/Internal EEPROM Reference
- 6. External EEPROM
- 7. Peripherals
Schematics