

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτοματοποίηση Σπιτιού Μέσω Κινητού Τηλεφώνου και Ethernet

Διπλωματική Εργασία Ιωάννη Αγαδάκου



Εξεταστική επιτροπή :

Επίκουρος Καθ Ι. Παπαευσταθίου (Επιβλέπων)

Καθ. Απ. Δόλλας

Καθ. Κ. Καλαϊντζάκης

Ευρετήριο**Περιεχόμενα**

Ευρετήριο	3
Πρόλογος	5
GSM Controlled Devices	6
IO Bridge	6
Industrial Ethernet SensorProbe-SecurityProbe Product	6
Abstract	8
Βασικές Αρχές Σχεδίασης	8
Σχεδιασμός	9
Συνοψίζοντας	10
Συσκευές και Υλικό Προτύπων	11
Εργαλεία Σχεδίασης	12
<i>Εργαλεία Ανάπτυξης:</i>	12
Λειτουργικό Σύστημα NutOS	14
Αρχιτεκτονική	15
Εστία Επισκόπηση Λειτουργίας	17
Network Initialization-Overview	17
Αρχικοποίηση συσκευής Master Device	17
Device Hub Initialization	17
Φάση Λειτουργίας Κεντρικής Μονάδας	18
Φάση Λειτουργίας Συνδεδεμένων Device Hubs	18
Έλεγχος Εσωτερικών Συσκευών Device Hub	18
Φάση Αναφοράς Συσκευών «Device Reporting»	18
Πρωτόκολλο Επικοινωνίας Συσκευών	19
Simple Device Messaging Protocol (SDMP)	20
<i>Προεσκόπηση Πρωτοκόλλου</i>	21
Βασικές Εντολές	23
Σύνταξη Βασικών Εντολών	23
Σύνθετες Εντολές	24
ΔΙΑΘΕΣΙΜΕΣ ΕΝΤΟΛΕΣ	25
<i>Περιγραφή Λειτουργίας Διαθέσιμων Εντολών</i>	26
Εστία-Υδρα Πρωτότυπες Συσκευές	32

ΕΣΤΙΑ

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet Εστία -GSM Interface A Master Device Implementation	33
Περιορισμοί Κεντρικής Συσκευής «Εστία»	34
«Ύδρα» A Simple Device Hub Implementation	36
Προεπιλεγμένες Ρυθμίσεις «Default Settings»	38
Σύνταξη Εντολών SDMP για τον έλεγχο της SDH «Ύδρα»	39
Προβλήματα Κατά την Υλοποίηση	41
Επίλογος – Συμπεράσματα	43
Appendix A. Ethernut 1.3h Schematics-Specs	45
Βιβλιογραφία	48
Ευρετήριο	49

Πρόλογος

Από τις αρχές του 1980 ορισμένοι πρωτοπόροι είχαν οραματιστεί ένα κόσμο, στον οποίο το σύνολο της πληροφορίας θα ήταν διαθέσιμο σε κάθε ενδιαφερόμενο. Όχι μόνο από ένα απομονωμένο τερματικό-υπολογιστή, αλλά προσβάσιμο ανεξαρτήτως τοποθεσίας και απόστασης από την πηγή της πληροφορίας. Επίσης, είχαν διαπιστώσει πως αυτή είναι η πραγματική δύναμη και μαγεία του ψηφιακού κόσμου, η δυνατότητα δηλαδή με την χρήση της τεχνολογίας, η πληροφορία που παράγεται σε ένα σημείο να μπορεί να μετατρέπεται και να μεταφέρεται σε ψηφιακή μορφή πέρα από τα φυσικά όρια της, να μπορεί να αναπαράγεται από τα κατάλληλα μέσα όταν χρειάζεται από κάθε ενδιαφερόμενο.

Σήμερα, 30 χρόνια αργότερα αυτό έχει γίνει πραγματικότητα. Το διαδίκτυο και οι τηλεπικοινωνίες έχουν φτάσει σε ένα επίπεδο που τότε αντιστοιχούσε σε ταινίες επιστημονικής φαντασίας. Χρήστες συναλλάσσονται μεταξύ τους με πολλές διαφορετικές μορφές ψηφιακών μέσων, και μεγάλο μέρος της σημερινής εργασίας γίνεται επίσης μέσω αυτών.

Τραπεζικοί λογαριασμοί, εργασίες και φορολογικές δηλώσεις, καθώς και τα νέα από προϊόντα και υπηρεσίες που μας αφορούν, είναι λίγα από αυτά που πλέον ένας χρήστης μπορεί να διαχειριστεί μέσω ψηφιακών μέσων χωρίς να απαιτείται η φυσική του παρουσία. Αυτό έχει αλλάξει την ζωή μας δραστικά προς το καλύτερο και είναι πλέον προφανές και αποδεκτό από όλη την ανθρωπότητα, ότι με την πρόοδο της τεχνολογίας ολοένα και περισσότερες από τις καθημερινές μας ασχολίες, γίνονται προσβάσιμες και διαχειρίσιμες, από απόσταση, είτε με την χρήση υπολογιστών είτε άλλων συσκευών.

Παραλή την πρόοδο της τεχνολογίας, ένα σημαντικό κομμάτι της καθημερινότητας μας δεν έχει εξελιχθεί, τουλάχιστον όχι με την μαζικότητα των προηγούμενων στο σημείο που περιγράψαμε. Να μπορεί δηλαδή ένας χρήστης να ενημερωθεί για την κατάσταση του και να το διαχειριστεί από απόσταση. Αυτό, είναι το σπίτι μας.

Ποιον δεν θα ενδιέφερε να μπορεί να δει το σπίτι του από απόσταση, να ενημερωθεί για τις φυσικές συνθήκες που επικρατούν, να επηρεάσει την κατάσταση συσκευών μέσα σε αυτό, με τον ίδιο τρόπο που μπορεί και ελέγχει και μεταβάλλει την κατάσταση των τραπεζικών του λογαριασμών; Αν και η τεχνολογία έχει φτάσει σε σημείο που ηλεκτρονικά και συσκευές κάποτε ήταν απρόσιτα και περιορισμένων δυνατοτήτων, πλέον είναι φθηνά και ταυτόχρονα αξιόπιστα με ποικίλες δυνατότητες. Παρόλα αυτά δεν έχει εμφανιστεί μια πρόταση αντάξια της σημερινής τεχνολογίας.

Διάφορες εταιρίες έχουν κατά καιρούς αναπτύξει προϊόντα για αυτοματισμούς σπιτιών αλλά καμία δεν έχει καταφέρει να ξεπεράσει ταυτόχρονα 3 βασικά προβλήματα: Κόστος, Προσβασιμότητα και Ευελιξία. Πριν συνεχίσουμε όμως ας δούμε μερικές τεχνολογίες του χώρου και τους περιορισμούς που αυτά αποτυγχάνουν να ξεπεράσουν.

GSM Controlled Devices

Οι εφαρμογές του χώρου αποτελούνται από ένα gsm module συνδεδεμένο με κάποιο relay ή κάποιο αισθητήριο όπως θερμομέτρα, υγρόμετρα και παρόμοια αισθητήρια, η λειτουργικότητα αυτών των τεχνολογιών είναι πολύ απλή και ο χρήστης είτε ελέγχει το relay με αναπάντητη κλήση είτε ενημερώνεται για τις τιμές του αισθητηρίου με sms. Το μειονέκτημα αυτής της προσέγγισης είναι ότι χρειάζεται ένα gsm modem για κάθε συσκευή, το οποίο πέρα από το κόστος, κάθε gsm κοστίζει πάνω από 150 ευρώ, είναι και πολύ δύσχερο καθώς ακόμα και αν περάσουμε το εμπόδιο του κόστους πρέπει ο χρήστης να απευθύνεται σε διαφορετικό αριθμό για κάθε συσκευή που θέλει να ελέγχει. Επίσης αυτές οι τεχνολογίες δεν υποστηρίζουν επικοινωνία μέσω Ethernet-IP.

IO Bridge

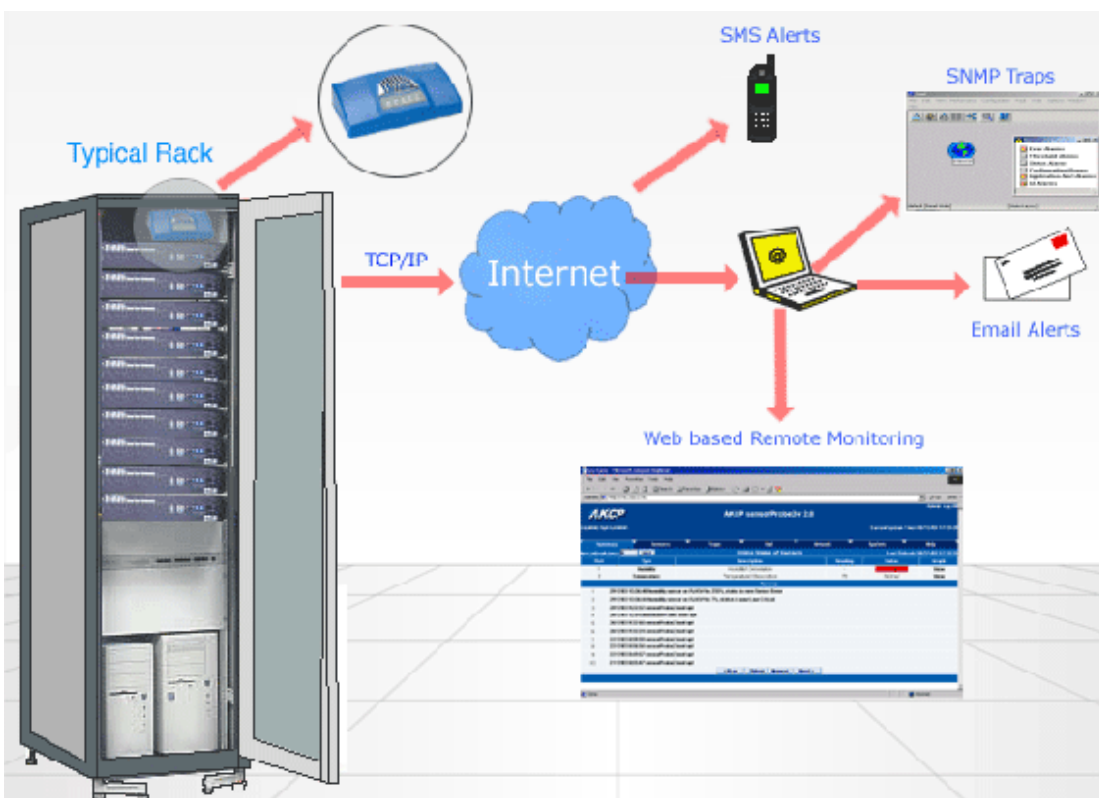
Η συγκεκριμένη εφαρμογή αποτελείται από μια πλατφόρμα βασισμένη σε ένα System ON a Chip παρόμοιο με το SoC Ethernet που χρησιμοποιήθηκε και σε αυτή την εργασία (Κεφ. Εστία –Υδρα ,appendix α). Πρόκειται για τεχνολογία που σκοπός της είναι η μετατροπή απλών αισθητηρίων και διακοπών(switches) σε ισοδύναμες Ethernet συσκευές, ο έλεγχος και η παρακολούθηση των οποίων γίνεται μέσω ενός web application της εταιρίας. Η συγκεκριμένη πρόταση είναι σχετικά φθηνή και προσιτή, με 4 ισοδύναμες Ethernet devices να κοστίζουν περί τα 150\$, μειονεκτήματα της είναι ότι δεν δίνει χειρισμό μέσω GSM και ότι το web application στο οποίο συνδέονται οι συσκευές είναι σε χώρο της εταιρίας, μη δίνοντας τη δυνατότητα στους χρήστες να έχουν προσωπικό χώρο. Επίσης το πρωτόκολλο δεν είναι ανοικτό, οπότε η επιλογή περιορίζεται σε προϊόντα της εταιρίας.

Industrial Ethernet SensorProbe-SecurityProbe Product

Πρόκειται για μια πρόταση με μεγάλες δυνατότητες αλλά και μεγάλο κόστος. Οι δυνατότητες των παραπάνω τεχνολογιών περιλαμβάνουν μεταξύ άλλων τον χειρισμό συσκευών μέσω Ethernet και συγκεκριμένα μέσα από ένα web interface το οποίο τρέχει στη συσκευή ελέγχου (cgi). Διαθέσιμες είναι και επεκτάσεις που δίνουν την δυνατότητα

χειρισμού μέσω GSM αυξάνοντας ωστόσο αρκετά το κόστος.

Συνοπτικά, σε αντίθεση με τα άλλα προϊόντα του χώρου, η συγκεκριμένη προσέγγιση χρησιμοποιεί ισχυρά SoCs, στα οποία τρέχει μια έκδοση του λειτουργικού linux, ενώ έχουν και ενσωματωμένο Apache Web



Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet Server, στον οποίο τρέχει και το web interface. Αυτό το δυνατό hardware δίνει τη δυνατότητα σε αυτό το προϊόν να παρέχει μεγάλο εύρος δυνατοτήτων, όπως mail alerts, sms notifications και περιγραφικά data charts για κάθε συνδεδεμένο αισθητήρα ή συσκευή. Το κόστος αυτής της προσέγγισης είναι αρκετά μεγάλο. Ενδεικτικά, για ένα δίκτυο αποτελούμενο από 4 αισθητήρια χωρίς υποστήριξη gsm το κόστος ανέρχεται στα 1200\$.

Οι διαθέσιμες προτάσεις σήμερα σε καμία περίπτωση δεν εξαντλούνται στην παραπάνω λίστα. Όλες όμως έχουν κοινά ελαττώματα, είναι ακριβές και έχουν φτιαχτεί στην πλειονότητα τους να δουλεύουν μόνο με προϊόντα του ίδιου κατασκευαστή.

Στην παρούσα εργασία αναπτύχθηκε μια ολοκληρωμένη τεχνολογία με σκοπό να αντιμετωπίσει τα προβλήματα που παρουσιάστηκαν προηγουμένως. Έγινε μια νέα προσέγγιση, πιο ευέλικτη και με χαμηλό κόστος.

Στο πλαίσιο της εργασίας αυτής αναπτύχθηκε και ένα πρωτόκολλο για να εξυπηρετήσει τις ανάγκες επικοινωνίας μεταξύ των συσκευών, το οποίο είναι δομημένο με τέτοιο τρόπο ώστε να είναι απλό και ταυτόχρονα αρκετά γενικό ώστε να μπορούν εύκολα να δημιουργηθούν συσκευές συμβατές με αυτήν την τεχνολογία.

Στα επόμενα κεφάλαια θα παρουσιασθεί λεπτομερώς σε όλα της τα στάδια αυτή η τεχνολογία, η οποία οδήγησε στη δημιουργία ενός προϊόντος χαμηλού κόστους χωρίς ωστόσο συμβιβασμό στις δυνατότητες. Συγκεκριμένα, κατασκευάστηκε ένα σύστημα αυτοματισμού σπιτιών, μέσω gsm και ethernet, με συνολικό κόστος για τα 14 αισθητήρια και το gsm modem μικρότερο των 300 Ευρώ. Ταυτόχρονα οι επιμέρους συσκευές του δικτύου, συγκεκριμένα η «Υδρα» μπορεί να χρησιμοποιηθεί και αυτόνομα για τη συλλογή δεδομένων από αισθητήρια και να ελέγξει μέχρι και 4 Relays με κόστος χαμηλότερο των 150\$ διοχετεύοντας τα συλεγόμενα δεδομένα σε άλλες εφαρμογές μέσω ethernet.

Abstract

Ο σχεδιασμός και η υλοποίηση ενός ολοκληρωμένου προϊόντος αυτοματισμού και ελέγχου σπιτιού μέσω IP και GSM(SMS), χρησιμοποιώντας ενσωματωμένες συσκευές πραγματικού χρόνου και χαμηλού κόστους, με δυνατότητα δικτύωσης των επιμέρους συσκευών και χειρισμό αυτών μέσω απλών SMS και IP.

Βασικές Αρχές Σχεδίασης

- Ενσωματωμένη Συσκευή
- Έλεγχος μέσω GSM/sms ->GSM module
- Υποστήριξη συσκευών over Ip
- Χαμηλό κόστος
- Υποστήριξη αρκετών συσκευών
- Αυτοματισμός - Έλεγχος σπιτιού
- Πραγματικού Χρόνου απαιτήσεις
- Επικοινωνία μεταξύ των συσκευών

Σχεδιασμός

Πριν προχωρήσουμε στο σχεδιασμό, πρέπει να ορίσουμε με πιο αυστηρό τρόπο τι εννοούμε με την έννοια «απαίτηση πραγματικού χρόνου», καθώς ανάλογα με την αυστηρότητα που θα τον ορίσουμε θα επηρεάσει (και έτσι είναι σωστό) το σύνολο της σχεδίασης. Εφόσον πρόκειται για μια συσκευή που σκοπός της είναι ουσιαστικά να μεσολαβεί μέσω άλλων (IP) συσκευών και του χρήστη, και η επικοινωνία αυτή να γίνεται μέσω Συντόμων Μηνυμάτων Κειμένου, τότε εκ των πραγμάτων αναφερόμαστε σε μια ασύγχρονη επικοινωνία με τον χρήστη. Αυτό συμβαίνει επειδή από τη στιγμή που μια συσκευή θα θελήσει να έρθει σε επικοινωνία με τον χρήστη, μια σειρά από απροσδιορίστου χρόνου βημάτων θα πρέπει να πραγματοποιηθεί. Να ξεκινήσει επικοινωνία με την Εστία (TCP/IP), να ακολουθήσει η μεταγλώττιση και επεξεργασία της εντολής από την εστία και τέλος να γίνει η μεταβίβαση του μηνύματος στο gsm module, μέσω του οποίου σε κάποιο χρόνο θα φτάσει στο χρήστη). Μέσα από μια απλή περιγραφή των παραπάνω βημάτων που υποχρεωτικά πρέπει να ακολουθηθούν, μπορούμε να συμπεράνουμε πως πρέπει το σύστημα μας να είναι «Μαλακών» χρονικών περιορισμών, καθώς μιλάμε για βήματα που έχουν απροσδιόριστο χρονικό κόστος και εξαρτώνται από έναν αριθμό μη προβλέψιμων παραγόντων.

Συνεχίζοντας στο επόμενο βήμα ας εξετάσουμε το θέμα της ενδιάμεσης ενσωματωμένης συσκευής και των χαρακτηριστικών που αυτή πρέπει να έχει για να υποστηρίζει τις απαιτήσεις μας. Θα πρέπει να έχει μια rs232 (σειριακή θύρα) για επικοινωνία με το GSM module και θα πρέπει να υποστηρίζει TCP/IP stack. Το TCP/IP stack πρέπει να σημειωθεί πως είναι αρκετά «ακριβό» για ενσωματωμένες συσκευές γιατί έχει μεγάλες απαιτήσεις σε επίπεδο Hardware και συγκεκριμένα σε θέμα μνήμης αλλά και ταχύτητας. Επίσης καθώς θα χρειαστεί να ελέγξουμε συσκευές μέσω IP και να υλοποιήσουμε ένα πρωτόκολλο επικοινωνίας μεταξύ συσκευών, θα είναι σωστό να χρησιμοποιήσουμε ένα λειτουργικό πραγματικού χρόνου, έτσι ώστε να έχουμε το πλεονέκτημα ,αρκετά χρονοβόρα και επιρρεπή στα λάθη κομμάτια, να τα αναλάβει το λειτουργικό και να κρύψει από εμάς τις περιττές λεπτομερές.

Επίσης θα πρέπει αυτή η συσκευή να είναι χαμηλού κόστους καθώς και να μπορεί να υποστηρίξει, εάν αυτό χρειαστεί, και αρκετές συσκευές TTL,I2C και SPI, έτσι ώστε χρησιμοποιώντας την τεχνογνωσία του RTOS να μπορούμε εύκολα να φτιάξουμε αποδοτικά και φθηνά SDH¹(Simple Device Hubs) για έλεγχο και αυτοματισμό Σπιτιών (Switches, thermometers, Relays, TriggerSensors, Hydrometers).

Μια συσκευή που ικανοποιεί όλες τις παραπάνω απαιτήσεις είναι η Ethernet ενσωματωμένη συσκευή, η οποία έχει και το εξής πλεονέκτημα: ένα λειτουργικό πραγματικού χρόνου το RTOS, που υποστηρίζεται και αναπτύσσεται από την ίδια εταιρία κατασκευής της συσκευής αυτής.

Συνοψίζοντας

Ο χρήστης θα μπορεί να επικοινωνεί μέσω απλού κινητού τηλεφώνου με τη συσκευή και να ελέγχει μέσω αυτής και άλλες συσκευές, οι οποίες με την σειρά τους θα ενώνονται με IP(μέσω router) με την κύρια συσκευή. Η κύρια συσκευή θα ενώνεται μέσω σειριακής θύρας με το GSM module και μέσω απλού καλωδίου UTP με το Ethernet router (Εικόνα 1).

Οι επιμέρους συσκευές μπορεί να αποτελούνται από παραπάνω από μια απλές συσκευές όπως ttl switches ή ttl controlled Relays. Αυτές οι συσκευές συνδέονται πάνω σε μια SDH συσκευή με αποτέλεσμα κάθε μία από αυτές να μεταμορφώνεται σε μια πλήρως συμβατή και ελεγχόμενη Ethernet device. Με αυτόν τον τρόπο έχουμε ένα πολύ οικονομικό και αποδοτικό μοντέλο. Μια τέτοια συσκευή υλοποιήθηκε ήδη και θα παρουσιαστεί στο δεύτερο μέρος αυτού του έγγραφου.

¹**Simple Device Hub** =Πρόκειται για μια συσκευή που πάνω της συνδέονται απλές ηλεκτρονικές συσκευές όπως switches, relays, thermometers(i2c-spi) και μετά χρησιμοποιώντας το Πρωτόκολλο επικοινωνίας η SDH μετατρέπει αυτές τις απλές συσκευές σε ισοδύναμες συσκευές IP. Μπορεί δηλαδή μετά κάποιος να ελέγξει το ttl switch με ip commands σαν να ήταν ip συσκευή.

Συσκευές και Υλικό Προτύπων

Για την υλοποίηση αυτής της εργασίας χρησιμοποιήθηκαν διάφορα εργαλεία ,τόσο σε επίπεδο λογισμικού , όσο και σε επίπεδο υλικού.

Οι συσκευές που χρησιμοποιήθηκαν είναι : ethernet 1.3-1.3h ως πρότυπες συσκευές (Κεντρική και Συσκευή εργάτης) . Η πλακέτα Ethernet , είναι ένα ολοκληρωμένο σύστημα ουσιαστικά ,βασισμένο στον atmega128 με ενσωματωμένη υποστήριξη ethernet ,χρησιμοποιώντας τον μικροελεγκτή της Realtek(Rtl8019as) και την εν γένει υποστήριξη ενός λειτουργικού συστήματος πραγματικού χρόνου (NutOS βλ επόμενο κεφάλαιο) .Η συγκεκριμένη πλακέτα επίσης έχει μια θύρα rs232 την οποία και θα χρησιμοποιήσουμε για να ελέγξουμε το GSM module.Τέλος οι ακροδέκτες του atmega είναι διαθέσιμοι και μπορούν να χρησιμοποιηθούν για σύνδεση των απλών αισθητήρων-συσκευών.

*PIR-Sensor:*Πρόκειται για ένα θερμοηλεκτρικό αισθητήρα , που χρησιμοποιήθηκε ως trigger sensor,όταν δηλαδή εντοπίζεται αρκετά μεγάλη θερμική ακτινοβολία συγκεκριμένου φάσματος,ο ακροδέκτης εξόδου αυτού του αισθητήρα παράγει ένα λογικό "1" .Η μέγιστη συχνότητα ειδοποίησης είναι ρυθμιζόμενη από 5 δευτερόλεπτα έως και 18 λεπτά.

*DS1621:*Ένας αισθητήρας θερμότητας και ένας θερμοστάτης ,στο ίδιο ολοκληρωμένο κύκλωμα σε μορφή dip 8.Η επικοινωνία με αυτή την συσκευή γίνεται μέσω i2c και ο θερμοστάτης παράγει ειδοποίηση ανάλογη trigger sensor.

Optical Isolated Relay Board: Για να καλυφθούν οι ανάγκες ελέγχου συσκευών on/off,χρησιμοποιήθηκε αυτή η πλακέτα που φέρει 2 Relay ελεγχόμενα μέσω TTL από τους ακροδέκτες του ethernet.Τα συγκεκριμένα Relay "αντέχουν" στις επαφές τους μέχρι 250 VAC (μέγιστο 5A).

Telit 862 Quad Python: Το GSM module που χρησιμοποιήθηκε.Το συγκεκριμένο GSM υποστηρίζει και την γλώσσα python ,μια δυνατότητα που δεν χρησιμοποιήθηκε στην παρούσα σχεδίαση.Ο έλεγχος του GSM γίνεται μέσω Attention Commands (AT COMMANDS),παρόμοιες με αυτές που χρησιμοποιούνται για τον έλεγχο modem.

Adsl Router: Για να καλυφθούν οι δικτυακές ανάγκες της τεχνολογίας που αναπτύχθηκε χρειάζεται ένα DHCP capable device, για να ελαχιστοποιηθεί το κόστος χρησιμοποιήθηκε ένα κοινό adsl router του εμπορίου οποιοδήποτε DHCP ROYTER μπορεί να καλύψει τις απαιτήσεις μας.

Nut/OS : Το λειτουργικό σύστημα που "τρέχει" στις συσκευές,τόσο στην Κεντρική συσκευή όσο και στην συσκευή εργάτη.

Εργαλεία Σχεδίασης

Ένα βασικό πρόβλημα που αντιμετωπίστηκε κατά την πορεία της εργασίας ήταν η έλλειψη συγκεκριμένων εργαλείων (toolchain) για την ανάπτυξη εφαρμογών με την χρήση του NutOS. Ως εκ τούτου τα εργαλεία που χρησιμοποιήθηκαν είναι γενικής φύσεως, δηλαδή χρησιμοποιούνται γενικά για την στον προγραμματισμό σε C των μικροελεγκτών της avr(και γενικά για προγραμματισμό μικροελεγκτών σε C).

Τα εργαλεία που χρησιμοποιήθηκαν χωρίζονται σε 2 κατηγορίες, στα εργαλεία ανάπτυξης του λογισμικού και στα εργαλεία ελέγχου και αποσφαλμάτωσης, αυτό δυστυχώς ήταν και ένα μεγάλο πρόβλημα καθώς τα προβλήματα του προγράμματος δεν μπορούσαν να εντοπισθούν μέσα από τα εργαλεία ανάπτυξης.

Εργαλεία Ανάπτυξης:

Compiler-Linker : Avr-gcc 4.3.2, το εργαλείο αυτό μεταγλωττίζει το πρόγραμμα που γράφεται σε C, σε γλώσσα μηχανής για τον μικροελεγκτή atmega128, επίσης επειδή ο κώδικας γράφεται σε αρχιτεκτονική 8086 μέσω της διαδικασίας cross compile, παράγεται "εκτελέσιμο" για αρχιτεκτονική RISC του atmega128. Να σημειωθεί επίσης ότι η συγκεκριμένη έκδοση δεν είναι απαραίτητη, το λογισμικό έχει παραχθεί επιτυχώς και με παλαιότερες εκδόσεις. Γλώσσα C :Το λογισμικό γράφτηκε σε απλή C Ansi99, και σε ορισμένα critical paths χρησιμοποιήθηκε και inline assembly.

NetBeans IDE 6.5 : Το περιβάλλον ανάπτυξης στο οποίο πραγματοποιήθηκε η ανάπτυξη, προσφέρει αρκετές ευκολίες στην εργασία και δυνατότητα χρήσης του προγράμματος make για την διαδικασία μεταγλώττισης και σύνδεσης(compile-linking), πράγμα που λόγω της χρήσης του NutOS, είναι απαραίτητη προϋπόθεση. Εναλλακτικά, θα μπορούσε να χρησιμοποιηθεί και το περιβάλλον Eclipse, αλλά, επιλέχθηκε το παρόν διότι προτείνεται από την κοινότητα και πολλά από τα τεχνικά κείμενα παρέχουν οδηγίες για αυτό.

Atmel avr studio 3.5 :Το εργαλείο αυτό χρησιμοποιήθηκε μόνο για τον προγραμματισμό της πλακέτας Ethernut με το παραγόμενο από την διαδικασία cross-compile πρόγραμμα. Λόγω του εξωτερικού υλικού που έχει συνδεθεί στην πλακέτα (εξωτερική φυσική μνήμη κλπ) δεν ήταν δυνατή η χρήση λειτουργιών όπως simulation, emulation του λογισμικού καθώς ο προσομοιωτής δεν μπορούσε να αναγνωρίσει το ξένο υλικό ούτε διέθετε τις ανάλογες βιβλιοθήκες. Να σημειωθεί επίσης ότι μεταγενέστερες εκδόσεις παρουσίαζαν δυσκολία στον προγραμματισμό της πλακέτας Ethernut, και χρειάζονται περισσότερες ρυθμίσεις για τον επιτυχή προγραμματισμό της με αυτές, προτείνεται για αυτό το λόγο η χρήση αυτής της έκδοσης(3.5).

Avr-gdb client-server: Σε πρώιμο στάδιο της ανάπτυξης, έγινε χρήση αυτών των εργαλείων για να καλυφθεί το κενό στην αποσφαλμάτωση και προσομοίωση, τα αποτελέσματα ωστόσο δεν ήταν τα αναμενόμενα και δεν χρησιμοποιήθηκαν περαιτέρω. Βασικά μειονεκτήματα της συγκεκριμένης μεθόδου ήταν η ασταθής συμπεριφορά, η δυσκολία στην χρήση και η δέσμευση πόρων από την περιορισμένων δυνατοτήτων πλακέτα(Ethernut).

Εργαλεία Αποσφαλμάτωσης-Ελέγχου

Hercules : Πρόκειται για μία εφαρμογή της εταιρείας HW group, που δίνεται δωρεάν για προσωπική χρήση. Η εφαρμογή αυτή περιλαμβάνει έναν ενσωματωμένο TCP server και client, που χρησιμοποιήθηκε κατά κόρον για να δοκιμαστεί η ορθότητα του πρωτοκόλλου επικοινωνίας και η δικτυακή συμπεριφορά των πρότυπων συσκευών. Επίσης παρέχει και ένα τερματικό παρακολούθησης rs232 με πολύ μεγαλύτερες δυνατότητες του HyperTerminal, όπως καταγραφή σε αρχείο και απεικόνιση σε δεκαεξαδικό σύστημα,

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet η συγκεκριμένη λειτουργία βοήθησε πολύ στην μελέτη της επικοινωνίας της κεντρικής συσκευής με το GSM modem. Να σημειωθεί επίσης ότι για το μεγαλύτερο μέρος της ανάπτυξης του λογισμικού η αποσφαλμάτωση γινόταν μέσω εκτυπώσεων (printf) στην σειριακή θύρα και για τις 2 πρότυπες συσκευές.

Tera Term: Ένα ακόμα εργαλείο που χρησιμοποιήθηκε για την παρακολούθηση της σειριακής θύρας(rs232).

Μια δυνατότητα που δεν χρησιμοποιήθηκε ,καθώς δεν ήταν διαθέσιμη κατά την διάρκεια ανάπτυξης, είναι η χρήση ενός ειδικού προγραμματιστή του JTAG-DEBUG ,ο οποίος σύμφωνα με την κοινότητα μπορεί να χρησιμοποιηθεί παράλληλα με τον simulator του avrstudio 4 για να παρέχει ένα σύμπλοκο εργαλείο αποσφαλμάτωσης και προσομοίωσης .

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet Λειτουργικό Σύστημα NutOS

Το λογισμικό που αναπτύχθηκε και στις δύο συσκευές ,βασίστηκε στο ενσωματωμένο λειτουργικό σύστημα πραγματικού χρόνου NutOs .

Οι λόγοι που κατέστησαν απαραίτητη την υιοθέτηση ενός λειτουργικού συστήματος είναι αρκετοί, οι σημαντικότεροι εκ των οποίων ήταν η χρήση του ethernet και η ανάγκη για multitasking.Το λειτουργικό σύστημα NutOs παρέχει ένα δυνατό application interface που καθιστά δυνατή την χρήση ethernet και αναλαμβάνει να διαχειριστεί την πολυ - πλοκότητα διαχείρισης πολλαπλών threads.Πιο συγκεκριμένα οι δυνατότητες που παρέχονται από το NutOS είναι οι ακόλουθες:

*Cooperative Multithreading:*Το λειτουργικό υποστηρίζει πολλαπλά threads, απλοποιώντας την διαδικασία αφού έχει υιοθετήσει την προσέγγιση αυτή ,καθώς η αλλαγή των διεργασιών (threads)γίνεται μόνο όταν το thread που εκτελείται παραχωρεί τον έλεγχο στο επόμενο διαθέσιμο,αν υπάρχουν περισσότερα του ενός που αναμένουν τότε τρέχει το πιο σημαντικό.

Mutex-Semaphore-Event Support: Παρέχεται υποστήριξη από το ίδιο το λειτουργικό για αυτά τα πολύ χρήσιμα προγραμματιστικά εργαλεία , για την ανάπτυξη του λογισμικού χρησιμοποιήθηκαν Mutex για συγχρονισμό των threads και Events για την εξυπηρέτηση των interrupts.

*Υποστήριξη πρωτοκόλλων επικοινωνίας i2c,spi,1wire:*Επεκτείνοντας την ήδη υπάρχουσα λειτουργικότητα του atmega για τα πρωτόκολλα επικοινωνίας i2c,spi,1wire το NutOS παρέχει εύχρηστες και υψηλότερου επιπέδου συναρτήσεις που αναλαμβάνουν να απλοποιήσουν την διαδικασία επικοινωνίας με αυτά .

*udp,tcp sockets :*Ίσως το μεγαλύτερο ατού αυτού του λειτουργικού είναι το πολύ δυνατό API,που περιβάλλει τα sockets,με την χρήση του οποίου κατέστη δυνατή η υλοποίηση του λογισμικού,για την εξυπηρέτηση των δικτυακών αναγκών της τεχνολογίας που αναπτύχθηκε.

File System(pnut): Άν και δεν χρησιμοποιήθηκε στην συγκεκριμένη εφαρμογή , υποστηρίζεται και ένα file system περιορισμένων δυνατοτήτων αν και αυτό πρέπει να επιλεγθεί κατά την διαδικασία χτισίματος του πυρήνα του λειτουργικού.

Avr-libc compliant: Σημαντικό προτέρημα είναι και η συμβατότητα με την standard avr-libc.

*Modular-Kernel:*Το NutOS μέσω ενός εργαλείου της κοινότητας που το αναπτύσσει (NutOS configurator), μπορεί να "χτιστεί" (build) για τις ανάγκες της εκάστοτε εφαρμογής ,επιλέγοντας κάθε φορά τις δυνατότητες που χρειάζονται ,με βάση τις εκάστοτε ανάγκες,έτσι γίνεται βελτιστοποίηση του πυρήνα και του ίδιου του λειτουργικού.

*Υλικό εκμάθησης -Τεχνικά Κείμενα:*Μια πολύ ενεργή κοινότητα αναπτύσσει καθημερινά το λειτουργικό και υπάρχουν άρτια τεκμηριωμένα τεχνικά κείμενα και παραδείγματα ανάπτυξης λογισμικού .

*Ελεύθερο Λογισμικό(Free Software):*Καθώς το NutOS ,καλύπτεται από την άδεια ελεύθερου λογισμικού δεν χρειάζεται να πληρώσουμε δικαιώματα για την χρήση του και για τον ίδιο λόγο μπορούμε ,αν χρειαστεί να αλλάξουμε κατά βούληση και τον κώδικα του παράγοντας αν θέλουμε δική μας έκδοση του λειτουργικού.

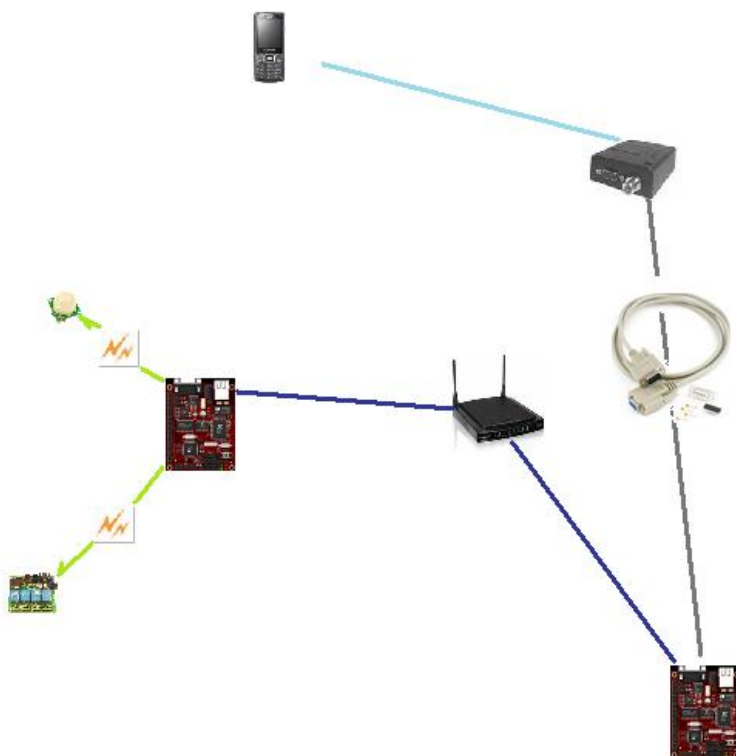
Όλα τα παραπάνω προτερήματα του NutOS ήταν ανεκτίμητης αξίας στην ανάπτυξη του λογισμικού μας.Τέλος ένα ακόμα βασικό κριτήριο για την επιλογή αυτού του λειτουργικού ήταν ότι η εταιρεία που κατασκευάζει την πλακέτα ethernut υποστηρίζει και το λειτουργικό σύστημα NutOS.

Αρχιτεκτονική

Η τοπολογία του δικτύου έχει 2 βασικά κομμάτια με διαφορετική σημασία το καθένα αλλά όπως θα γίνει φανερό είναι εξίσου σημαντικά. Μπορούμε να διαχωρίσουμε την εργασία(project) σε δυο επίπεδα , το επίπεδο του δικτύου Ip και την Κεντρική συσκευή .

Το δίκτυο IP είναι βασικό γιατί αυτό θα είναι το κανάλι μέσω του οποίου θα γίνεται η επικοινωνία μεταξύ των συσκευών.

Για να επιτευχθεί αυτό χρειάζεται ένας DHCP SERVER, ένα router δηλαδή που θα ασχολείται με την απόδοση Ip σε κάθε συσκευή που συνδέεται στο δίκτυο και θα δρομολογεί τα μηνύματα στους σωστούς



Εικόνα 1 Αρχιτεκτονική ΕΣΤΙΑ

παραλήπτες. Καθώς βασικός άξονας σχεδιασμού είναι και το κόστος, το ρόλο του switch μπορεί να παίξει ένα κοινό ADSL MODEM SWITCH. Πέρα από προσιτή τιμή, τέτοιες συσκευές είναι πλέον πολύ άμεσα διαθέσιμες στο ευρύ κοινό καθώς δίνονται από όλους τους παρόχους (isps) με την έναρξη κάποιας σύνδεσης. Αυτό έχει το επιπρόσθετο όφελος ότι χωρίς επιπλέον κόστος μπορεί να χρησιμοποιηθεί και από την Εστία.

«Κεντρική Συσκευή» είναι η «καρδιά» του δικτύου ΕΣΤΙΑ, καθώς πρόκειται για την συσκευή που συνδέεται και ελέγχει το GSM modem. Επίσης έχει το φόρτο ελέγχου και συντονισμού των άλλων συσκευών όπως τα SDH και γενικά οποιασδήποτε συσκευής που είναι μέλος του Δικτύου Συσκευών.

Η Κεντρική συσκευή καταχωρεί όλες αυτές τις συσκευές σε ένα εσωτερικό πίνακα, αποθηκεύοντας σημαντικές πληροφορίες όπως όνομα, διεύθυνση, δεδομένα και group στο οποίο ανήκουν. Αυτό είναι και η βάση του δικτύου ΕΣΤΙΑ. Να σημειωθεί ότι τη διεύθυνση Ip

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet κάθε συσκευή την έχει πρώτα λάβει από το ADSL MODEM (ή γενικά το router πάνω στο οποίο συνδέουμε τις συσκευές) και ύστερα ενημερώνει την Κεντρική Συσκευή.

Η τοπολογία όπως φαίνεται στην εικόνα 2 είναι η εξής: στο router συνδέονται όλες οι συσκευές ip (τα ethernet), στην κεντρική συσκευή συνδέουμε με σειριακή θύρα το gsm modem και στις επιμέρους ip συσκευές συνδέουμε τα απλά ηλεκτρονικά κομμάτια ttl, i2c, στα ανάλογα pins όπως αυτά ορίζονται στο specification της συσκευής. Στα πλαίσια αυτής της εργασίας υλοποιήθηκε μια τέτοια συσκευή με το κωδικό όνομα «Ύδρα», αναλυτικές πληροφορίες για την οποία υπάρχουν στο αντίστοιχο κεφάλαιο. Με αυτόν τον τρόπο μπορούμε να έχουμε λίγες μονάδες ip, που άλλωστε είναι και οι ακριβότερες, συνδέοντας σε αυτές πολλά φθηνά αισθητήρια και διακόπτες.

Ένα άλλο μεγάλο προτέρημα της παραπάνω σχεδίασης είναι ότι μπορούμε να έχουμε διάφορα φυσικά μέσα επικοινωνίας αφού το ip Layer μας δίνει την δυνατότητα για abstraction. Θα μπορούσαμε να έχουμε δηλαδή ασύρματη σύνδεση μεταξύ των συσκευών είτε ενσύρματη, είτε ακόμα και σύνδεση μέσω power lines, το οποίο σε συνδυασμό με το χαμηλό κατά μονάδα κόστος μας δίνει μεγάλη ευελιξία.

Λέγοντας κατά μονάδα κόστος, εννοούμε ότι καθώς συνδέουμε πολλές απλές συσκευές (αισθητήρια, διακόπτες) πάνω σε μια ip συσκευή ethernet, ουσιαστικά καταφέρνουμε να διαιρούμε το κόστος του Ip δικτύου μας. Αφού ενώ έχουμε μια πραγματική συσκευή ip με χρήση της συγκεκριμένης αρχιτεκτονικής- τεχνολογίας, μετατρέπουμε όλα τα αισθητήρια και τους διακόπτες σε ισοδύναμες Ethernet συσκευές.

Εστία Επισκόπηση Λειτουργίας

Network Initialization-Overview

Κάθε συσκευή πέραν της Κ.Σ (κεντρική συσκευή), αφού λάβει ip περιμένει για ένα σήμα συγχρονισμού το οποίο στέλνει η ΚΣ και μεταφέρει, μεταξύ άλλων, το αναγνωριστικό της και την ip της. Έτσι κάθε συσκευή γνωρίζει ποια είναι η Κεντρική Μονάδα, κάτι το οποίο είναι απαραίτητο. Ύστερα από αυτή τη διαδικασία έρχεται το στάδιο registration, που κάθε συσκευή αναγνωρίζει πρώτα τις συνδεδεμένες σε αυτήν ηλεκτρονικές συσκευές και ύστερα τις καταχωρεί στην Κεντρική Μονάδα με τη χρήση του κατάλληλου μηνύματος. Με αυτόν τον τρόπο, μπορούμε να μετατρέψουμε ουσιαστικά πολλές απλές ηλεκτρονικές συσκευές σε Ip συσκευές χρησιμοποιώντας μόνο ένα ip interface. Ύστερα από το βήμα registration, κάθε συσκευή συγχρονίζεται με την κεντρική μονάδα και έτσι υπάρχει χρονική σταθερότητα σε όλο το δίκτυο συσκευών. Αυτό είναι πολύ σημαντικό καθώς πρέπει να υπάρχει κοινή βάση αναφοράς χρόνου αλλιώς δεν είναι δυνατή η ωφέλιμη συνεργασία μεταξύ των συσκευών. Να σημειώσουμε σε αυτό το σημείο ότι ο χρόνος λαμβάνεται από την πύλη του GSM provider. Όταν η Κεντρική Μονάδα τεθεί σε λειτουργία, στέλνει ένα μήνυμα στο χρήστη ότι τέθηκε σε ισχύ και με την λήψη της αναφοράς αυτού του μηνύματος λαμβάνει το σωστό χρόνο.

Αρχικοποίηση συσκευής Master Device

- Φόρτωση στοιχείων συσκευής, χρηστή και PIN(για το συνδεδεμένο GSM) από την σταθερή μνήμη
- Αρχικοποίηση του GSM modem σε περίπτωση αποτυχίας, επανάληψη, πρόκειται για απαραίτητη προϋπόθεση. Εάν δεν ολοκληρωθεί αυτό το βήμα η διαδικασία δεν συνεχίζεται
- Αποστολή μηνύματος λειτουργίας στον χρήστη
- Ρύθμιση χρόνου από την αναφορά λήψης του μηνύματος λειτουργίας στον χρήστη
- Φόρτωση και αρχικοποίηση των χρονικών εντολών, αν υπάρχουν, από την μη μεταβαλλόμενη μνήμη
- Broadcast Master Signal στο δίκτυο στο port 23
- Μετάβαση στην φάση Operation.

Device Hub Initialization

- Φόρτωση στοιχείων συσκευής(alias)
- Αναγνώριση και αρχικοποίηση εσωτερικών συσκευών
- Φόρτωση χρονικών εντολών από τη μνήμη και αρχικοποίηση αυτών
- Αναμονή σήματος Master, η διαδικασία δεν προχωράει αν δεν ολοκληρωθεί αυτό το βήμα
- Καταχώρηση εσωτερικών συσκευών στην Κεντρική Συσκευή
- Συγχρονισμός με την κεντρική συσκευή
- Συνέχεια στο βήμα Operation

Φάση Λειτουργίας Κεντρικής Μονάδας

- Αναμονή για εισερχόμενα δεδομένα από το gsm modem και εξυπηρέτηση εισερχόμενων εντολών από sms
- Εκτέλεση αποθηκευμένων χρονικών εντολών
- Μέτρηση χρόνου και σε περίπτωση λήψης καινούργιου χρόνου από το δίκτυο GSM ενημέρωση του δικτύου συσκευών
- Σε περιοδικά διαστήματα εκπομπή σήματος αναγνώρισης για καινούργιες συσκευές
- Εξυπηρέτηση εισερχόμενων μηνυμάτων από το Ip interface και διευθέτηση τους
- Κατά την διάρκεια του βήματος Device Reporting των άλλων συσκευών γίνεται ενημέρωση του κεντρικού πίνακα κατάστασης συσκευών
- Ενημέρωση του εσωτερικού πίνακα συσκευών με καινούργιες συσκευές κατά το βήμα network initialization των καινούργιων στο δίκτυο συσκευών

Φάση Λειτουργίας Συνδεδεμένων Device Hubs

Μετά την ολοκλήρωση του σταδίου «Network Initialization», κάθε συσκευή προχωράει στο στάδιο Operation. Κατά τη διάρκεια του υπάρχουν 2 βασικές εργασίες (tasks) για να ελέγχονται οι συνδεδεμένες σε αυτήν ηλεκτρονικές συσκευές, και να αναφέρει περιοδικά την κατάσταση τους στην Κεντρική Μονάδα. Το βήμα της λειτουργίας (operation) είναι διαφορετικό για την κεντρική μονάδα.

Έλεγχος Εσωτερικών Συσκευών Device Hub

Σε αυτήν την φάση πραγματοποιούνται τα ακόλουθα βήματα

- Προσοχή για εισερχόμενες εντολές που αφορούν την λειτουργία των συνδεδεμένων στην συσκευή αισθητήριων και switches
- Εξυπηρέτηση των εξωτερικών interrupts και εκτέλεση των συνδεδεμένων με αυτά εντολών (βλ. Κεφάλαιο Ύδρα)
- Εκτέλεση των αποθηκευμένων χρονικών εντολών
- Περιοδική αυτοσυντήρηση
- Μέτρηση του χρόνου από ανεξάρτητη διεργασία με χρήση internal timers

Φάση Αναφοράς Συσκευών «Device Reporting»

Ανά τακτά χρονικά διαστήματα οι συσκευές αναφέρουν την κατάσταση των εσωτερικών τους συσκευών στην κεντρική μονάδα έτσι ώστε να υπάρχει μια σύγχρονη εικόνα της κατάστασης των συσκευών στο πεδίο, άμεσα προσβάσιμη στον χρήστη.

Πρωτόκολλο Επικοινωνίας Συσκευών

Εφόσον σκοπεύουμε να έχουμε μια δικτύωση συσκευών και επιθυμούμε αυτές να συνεργάζονται, κάτι τέτοιο είναι πολύ επιθυμητό και αναγκαίο, πρέπει να έχουμε ένα πρωτόκολλο επικοινωνίας μέσω του οποίου οι συσκευές να ανταλλάσσουν δεδομένα .

Βήματα Πρωτοκόλλου

1. Εύρεση παραλήπτη από τον Κεντρικό Πίνακα
2. Αποστολή μηνύματος
3. Αναμονή απάντησης
4. Επεξεργασία απάντησης
5. Επανάληψη βημάτων 1-3

Τα παραπάνω βήματα εν γένει είναι κοινά σε πολλά πρωτόκολλα επικοινωνίας, κάτι το οποίο είναι αναμενόμενο αφού είναι και αρκετά προσφιλή σε μας και αρκετά απλά και λογικά. Παρόλα αυτά συνήθως η υλοποίηση αυτών δεν συνάδει με την απλότητα της σκέψης αυτής. Στο συγκεκριμένο πρωτόκολλο έχει γίνει μεγάλη προσπάθεια να παραμείνει απλό αλλά δυνατό μέσα στα όρια των περιορισμών που μας θέτει το γεγονός ότι πρόκειται για πρωτόκολλο επικοινωνίας μεταξύ ενσωματωμένων συσκευών με περιορισμένες δυνατότητες και μεγάλο φόρτο εργασίας. Τέλος αλλά εξίσου σημαντικό είναι το γεγονός πως θα πρέπει να ληφθεί υπόψιν και ο περιορισμός των 170 χαρακτήρων που έχει ένα sms.

Με γνώμονα τα παραπάνω υλοποιήθηκε το SDMP (Simple Device Messaging Protocol), αναλυτική περιγραφή του οποίου ακολουθεί στο επόμενο κεφάλαιο.

Simple Device Messaging Protocol (SDMP)

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, η επικοινωνία μεταξύ των συσκευών γίνεται μέσω ενός πρωτοκόλλου που δημιουργήθηκε στα πλαίσια αυτής της εργασίας για να καλύψει τις ανάγκες του σχεδιασμού.

Οι λόγοι που επέβαλαν τον σχεδιασμό ενός καινούργιου πρωτοκόλλου επικοινωνίας είναι αρκετοί, σημαντικότεροι εκ των οποίων είναι ότι τα περισσότερα ολοκληρωμένα προϊόντα είναι κλειστής αρχιτεκτονικής και αυτό συμπεριλαμβάνει και το πρωτόκολλο που χρησιμοποιούν για επικοινωνία, ενώ άλλα έχουν ένα σύστημα επικοινωνίας το οποίο δεν εφαρμόζεται σε συσκευές με περιορισμένες λειτουργικές ικανότητες.

Αρχές σχεδίασης :

1. Απλό αλλά δυνατό
2. Κύριος στόχος ενσωματωμένες συσκευές με περιορισμένους πόρους
3. Εύκολα επεκτάσιμο
4. Device independent
5. Τα μηνύματα (εντολές) θα πρέπει να είναι περιορισμένου μεγέθους

Περιγραφή:

Η επικοινωνία με το SDMP γίνεται μέσω μηνυμάτων. Όλα τα μηνύματα μεταξύ συσκευών θα μπορούσαν να παραλληλιστούν με εντολές που μεταφέρουν, μαζί με την λειτουργία που πρέπει να εκτελεστεί και διάφορες παραμέτρους που δίδουν την εκτέλεση της εντολής αυτής. Τα μηνύματα αυτά είναι περιορισμένου μεγέθους, και είναι φτιαγμένα έτσι η σύνταξη τους, που η αποκωδικοποίηση τους να είναι εύκολη χωρίς να χρειάζεται πολλά βήματα από τον παραλήπτη. Το πιο σημαντικό είναι να είναι μονοσήμαντη το πολύ σε 1 βήμα¹. Επίσης κάθε μήνυμα είναι αυτούσιο και δε χρειάζεται αλλά δεδομένα, δηλαδή η πληροφορία που χρειάζεται για την εκτέλεση τους εμπεριέχεται σε αυτά.

Τα πεδία παραμέτρων έχουν διαφορετικές χρήσεις ανάλογα με το μήνυμα, στην περίπτωση του μηνύματος πρόσθεσης συσκευής² τα πεδία παραμέτρων μεταφέρουν πληροφορίες για την συσκευή που πρέπει να προστεθεί στο σύστημα, ενώ στην περίπτωση του μηνύματος συγχρονισμού τα πεδία μεταφέρουν πληροφορία χρόνου. Ακόμα σε κάποιες περιπτώσεις έχουν και υποπεδία για να ικανοποιηθούν οι ανάγκες των πιο πολύπλοκων εντολών, όπως οι χρονικές εντολές³.

Προεσκόπηση Πρωτοκόλλου

Τα μηνύματα (εντολές) χωρίζονται σε 3 κατηγορίες ανάλογα με την πολυπλοκότητα των λειτουργιών που καλούνται να υποστηρίξουν. Επειδή χρειάζονται και περισσότερα δεδομένα υπάρχουν περισσότερες παράμετροι που πρέπει να περάσουν στον παραλήπτη.

Απλά Μηνύματα Επικοινωνίας(Simple Messages)

1. Μηνύματα σύνθετων λειτουργιών (Complex Messages)
2. Ειδικά Μηνύματα Συσκευών (Special Messages)

Οι διαφορές μεταξύ των μηνυμάτων αυτών, είναι σε συντακτικό επίπεδο, η ύπαρξη περισσότερων παραμέτρων. Η ιδέα παραμένει η ίδια σε όλα, δηλαδή κάθε μήνυμα-εντολή μεταφέρει όλη την πληροφορία που χρειάζεται για την εκτέλεση του σε ένα βήμα. Επειδή όμως από την φύση τους κάποιες λειτουργίες είναι πιο πολύπλοκες, χρειάζονται και περισσότερες παραμέτρους. Αυτό θα γίνει πιο ξεκάθαρο στο επόμενο κεφάλαιο με την παρουσίαση των εντολών του SDMP.

Το SDMP είναι ανεξάρτητο τόσο από το φυσικό μέσο μετάδοσης, όσο και από την τεχνολογία που χρησιμοποιείται για να μεταδώσει το μήνυμα (GSM NETWORK ,IP layer,rs232), αν και σχεδιάστηκε για να υποστηρίξει καλύτερα το IP πρωτόκολλο .

SDMP Messages General Syntax

COMMAND_NAME*TARGET*MESSAGE_BODY*ATTRIBUTE1*ATTRIBUTE2

Μέγεθος μηνύματος: Το μέγεθος του παραπάνω μηνύματος σε περίπτωση που μεταδίδεται μέσω SmS δεν πρέπει να υπερβαίνει τους 160 χαρακτήρες σε μέγεθος και σε περίπτωση μετάδοσης μέσω άλλων μέσων, τους 256 χαρακτήρες. Σε οποιαδήποτε περίπτωση το μέγεθος του Message_body δεν πρέπει να υπερβαίνει τους 170 χαρακτήρες. Ο λόγος για τους παραπάνω περιορισμούς είναι ότι τα μηνύματα πρέπει να μπορούν να μεταδίδονται από ένα sms, για λογούς κόστους αλλά και απόδοσης το μέγεθος των μηνυμάτων πρέπει να υπόκειται στα παραπάνω όρια. Ταυτόχρονα υπάρχει κέρδος και σε απόδοση αλλά και σε κόστος επικοινωνίας.

Επεκτασιμότητα Πρωτοκόλλου: Το SDMP είναι αρκετά ευέλικτο όσον αφορά τη δημιουργία καινούργιων τύπων μηνυμάτων (εντολών), καθώς κάθε μήνυμα καθορίζεται από το πρώτο πεδίο (πεδίο command_name). Ένας κατασκευαστής αρκεί να επεκτείνει το πρωτόκολλο δημιουργώντας μια καινούργια περίπτωση λειτουργίας επεκτείνοντας τον βασικό message handler. Υπάρχει όμως και ένας τρόπος να προστεθούν νέες εντολές χωρίς να χρειαστεί να γίνει αυτό, καθώς το ίδιο το SDMP έχει ένα ειδικό μήνυμα που ονομάζεται SPECIAL_COMMAND και σκοπός αυτού του μηνύματος είναι να μεταφέρει οποιαδήποτε πληροφορία στον παραλήπτη χωρίς οι διαμεσολαβητές να γνωρίζουν τίποτα άλλο. Έτσι μπορούν καινούργιες συσκευές να έχουν δικές τους λειτουργίες, άγνωστες στο υπόλοιπο σύστημα. Με τη χρήση αυτού του ειδικού μηνύματος αυτές οι λειτουργίες γίνονται προσβάσιμες από τις άλλες συσκευές και τον χρηστή.

Σημασία Πεδίων Μηνύματος: Όπως αναφέρθηκε και στην αρχή αυτού του κεφαλαίου, βασικό χαρακτηριστικό του SDMP είναι ότι όλη η πληροφορία που απαιτείται για την εργασία ενός μηνύματος εμπεριέχεται όλη σε αυτό. Κάτι τέτοιο επιτυγχάνεται με 2 τρόπους: Καταρχάς κάθε μήνυμα μπορεί να μεταφέρει πληροφορία στο πεδίο MESSAGE_BODY και ο παραλήπτης την χρησιμοποιεί για την εκτέλεση της εργασίας του. Υπάρχουν όμως άλλα 3 πεδία τα οποία μεταφέρουν πληροφορία αν αυτή χρειάζεται, τα πεδία attribute1, attribute2

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet και target. Θα παρατηρήσατε τη χρήση της φράσης «αν αυτή χρειάζεται», λόγω περιορισμένου μεγέθους (Θυμηθείτε ότι το μήνυμα δεν μπορεί να υπερβαίνει τους 160 χαρακτήρες όταν μεταδίδεται από sms και τους 256 από αλλά μέσα. Κάθε τύπος μηνύματος μπορεί να παραλείπει κάποια πεδία χρησιμοποιώντας το χώρο τους για άλλα πεδία όπου χρειάζεται. Επίσης, σε κάποιες περιπτώσεις, κάποια πεδία μπορεί να παραληφθούν τελείως μεταφέροντας ωστόσο πληροφορία. (Δείτε SEND_DATA, REQUEST_DATA)

Πεδίο Target: Οι τιμές αυτού του πεδίου είναι 3 ('A', 'G', 'I') και η σημασία τους είναι:

- 'A'=O παραλήπτης στο Πεδίο Attribute 2 είναι με το ψευδώνυμο του (alias) και είναι μια συσκευή
- 'G'= O παραλήπτης στο Πεδίο Attribute 2 είναι με το ψευδώνυμο του (alias) και είναι ένα σύνολο από συσκευές (GROUP)
- 'I'= O παραλήπτης στο Πεδίο Attribute 2 είναι με το IP και είναι μια συσκευή

Σύνολα συσκευών με χρήση IP δεν μπορούν να δοθούν .

Πεδία Attribute1,2: Τα πεδία αυτά μεταφέρουν διαφορετικού είδους πληροφορία, ανάλογα με το είδος του μηνύματος, στα μηνύματα μεταφοράς δεδομένων (SEND_DATA ,REQUEST_DATA.

Το Πεδίο attribute1 είναι το Πεδίο πηγής (source) και το Πεδίο attribute 2 είναι ο παραλήπτης. Ανάλογα με την τιμή του Πεδίου target ο παραλήπτης μπορεί να είναι μια συσκευή ή ένα σύνολο από συσκευές. Στις πιο πολύπλοκες εντολές, όπως είναι οι χρονικές εντολές τα πεδία αυτά έχουν διαφορετική σημασία, δηλαδή το πρώτο Πεδίο είναι ο χρόνος έναρξης της εντολής, ενώ το δεύτερο Πεδίο είναι ο χρόνος τερματισμού της χρονικής εντολής.

Μεταβλητά Πεδία: Όπως φάνηκε μέχρι τώρα, τα πεδία έχουν διαφορετική σημασία και χρησιμότητα ανάλογα με την κατάσταση (context) που χρησιμοποιούνται. Όπως θα φανεί στην πορεία έχουν και διαφορετική σύνταξη σε κάποιες περιπτώσεις. Κάτι τέτοιο ουσιαστικά επιβάλλεται από τον περιορισμό μεγέθους των 160 χαρακτήρων, εφόσον λόγω μεγέθους δεν μπορεί να μεταφερθεί, αλλά με τη χρήση της παραπάνω μεταβλητότητας μπορεί να εννοηθεί κατά περίπτωση.

Βασικές Εντολές

Κάθε μήνυμα χωρίζεται σε πεδία με τη χρήση ειδικών διαχωριστών (delimiters). Στα βασικά μηνύματα ο διαχωριστής των πεδίων είναι ο δεσμευμένος χαρακτήρας '*' και κάθε βασικό μήνυμα είναι της μορφής

COMMAND_NAME*TARGET*MESSAGE_BODY*ATTRIBUTE1*ATTRIBUTE2

Σύνταξη Βασικών Εντολών

Μία εντολή αποτελείται από 5 πεδία και ο χαρακτήρας διαχωρισμού των πεδίων είναι ο '*'.

COMMAND_NAME*[ENTITY]*COMMAND_BODY*CMD. Attribute1*CMD. Attribute2

Τα πεδία έχουν τους εξής ορισμούς:

(να σημειωθεί ότι ο '*' είναι **υποχρεωτικός** μεταξύ των πεδίων)

COMMAND_NAME = Πρόκειται για το όνομα της εντολής και είναι υποχρεωτικό σε κάθε περίπτωση. Εντολή που έχει μη αναγνωρίσιμο όνομα αγνοείται.

Entity=Πρόκειται για το σύνολο στο οποίο αναφέρεται η εντολή, αποδεκτές τιμές είναι:

- G, g=group, η εντολή αναφέρετε σε ένα group συσκευών (σημ.)
- A,a=alias ,η εντολή αναφέρετε σε μια συσκευή η οποία αναφέρεται με το όνομα της(alias)
- I, I=I, η εντολή αναφέρετε σε μια συσκευή και αυτή δίνετε με το Ip, της

Command_body= μέγεθος μέχρι και 160 χαρακτήρες (σημ.). Αυτό είναι το πεδίο που χρησιμοποιείται για να μεταφερθούν τα (κύρια) δεδομένα της εντολής

CMD.Attribute1= Μέχρι και 33 χαρακτήρες. Στις περισσότερες εντολές αναφέρετε η συσκευή που έστειλε αυτήν την εντολή, ενώ κάποιες φορές αυτό το πεδίο μπορεί να παραληφθεί.

CMD. Attribute2= Μέχρι και 33 χαρακτήρες. Στις περισσότερες εντολές αναφέρετε η συσκευή –group που είναι ο αποδέκτης αυτής της εντολής, κάποιες φορές όμως αυτό το πεδίο μπορεί να παραληφθεί.

Σύνθετες Εντολές

Πιο σύνθετες εντολές χρησιμοποιούν μια πιο περίπλοκη σύνταξη του SDMP ,πραγματοποιώντας διαίρεση των βασικών πεδίων σε μικρότερα και αποθηκεύοντας περισσότερες πληροφορίες. Ο αριθμός αυτών ποικίλει κατά περίπτωση.

Ειδικές περιπτώσεις πεδίων Body. ATTRIBUTE1.ATTRIBUTE2

Το πρωτόκολλο SDMP επιτρέπει, αν χρειάζεται, να μεταφερθεί παραπάνω πληροφορία, όπως στην περίπτωση της εντολής SCHEDULED_COMMAND ATTRIBUTE2(END_TIME) και να γίνει επιπλέον μορφοποίηση του πεδίου για να ικανοποιηθούν οι ανάγκες αυτών των πιο πολύπλοκων εντολών. Με τη χρήση του δεσμευμένου χαρακτήρα '&' η πληροφορία μορφοποιείται με αυτόν τον τρόπο παρόμοια με την γενική σύνταξη π.χ.

SCHEDULED_COMMAND*TARGET*COMMAND_BODY*START_TIME*END_TIME

COMMAND_BODY*=*&COMMAND_NAME&TARGET&CMD.ATTRIBUTE1&CMD.ATTRIBUTE2&

END_TIME *=*&_YYYY_MM_DD_hh_mm_ss_&?hh?mm?ss?&

Το πεδίο Attribute2 εννοιολογικά μεταφέρει την πληροφορία του χρόνου λήξης της παραπάνω εντολής(END_TIME).

Το Πεδίο commandBody μεταφέρει την εντολή που θα εκτελεστεί σε κάθε περίοδο της χρονικής εντολής. Παρηρούμε ότι ουσιαστικά πρόκειται για μια βασική σύνταξη με μοναδική διαφορά την χρήση του χαρακτήρα '&' αντί του '*', για το διαχωρισμό της πληροφορίας.

Με την χρήση του χαρακτήρα '&' βλέπουμε ότι έχουμε 2 καινούργια πεδία μέσα στο υποπεδίο Attribute2, υποπεδίο1 = _YYYY_MM_DD_hh_mm_ss_ που είναι η ημερομηνία λήξης της χρονικής εντολής και το υποπεδίο 2 = ?hh?mm?ss?, που είναι η περίοδος επανάληψης της χρονικής εντολής. Στο υποπεδίο2 βλέπουμε και έναν ακόμη δεσμευμένο χαρακτήρα '?', που είναι και ο τελευταίος και πιο σπάνια χρησιμοποιούμενος χαρακτήρας διαχωρισμού (delimiter). Αυτός ο χαρακτήρας διαιρεί περαιτέρω το υποπεδίο2 σε 3 καινούργια πεδία, που μεταφέρουν την πληροφορία επανάληψης της εντολής.

Η χρονική εντολή δεν είναι η μόνη που χρησιμοποιεί την δυνατότητα διαίρεσης των βασικών πεδίων για να μεταφέρει περισσότερες πληροφορίες. Οποτεδήποτε η απλή σύνταξη δεν αρκεί για την ικανοποίηση των αναγκών κάποιας εντολής γίνεται διαίρεση των πεδίων.

Ακολουθεί Πίνακας με τις **Γενικά Διαθέσιμες εντολές**, που αποτελεί τη βάση του πρωτόκολλου.

Στο μέλλον φυσικά μπορεί να προστεθούν και άλλες εντολές ή μπορεί να επαυξηθεί για να εξυπηρετήσει ανάγκες συγκεκριμένων εντολών. Πρέπει όμως τις ακόλουθες εντολές να τις υποστηρίζουν όλες οι συσκευές για να υπάρχει μια κοινή βάση επικοινωνίας.

ΔΙΑΘΕΣΙΜΕΣ ΕΝΤΟΛΕΣ				
COMMAND_NAME	BODY	TARGET	ATTRIBUTE 1	ATTRIBUTE 2
<i>FIND</i>	ΔΕ ΧΡΕΙΑΖΕΤΑΙ	A,I,Σ	Ζητών	Στόχος
<i>SEND_DATA</i>	Δεδομένα προς μεταφορά	A,I,Σ	Αφετηρία	Στόχος
<i>REQUEST_DATA</i>	Δεν χρειάζεται	A,I,Σ	Αφετηρία	Στόχος
<i>SWITCH_OFF</i>	Δε χρειάζεται	A,I,Σ	Αφετηρία	Στόχος
<i>SWITCH_ON</i>	Δε χρειάζεται	A,I,Σ	Αφετηρία	Στόχος
<i>ABORT_SCHEDULED</i>	Αριθμός Εντολής Προς διαγραφή	Δε χρειάζεται	Δε χρειάζεται	Δε χρειάζεται
<i>SCHEDULED_COMMAND</i>	Εντολή που θα εκτελεστεί	Δε χρειάζεται	Χρόνος Εκκίνησης	Χρόνος Τερματισμού*
<i>SEND_SMS</i>	Μήνυμα Κειμένου	A	Αριθμός τηλεφώνου	Ζητών
<i>ADD_ME</i>	IP Συσκευής	A	Alias Συσκευής	Κατηγορία Που ανήκει
<i>REMOVE_ME</i>	IP /alias	A,i	Δε Χρειάζεται	Δε Χρειάζεται
<i>ADD_TO_GROUP</i>	IP /alias	A,i	Όνομα του Group	Δε Χρειάζεται
<i>REMOVE_FROM_GROUP</i>	IP /alias	A,i	Όνομα του Group	Δε Χρειάζεται
<i>SET_TIME</i>	Χρόνος	A	Αφετηρία	Στόχος
<i>SET_OWNER</i>	Δεδομένα	A	Δε Χρειάζεται	Δε Χρειάζεται
<i>SET_DEVICE_INFO</i>	Δεδομένα	A	Δε Χρειάζεται	Δε Χρειάζεται
<i>SPECIAL_COMMAND</i>	Κατά Περίπτωση	A,I,Σ	Κατά Περίπτωση	Κατά Περίπτωση

Πίνακας 1 Οι εντολές αυτές είναι Υποχρεωτικό να υποστηρίζονται από όλες τις συσκευές

Περιγραφή Λειτουργίας Διαθέσιμων Εντολών

SEND_DATA :

Syntax: *SEND_DATA*TARGET*<DATA>*SOURCE*TARGET*

Σκοπός: Μεταφορά δεδομένων από το device source στο device target, εναλλακτικά αν το target είναι Group, τα Δεδομένα θα μεταφερθούν σε όλες τις συσκευές του group.

Λειτουργία: Τα πεδία Target και Source μπορούν να παραληφθούν και να χρησιμοποιηθεί ο χώρος τους στο πεδίο <Data>. Κατά σύμβαση όταν μια συσκευή λαμβάνει δεδομένα χωρίς να είναι ορισμένος ο target, τότε τα κρατάει αυτή και προσπαθεί να τα επεξεργαστεί με τον default τρόπο. Κάθε συσκευή πρέπει να υποστηρίζει μια default επεξεργασία, πχ. ένας θερμοστάτης αν λάμβανε δεδομένα μπορεί να έθετε το καινούργιο threshold. Παράλληλα εάν απουσιάζει ο αποστολέας, τότε για να χρησιμοποιηθούν τα δεδομένα πρέπει η default action να μην χρειάζεται σαν δεδομένο τον αποστολέα. Καμία απάντηση δε δίδετε σαν αποτέλεσμα αυτού του μηνύματος, Φυσικά η εκάστοτε συσκευή μπορεί να κάνει κάτι σαν αποτέλεσμα της εντολής αυτής, αλλά το πρωτόκολλο δεν το απαιτεί.

REQUEST_DATA

Syntax : *REQUEST_DATA*TARGET*<EMPTY>*SOURCE*TARGET*

Σκοπός: Αντίστοιχα με την προηγούμενη Εντολή, προκαλεί τον παραλήπτη (target) να του στείλει δεδομένα. Κάθε συσκευή έχει ένα σύνολο από πληροφορία που θεωρεί δεδομένα, αυτά θα επιστραφούν με μια ανάλογη Εντολή SEND_DATA. Εδώ το πεδίο SOURCE πρέπει να συμπληρωθεί, αλλιώς ο target δε θα γνωρίζει που να στείλει τα δεδομένα. Ανάλογα με τη χρήση του πεδίου target, ο Στόχος μπορεί να είναι ένα group από συσκευές

SWITCH_OFF

Syntax : *SWITCH _OFF*TARGET*<EMPTY>*TARGET* *

Σκοπός: Προκαλεί συσκευές που υποστηρίζουν τη λειτουργία να αλλάξουν την κατάσταση σε off. Σε συσκευές όπως switches πρέπει να είναι το LOW state. Αν το target field είναι G (Group) τότε η λειτουργία εκτελείται σε όλες τις συσκευές του group.

SWITCH_ON

Syntax : *SWITCH _OFF*TARGET*<EMPTY>*TARGET* *

Σκοπός: Προκαλεί συσκευές που υποστηρίζουν τη λειτουργία να αλλάξουν κατάσταση σε on. Σε συσκευές όπως switches πρέπει να είναι το HIGH state. Αν το target field είναι G (Group) τότε η λειτουργία εκτελείται σε όλες τις συσκευές του group.

SCHEDULED_COMMAND

Syntax : *SCHEDULED_COMMAND *TARGET*COMMAND*START_TIME*END_TIME *

Σκοπός: Πρόκειται για μια συνθέτη Εντολή με σκοπό να εκτελεί άλλες απλές εντολές στο χρόνο. Η Εντολή που θα εκτελεστεί όταν έρθει ο Χρόνος START_TIME, είναι η Εντολή που υπάρχει στο πεδίο BODY. Η Εντολή αυτή μεταφέρει όλη την πληροφορία που χρειάζεται μορφοποιώντας περαιτέρω το BODY σε ένα καινούργιο μικρότερο μήνυμα με τον ίδιο αριθμό πεδίων, με την χρήση όμως του δεσμευμένου χαρακτήρα '&' (delimiter). Επίσης και στο πεδίο Attribute 2, δηλαδή στο END_TIME, εκτός από την ώρα λήξης εμπεριέχεται και η περίοδος επανάληψης με τη χρήση του τρίτου χαρακτήρα διαχωρισμού '?'.
 COMMAND_BODY=[&COMMAND_NAME&TARGET&BODY&ATTRIBUTE1&ATTRIBUTE2&]*

Όπως είναι φανερό πρόκειται για την σύνταξη ενός κανονικού μηνύματος που εμπεριέχεται στο COMMAND_BODY του προηγούμενου μηνύματος, με την χρήση όμως του δεύτερου χαρακτήρα διαχωρισμού '&'. Αυτός είναι και ο τρόπος που το SDMP μεταφέρει όταν χρειάζεται περισσότερη πληροφορία.

END_TIME=&_YEAR_MONTH_DAY_HOUR_MINUTE_SECONDS_&?HOURS?MINUTES:SECONDS?&*

Σημείωση: Πρέπει αυτή η Εντολή να αποθηκεύεται από την συσκευή στην eeprom (ή σε κάποιο άλλο είδος μη μεταβαλλόμενης μνήμης), καθώς η εντολή ολοκληρώνεται όταν έρθει ο Χρόνος τέλους, είτε ακυρωθεί από τον χρήστη με την ABORT_SCHEDULED. Είναι υποχρεωτικό όλες οι συσκευές να την υποστηρίζουν και με κάποιο τρόπο να εγγυώνται την αποθήκευση και επανεκκίνηση της μετά από διακοπή ρεύματος ή επανεκκίνηση της συσκευής.

ABORT_SCHEDULED

Syntax : * ABORT_SCHEDULED *TARGET*SCHEDULED_COMMAND NO* n/a* n/a*

Σκοπός: Πρόκειται για μια Εντολή που σκοπός της είναι να τερματίσει μια προηγούμενα δημιουργημένη χρονική Εντολή (scheduled command). Η μοναδική απαιτούμενη πληροφορία είναι ο Αριθμός της εντολής που επιθυμούμε να διαγράψουμε και αυτό ορίζεται στο πεδίο COMMAND_BODY

COMMAND_BODY=[1,2,3,4]*

Σημείωση: Καθώς κάθε Χρονική Εντολή είναι Υποχρεωτικό να αποθηκεύεται σε μια μη μεταβαλλόμενη μνήμη και το μέγεθος της μπορεί να είναι αρκετά μεγάλο, π.χ. 280 bytes για κάθε Εντολή, μόνο 4 εντολές είναι Υποχρεωτικό να μπορούν να δημιουργηθούν. Η εκάστοτε συσκευή μπορεί να δημιουργεί και άλλες, αυτό τίθεται στη διακριτική ικανότητα του κατασκευαστή κάθε συσκευής.

SEND_SMS

Syntax : * SEND_SMS*TARGET*SMS_BODY *PHONE_NUMBER or owner*DeviceName*

Σκοπός: Πρόκειται για μια Εντολή που σκοπός της είναι η αποστολή ενός SMS σε κάποιον αριθμό ή στο χρήστη με το συνθηματικό owner (δηλαδή στο αποθηκευμένο νούμερο που αντιστοιχεί στον κεντρικό χρήστη). Όπως σε όλες τις εντολές, πέραν φυσικά της REQUEST_DATA, η συσκευή που στέλνει το SMS δεν είναι υποχρεωμένη να απαντήσει για την επιτυχία ή μη της αποστολής.

*SMS_BODY *=*COMMAND_BODY*

Το μέγεθος του μηνύματος ΔΕΝ πρέπει να υπερβαίνει τους 160 χαρακτήρες.

PHONE_NUMBER or owner=* αριθμός τηλεφώνου σε διεθνή μορφή, είτε το συνθηματικό **owner**, το οποίο θα αναγκάσει την συσκευή να στείλει το μήνυμα στον υποθηκευμένο αριθμό του κύριου χρήστη.

DeviceName= Η συσκευή που προκάλεσε την αποστολή του μηνύματος, είναι προαιρετικό πεδίο για αυτήν την Εντολή.

ADD_ME

Syntax : * ADD_ME*TARGET*DEVICE_IP *DEVICE_ALIAS *DEVICE_TYPE *

Σκοπός: Πρόκειται για μια Εντολή που σκοπός της είναι η εισαγωγή (registration) μιας καινούργιας συσκευής στον πίνακα της κεντρικής συσκευής (master device). Κάθε DEVICE_ALIAS πρέπει να είναι μοναδικό. Σε περίπτωση σύγκρουσης (conflict) με ήδη υπάρχοντα alias, έχει μεγαλύτερη ισχύ το καινούργιο και η παλαιότερη συσκευή αντικαθίσταται από την καινούργια.

*DEVICE_IP *=*0-255.0-255.0-255.0255* πρόκειται για το IP address της συσκευής. Δεν είναι υποχρεωτικό να είναι μοναδικό, αφού σε κάποιες συσκευές όπως τα SDH(Simple Device Hubs) οι εσωτερικές συσκευές έχουν όλες το ίδιο IP και διαφορετικό alias. Αυτό είναι κάτι λογικό, καθώς το Ip τους είναι το Ip του hub.

*DEVICE_ALIAS *=*alphanumeric* μέχρι 33 χαρακτήρες και πρέπει να είναι μοναδικό σε κάθε δίκτυο.

*DEVICE_TYPE *=*alphanumeric* μέχρι 33 χαρακτήρες. Οι τύποι συσκευών που υποστηρίζονται εξ αρχής είναι οι εξής: "SENSORS", "ON_OFF", "DISPLAY", "SPECIAL_TYPE".

Μπορεί ωστόσο μια συσκευή να δηλώσει διαφορετικό τύπο. Συσκευές που δηλώνουν έναν από τους παραπάνω υποστηριζόμενους τύπους χρίζουν ιδιαίτερης διαχείρισης, καθώς εισάγονται αυτόματα στο ανάλογο group. Για παράδειγμα όλες οι συσκευές που είναι τύπου SENSORS μπαίνουν στο group SENSORS και ως αποτέλεσμα μπορούμε να ανακτήσουμε τα στοιχεία όλων των sensors που έχουν εισαχθεί στο σύστημα, χρησιμοποιώντας την Εντολή *FIND*G*SENSORS***, αντίστοιχα και στις άλλες κατηγορίες.

REMOVE_ME

Syntax : * REMOVE_ME*TARGET*DEVICE * *

Σκοπός: Πρόκειται για μια Εντολή με σκοπό να διαγράφει από τον πίνακα συσκευών της κεντρικής συσκευής. Αν το target είναι α, τότε πρέπει το αναγνωριστικό με το οποίο η κεντρική συσκευή θα αναζητήσει και θα διαγράψει την συσκευή να είναι το alias, ενώ αν το πεδίο target είναι Ι, τότε το αναγνωριστικό της συσκευής πρέπει να είναι το ip της.

ΠΡΟΣΟΧΗ: σε περίπτωση πολλαπλών εγγράφων με το ίδιο ip, όπως στην Περίπτωση SDH, η εγγραφή που θα αναιρεθεί θα είναι η πρώτη που έγινε με αυτή την ip. Γι' αυτό το λόγο προτείνεται να χρησιμοποιείται πάντα η μορφή μηνύματος με χρήση του alias.

*REMOVE_ME*A*DEV_ALIAS***

ADD_TO_GROUP

Syntax : * ADD_TO_GROUP*DEVICE_ALIAS *GROUP_NAME* *

Σκοπός: Πρόκειται για μια Εντολή που σκοπός της είναι η εισαγωγή της συσκευής που αναφέρεται στο πεδίο COMMAND_BODY να γίνει μέλος του group που αναφέρεται στο πεδίο attribute1. Θυμηθείτε πως αν το πεδίο target είναι i τότε πρέπει η συσκευή να δίνεται με το ip της, ενώ αν το πεδίο target είναι 'α' τότε πρέπει να δίδεται με το alias.

Σημείωση: Τα groups που υποστηρίζονται είναι τα ακόλουθα: SENSORS_GROUP, ON_OFF_GROUP, GROUP_A, GROUP_B, GROUP_C, GROUP_D. Τα groups sensors, on_off έχουν άμεση σχέση με τους ανάλογους τύπους συσκευών και όπως αναφέρθηκε και στην Εντολή ADD_ME, εισάγονται αυτόματα σε αυτά τα groups κατά την διαδικασία εισαγωγής τους στον πίνακα της κεντρικής συσκευής. Τα υπόλοιπα groups είναι τα λεγόμενα user_groups και όπως το όνομα τους προδίδει είναι απόλυτα στον έλεγχο του χρηστή το ποιες συσκευές ανήκουν σε αυτά τα groups .

Σημείωση: Κάθε συσκευή μπορεί να ανήκει σε ένα και μόνο ένα group.

REMOVE_FROM_GROUP

Syntax : * REMOVE_FROM_GROUP*DEVICE_ALIAS ** *

ΣΚΟΠΟΣ: Αφαιρεί τη συσκευή από το group στο οποίο αυτή ανήκει.

SET_TIME

Syntax : * SET_TIME*TARGET*TIME *SOURCE_DEV *TARGET_DEV *

ΣΚΟΠΟΣ: Ενημερώνει τη συσκευή που ορίζεται στο attribute2, ότι υπάρχει ενημέρωση χρόνου και αυτή πρέπει να ενημερώσει την εσωτερική της ώρα.

ΕΣΤΙΑ

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet

Σημείωση: Ο χρόνος όταν επικοινωνούν μεταξύ τους συσκευές έχει την ακολουθεί μορφή :

_YEAR_MONTH_DAY_HOUR_MINUTE_SECONDS_

Προσοχή!! Ο Χρόνος(Year) είναι τα χρόνια που πέρασαν από το 1900^{1(Appendix a)}

GET_TIME

Syntax: *GET_TIME*TARGET*DEVICE_REQUESTING*DEVICE_TARGET**

Σκοπός: Σε αντιστοιχία με την προηγούμενη Εντολή, μια συσκευή ζητά να συγχρονιστεί. Μια σημαντική διαφορά με τις άλλες εντολές είναι ότι το DEVICE_TARGET, μπορεί να πάρει την ειδική τιμή owner και αυτό σημαίνει ότι πρέπει να επικοινωνήσει η συσκευή με sms για να ενημερωθεί από τον χρηστή.

Eg *GET_TIME*A*PC1*owner*

SET_OWNER

Syntax: *SET_OWNER*TARGET*USER_NAME*PHONE_NUMBER*GEM PIN*

Σκοπός: Η εισαγωγή του master user στην κύρια συσκευή γίνεται με αυτήν την Εντολή και σε Περίπτωση αλλαγής χρηστή το ίδιο. Αυτή η Εντολή σώζεται στη μνήμη και έχει ισχύ από τη στιγμή της εκτέλεσης της, οπότε ο προηγούμενος χρήστης διαγράφεται εντελώς. Στοιχεία που μεταφέρονται επίσης είναι το τηλέφωνο επικοινωνίας με τον χρηστή, καθώς και το Pin της sim κάρτας του συνδεδεμένου στην κύρια συσκευή gsm modem.

Πχ *SET_OWNER*A*Agadakos_Ioannis*+306977400374*8272*

Σημείωση: Το τηλέφωνο πρέπει να δοθεί σε διεθνή μορφή.

SET_DEVICE_INFO

Syntax: *SET_DEVICE_INFO*target**DEVICE_INFO**

Σκοπός: Κάθε συσκευή πρέπει να έχει κάποιες πληροφορίες που αποθηκεύονται σε μια σταθερή μνήμη (eeprom). Αυτές είναι πληροφορίες όπως -κυρίως- το alias της συσκευής, όμως ανάλογα με την εκάστοτε συσκευή μπορεί να είναι και άλλες πληροφορίες που θα κρίνει απαραίτητες ο κάθε κατασκευαστής. Αυτή η Εντολή αλλάζει τις αποθηκευμένες πληροφορίες της συσκευής με το περιεχόμενο του πεδίου attribute1.

Ο λόγος που χρησιμοποιείται το πεδίο attribute1, και όχι το πεδίο body, για την μεταφορά αυτής της πληροφορίας είναι ότι η κύρια χρησιμότητα αυτής της εντολής είναι να αλλάζει το alias των συσκευών. Αυτό το πρωτόκολλο φτιάχτηκε σύμφωνα με την αρχιτεκτονική να υποστηρίζει κάθε συσκευή, γι' αυτό το λόγο έχει ένα alias το **πολύ** μέχρι 33 χαρακτήρες. Χρησιμοποιήθηκε λοιπόν αυτό το πεδίο για την αποφυγή λαθών (θυμηθείτε ότι το πεδίο attribute1 μπορεί να περιέχει το πολύ 33 χαρακτήρες).

SPECIAL_COMMAND

Syntax: *SPECIAL_COMMAND*TARGET*COMMAND_BODY*ATTRIBUTE1*ATTRIBUTE2*

Σκοπός: Κάθε συσκευή έχει διαφορετικό σκοπό για αυτήν την Εντολή και η μορφή και η σημασία του κάθε πεδίου είναι εξαρτημένη με την κάθε συσκευή. Ένα παράδειγμα αυτής θα αναλυθεί στο κεφάλαιο όπου παρουσιάζουμε το SDH (Υδρα) που αναπτύχθηκε στο πλαίσιο αυτής της εργασίας.

Εστία-Ύδρα Πρωτότυπες Συσκευές

Χρησιμοποιώντας την τεχνολογία που αναπτύχθηκε στα προηγούμενα κεφάλαια υλοποιήθηκαν 2 συσκευές, η μία είναι μια μορφή Κύριας Συσκευής με τη δυνατότητα ελέγχου GSM MODEM. Στη συσκευή αυτή δόθηκε το κωδικό όνομα «Εστία», ενώ η δεύτερη συσκευή με κωδικό όνομα «Ύδρα» είναι μια υλοποίηση των Simple Device Hubs, με δυνατότητα να ελέγχει απλές ηλεκτρονικές συσκευές με εισόδους – εξόδους TTL και I2C.

Οι συσκευές αυτές κατασκευάστηκαν βασισμένες στα πρότυπα που περιγράφηκαν στο κεφάλαιο «Αρχιτεκτονική» και η καθεμιά υλοποιεί τα αντίστοιχα σενάρια λειτουργίας. Σκοπός τους είναι να δείξουν τη λειτουργικότητα της παρούσας εργασίας και να αποτελέσουν τα πρώτα παραδείγματα συσκευών που δουλεύουν με το πρωτόκολλο SDMP.

Ιδιαίτερο ενδιαφέρον παρουσιάστηκε στην πορεία ανάπτυξης και κατασκευής της «Ύδρα» καθώς η συγκεκριμένη συσκευή παρουσιάζει ιδιότητες που υπερβαίνουν το πλαίσιο του αυτοματισμού σπιτιών – τοποθεσιών και μπορούν να είναι επιθυμητές σε πολλές εφαρμογές συλλογής δεδομένων. Η μεταγωγή των δεδομένων από απλές ηλεκτρονικές συσκευές είναι δύσκολη και με τις περισσότερες παρούσες εφαρμογές είναι και ακριβή (Ethernet sensors).

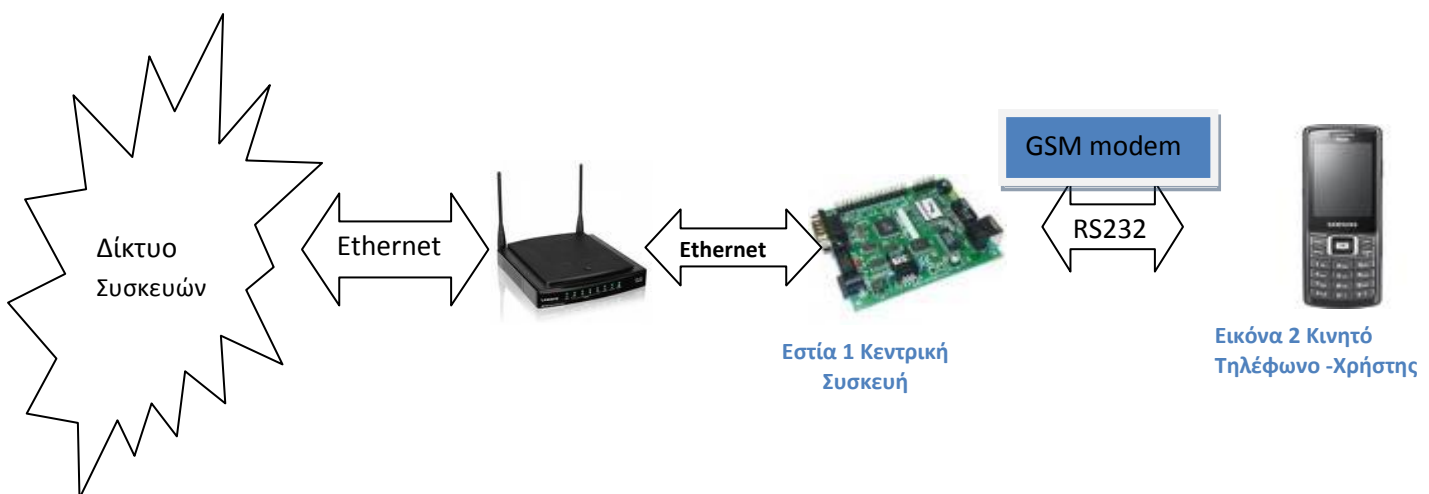
Αναλυτική περιγραφή ακολουθεί για την κάθε συσκευή στο αντίστοιχο κεφάλαιο που ακολουθεί.

Εστία -GSM Interface A Master Device Implementation



Η «Εστία» είναι μια συσκευή, ομώνυμη με την τεχνολογία που αναπτύχθηκε σε αυτήν την εργασία, βασισμένη στο webcat project. Πρόκειται για ένα system on a chip με τον μικροελεγκτή atmega128 και τον ελεγκτή solid state RTL 8019AS (Realtek), έχει επίσης μια rs232 θύρα επικοινωνίας που θα χρησιμοποιηθεί για τον έλεγχο του GSM modem.

Χρησιμοποιώντας το λειτουργικό πραγματικού χρόνου NutOS, υλοποιήθηκε μια **Κεντρική Συσκευή** με δυνατότητες **gsm control**, σκοπός της οποίας είναι να καταχωρεί τις υπόλοιπες συσκευές του δικτύου στον πίνακα Συσκευών. Επιπρόσθετα, πραγματώνει τα σενάρια λειτουργίας όπως αυτά περιγράφηκαν στο κεφάλαιο «Αρχιτεκτονικής» της συγκεκριμένης τεχνολογίας (Εστία).



Ο σκοπός αυτής της συσκευής όπως αναπτύχθηκε σε προηγούμενα κεφάλαια είναι να λειτουργήσει σαν Κεντρική Συσκευή της τεχνολογίας ΕΣΤΙΑ, και ταυτόχρονα να είναι ο

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet διαμεσολαβητής μεταξύ του IP layer και του GSM Network. Όπως είναι προφανές ο φόρτος εργασίας που καλείται να εκτελέσει αυτή η συσκευή είναι μεγάλος, με τη χρήση όμως κάποιων λογικών περιορισμών κατέστη επιτυχής η κατασκευή της με στην πλατφόρμα webcat.

Η «Εστία» υλοποιεί όλα τα βήματα των Κεντρικών Συσκευών της τεχνολογίας ΕΣΤΙΑ και υποστηρίζει το πρωτόκολλο SDMP, με όλες τις βασικές εντολές και τους ακόλουθους περιορισμούς.

Περιορισμοί Κεντρικής Συσκευής «Εστία»

1. Υποστηρίζει 4 Χρονικές Εντολές
2. Τα user group μπορούν να έχουν μέχρι 8 συσκευές
3. Τα ειδικά group τύπων συσκευών, SENSORS, ON_OFF μπορούν να έχουν μέχρι και 16 συσκευές
4. Τα δεδομένα κάθε συσκευής πρέπει να αναφέρονται με χρήση της εντολής SEND_DATA, χωρίς να ξεπερνούν τους 40 χαρακτήρες
5. Οι ρυθμίσεις των group δεν αποθηκεύονται στη μη μεταβαλλόμενη μνήμη
6. Οι υπάρχουσες συσκευές που έχουν καταχωρηθεί στον Εσωτερικό Πίνακα Συσκευών δεν αφαιρούνται περιοδικά και σε περίπτωση αποσύνδεσης τους πρέπει να ενημερώσουν την «Εστία» με χρήση της εντολής REMOVE_ME
7. Σε περίπτωση δυσλειτουργίας του GSM Modem, δεν παρέχεται καμία εγγύηση για την ορθή λειτουργία του συστήματος
8. Τα δεδομένα των συσκευών δεν αποθηκεύονται στην μη μεταβαλλόμενη μνήμη.
9. Για τη σωστή λειτουργία του δικτύου πρέπει το initialization της «Εστία» να προηγείται της έναρξης λειτουργίας των άλλων συσκευών
10. Συνολικά οι συνδεδεμένες σε αυτήν συσκευές προτείνεται να μην υπερβαίνουν τις 32

Καινούργιες Δυνατότητες και Ειδικές λειτουργίες της «Εστία»

- 1 Σε περίπτωση τηλεφωνικής κλήσης, απάντηση status συνδεδεμένων εντολών με sms στον χρήστη
- 2 Η SPECIAL COMMAND προωθείται όπως είναι στην συσκευή που αναφέρεται στο πεδίο attribute2, χωρίς να γίνει οποιαδήποτε επεξεργασία

Αυτοί οι περιορισμοί προκύπτουν από τις περιορισμένες δυνατότητες της συσκευής webcat, λεπτομερώς αυτές αναφέρονται στο ανάλογο παράρτημα. Συνοπτικά ο κυριότερος λόγος που επέβαλε την ανάγκη για τους παραπάνω περιορισμούς είναι η περιορισμένη eeprom 4kb, γεγονός που απαγορεύει την αποθήκευση πολλών δεδομένων όπως πρόκειται για τις χρονικές εντολές. Κάθε χρονική εντολή χρειάζεται 256 bytes δεδομένων κατά την διαδικασία αποθήκευσης της και ανάλογα τα groups για να οριστεί πλήρως κάθε συσκευή, οι απιτήσεις της καυμαίνονται ανάλογα το μέγεθος του alias (μέχρι και 33 χαρακτήρες) και το μέγεθος του ονόματος του τύπου της (33 χαρακτήρες) μέχρι και 100 bytes αν σκεφτούμε ότι και τα δεδομένα είναι 40 χαρακτήρες σε αυτήν την υλοποίηση κεντρικής συσκευής. Είναι λοιπόν αδύνατον να αποθηκευτούν λόγω έλλειψης χώρου τα παραπάνω δεδομένα. Ένας ακόμα περιορισμός που δεν ήταν ωστόσο τόσο καθοριστικός όσο η έλλειψη σταθερής μνήμης είναι η περιορισμένη επεξεργαστική ισχύς της πλατφόρμας webcat.

Συμπεριφορά Σε Κανονικές Συνθήκες Λειτουργίας

Όταν λέμε κανονικές συνθήκες λειτουργίας εννοούμε ότι τα επιμέρους τμήματα δουλεύουν άρτια (gsm modem, router, SDH) και δεν παραβαίνουμε τους προαναφερθείς περιορισμούς.

Η συμπεριφορά της συσκευής σε αυτήν την κατάσταση είναι αρκετά ικανοποιητική με γρήγορη απόκριση στα εισερχόμενα μηνύματα και εκτέλεση τους μέσα σε αποδεκτά χρονικά όρια τις τάξης των μερικών δευτερολέπτων. Δεδομένης της διαδρομής της εντολής από το κινητό τηλέφωνο ως την εκτέλεση της πιθανώς σε κάποια συσκευή στο δίκτυο, είναι πολύ καλή απόκριση. Να τονίσουμε εδώ ότι η απόκριση και γενικά η άρτια συμπεριφορά δεν φαίνεται να επηρεάζεται ιδιαίτερα από το πλήθος των συνδεδεμένων σε αυτήν συσκευών. Στα test που πραγματοποιήθηκαν ήταν μέχρι και 14 συσκευές συνδεδεμένες. Υπό αυτές τις συνθήκες, η κατάσταση της RAM ήταν από τα 32768 συνολικά διαθέσιμα bytes να είναι ελεύθερα τα 16580, με μια διακύμανση φυσικά. Σε φυσιολογικές συνθήκες εργασίας, η χαμηλότερη μέτρηση σε συνολικό χρόνο παρακολούθησης της συσκευής μιας εβδομάδας ήταν 14957 bytes.

Test workload

- 14 registered devices
- 2 χρονικές εντολές με περίοδο 2 λεπτών
- 1 user group με 4 από τις προηγούμενες συσκευές καταχωρημένες
- Εξυπηρέτηση και εισερχόμενων μηνυμάτων από τον χρήστη
- Οι συνδεδεμένες συσκευές κάνανε αναφορά κάθε 2 λεπτά
- Ορισμένες προκαλούσαν και extra network traffic on demand

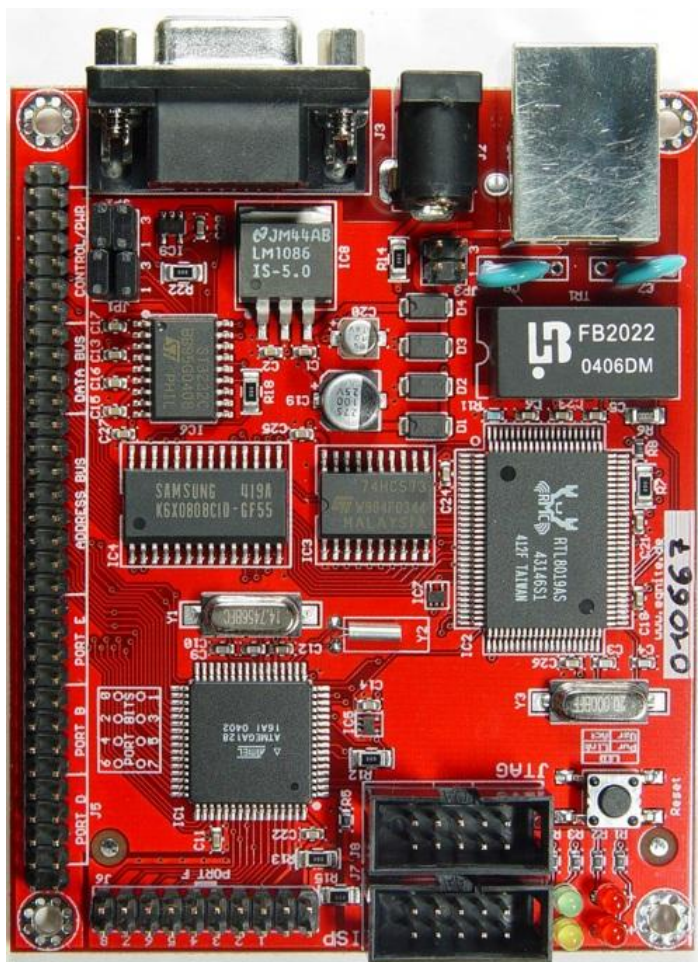
Average response time to gsm messages: 7,8 seconds

Average response time to Ethernet messages: 0,5 seconds.

Average Free Memory $\approx$ 15600 bytes

Eeprom $\approx$ 2800 bytes

«Υδρα» A Simple Device Hub Implementation



Η δεύτερη συσκευή που κατασκευάστηκε στο πλαίσιο της εργασίας είναι η συσκευή με το κωδικό όνομα «Υδρα» και πρόκειται για μια υλοποίηση του πρότυπου Simple Device Hub. Είναι δηλαδή μια πλατφόρμα πάνω στην οποία συνδέονται απλές ηλεκτρονικές συσκευές (βλ. κεφ. «Αρχιτεκτονική») των τεχνολογιών I2C ή αλλιώς TWI (Two wire Interface) και απλές TTL. Δεν θα ήταν δύσκολο να υποστηριχθούν και SPI devices και είναι πιθανό σε επόμενα revisions να συμπεριληφθούν και αυτές.

Οι συνδέσεις των συσκευών αυτών γίνονται στα pins: PortB 0-7, PORTE 7,4,5, PortD 2,3 και στα Pins D0 (SCL) PinD1(SDA) συνδεουμε τις I2C συσκευές. Τα Pins του PortB είναι όλα ορισμένα σαν switches δηλαδή μπορούν να οδηγήσουν συσκευές που δέχονται σαν είσοδο απλό TTL σήμα τάσης 5V και εντάσεις όχι μεγαλύτερης από 20 mA.

Τα υπόλοιπα pins, πέραν αυτών που χρησιμοποιούνται για την επικοινωνία με τις I2C συσκευές, είναι ορισμένα σαν TTL Inputs και συγκεκριμένα Interrupt Driven Logic Level Change. Η χρησιμότητα αυτών είναι να λειτουργούν ως Sensors π.χ. Motion Sensors, Fire Detectors και γενικά για λειτουργίες που η τεχνική polling δεν είναι επαρκής και θέλουμε άμεση δράση στον εντοπισμό κάποιου hazard. Συνολικά δηλαδή έχουμε 13 TTL devices ή ισοδύναμες αυτών και μέχρι και 127 i2c devices, θεωρητικά πάντα. Όπως θα δούμε στην συνέχεια κάθε συνδεδεμένη απλή ηλεκτρονική συσκευή με την τεχνολογία ΕΣΤΙΑ μετασχηματίζεται σε μια ETHERNET Device από την πλατφόρμα «Υδρα» και αυτό «κοστίζει» περίπου 100bytes RAM για κάθε συσκευή. Σε περίπτωση που συνδέσουμε τόσες I2C devices θα εξαντλήσουμε την περιορισμένη μνήμη αυτής της πλατφόρμας. Το πραγματικό όριο των I2C devices για αυτή την υλοποίηση είναι 16 συσκευές, για να έχουμε και την απόδοση που επιθυμούμε.

Σημείωση : Για περισσότερες πληροφορίες για τα Ports, τα Pins και τα Interrupts ανατρέξτε στο manual του atmega128, καθώς τα διαθέσιμα pins αυτού του SoC (System On a Chip) είναι του atmega128.

Οι συσκευές με τις οποίες συνδέθηκε η «Υδρα», δοκιμάστηκε επιτυχώς κατά την διάρκεια του development και εν τέλει χρησιμοποιήθηκαν στην παρουσίαση λειτουργικότητας της συσκευής είναι:

- PIR sensor, TTL signal on Detection Frenshel Lens attached.



- Optical Relay Switches (boards of 2 and 4 Relays) ttl 125 VAC on load.



driven

- Dallas DS1621 Thermometer-Thermostat I2C, on Threshold TTL Signal .



pass

\

Ειδικές Εντολές Και Ρυθμίσεις της S.D.H “Υδρα”

Χρησιμοποιώντας μηνύματα SDMP ο χρήστης έχει πρόσβαση τόσο στην ΎΔΡΑ όσο και στις λειτουργίες των συνδεδεμένων σε αυτήν συσκευών. Οι παρακάτω λειτουργίες είναι διαθέσιμες στον χρήστη:

- Αλλαγή κατάστασης όλων των switches (8 switches στο PortB), είτε διακριτά ένα switch την φορά, είτε περισσότερα του ενός με τη χρήση groups
- Αναφορά κατάστασης switch (ON-OFF)
- Ανάθεση στους Trigger Sensors, (PINS E4,5,7 D2,3) οποιασδήποτε εντολής του SDMP, που πραγματοποιείται κάθε φορά που γίνεται Triggered ανάλογο Interrupt
- Δημιουργία μέχρι και 4 Χρονικών Εντολών προς ικανοποίηση περιοδικών διεργασιών (π.χ. άνοιξε το switch 2 (θερμοσίφωνας) στις 20:00 κάθε μέρα)
- Αλλαγή του alias της συγκεκριμένης Ύδρας
- Αναφορά status i2c thermometer
- Ρύθμιση i2c θερμοστάτη (trigger threshold)

Με τη χρήση των παραπάνω ο χρήστης την πρώτη φορά λειτουργίας πρέπει να ρυθμίσει την συσκευή. Σε περίπτωση μη ρύθμισης υπάρχουν default settings, με βάση τα οποία θα λειτουργήσει αυτή η συσκευή.

Προεπιλεγμένες Ρυθμίσεις «Default Settings»

- Alias Hydra
- Trigger Sensors, On trigger Execute -
>*SEND_SMS*A*[SENSOR_ALIAS]Triggered*owner*Estia*
- Καμία χρονική εντολή
- Θερμοστάτης i2c στους 90° C
- Όλα τα Switches off

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet Σύνταξη Εντολών SDMP για τον έλεγχο της SDH «Υδρα»

Χρησιμοποιώντας εντολές SDMP ο χρήστης μπορεί να ρυθμίσει τη λειτουργία των εσωτερικών συσκευών σύμφωνα με τις ανάγκες του. Συγκεκριμένα, μπορεί να επηρεάσει την πράξη που εκτελούν τα *trigger sensors* έτσι ώστε να εκτελείται διαφορετική *action* (πράξη) κατά περίπτωση. Επίσης μπορεί να ρυθμίσει την συχνότητα που θα γίνεται *trigger* σήμα από οποιονδήποτε *trigger sensor* μεταβάλλοντας το χρονικό όριο επανάληψης. Μπορεί δηλαδή να θέσει ότι δεν θα γίνεται *trigger* περισσότερο από μια φορά στα 60 δευτερόλεπτα. Φυσικά κάθε αισθητήρας μπορεί να έχει διαφορετική ρύθμιση από τους άλλους, αφού οι ρυθμίσεις του χρηστή αποθηκεύονται στη σταθερή μνήμη (*eprom*) και σε περίπτωση επανεκκίνησης αντικαθιστούν τις *default* πράξεις. Επίσης ο χρήστης μπορεί να ρυθμίσει, όπως αναφέρθηκε και πρωτύτερα, το όριο του θερμοστάτη I2C κατά βούληση φυσικά μέσα στα επιτρεπτά από την συσκευή όρια.

Στην περίπτωση των αισθητήρων η εντολή είναι η «*SEND_DATA*» και πρέπει να έχει την ακόλουθη σύνταξη:

SEND_DATA*A*[ACTION_TO_BE_PERFORMED?TIME_THRESHOLD?]*SENSOR_ALIAS

Η εντολή που βρίσκεται στο πεδίο *commandBody*, με τη χρήση της διαίρεσης πεδίων και τον δεύτερο ειδικό τελεστή '&' γράφεται στην ακόλουθη μορφή:

&COMMAND_NAME&COMMAND_TARGET&COMMAND_BODY&ATTRIBUTE1&ATTRIBUTE2&

Η μέγιστη επιτρεπόμενη συχνότητα έπεται της εντολής στο πεδίο *body*, πρέπει να περικλείεται από το διαχωριστή '?' και η μονάδα μέτρησης είναι τα δευτερόλεπτα.

?SECONDS?

Ένα παράδειγμα χρήσης είναι το ακόλουθο:

SEND_DATA*A*&SEND_SMS&A&Intruder-Location:MainHall&&?3600?&*Estia*Hydr_SN1

Επεξήγηση: Η παραπάνω εντολή θα έχει ως αποτέλεσμα οπότε ο αισθητήρας SN1 εντοπίζει κίνηση, να αποστέλλεται ένα μήνυμα στον χρηστή με το κείμενο **Intruder-Location: MainHall**. Η ελάχιστη περίοδος σύμφωνα με την εντολή είναι 3600 δευτερόλεπτα, που σημαίνει ότι δε θα σταλεί πάνω από ένα μήνυμα ανά ώρα σε περίπτωση συνεχόμενων εντοπισμών.

Σημείωση: Η SDH «Υδρα» αναθέτει στις εσωτερικές τις συσκευές ένα όνομα (*alias*) για να είναι προσβάσιμες. Το όνομα στη συγκεκριμένη υλοποίηση προκύπτει με την ακόλουθη ονοματολογία: First 4 letters of Device Alias+'_'+[SN|SW]_[Number of Device] DeviceAlias είναι το όνομα της SDH Υδρας, το οποίο έχει την αρχική τιμή «Hydra». Ο χρήστης μπορεί να αλλάξει το όνομα της συσκευής χρησιμοποιώντας την εντολή **SET_DEVICE_INFO**.

SN=Από την λέξη Sensor, δίδεται στους αισθητήρες

SW= Από την λέξη Switch, δίδεται στα TTL Switches (θυμηθείτε PortB)

Όσον αφορά τον I2c θερμοστάτη η εντολή και η ιδέα είναι η ίδια:

ΕΣΤΙΑ

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet

*SEND_DATA*A* [&VALUE 1&VALUE2&]**Hydr_SN_6*

Η τιμή είναι μια απλή δεκαδική τιμή π.χ. 26. Η μικρότερη τιμή από της 2 είναι η ελάχιστη τιμή και η μεγαλύτερη η μέγιστη θερμοκρασία.

Οποιαδήποτε από τις βασικές εντολές SDMP μπορεί να γίνει πράξη εκτέλεσης του αισθητήρα, ακολουθώντας πάντα τους περιορισμούς τους μεγέθους των πεδίων που έχουν αναφερθεί στην ενότητα SDMP PROTOCOL.

Προβλήματα Κατά την Υλοποίηση

Κατά τον σχεδιασμό και ανάπτυξη αυτής της τεχνολογίας υπήρξαν ,όπως πάντα, κάποιες δυσκολίες και προκλήσεις,σημαντικότερες εκ των οποίων ήταν, η έλλειψη εργαλείων επισκόπησης της πληροφορίας και η έλλειψη εφαρμογών προσομοίωσης και αποσφαλμάτωσης.

Στην πορεία ενός μηνύματος απο το κινητό του χρήστη έως και την συσκευή εργάτη ,μπορούμε να διαγράψουμε τα εξής στάδια : πύλη του διακομιστή δικτύου GSM-GPRS(πχ cosmote,vodafone) ,GSM modem,Κεντρική Συσκευή "Εστια", dhcp router ,συσκευή εργάτη(πχ "υδρα") και τέλος η απλή ηλεκτρονική συσκευή που είναι συνδεδεμένη στην "Υδρα" όπως το relay το οποίο και εν τέλει ,αν πρόκειται για εντολή SWITCH_ON, ανάβει.

Όπως φαίνεται από την παραπάνω περιγραφή η πορεία που ακολουθεί η πληροφορία ,όχι μόνο είναι εκτενέστατη αλλά,περιλαμβάνει και πολλές διαφορετικές οντότητες,η αδυναμία λοιπόν ,λόγω της πολυπλοκότητας αυτής ήταν ένα μεγάλο πρόβλημα καθώς ήταν ιδιαίτερα δύσκολο να εντοπισθεί όταν κάτι πήγαινε λάθος ποιος κρίκος στην παραπάνω αλυσίδα ήταν προβληματικός.

Το επόμενο ,αλλά όχι λιγότερο σημαντικό πρόβλημα ,ήταν η έλλειψη εργαλείων αποσφαλμάτωσης, όπως είχε αναφερθεί και στο κεφάλαιο Λογισμικό Σχεδίασης,δεν υπήρχε η δυνατότητα που δίνεται σε ανάλογες εφαρμογές PC,καθώς το πρόγραμμα που αναπτύσσαμε ήταν για άλλη αρχιτεκτονική και μάλιστα για ειδική ολοκληρωμένη πλακέτα που μας απέτρεπε να χρησιμοποιήσουμε το λογισμικό της Altera!

Η αποσφαλμάτωση έγινε εξ ολοκλήρου χρησιμοποιώντας debug output στην rs232 με χρήση formatted output(printf ,puts) και η προσομοίωση ,υλοποιώντας ένα πρόγραμμα οδηγό σε C για αρχιτεκτονική 8086,που βοήθησε μόνο στον συγχρονισμό των threads και για να ανακαλυφθούν και να διορθωθούν τα λογικά λάθη και για την ωρίμανση κάποιων αλγορίθμων(πργραμματιστικών λαθών).Όμως συμπεράσματα απόδοσης συμπεριφοράς και προβλήματα που προέκυψαν λόγω διαφοράς λειτουργικού και υλικού δεν μπορούσαν να εντοπισθούν απο την παραπάνω διαδικασία.

Τέλος, κάποια προγραμματιστικά προβλήματα που αντιμετωπίστηκαν ,και προέκυψαν ακριβώς λόγω της αδυναμίας του μοντέλου προσομοίωσης να εκφράσει επακριβώς το πραγματικό λογισμικό όπως αυτό παράγεται από τον cross compiler μετά απο linking και με τις βιβλιοθήκες του NutOS ήταν τα εξής:

"Κρυφές" Βελτιστοποιήσεις avr-gcc ,λόγω του περιορισμένου μεγέθους της μνήμης flash ,που είναι η μνήμη που προγραμματίζεται με την εφαρμογή,χρησιμοποιήθηκε η βελτιστοποίηση του παραγόμενου κώδικα ως προς το μέγεθος(flag -Os),αυτό έχει σαν αποτέλεσμα ο μεταγλωττιστής να αφαιρεί κομμάτια κώδικα όποτε μπορεί κάνοντας κάποιες υποθέσεις ,συγκεκριμένα στην περίπτωση των interrupts υπήρχαν μεταβλητές που αλλάζαν την τιμή τους μόνο από το κομμάτι κώδικα του interrupt handler,ο μεταγλωττιστής θεωρούσε λοιπόν ότι εφόσον δεν αλλάζουν τιμή μέσα στο κυρίως πρόγραμμα μπορεί με ασφάλεια να αντικατασταθεί τις μεταβλητές αυτές με σταθερές,αυτό είχε ως αποτέλεσμα ορισμένες διακλαδώσεις να μην πέρνωνται ποτέ καθώς η τιμή της μεταβλητής δεν άλλαζε ποτέ(είχε αντικατασταθεί στο στάδιο μεταγλώττισης από σταθερά) η λύση στο πρόβλημα αυτό ήταν η χρήση της ειδικής λέξης volatile που ενημερώνει τον μεταγλωττιστή οτι αυτή η μεταβλητή ενδέχεται να αλλάξει εκτός του κυρίως προγράμματος.

Ένα άλλο πρόβλημα ήταν ότι το λειτουργικό σύστημα για να ξεκινήσει καλούσε μια συνάρτηση με όνομα main, ένα πρόγραμμα σε c όμως μπορεί να έχει μόνο μια main, οπότε η δική μας εφαρμογή δε μπορούσε να κάνει link με τις βιβλιοθήκες του λειτουργικού διότι υπήρχε ήδη main,αυτό λύθηκε χρησιμοποιώντας την ειδική εντολή -defsymbol main 0 .

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet
Άλλα προβλήματα που αντιμετωπίστηκαν ήταν η πρόβλεψη heap memory από τον μεταγλωτιστή, αυτό οδηγούσε σε out of heap error, και οδηγούσε το σύστημα σε επανεκκίνηση, αυτό αντιμετωπίστηκε γράφοντας μικρότερες συναρτήσεις και καλώντας τις σειριακά, περιορίζοντας επίσης στο ελάχιστο το βάθος κλήσεων ανά συνάρτηση (το πολύ 2)

Επίλογος – Συμπεράσματα

Κατά τη διάρκεια ανάπτυξης αυτής της τεχνολογίας υπήρξαν πολλά προβλήματα μέσα από τα οποία δημιουργήθηκε ένα προϊόν, το οποίο καταφέρνει να αντιμετωπίσει τα προβλήματα που ξεκινώντας αυτήν την εργασία είχαμε θέσει ως στόχους.

Το κόστος συγκεκριμένα, ένας πολύ σημαντικός παράγοντας που η αυτοματοποίηση σπιτιών δεν έχει προχωρήσει σε μεγάλο βαθμό, έπεσε δραματικά ανά συσκευή. Αναλυτικά, το κόστος αυτοματοποίησης για ένα μεσαίο σπίτι είναι:
 Ethernet 1.3h (80) + 8 * Optical Relays 250 VAC (8*3=24) + 5 * TTL triggerSensor(avg(6)*5=30 + 1 i2c Termostat Ds1621 (2) =80+24+40+2=142 Euro

Προσθέτοντας GSM δυνατότητες χρειαζόμαστε τα εξής :
 Ethernet 1.3h(80) + TELITGSM862 (150)=230Euro.

Συνολικό κόστος :230+142=372 Euro.

Με τη χρήση της τεχνολογίας «ΕΣΤΙΑ» και τον παραπάνω εξοπλισμό έχουμε δυνατότητα ελέγχου όλων των ανωτέρω συσκευών μέσω Ethernet και GSM. Επιπρόσθετα, αν το router είναι συνδεδεμένο στο Internet τότε μπορούμε να έχουμε πρόσβαση και μέσω web interface.

Αν προσπαθήσουμε να πετύχουμε την παραπάνω λειτουργικότητα με άλλα προϊόντα του χώρου η τιμή θα είναι σε κάποιες περιπτώσεις μιας τάξης μεγέθους μεγαλύτερη. Ενδεικτικά προσπαθώντας να έχουμε τον ίδιο αριθμό συσκευών με την τεχνολογία και τα προϊόντα της industriaethernet, το κόστος ανέρχεται:

SensorProbe8 642\$

NetDio-Digital I/O 143\$

Temperature Sensor 75\$

2 Motion Detector Sensors 250\$

3 OtherSensors(average price of any 3 sensors)250\$

Σύνολο :1360\$ σε ευρώ :816

Σημειώνεται ότι το κόστος αυτό είναι χωρίς την δυνατότητα ελέγχου μέσω sms, αν και sms notification υποστηρίζεται από την συγκεκριμένη τεχνολογία μέσω του ιστοχώρου.

Το κόστος είναι ακόμα μεγαλύτερο αν πάμε στις πολλές διαφορετικές υλοποιήσεις που υπάρχουν και διαχειρίζονται μόνο ένα αισθητήριο ή διακόπτη (relay) με κάθε gsm modem. Ένας μέσος όρος των 150 ευρώ ανά συσκευή εκτινάσσει το κόστος, για τον ίδιο αριθμό συσκευών, στα 2100 ευρώ χωρίς να υπολογίσουμε τα υπόλοιπα έξοδα (π.χ. μια sim για κάθε gsm modem κλπ), και χωρίς φυσικά να δίνεται η δυνατότητα χειρισμού μέσω Ethernet.

Η λειτουργικότητα της τεχνολογίας που αναπτύχθηκε δεν εξαντλείται μόνο στο πεδίο του αυτοματισμού σπιτιών, αλλά βρίσκει εφαρμογή και σε πεδία όπως η συλλογή δεδομένων από αισθητήρια και η μεταφορά τους μέσω δικτύου σε έναν απομακρυσμένο υπολογιστή. Λόγω του πολύ χαμηλού κόστους της «Υδρας» και των μεγάλων ωστόσο δυνατοτήτων (θυμηθείτε χρονοπρογραμματισμός εντολών, αλλαγή πράξεων trigger sensor κ.λ.π.), θα μπορούσε να δημιουργηθεί ένα δίκτυο από τέτοιες πλατφόρμες συλλογής δεδομένων και με την χρήση του πρωτοκόλλου SDMP να γίνεται ένα interfacing με εφαρμογές τρίτων κατασκευαστών, είτε για συλλογή και περαιτέρω επεξεργασία των δεδομένων, είτε και με άλλες ενσωματωμένες συσκευές.

Τέλος, ένα ακόμα προτέρημα της «ΕΣΤΙΑ» έναντι των περισσότερων άλλων προσεγγίσεων, είναι η χαμηλή της κατανάλωση, αφού αυτή δεν ξεπερνάει σε καμία

Αυτοματισμός Ελεγχόμενος Μέσω Κινητού Τηλεφώνου Και Ethernet περίπτωση τα 12watt(στα καινούργια μοντέλα δε 1.3h η κατανάλωση είναι μικρότερη των 4 watt). Αυτό δίνει τη δυνατότητα στις συσκευές που κατασκευάστηκαν (βλ. Εστία-Υδρα) να μπορούν να λειτουργήσουν από ηλιακά panels και άλλες μορφές ανανεώσιμων πηγών ενέργειας.

Μελλοντική Εργασία

Κατασκευή ενός web-interface για έλεγχο μέσω Internet. Προτεινόμενη τεχνολογία JSF,Richfaces.

Μελέτη και σχεδιασμός τροφοδοσίας των συσκευών,Υδρα και Εστία, μέσω ανανεώσιμων πηγών ενέργειας.

Περαιτέρω ανάπτυξη του πρωτοκόλλου SDMP με περισσότερες εντολές και υποστήριξη streaming data.

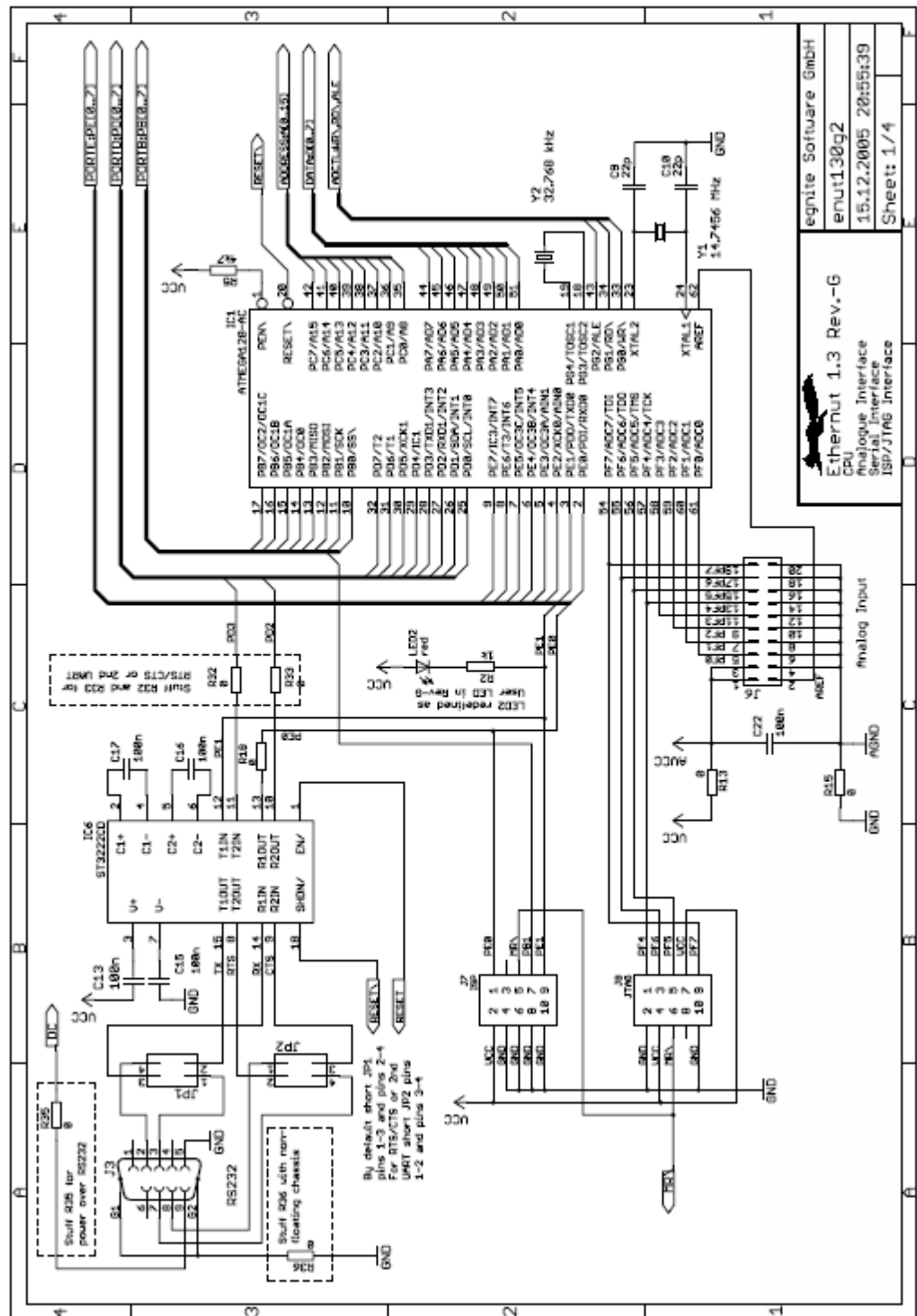
Υλοποίηση υποστήριξης του πρωτοκόλλου 3-wire και 1-wire, επεκτείνοντας έτσι την γκάμα συσκευών που είναι στην διάθεση μας. Η αλλαγή αυτή γίνεται εξελίσσοντας το βασικό κώδικα της «Υδρας» και γενικά των SDH (Simpe Device Hubs).

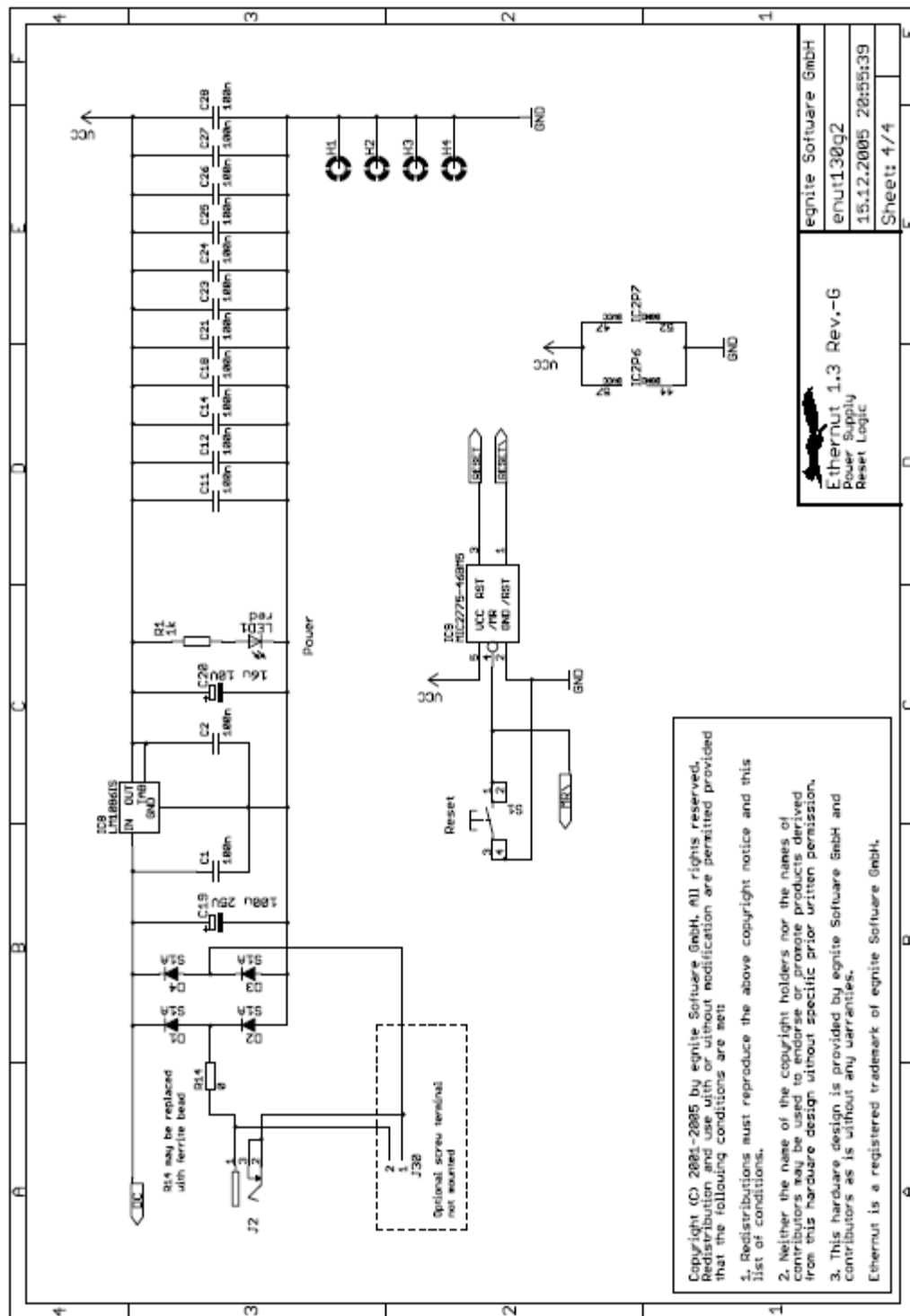
Κατασκευή μιας ενδιάμεσης συσκευής για χρήση σε εφαρμογές περιορισμένων απαιτήσεων , για ελαχιστοποίηση κόστους.Συγκεκριμένα σε εφαρμογές που οι ανάγκες δεν ξεπερνούν τα 8 relay και συνολικά τις 16 συσκευές , θα μπορούσε να δημιουργηθεί μια Κ.Σ βασισμένη στην «Εστία» συνδιάζοντας σε αυτήν και την λειτουργικότητα της «Υδρας» .

Appendix A. Ethernut 1.3h Schematics-Specs

Specs

- ATmega 128 RISC microcontroller with up to 16 MIPS throughput
- Full duplex IEEE 802.3 compliant 10 Mbps Ethernet controller with on-board RJ-45 connector
- Two serial ports, one RS-232 at DB-9 connector
- 128 kByte in-system programmable Flash ROM
- 4 kByte in-system programmable EEPROM
- 32 kByte external SRAM
- 22 programmable digital I/O lines
- 8-channel, 10-bit analog/digital converter
- Two 8-bit and two 16-bit timers/counters
- Watchdog timer for enhanced reliability
- LED indicators for power supply and Ethernet activity
- Single power supply DC 9-12V





Περισσότερες πληροφορίες :

<http://www.ethernut.de/en/hardware/enut1>

Βιβλιογραφία

Λειτουργικό NutOs : <http://www.ethernut.de/en/software/index.html>

Ethernut 1.3.h : <http://www.egnite.de/en/produkte/ethernut-familie/ethernut-13.html>

IndustrialEthernet devices : <http://www.industrialethernet.com/ipmonitoring.html>

IObridge : www.iobridge.com

AtMega128 : www.atmel.com

«

«Υδρα» A Simple Device Hub Implementation, 36

A

Abstract, 8

Appendix A. Ethernut 1.3h Schematics-Specs, 45

G

GSM Controlled Devices, 6

S

Simple Device Messaging Protocol (SDMP), 20

A

Αρχιτεκτονική, 15

B

Βασικές Εντολές, 23

E

Επίλογος – Συμπεράσματα, 43

Εργαλεία Σχεδίασης, 12

Εστία Επισκόπηση Λειτουργίας, 17

Εστία-Υδρα Πρωτότυπες Συσκευές, 32

Λ

Λειτουργικό Σύστημα NutOS, 14

M

Μελλοντική Εργασία, 44

Π

Προβλήματα Κατά την Υλοποίηση, 41

Προεσκόπηση Πρωτοκόλλου, 21

Πρόλογος, 5

Πρωτόκολλο Επικοινωνίας Συσκευών, 19

Σ

Σύνθετες Εντολές, 24

Σύνταξη Εντολών SDMP για τον έλεγχο της SDH «Υδρα», 39

Συσκευές και Υλικό Προτύπων, 11

Σχεδιασμός, 9

