

Ανάπτυξη Δικτύου Ελέγχου Κτιρίων με το Πρότυπο LON
για Εξοικονόμηση Ενέργειας

Σιδεράτος Ιωάννης

Πολυτεχνείο Κρήτης
Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

14 Ιανουαρίου 2012

Εγκρίθηκε από την τριμελή επιτροπή:

1. Καθ. Καλαϊτζάκης Κωνσταντίνος (Επιβλέπων)
2. Καθ. Σταυρακάκης Γεώργιος
3. Επικ. Καθ. Κολοκοτσά Διονυσία

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Καλαϊτζάκη Κωνσταντίνο καθώς επίσης και την καθηγήτριά μου κα. Κολοκοτσά Διονυσία, για την αμέριστη βοήθεια και την πολύτιμη καθοδήγησή τους, κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την στήριξή τους όλα αυτά τα χρόνια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Ένα δίκτυο ελέγχου αποτελείται από μία ομάδα συσκευών η οποία λειτουργεί με βάση ένα peer-to-peer μοντέλο επικοινωνίας με σκοπό την παρακολούθηση αισθητήρων και ενεργοποιητών ελέγχου. Το σύστημα LONWORKS αποτελεί την κορυφαία λύση για τον αυτόματο έλεγχο σπιτιών, κτιρίων, βιομηχανιών και μεταφορών. Το δίκτυο LONWORKS χρησιμοποιεί το πρωτόκολλο LonTalk, το οποίο είναι επίσης γνωστό και ως ANSI/EIA 709.1 Control Networking Standard με σκοπό την εκτέλεση διαφόρων λειτουργιών. Τα δίκτυα ελέγχου τεχνολογίας LONWORKS βασίστηκαν στις ακόλουθες αρχές :

- Ένα δικτυωμένο σύστημα ελέγχου είναι σημαντικά πιο ισχυρό και πιο ευέλικτο από ένα μη-δικτυωμένο σύστημα ελέγχου.
- Οι επιχειρήσεις μπορούν μακροπρόθεσμα να εξοικονομήσουν περισσότερα χρήματα χρησιμοποιώντας δικτυωμένα συστήματα ελέγχου.
- Τα δικτυωμένα συστήματα ελέγχου μπορούν να αξιοποιήσουν τις δυνατότητές τους για να ελέγξουν και να αντιμετωπίσουν εύκολα νέες εφαρμογές.

Τα LONWORKS δίκτυα διακυμαίνονται μεταξύ μικρών ενσωματωμένων δικτύων σε διάφορες μηχανές, αλλά και μεγάλων δικτύων με χιλιάδες συσκευές, βρίσκοντας εφαρμογή σε τραίνα, αεροπλάνα, κτίρια και εργοστάσια, με κύριο σκοπό βέβαια την μέγιστη δυνατή αξιοπιστία και ευελιξία, αλλά και την ελαχιστοποίηση του κόστους ανάπτυξής τους.

Περιεχόμενα

Εισαγωγή	13
I ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	14
1 Δίκτυα ελέγχου	15
1.1 Επισκόπηση	15
1.2 Η αξία των δικτυωμένων συστημάτων ελέγχου σε πραγματικές εφαρμογές	17
1.3 Η παραδοσιακή προσέγγιση σχεδιασμού συστημάτων ελέγχου	18
1.3.1 Η νέα προσέγγιση των δικτύων ελέγχου	19
1.3.2 Η μετάβαση από τα δίκτυα δεδομένων στα δίκτυα ελέγχου	20
1.3.3 Συστατικά των δικτύων ελέγχου	20
1.4 Χρησιμοποιώντας νέες τεχνολογίες σε ιεραρχικά συστήματα	22
2 Δίκτυα LONWORKS τεχνολογίας	25
2.1 Επισκόπηση	25
2.2 Υλοποιώντας ένα δίκτυο ελέγχου	27
2.2.1 Οικονομικά αποδοτική καλωδίωση του δικτύου	28
2.2.2 Συμβατές συσκευές στο δίκτυο	28
2.2.3 Αποτελεσματικός σχεδιασμός του συστήματος	28
2.2.4 Πρότυπο διαχείρισης του δικτύου	29
2.2.5 Πρότυπο εργαλείων δικτύου	29
2.2.6 Πρότυπο ρύθμισης συσκευών	30
2.2.7 Υποστήριξη IP	30
2.3 Σηματοδοσία μέσω γραμμών μεταφοράς ηλεκτρικής ενέργειας	32

3	Η Πλατφόρμα LONWORKS	35
3.1	Επισκόπηση	35
3.1.1	Το Neuron chip	37
3.1.1.1	Πομποδέκτες	38
3.1.2	Εργαλεία ανάπτυξης	38
3.1.3	Δρομολογητές	39
3.1.4	Διασυνδέσεις δικτύου– Network Interfaces	40
3.1.5	Διαχείριση της πλατφόρμας	41
3.1.5.1	Το λειτουργικό σύστημα του δικτύου	43
3.1.5.2	ISI	43
3.1.6	Εργαλεία δικτύου	44
3.1.6.1	LonMaker Integration Tool	44
3.1.6.2	LonScanner Protocol Analyzer	45
4	Το πρωτόκολλο LonTalk	46
4.1	Επισκόπηση	46
4.2	Φυσικό επίπεδο	49
4.2.1	Τύποι καναλιών	49
4.2.1.1	PL-20	49
4.3	Επίπεδο ζεύξης δεδομένων	52
4.3.1	Πρόσβαση στο μέσο	52
4.3.1.1	Αλγόριθμος πρόσβασης CSMA	52
4.3.1.2	Προτεραιότητα	54
4.3.2	Ρυθμαπόδοση καναλιών	55
4.4	Επίπεδο δικτύου	56
4.4.1	Τύποι διευθύνσεων	56
4.4.1.1	Φυσική διεύθυνση	57
4.4.1.2	Διεύθυνση τομέα	58
4.4.1.3	Διεύθυνση υποδικτύου	58
4.4.1.4	Διεύθυνση κόμβου	59
4.4.1.5	Ομάδες συσκευών	60
4.5	Επίπεδο μεταφοράς	61
4.5.1	Υπηρεσίες μηνυμάτων	61
4.6	Επίπεδο συνόδου	62

4.6.1	Αίτηση-Απόκριση	63
4.6.2	Υπηρεσία ελέγχου ταυτότητας	63
4.7	Επίπεδο παρουσίασης	64
4.7.1	Μηνύματα	64
4.7.2	Μεταβλητές δικτύου	65
4.8	Επίπεδο εφαρμογής	68
4.8.1	Program ID	70
4.9	Σύνοψη πρωτοκόλλου	72
5	Το πρωτόκολλο ISI	74
5.1	Επισκόπηση	74
5.2	Λειτουργία του πρωτοκόλλου	75
5.2.1	Αλγόριθμος Fire-and-Forget	76
5.2.2	Το μοντέλο σύνδεσης	78
5.2.3	Διαχείριση συνδέσεων	80
5.2.3.1	Manual Enrollment	80
5.2.3.2	Controlled Enrollment	81
5.2.3.3	Automatic Enrollment	81
5.2.4	Μηνύματα ISI	82
5.3	Μηχανή ISI	84
5.3.1	Βασικές διαδικασίες	84
6	Η γλώσσα προγραμματισμού Neuron C	88
6.1	Επισκόπηση	88
6.2	Χρονοπρογραμματιστής	90
6.3	Timers	93
6.4	Αντικείμενα εισόδου/εξόδου	93
II	ΥΛΟΠΟΙΗΣΗ	95
7	Ανάλυση του υλικού	96
7.1	Επισκόπηση	96
7.2	Μπλοκ διάγραμμα των PL 3120 / PL 3150 Power Line Smart Transceivers	98
7.2.1	Πυρήνας Neuron	98

7.2.1.1	Μνήμη	100
7.2.1.2	Είσοδος/ Έξοδος	102
7.2.2	Κύκλωμα ζεύξης	106
7.2.3	Τροφοδοσία	111
7.2.4	Κυκλωματικά διαγράμματα	114
7.3	Mini Gizmo I/O board	118
7.4	Σύνδεση δικτύου ελέγχου με τον υπολογιστή	120
8	Ανάλυση της εφαρμογής	122
8.1	Εισαγωγή	122
8.2	Digital Temperature Alarm Sensor	124
8.2.1	Σύνδεση ISI	124
8.2.2	Λειτουργία	128
8.2.2.1	Λήψη δειγμάτων θερμοκρασίας	128
8.2.2.2	Διαχείριση του Piezo Buzzer και των LEDs	132
8.2.2.3	Ελεγκτής	136
8.3	Monitor	140
8.3.1	Σύνδεση ISI	141
8.3.2	Λειτουργία	143
8.4	Διαγράμματα ροής	146
8.5	Πηγαίοι κωδικες	148
	Συμπεράσματα	166
	Βιβλιογραφία	169

Κατάλογος Σχημάτων

1.1	Σύστημα κεντρικής αρχιτεκτονικής	18
1.2	Ανοιχτό δίκτυο ελέγχου	19
1.3	Συστατικά των δικτύων ελέγχου	21
1.4	Σύστημα ιεραρχικής αρχιτεκτονικής	23
2.1	Δίκτυο ελέγχου LONWORKS	26
2.2	Χαρακτηριστικά πομποδεκτών	31
2.3	Γενική σχεδίαση του εύρους συχνοτήτων	33
2.4	Λειτουργία διπλής φέρουσας συχνότητας	33
3.1	Μια άποψη της λειτουργίας του LONWORKS συστήματος βάσει των επιπέδων του πρωτοκόλλου	36
3.2	Το εσωτερικό μιας τυπικής συσκευής ελέγχου LONWORKS	39
4.1	PL-20 φέρουσες συχνότητες	50
4.2	Πομποδέκτες PL3120 και PL3150	51
4.3	Beta 2 Slots	54
4.4	Slots προτεραιότητας και μη-προτεραιότητας	55
4.5	Δύο υποδίκτυα στο ίδιο κανάλι	58
4.6	Δύο υποδίκτυα σε κοινό segment	59
4.7	Τρία υποδίκτυα σε δύο segments	59
4.8	Ομάδες συσκευών	60
4.9	Επιτρεπτό πλήθος συσκευών σε ένα δίκτυο LONWORKS	61
4.10	Σύνδεση δύο μεταβλητών δικτύου	66
4.11	Σύνδεση πολλών μεταβλητών δικτύου	67
4.12	Σύνδεση δύο συσκευών	68
4.13	Functional Profile	70

5.1	Single Assembly	78
5.2	Compound Assembly with 1 functional block	79
5.3	Compound Assembly with 2 functional blocks	79
5.4	Ακολουθία των γεγονότων σε manual και controlled συνδέσεις	86
6.1	Αποστολή μηνυμάτων	89
6.2	When clause	90
6.3	Παράδειγμα λειτουργίας χρονοπρογραμματιστή	92
7.1	Μπλοκ διάγραμμα των PL3120/PL3150 Smart Transceivers	98
7.2	Μια άποψη της αρχιτεκτονικής του πυρήνα Neuron	99
7.3	Λειτουργία των επεξεργαστών κατά τη διάρκεια μιας φάσης ενός minor cycle	100
7.4	To interface των πυρήνων PL 3120 / PL 3150	101
7.5	Κατανομή μνήμης στα PL 3150 και PL 3120	102
7.6	Εσωτερικά κυκλώματα των timer/counters	103
7.7	Τυπικό κύκλωμα της λειτουργίας service	105
7.8	Πιθανή διαδρομή σήματος σε ένα PL δίκτυο	106
7.9	Μοντέλο εξασθένησης	107
7.10	Μοντέλο κυκλώματος ζεύξης	108
7.11	Μοντέλο κυκλώματος ζεύξης Line-to-Neutral	109
7.12	Προσθήκη πυκνωτή C2 στο αρχικό κύκλωμα ζεύξης	109
7.13	Πλήρως λειτουργικό line-to-neutral κύκλωμα ζεύξης	110
7.14	Line-to-neutral κύκλωμα ζεύξης συνδεδεμένο με τροφοδοσία γραμμικού τύπου (wall-plug Tamura Corporation)	113
7.15	Τιμές των στοιχείων του wall-plug coupling circuit	114
7.16	PL 3120 Smart Transceiver	115
7.17	PL 3150 Smart Transceiver	115
7.18	Εσωτερικό ρολόι 10MHz και σήματα δέκτη	116
7.19	Σήματα που αφορούν τη μετάδοση δεδομένων από τους PL Smart Transceivers— I/O objects	117
7.20	I/O connector	118
7.21	Mini Gizmo	119
7.22	Κυκλωματική αναπαράσταση LEDs και Piezzo Buzzer.	119
7.23	Αισθητήρας θερμοκρασίας Dallas DS18S20	120

7.24	Διεπαφή δικτύου U20 USB	121
7.25	Ολοκληρωμένη άποψη του υλικού	121
8.1	Τοποθέτηση των LONWORKS συσκευών σε ένα κτίριο	123
8.2	Functional profile SFPTswitch	125
8.3	Διάγραμμα σύνδεσης αισθητήρα με πομποδέκτη	129
8.4	Χρονικό διάγραμμα μετάδοσης παλμών	130
8.5	Παλμοί reset και presence	131
8.6	Διάγραμμα της συνάρτησης io_out για το μοντέλο εισόδου-εξόδου frequency	133
8.7	Καταχωρητής ολίσθησης με 4 επίπεδα	134
8.8	Διαδικασία ολίσθησης των bits	135
8.9	Χρόνος λειτουργίας της io_out() με το μοντέλο bit	135
8.10	Διαδικασία φόρτωσης του κώδικα της εφαρμογής στο PL3150 Neuron Chip	140
8.11	Functional profile: Open Loop Actuator	141
8.12	Μήνυμα σύνδεσης CSMO/A	143
8.13	Διαδικασία φόρτωσης του κώδικα της εφαρμογής στο PL3120 Neuron Chip	145
8.14	Επιλογή του network interface	146
8.15	Σύνδεση του Monitor με το PL3120 Neuron Chip	146
8.16	Διάγραμμα ροής για το Monitor	147
8.17	Διάγραμμα ροής για τον digital temperature alarm sensor	148

Κατάλογος Πινάκων

4.1	Τα τρία πρώτα επίπεδα του μοντέλου αναφοράς OSI	47
4.2	Τα τέσσερα τελευταία επίπεδα του μοντέλου αναφοράς OSI	48
4.3	Τύποι καναλιών LONWORKS	50
4.4	Προδιαγραφές καναλιών PL-20	51
4.5	Ρυθμαπόδοση καναλιού με 12-byte πακέτα	56
4.6	Ρυθμαπόδοση καναλιού με 64-byte πακέτα	57
4.7	Κωδικοί μηνυμάτων	64
4.8	Μηνύματα εφαρμογών	65
5.1	Ρυθμός πακέτων σε ένα ISI δίκτυο	76
7.1	Χαρακτηριστικά των PL Smart Transceivers	97
7.2	Χαρακτηριστικές τιμές για την προσέγγιση της μέσης τάσης τροφοδοσίας .	112
8.1	Αποτύπωση ολοκληρωμένου εύρους θερμοκρασιών πάνω στα LEDs	138

Εισαγωγή

Στόχος της εργασίας αυτής είναι ο σχεδιασμός και η ανάπτυξη ενός δικτύου ελέγχου βασισμένο στην τεχνολογία LONWORKS. Οι συσκευές του δικτύου αυτού χρησιμοποιούν το πρωτόκολλο ISI για την εγκατάστασή τους, καθώς επίσης το πρωτόκολλο LonTalk και τις γραμμές μεταφοράς ηλεκτρικής ισχύος ενός κτιρίου για την επικοινωνία τους.

Στο πρώτο μέρος της εργασίας οργανώνεται το θεωρητικό υπόβαθρο. Το υπόβαθρο αυτό προσφέρει στον αναγνώστη τις απαραίτητες γνώσεις για την κατανόηση της εργασίας. Στα τρία πρώτα κεφάλαια δίνονται αρκετά στοιχεία όσον αφορά γενικά τα δίκτυα ελέγχου, αλλά και ειδικότερα τα δίκτυα LONWORKS τεχνολογίας. Στα επόμενα τρία κεφάλαια του πρώτου μέρους παρουσιάζονται τα πρωτόκολλα LonTalk και ISI, τα οποία χρησιμοποιούνται για την επικοινωνία των συσκευών του δικτύου, αλλά και κάποια βασικά στοιχεία της γλώσσας προγραμματισμού Neuron C, η οποία χρησιμοποιήθηκε για την ανάπτυξη των δύο εφαρμογών της εργασίας.

Στο δεύτερο μέρος της εργασίας και συγκεκριμένα στο 7ο κεφάλαιο παρουσιάζεται το υλικό που χρησιμοποιήθηκε στην εργασίας μας. Δίνονται σχήματα καθώς και διάφορα κυκλωματικά διαγράμματα των επιμέρους τμημάτων του υλικού, με σκοπό την πλήρη κατανόηση της λειτουργίας ενός δικτύου LONWORKS. Η μελέτη του υλικού αποτέλεσε σημαντικό κομμάτι για την εργασία μας, διότι με βάση αυτό έγινε η ανάπτυξη των συσκευών του δικτύου, όσον αφορά τον κώδικά τους. Στο κεφάλαιο 8 γίνεται η ανάλυση των δύο εφαρμογών των συσκευών του δικτύου από πλευράς κώδικα. Αναλύεται εκτενώς ο τρόπος εγκατάστασής τους στο δίκτυο (πρωτόκολλο ISI), η μεθοδολογία της μεταξύ τους σύνδεσης αλλά και ο τρόπος λειτουργίας τους μετά την πραγματοποίηση της σύνδεσης.

Μέρος Ι

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Κεφάλαιο 1

Δίκτυα ελέγχου

1.1 Επισκόπηση

Η θεωρία ελέγχου είναι ένας διεπιστημονικός κλάδος της μηχανικής και των μαθηματικών, ο οποίος ασχολείται με την συμπεριφορά των δυναμικών συστημάτων. Η επιθυμητή έξοδος ενός συστήματος καλείται αναφορά. Όταν μία ή περισσότερες μεταβλητές εξόδου ενός συστήματος πρέπει να ακολουθούν μια συγκεκριμένη αναφορά στον χρόνο, ένα ελεγκτής χειρίζεται τις εισόδους σε ένα σύστημα ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα στην έξοδο του συστήματος.

Σε ένα ευρύ φάσμα προϊόντων και συστημάτων, από συστήματα αυτοματισμού σε εργοστάσια μέχρι συστήματα ελέγχου κτιρίων καθώς και σε ενσωματωμένα συστήματα ελέγχου μηχανών υπάρχει η τάση απομάκρυνσης από τα κεντρικά συστήματα ελέγχου. Οι κατασκευαστές φτιάχνουν πλέον προϊόντα τα οποία είναι βασισμένα σε νέες αρχιτεκτονικές ελέγχου δικτύων οι οποίες ονομάζονται open standard αρχιτεκτονικές. Αυτές οι αρχιτεκτονικές υλοποιούν ένα έξυπνο και καταναμημένο είδος ελέγχου χρησιμοποιώντας ένα τυπικό πρωτόκολλο επικοινωνίας χαμηλού κόστους καθώς και πομποδέκτες για την επικοινωνία. Αυτή η υλοποίηση εξασφαλίζει αξιοπιστία, ευελιξία, χαμηλότερο κόστος, ταχύτερη ανάπτυξη όπως επίσης δίνει τη δυνατότητα για καλύτερη παρακολούθηση και έλεγχο της ενέργειας.

Σε ένα κεντρικό σύστημα ελέγχου, απομακρυσμένοι αισθητήρες παρέχουν ανατροφοδότηση (feedback) σε μικροελεγκτές, σε ελεγκτές προγραμματιζόμενης λογικής, ή σε άλλους ελεγκτές οι οποίοι με τη σειρά τους στέλνουν ερεθίσματα σε LEDs ή ενεργοποιητές. Κάθε κεντρικό σύστημα ελέγχου έχει τις δικές του μοναδικές απαιτήσεις εισόδου/εξόδου και

επεξεργασίας. Μεγάλα και πολύπλοκα συστήματα ελέγχου είναι δυνατό να χωριστούν σε δύο ή και περισσότερα κεντρικά συστήματα, οι ελεγκτές των οποίων πρέπει να επικοινωνούν συνέχεια. Ήττε διαχωρίζονται είτε όχι, τα συστήματα αυτά είναι πολύ ακριβά για να μπορέσουν να αναπτυχθούν, δαπανηρά όσον αφορά την εγκατάστασή τους καθώς και αρκετά πολύπλοκα για την επέκτασή τους.

Σε ένα σύστημα ελέγχου(όχι κεντρικό), ευφυείς συσκευές ελέγχου επικοινωνούν χρησιμοποιώντας ένα κοινό πρωτόκολλο. Κάθε συσκευή στο δίκτυο ελέγχου περιέχει ενσωματωμένη λογική που υλοποιεί το κοινό αυτό πρωτόκολλο και εκτελεί τις λειτουργίες ελέγχου. Επιπλέον κάθε συσκευή περιέχει και ένα πομποδέκτη που ενώνει τη συσκευή με το μέσο επικοινωνίας. Οι συσκευές σε ένα δίκτυο ελέγχου μπορούν να εκτελέσουν απλές λειτουργίες αλλά και ένα σύνολο από άλλες πιο πολύπλοκες εργασίες ενώ μπορεί να είναι απλοί αισθητήρες, ενεργοποιητές, διακόπτες, ανιχνευτές κίνησης, ή ρελέ. Επίσης ενδέχεται να είναι συσκευές εποπτικού ελέγχου αλλά και συσκευές συλλογής δεδομένων με στόχο την εποπτεία άλλων συσκευών ενός δικτύου. Ενώ κάποιες μεμονωμένες συσκευές μπορούν να εκτελούν απλές λειτουργίες μπορεί το σύστημα που τις περιλαμβάνει να εκτελέσει μία πιο πολύπλοκη εφαρμογή όπως η λειτουργία μιας γραμμής παραγωγής. Τα δίκτυα ελέγχου απαιτούν μία διαφορετική πλατφόρμα από αυτή των δικτύων επεξεργασίας δεδομένων και ένα στοιχείο που τα διακρίνει είναι τα μικρά σε μέγεθος μηνύματα που μεταδίδονται συχνά μεταξύ των κόμβων και τα οποία απαιτούν μεγάλη αξιοπιστία προσφέροντας χαμηλή επιβάρυνση στο σύστημα.

Για παράδειγμα ένα δίκτυο ελέγχου μπορεί να έχει ένα μεγάλο αριθμό από αισθητήρες πίεσης και θερμοκρασίας οι οποίοι παρέχουν δεδομένα πίεσης και θερμοκρασίας σε ένα ελεγκτή. Ένα τέτοιο σύστημα μπορεί να μην έχει μεγάλα σε μέγεθος δεδομένα αλλά απαιτεί αξιόπιστη μετάδοση των δεδομένων και των μηνυμάτων από τον ελεγκτή για να μπορέσει να λειτουργήσει σωστά.

Πολλοί κατασκευαστές κατανοούν τη σημασία αυτών των ζητημάτων αλλά και τα οφέλη των δικτύων ελέγχου προσπαθώντας όμως (συνήθως μάταια) να δημιουργήσουν τις δικές τους πλατφόρμες ανάπτυξης δικτύων, με αποτέλεσμα η περισσότερη προσπάθεια να δαπανάται στη σχεδίαση και όχι στην ανάπτυξη χρήσιμων εφαρμογών.

1.2 Η αξία των δικτυωμένων συστημάτων ελέγχου σε πραγματικές εφαρμογές

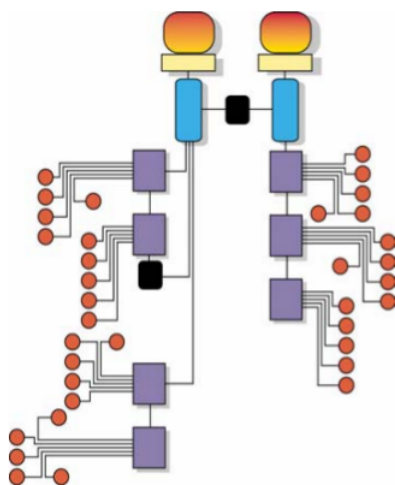
Πολλές εφαρμογές ελέγχου ακολουθούν την ίδια εξελικτική διαδρομή που ακολούθησε και η βιομηχανία των υπολογιστών. Αυτή η διαδρομή επέτρεψε στη βιομηχανία των υπολογιστών να αναπτυχθεί και να δημιουργηθούν συστήματα γενικού σκοπού που να υπερβούν τις προσδοκίες των σχεδιαστών ηλεκτρονικών υπολογιστών. Μία παρόμοια εξέλιξη είχαν και τα δίκτυα ελέγχου. Ένα παράδειγμα της εξέλιξης των συστημάτων ελέγχου, από συστήματα ειδικού σκοπού σε συστήματα γενικού σκοπού είναι η εξέλιξη των ελέγχων φωτισμού. Ένα τυπικό σύστημα ελέγχου φωτισμού μιας πόλης είναι βελτιστοποιημένο ως προς την αξιοπιστία και το κόστος. Ένα τυπικό σύστημα ελέγχου φωτισμού ειδικού σκοπού βασίζεται αποκλειστικά σε ένα ειδικό σύστημα επικοινωνίας και παρακολούθησης. Η εναλλακτική λύση για ένα τέτοιο σύστημα ειδικού σκοπού είναι ένα σύστημα ελέγχου πάνω σε δίκτυο. Το δίκτυο που παρέχει πρόσβαση στο φωτισμό της πόλης είναι η υπάρχουσα υποδομή επικοινωνιών που διαθέτει και μαζί με τον ελεγκτή διαχειρίζονται όλο το σύστημα φωτισμού μιας πόλης. Ο φωτισμός μπορεί να είναι μόνο ένα κομμάτι του συστήματος γιατί αργότερα μπορούν να προστεθούν και άλλα κομμάτια, τα οποία πρέπει σαφώς να υποστηρίζουν την ίδια τεχνολογία, όπως για παράδειγμα αισθητήρες εντοπισμού πεζών. Τα άλλα κομμάτια του συστήματος μπορούν να χρησιμοποιηθούν για να επεκτείνουν τη λειτουργικότητα του αρχικού συστήματος φωτισμού και να παράσχουν αξιοποιήσιμα δεδομένα στον ελεγκτή, καθώς και άλλες λειτουργικότητες που αφορούν για παράδειγμα την ασφάλεια των πολιτών.

Με τη χρήση του παραπάνω παραδείγματος διαπιστώνουμε πώς από ένα πολύ απλό σύστημα ελέγχου μπορούμε να καταλήξουμε σε ένα αρκετά λειτουργικότερο σύστημα με πολλά πλεονεκτήματα και ωφέλη. Η αξία λοιπόν των συστημάτων αυτών στο τομέα του ελέγχου είναι αρκετά μεγάλη καθώς τα συστήματα αυτά είναι φτιαγμένα πάνω σε πλατφόρμες οι οποίες μπορούν να εξελιχθούν αρκετά και να υποστηρίξουν νέες εφαρμογές και υπηρεσίες.

1.3 Η παραδοσιακή προσέγγιση σχεδιασμού συστημάτων ελέγχου

Η λογική του ελέγχου προήλθε μέσω ηλεκτρομηχανικών ρελέ αλλά και μέσω του σχεδιασμού διαφόρων ελεγκτών. Η έλευση της τεχνολογίας στερεάς κατάστασης προσέφερε ένα μέσο για την μείωση του κόστους και για την αύξηση της ευελιξίας χρησιμοποιώντας λογικά κυκλώματα για την αντικατάσταση των καλωδίων. Όλο και πιο ισχυροί αλγόριθμοι αναπτύχθηκαν προσφέροντας στην ουσία αυστηρότερο και καλύτερο έλεγχο των διεργασιών. Ωστόσο τα ζητήματα που είχαν να κάνουν με διάφορες αλλαγές στον τρόπο λειτουργίας παρέμειναν όσο τα συστήματα μεγάλωναν σε μέγεθος.

Αυτό που δημιουργούσε το πρόβλημα ήταν το πώς καταλάβαινε κανείς την έννοια της ανάπτυξης του υλικού και του λογισμικού. Κάθε κατασκευαστής έφτιαχνε τα δικά του συστήματα και παρείχε σε αυτά όλες τις ευφυείς συσκευές που ήταν απαραίτητες για να λειτουργήσει, κυρίως ελεγκτές. Όμως η σύνδεση συστημάτων μεταξύ διαφορετικών κατασκευαστών ήταν σχεδόν πάντα δύσκολη κυρίως λόγω των διαφορετικών πρωτοκόλλων επικοινωνίας που χρησιμοποιούσαν τα εκάστοτε συστήματα ελέγχου. Η επικοινωνία ήταν δυνατή μόνο για κάποιες πολύ περιορισμένες πληροφορίες ελέγχου καθώς τα δεδομένα από διαφορετικούς αισθητήρες δεν ήταν πάντα προσβάσιμα. Το αποτέλεσμα ήταν να δημιουργούνται συστήματα μη αποδοτικά όσον αφορά τη λειτουργία αλλά και το κόστος τους, συστήματα δηλαδή όχι και τόσο αξιόπιστα.



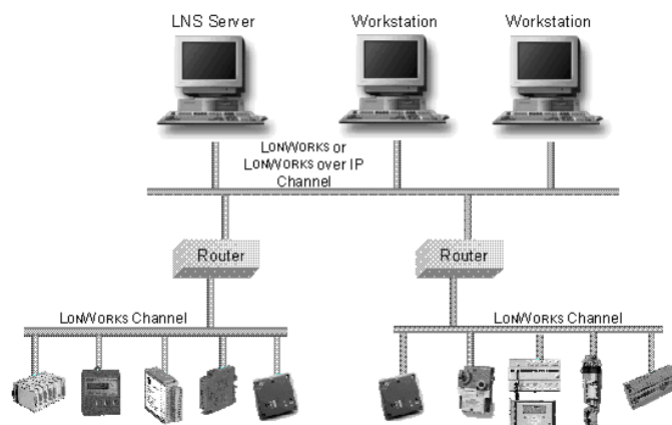
Σχήμα 1.1: Σύστημα κεντρικής αρχιτεκτονικής

Το παραπάνω σχήμα δείχνει την αρχιτεκτονική ενός κεντρικού συστήματος ελέγχου η οποία μέχρι και πρόσφατα ήταν η τυπική αρχιτεκτονική των περισσότερων συστημάτων ελέγχου σε εμπορικές και βιομηχανικές εφαρμογές. Αισθητήρες και ενεργοποιητές συνδέονται μεταξύ τους με καλώδια σε ένα υποσύστημα, το οποίο με τη σειρά του συνδέεται στο σύστημα του ελεγκτή μέσω ενός διαύλου επικοινωνίας. Το σύστημα του ελεγκτή αποτελείται από ένα υψηλής απόδοσης μικρο-επεξεργαστή ο οποίος τρέχει μία τυπική εφαρμογή ελέγχου, η οποία υλοποιεί όλες τις απαιτήσεις εισόδου/εξόδου του συστήματος. Σε μεγάλα συστήματα ο ελεγκτής έχει τη δυνατότητα να συνδέεται μέσω ενός άλλου διαύλου επικοινωνίας με άλλους ελεγκτές. Κάθε σύστημα πρέπει να διαθέτει μία εφαρμογή η οποία υλοποιείται μέσω μίας, κατάλληλης για το υλικό, γλώσσας προγραμματισμού.

1.3.1 Η νέα προσέγγιση των δικτύων ελέγχου

Η νέα προσέγγιση στα δίκτυα ελέγχου βασίζεται σε ένα άλλο είδος δικτύων, τα ανοιχτά δίκτυα ελέγχου (Open Control Networks). Αυτά τα δίκτυα ελέγχου επιτρέπουν σε έναν αριθμό απο ευφρείς συσκευές να επικοινωνούν κατευθείαν μεταξύ τους. Δεν απαιτείται κάποιο εποπτικό σύστημα ελεγκτή το οποίο να λαμβάνει μηνύματα από κάποια συσκευή και να τα προωθεί σε κάποια άλλη. Αυτό σημαίνει ότι οι συσκευές του δικτύου έχουν την ικανότητα να μεταβιβάζουν, μόνες τους, την πληροφορία απευθείας σε άλλες συσκευές. Αυτή η πληροφορία μεταδίδεται από την συσκευή-αποστολέα προς μία ή περισσότερες συσκευές -παραλήπτες σε μορφή πακέτων δεδομένων (packets of data).

Το σχήμα που ακολουθεί μας δίνει τη γενική μορφή ενός Open Control Network.



Σχήμα 1.2: Ανοιχτό δίκτυο ελέγχου

1.3.2 Η μετάβαση από τα δίκτυα δεδομένων στα δίκτυα ελέγχου

Τα δίκτυα στη γενική τους μορφή υπήρχαν εδώ και πολλά χρόνια, όχι για τον έλεγχο συσκευών αλλά μόνο για τις ανάγκες μεγάλων υπολογιστικών συστημάτων. Δημιουργήθηκαν πρωτόκολλα επικοινωνίας τα οποία σχεδιάστηκαν και βελτιστοποιήθηκαν με σκοπό τη διοχέτευση μεγάλων ποσοτήτων δεδομένων μεταξύ υπολογιστών σε συστήματα δέσμης. Προϊόντος του χρόνου, τα πρωτόκολλα αυτά εξελίχθηκαν ενσωματώνοντας μεγαλύτερη ευελιξία και λειτουργικότητα. Τα περισσότερα όμως συνέχισαν να σχεδιάζονται με κύριο σκοπό την επικοινωνία των υπολογιστών μέσω δεδομένων. Τελικά το κόστος των μικροεπεξεργαστών έφτασε στο σημείο, ώστε να μπορούν να ενσωματώνονται ως κομμάτια υλοποίησης σε φθηνούς ελεγκτές και φθηνές συσκευές ελέγχου. Από τότε οι σχεδιαστές άρχισαν να συνειδητοποιούν ότι τα πρωτόκολλα επικοινωνίας που χρησιμοποιούσαν δεν ήταν ρυθμισμένα κατάλληλα ώστε να βελτιστοποιούν την απόδοση των συστημάτων ελέγχου. Τα δίκτυα ελέγχου έχουν μία σειρά από μοναδικές απαιτήσεις που τα κάνουν εντελώς διαφορετικά από τα δίκτυα δεδομένων και είναι :

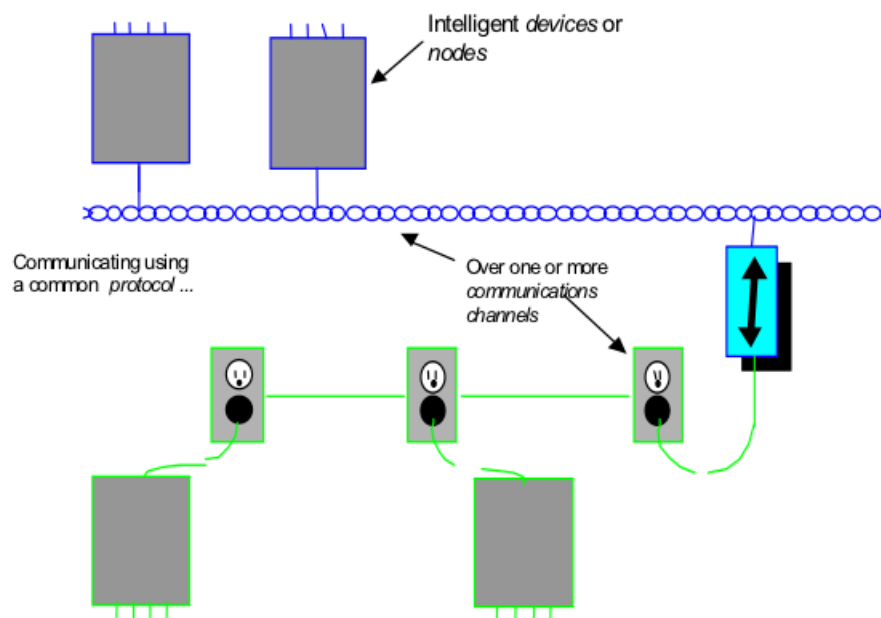
- Συχνή, ασφαλή και αξιόπιστη επικοινωνία μεταξύ των συσκευών του δικτύου.
- Σύντομη μορφή μηνυμάτων για την πληροφορία που πρόκειται να μεταδοθεί.
- Λειτουργία peer-to-peer για την κάθε συσκευή του δικτύου.
- Μικρές και χαμηλού κόστους συσκευές.

Ήταν η ανάγκη για την αντιμετώπιση αυτών των συγκεκριμένων απαιτήσεων που έχουν τα δίκτυα ελέγχου, μαζί με την πίστη ότι ένα πρότυπο για την επικοινωνία θα επέτρεπε την καλύτερη δυνατή συσχέτιση μεταξύ συσκευών ενός δικτύου που οδήγησαν στην ανάπτυξη του πρωτοκόλλου LONWORKS.

1.3.3 Συστατικά των δικτύων ελέγχου

Το επόμενο σχήμα απεικονίζει τα βασικά συστατικά ενός δικτύου ελέγχου, το οποίο αποτελείται από συσκευές οι οποίες επικοινωνούν μεταξύ τους χρησιμοποιώντας ένα κοινό

πρωτόκολλο, πάνω σε ένα ή περισσότερα κανάλια επικοινωνίας. Οι συσκευές του δικτύου συχνά ονομάζονται και κόμβοι.



Σχήμα 1.3: Συστατικά των δικτύων ελέγχου

Κάθε συσκευή περιλαμβάνει ένα ή περισσότερους επεξεργαστές οι οποίοι παρέχουν την λογική τους ώστε να υλοποιηθεί το πρωτόκολλο επικοινωνίας. Επίσης κάθε συσκευή περιλαμβάνει και ένα πομποδέκτη (transceiver), με σκοπό την επικοινωνία της συσκευής με το εκάστοτε κανάλι, για την υλοποίηση μιας συγκεκριμένης εφαρμογής. Οι εφαρμογές που τρέχουν σε ένα σύστημα δεν είναι συγχρονισμένες και είναι πολύ πιθανό οι συσκευές που υλοποιούν αυτές τις εφαρμογές να προσπαθήσουν να “μιλήσουν” την ίδια χρονική στιγμή. Ωστόσο η εποικοδομητική μεταφορά της πληροφορίας μεταξύ των συσκευών ενός δικτύου, απαιτεί μία καλή οργάνωση όσον αφορά ένα σύνολο από κανόνες και διαδικασίες. Αυτό το σύνολο των κανόνων και των διαδικασιών είναι το πρωτόκολλο επικοινωνίας, το οποίο συντομογραφικά αναφέρεται ως πρωτόκολλο.

Το πρωτόκολλο ορίζει τη μορφή των μηνυμάτων τα οποία ανταλλάσσονται μεταξύ των συσκευών, καθώς επίσης και τις ενέργειες που πρόκειται να λάβουν χώρα, όταν μια συσκευή στέλνει μήνυμα σε μια άλλη, έχοντας τη μορφή ενσωματωμένου λογισμικού.

Το μονοπάτι που ακολουθεί η πληροφορία για να φτάσει από μία συσκευή σε μία άλλη

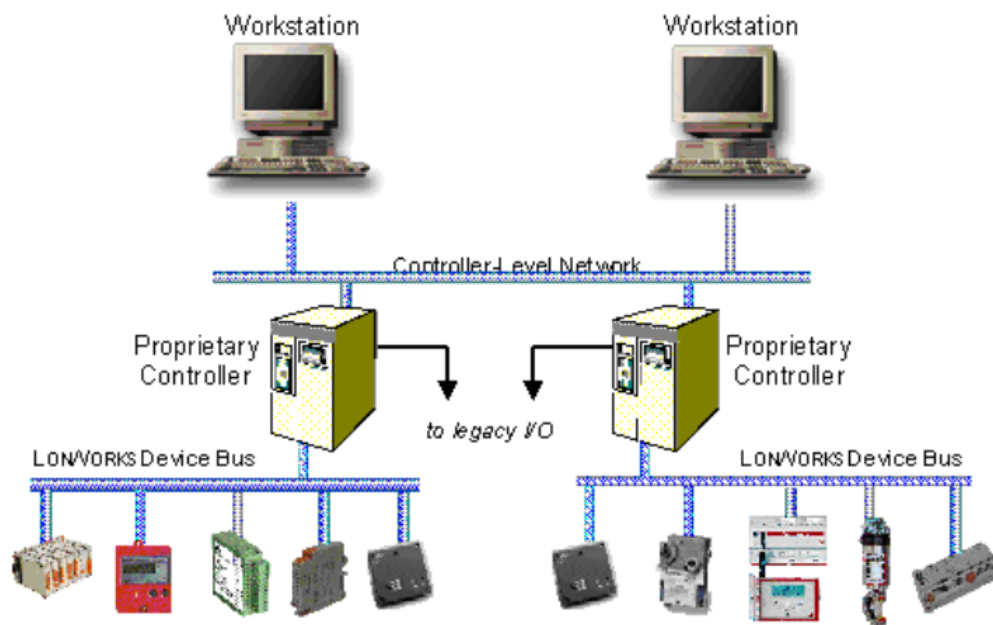
συνθέτει το κανάλι επικοινωνίας, ή απλά το *κανάλι*. Διαφορετικής τεχνολογίας πομποδέκτες μπορεί να είναι σε θέση να επικοινωνούν πάνω στο ίδιο κανάλι. Τα κανάλια κατηγοριοποιούνται με βάση τον τύπο τους και οι πομποδέκτες πρέπει να προσδιορίσουν τον τύπο του καναλιού ή των καναλιών που μπορούν να υποστηρίξουν. Η επιλογή του κατάλληλου τύπου καναλιού επηρεάζει σημαντικά την ταχύτητα της μετάδοσης όπως επίσης και τη γενικότερη τοπολογία του δικτύου.

Όλες οι συσκευές οι οποίες συνδέονται σε ένα συγκεκριμένο κανάλι πρέπει να έχουν συμβατούς πομποδέκτες, οι οποίοι θα πρέπει να έχουν ρυθμιστεί κατάλληλα. Μπορούμε να φτιάξουμε ένα πομποδέκτη για κάθε μέσο αν και μερικοί είναι δύσκολο να υλοποιηθούν και ως εκ τούτου πάρα πολύ ακριβοί. Υπάρχουν διαθέσιμοι πομποδέκτες για κάθε τύπο καναλιού στα οποία συμπεριλαμβάνονται : συνεστραμμένο ζεύγος καλωδίων, γραμμή ισχύος (Power Line), ραδιοσυχνότητες (RF), οπτικές ίνες και ομοαξονικό καλώδιο.

1.4 Χρησιμοποιώντας νέες τεχνολογίες σε ιεραρχικά συστήματα

Μια από τις πιο κοινές αρχιτεκτονικές συστημάτων ελέγχου η οποία αναπτύσσεται σήμερα από διάφορους κατασκευαστές είναι η ιεραρχική αρχιτεκτονική ελέγχου η οποία φαίνεται στο σχήμα 1.4 .

Σε αυτό το σχήμα παρατηρούμε διάφορους ελεγκτές, οι οποίοι ενδέχεται να μπορούν να ενσωματώσουν στοιχεία της LONWORKS τεχνολογίας και να είναι συνδεδεμένοι σε απομονωμένα (isolated) δίκτυα. Αυτά τα δίκτυα πολλές φορές ονομάζονται δίαυλοι συσκευών ή δίκτυα συσκευών. Η ουσία της λογικής του συστήματος αυτού εξακολουθεί να έγκειται στην παροχή πρόσβασης σε διάφορους αισθητήρες και ενεργοποιητές και όχι στην παροχή πρόσβασης σε οποιοδήποτε σημείο του δικτύου από ελεγκτές και άλλους σταθμούς εργασίας οι οποίοι μπορεί να βρίσκονται σε οποιοδήποτε σημείο της ιεραρχίας. Επίσης υπάρχει και ένας μηχανισμός ο οποίος παρέχει το κατάλληλο λογισμικό για τη λειτουργία των ελεγκτών/πυλών, των οποίων η τεχνολογία προγραμματισμού μπορεί να είναι αρκετά καινούργια, όπως για παράδειγμα η Java. Αυτές οι πύλες πολλές φορές επικοινωνούν πάνω σε ένα ανοικτό δίκτυο (Open network) το οποίο ονομάζεται δίαυλος ελέγχου. Όμως το συμπέρασμα που προκύπτει είναι ότι η ιεραρχική αρχιτεκτονική οδηγεί πάλι σε κλειστά συστήματα ελέγχου.



Σχήμα 1.4: Σύστημα ιεραρχικής αρχιτεκτονικής

Ακόμη και όταν χρησιμοποιηθούν LONWORKS συσκευές, η αρχιτεκτονική αυτής της ιεραρχικής δομής δεν μπορεί να επωφεληθεί από τη δύναμη που μπορεί να δώσει το LONWORKS σύστημα. Οι LONWORKS συσκευές σε αυτή την αρχιτεκτονική έχουν περιορισμένο ρόλο στη λήψη αποφάσεων και περιορισμένη αλληλεπίδραση με συσκευές σε άλλα σημεία της ιεραρχίας. Η μόνη διαδρομή την οποία μπορούν να χρησιμοποιήσουν για να επικοινωνήσουν είναι η επικοινωνία μέσω των πυλών. Αυτό το στοιχείο φέρνει την ιεραρχική αρχιτεκτονική συστημάτων ελέγχου ακόμα πιο μακριά από αυτό που ονομάσαμε ανοικτό σύστημα ελέγχου. Το σύστημα ελέγχου λοιπόν παραμένει “κλειστό” ενώ οι ελεγκτές του συστήματος οι οποίοι υλοποιούν το μεγαλύτερο κομμάτι της λογικής του συστήματος, δρουν σαν πλήρως εποπτικές συσκευές. Ελέγχουν τις σχέσεις αλληλεπίδρασης μεταξύ των συσκευών εισόδου/εξόδου, παρέχοντας επίσης οδηγούς (drivers) για την σύνδεση των συσκευών σε άλλους διαύλους επικοινωνίας.

Έτσι λοιπόν μια τέτοια ιεραρχική αρχιτεκτονική συστήματος δεν αποτελεί τη βέλτιστη λύση για διάφορους λόγους. Οι πιο σημαντικοί λόγοι είναι οι εξής :

- Το σύστημα γίνεται χωρίς λόγο αρκετά πολύπλοκο. Αν η αρχιτεκτονική του συστήματος ελέγχου είχε υλοποιηθεί με μία πραγματική peer-to-peer λογική, το επίπεδο του ελεγκτή στο δίκτυο θα μπορούσε να απαλειφεί χωρίς κάποια απώλεια στη λειτουργικότητα. Ο τελικός χρήστης του συστήματος δεν λαμβάνει κάποιο όφελος από

το επιπλέον επίπεδο της ιεραρχίας και στην πραγματικότητα επηρεάζεται μόνο αρνητικά από το κόστος και την πολυπλοκότητα, στοιχεία τα οποία διαμορφώνονται με την εγκατάσταση και τη συντήρηση ενός δεύτερου επιπέδου ελέγχου στο δίκτυο βασισμένο σε διαφορετική τεχνολογία.

- *Εξακολουθεί να είναι κλειστό σύστημα.* Ακόμα και αν οι συσκευές του δικτύου είναι LONWORKS συσκευές, οι ελεγκτές και οι αλγόριθμοι ελέγχου του συστήματος αυτής της αρχιτεκτονικής δεν ανήκουν στην τεχνολογία LONWORKS. Απαιτούν προγραμματισμό και ρύθμιση στις παραμέτρους τους με τη βοήθεια κάποιων εργαλείων δικτύου (network tools). Αυτό αποτρέπει τον τελικό χρήστη από την επίτευξη ενός από τους κύριους στόχους των ανοικτών προτύπων (open standards), την ελευθερία της υλοποίησης νέων συναρτήσεων και λειτουργιών.
- *Δεν είναι δυνατή η επικοινωνία με ένα οποιοδήποτε άκρο του συστήματος, οποιαδήποτε χρονική στιγμή.* Επειδή αυτού του είδους η αρχιτεκτονική αποτελείται από πολλαπλά επίπεδα ελέγχου δεν είναι δυνατή η απευθείας επικοινωνία συσκευών από διαφορετικά κανάλια. Αυτή η αρχιτεκτονική οριοθετεί τη ροή της πληροφορίας μεταξύ των συσκευών, την ευκολία στην υλοποίηση αλγορίθμων ελέγχου, συμβάλει στην αύξηση του χρόνου εγκατάστασης και ουσιαστικά υποβαθμίζει την όλη χρησιμότητα του συστήματος.

Η ιεραρχική αρχιτεκτονική του συστήματος είναι για τον τελικό χρήστη αρκετά πολύπλοκη και δαπανηρή, μπερδεύοντας επίσης τον ανενήμερω αγοραστή ο οποίος έχει αφηθεί στο να πιστέψει ότι έχει αγοράσει ένα σύστημα ανοικτής αρχιτεκτονικής.

Ο αναγνώστης μπορεί να ανατρέξει στις πηγές [2], [8] για επιπλέον πληροφορίες.

Κεφάλαιο 2

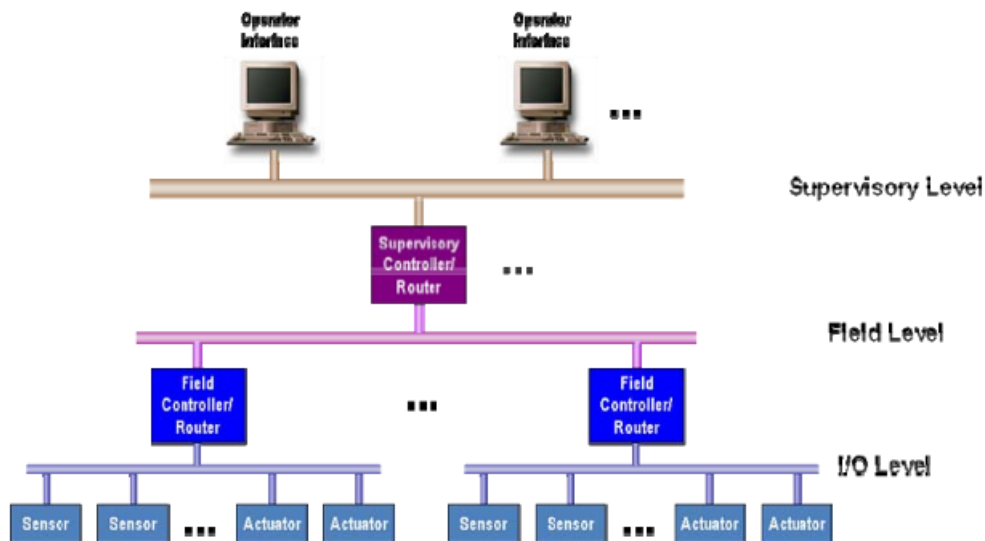
Δίκτυα LONWORKS τεχνολογίας

2.1 Επισκόπηση

Η βασική ιδέα της LONWORKS πλατφόρμας είναι ότι η πληροφορία που μεταδίδεται σε μια εφαρμογή ελέγχου ή παρακολούθησης (monitoring) είναι η ίδια σε όλες τις αγορές και βιομηχανίες. Για παράδειγμα η πόρτα ενός γκαράζ ή η πόρτα των επιβατών ενός πλοίου στέλλουν ουσιαστικά την ίδια πληροφορία (ανοικτή ή κλειστή πόρτα). Μια δεύτερη ιδέα πάνω στην οποία βασίζεται η πλατφόρμα LONWORKS είναι η γνώση ότι τα δίκτυα, ανεξάρτητα από τη λειτουργία τους, αυξάνουν την ισχύ τους καθώς προστίθενται νέοι κόμβοι στο σύστημα.

Από πολλές απόψεις, ένα δίκτυο LONWORKS μοιάζει πολύ με ένα παραδοσιακό δίκτυο δεδομένων. Τα δίκτυα δεδομένων αποτελούνται από πολλούς υπολογιστές συνδεδεμένους σε διάφορα μέσα επικοινωνίας, τα οποία με τη σειρά τους είναι συνδεδεμένα σε δρομολογητές. Οι υπολογιστές επικοινωνούν μεταξύ τους μέσω ενός κοινού πρωτοκόλλου όπως είναι το TCP/IP. Τα δίκτυα δεδομένων είναι βελτιστοποιημένα στο να μεταφέρουν μεγάλο όγκο δεδομένων και η σχεδίαση των πρωτοκόλλων που υλοποιούνται σε αυτά τα δίκτυα υποθέτουν πως οι καθυστερήσεις στην αποστολή των δεδομένων αλλά και στις αποκρίσεις είναι αποδεκτές. Παρά το γεγονός ότι τα δίκτυα αυτά βασίζονται σε “ ανοικτά ” πρωτόκολλα, οι κατασκευαστές δεν επιλέγουν να αναπτύξουν τα δικά τους στοιχεία δικτύωσης όπως πομποδέκτες και δρομολογητές γιατί είναι αποδοτικότερο οικονομικά να αγοράσουν αυτά τα στοιχεία από κάποιον αξιόπιστο προμηθευτή.

Όπως φαίνεται στο παρακάτω σχήμα τα δίκτυα ελέγχου περιέχουν παρόμοια στοιχεία ελέγχου με τα δίκτυα δεδομένων με τη διαφορά ότι τα στοιχεία στα δίκτυα ελέγχου είναι βελτιστοποιημένα όσον αφορά το κόστος, την απόδοση, το μέγεθος και το χρόνο απόκρισης. Τα δίκτυα ελέγχου επιτρέπουν στα δικτυωμένα συστήματα να επεκταθούν σε μια κατηγορία εφαρμογών στην οποία η τεχνολογία των δικτύων δεδομένων δεν είναι απαραίτητη. Οι κατασκευαστές των συστημάτων ελέγχου είναι πλέον σε θέση να μειώσουν τον χρόνο ανάπτυξης βάζοντας LONWORKS τεχνολογία στα προϊόντα που δημιουργούν. Το αποτέλεσμα είναι μια αποδοτικότερη οικονομικά ανάπτυξη καθώς και η διαπίστωση ότι συσκευές από διαφορετικούς κατασκευαστές είναι σε θέση να επικοινωνούν μεταξύ τους.



Σχήμα 2.1: Δίκτυο ελέγχου LONWORKS

Τα LONWORKS δίκτυα κυμαίνονται από μικρά δίκτυα ενσωματωμένα σε μηχανές, μέχρι μεγάλα δίκτυα αποτελούμενα από χιλιάδες συσκευές οι οποίες χρησιμοποιούνται για τον έλεγχο αυτοματισμού κτιρίων. Χρησιμοποιούνται ευρέως στη βιομηχανία αλλά και σε οικιστικές εφαρμογές με κύριο σκοπό βέβαια την μέγιστη δυνατή αξιοπιστία και ευελιξία αλλά και την ελαχιστοποίηση του κόστους ανάπτυξης τους.

Η εταιρεία Echelon κατασκευάζει μια μεγάλη ποικιλία LONWORKS προϊόντων με σκοπό να βοηθήσει τους προγραμματιστές και τους "τελικούς" χρήστες να υλοποιούν LONWORKS δίκτυα. Αυτά τα προϊόντα παρέχουν μια ολοκληρωμένη LONWORKS λύση συμπεριλαμβανομένων και κάποιων εργαλείων ανάπτυξης, λογισμικών διαχείρισης δικτύων, πομποδεκτών για τα διάφορα είδη καναλιών (Power Line και twisted pair cable), δρομολογητών και ελεγκτών.

2.2 Υλοποιώντας ένα δίκτυο ελέγχου

Κατανοώντας τη μεγάλη δύναμη της LONWORKS πλατφόρμας και αντλώντας αυτή τη δύναμη από το γεγονός ότι μπορεί να γίνει η πλατφόρμα ανάπτυξης οποιασδήποτε λειτουργίας ελέγχου, είμαστε σε θέση να αναπτύξουμε το πιο οικονομικά αλλά και όχι μόνο, σύστημα ελέγχου. Έτσι δημιουργούμε ένα δικτυωμένο σύστημα ελέγχου γενικού σκοπού, το οποίο μπορεί πάρα πολύ εύκολα να εξελίσσεται (αλλάζοντας απλά κάποιες παραμέτρους). Ένα δικτυωμένο σύστημα ελέγχου αξιοποιεί μια φυσική αλλά και λογική υποδομή, ώστε να παρέχει πολύ εύκολα μια πλήρη λειτουργία ελέγχου, η οποία θα ανταποκριθεί στις απαιτήσεις των χρηστών.

Έτσι λοιπόν ολόκληρο το σύστημα ελέγχεται από μία ενιαία υποδομή ελέγχου. Οι συσκευές σε ένα τέτοιο σύστημα αποκτούν πολύ εύκολα πρόσβαση στο κανάλι επικοινωνίας. Υπάρχουν επίσης και ορισμένες υπηρεσίες δικτύου, οι οποίες κάνουν πιο εύκολη τη διαδικασία εγκατάστασης του δικτύου, εξασφαλίζοντας παράλληλα και την συμβατότητα συσκευών από διαφορετικούς κατασκευαστές. Τέλος στο επίπεδο της εφαρμογής υπάρχει και ένα κοινό πρότυπο, το οποίο εξασφαλίζει την ασφαλή επικοινωνία μεταξύ των συσκευών του δικτύου. Πρέπει να καταστεί σαφές ότι διαφορετικής λογικής δίκτυα ελέγχου έχουν και διαφορετικές απαιτήσεις και ανάγκες, όπως επίσης και ότι δεν έχουν εκπαιδευτεί όλοι οι χρήστες στα ίδια εργαλεία δικτύωσης.

Οι επόμενες υποενότητες παρέχουν μία ολοκληρωμένη αναφορά σχετικά με τα απαραίτητα στοιχεία που πρέπει να έχει κάποιος κατά νου, όταν σχεδιάζει ολοκληρωμένα δίκτυα ελέγχου.

Ακολουθεί μια συνοπτική λίστα :

- Οικονομικά αποδοτική καλωδίωση του δικτύου
- Συμβατές συσκευές στο δίκτυο
- Αποτελεσματικός σχεδιασμός του συστήματος
- Πρότυπο διαχείρισης του δικτύου
- Πρότυπο εργαλείων δικτύου

2.2.1 Οικονομικά αποδοτική καλωδίωση του δικτύου

Η βάση ενός δικτυωμένου συστήματος ελέγχου είναι η οικονομικά αποδοτική καλωδίωση. Πολλά συστήματα ελέγχου δημιουργούνται με δαπανηρές καλωδιώσεις ή καλωδιώσεις οι οποίες απαιτούν δαπανηρούς συνδέσμους αλλά πολλές φορές και αρκετά δαπανηρούς διακόπτες ή κόμβους (hubs). Η πλέον οικονομικά αποδοτική καλωδίωση για εμπορικά και βιομηχανικά δίκτυα είναι ένα απλό καλώδιο συνεστραμμένου ζεύγους (twisted pair wire), το οποίο μπορεί να συνδεθεί σε οποιαδήποτε τοπολογία δικτύου. Η πιο οικονομικά αποδοτική καλωδίωση για οικιστικές εφαρμογές αλλά και εφαρμογές φωτισμού είναι το ήδη υπάρχον σύστημα των γραμμών μεταφοράς ηλεκτρικής ισχύος (Power Line). Αυτή η λύση επιτρέπει την δημιουργία ενός δικτυωμένου συστήματος ελέγχου χωρίς να χρησιμοποιηθούν νέα καλώδια.

2.2.2 Συμβατές συσκευές στο δίκτυο

Είναι ζωτικής σημασίας οι συσκευές οι οποίες εγκαθίστανται στο δίκτυο ελέγχου, να μπορούν να αλληλεπιδρούν και να ανταλλάζουν πληροφορία χωρίς ιδιαίτερη προσπάθεια. Έτσι λοιπόν ένα από τα κρίσιμα σημεία στην σχεδίαση των δικτύων ελέγχου είναι ο σχεδιασμός και η επιλογή των κατάλληλων συσκευών, στον άξονα μιας κατευθυντήριας γραμμής που πρέπει να ακολουθείται όσον αφορά την καλύτερη επικοινωνία. Αυτό λοιπόν είναι δυνατό να επιτευχθεί μέσω της πλατφόρμας LONWORKS επιλέγοντας προϊόντα που έχουν πιστοποιηθεί από τη LonMark International.

2.2.3 Αποτελεσματικός σχεδιασμός του συστήματος

Όπως ακριβώς ένα σύστημα ελέγχου το οποίο είναι υλοποιημένο πάνω σε ένα μονολιθικό επεξεργαστή, σχεδιάζεται λαμβάνοντας υπόψη τις δυνατότητες του επεξεργαστή έτσι και ένα δικτυωμένο σύστημα ελέγχου πρέπει να σχεδιάζεται λαμβάνοντας υπόψη τις δυνατότητες που έχει το δίκτυο. Οι δυνατότητες που έχει ένα δίκτυο, σύννηθως εκφράζονται και μετριούνται με βάση το εύρος ζώνης (bandwidth) , δηλαδή πακέτα ανά δευτερόλεπτα. Το εύρος ζώνης καθορίζεται από τον τύπο καλωδίωσης που χρησιμοποιείται στο δίκτυο, όπως αναφερθηκε στην πρώτη υποενότητα. Ο αποτελεσματικός σχεδιασμός του συστήματος εξασφαλίζει ότι κάθε συσκευή σε ένα δίκτυο ελέγχου αξιοποιεί το μερίδιο από το συνολικό εύρος ζώνης που της αναλογεί. Τέλος βοηθάει στην διαμέριση μεγάλων δικτύων σε άλλα

μικρότερα, με σκοπό να αυξήσει το συνολικά διαθέσιμο εύρος ζώνης.

2.2.4 Πρότυπο διαχείρισης του δικτύου

Το πρότυπο διαχείρισης του δικτύου παρέχει όλες τις απαραίτητες υπηρεσίες δικτύου καθώς και τις διάφορες διεπαφές για την υποδομή του. Αυτές οι υπηρεσίες επιτρέπουν σε διάφορα εργαλεία και εφαρμογές από διαφορετικούς προμηθευτές να συνυπάρχουν αρμονικά στο ίδιο δίκτυο. Το πιο σημαντικό είναι ότι επιτρέπει σε διάφορα εργαλεία να μοιράζονται τα δεδομένα του δικτύου. Δύο λύσεις είναι διαθέσιμες στα LONWORKS δίκτυα – το LNS Network Operating System, για εμπορικές και βιομηχανικές εφαρμογές (αλλά και για συστήματα μεταφοράς) και το πρωτόκολλο ISI (Interoperable Self-Installation), κυρίως για οικιστικές εφαρμογές. Ένα πρότυπο διαχείρισης του δικτύου όπως το LNS ή το ISI επιτρέπει στους κατασκευαστές των διαφόρων συσκευών να επικεντρώνονται στη σχεδίαση των συσκευών αυτών και όχι στη δημιουργία ολόκληρου του συστήματος ελέγχου.

2.2.5 Πρότυπο εργαλείων δικτύου

Στα εργαλεία ενός δικτύου συμπεριλαμβάνονται εργαλεία ενοποίησης του δικτύου με άλλες εφαρμογές, καταγραφείς δεδομένων καθώς και HMI εφαρμογές. Τα εργαλεία δικτύου μπορούν είτε να αναπτυχθούν με κάποιο τρόπο είτε να αγοραστούν, αλλά και στις δύο περιπτώσεις πρέπει να είναι συμβατά με το πρότυπο διαχείρισης του δικτύου που έχει επιλεγεί (LNS ή ISI). Στην ουσία το πρότυπο διαχείρισης του δικτύου καθορίζει και το πρότυπο που θα χρησιμοποιηθεί για τα εργαλεία. Υπάρχει η δυνατότητα να χρησιμοποιηθεί ένα μόνο εργαλείο σε ένα σύστημα ελέγχου, υπάρχει όμως και η δυνατότητα να συνδυαστούν δύο εργαλεία για τη διαχείριση του συστήματος. Τα εργαλεία επίσης μπορούν να επιλεγθούν με βάση τη λειτουργικότητα αλλά και τη χρηστικότητα τους. Για εμπορικά και βιομηχανικά δίκτυα πρέπει να επιλεγούν εργαλεία που υποστηρίζουν το πρότυπο λειτουργίας δικτύων plug-in, όπως είναι το LonMaker Integration Tool. Το εργαλείο αυτό το οποίο είναι ευρέως διαδεδομένο για τέτοιου είδους εφαρμογές, υποστηρίζει πλήρως τα λεγόμενα functional profiles (λειτουργικά πρότυπα) και configuration properties (ιδιότητες διαμόρφωσης) της LONMARK και επιτρέπει στον σχεδιαστή την επαναχρησιμοποίηση ορισμένων κομματιών της σχεδίασης του δικτύου για άλλους σκοπούς.

Τα εργαλεία και η διαχείριση δικτύων αναλύονται στο επόμενο κεφάλαιο μιας και αποτελούν μέρος της πλατφόρμας LONWORKS.

2.2.6 Πρότυπο ρύθμισης συσκευών

Έχοντας σαν σκοπό τη μείωση του κόστους του συστήματος, μια συσκευή δεν πρέπει απλά να υποστηρίζει ένα πρότυπο επικοινωνίας αλλά και ένα πρότυπο για τη σωστή ρύθμισή της. Και πάλι η LonMark λειτουργεί προς αυτή την κατεύθυνση διότι προσφέρει κατευθυντήριες γραμμές όσον αφορά τις απαιτήσεις του φυσικού επιπέδου των συσκευών, όπως επίσης και τα λειτουργικά προφίλ (functional profiles), τις ιδιότητες διαμόρφωσης (configuration properties) αλλά και ορισμένες χρήσιμες μεθοδολογίες εγκατάστασης.

2.2.7 Υποστήριξη IP

Το πρωτόκολλο διαδικτύου (IP) είναι το πρωτόκολλο πάνω στο οποίο είναι “κτισμένο” το Internet. Ένα δίκτυο ελέγχου πρέπει συχνά να παρέχει, για την ενθυλάκωση του συστήματος ελέγχου, μηνύματα ή πακέτα μηνυμάτων σε μορφή IP πακέτων. Έτσι με αυτό το τρόπο τα μηνύματα μπορούν να περνούν σε όλο τον κόσμο χωρίς να πρέπει να μεταφραστούν σε ξένα πρωτόκολλα. Το κόστος μεταφοράς είναι ελάχιστο. IP υποστήριξη για τα δίκτυα LONWORKS παρέχεται είτε μέσω του LNS server (i.LON SmartServer), είτε μέσω του LonBridge Server, ο οποίος συνδέεται και στο δίκτυο LONWORKS και στο δίκτυο IP.

Τα LONWORKS δίκτυα παρέχουν ένα ολοκληρωμένο πακέτο υπηρεσιών συμπεριλαμβανομένης και της αναγνώρισης της ταυτότητας των μηνυμάτων (authentication) από άκρο σε άκρο καθώς και παράδοση μηνυμάτων με προτεραιότητα. Οι υπηρεσίες διαχείρισης του δικτύου επιτρέπουν στα εργαλεία δικτύου να αλληλεπιδρούν με τις συσκευές που βρίσκονται πάνω στο δίκτυο και περιλαμβάνουν μεταξύ άλλων υπηρεσίες για την διάγνωση προβλημάτων στο δίκτυο καθώς και διάφορες λειτουργίες για την εκκίνηση, την διακοπή και την επαναφορά της εφαρμογής που τρέχει σε μια συσκευή ελέγχου.

Τα Neuron Chips , μια οικογένεια μικροεπεξεργαστών οι οποίοι σχεδιάστηκαν από

την εταιρεία Echelon , συνδυάζουν ένα πυρήνα επεξεργαστή ο οποίος είναι συμβατός με το ANSI/EIA 709.1 για την υλοποίηση διάφορων εφαρμογών, με ένα πομποδέκτη. Αυτά τα τσιπ διαθέτουν ένα μοναδικό 48-bit αριθμό αναγνώρισης ο οποίος ονομάζεται Neuron-id. Τα Neuron 3120 chip περιλαμβάνουν μια αυτόνομη μνήμη για το πρόγραμμα της εφαρμογής (δεν περιλαμβάνεται εξωτερική μνήμη), σε αντίθεση με τα 3150 τα οποία περιλαμβάνουν εξωτερική μνήμη παράλληλα με την εσωτερική. Τα chip αυτά εκτός του επεξεργαστή περιλαμβάνουν και πομποδέκτες οι οποίοι είναι γνωστοί ως *PL Smart Transceivers* και αποτελούν κομμάτι της LONWORKS πλατφόρμας, τα χαρακτηριστικά των οποίων δίνονται αναλυτικά στο επόμενο κεφάλαιο.

Οι πομποδέκτες αυτοί επικοινωνούν μεταξύ τους μέσω των γραμμών μεταφοράς του ηλεκτρικού ρεύματος (εξ' ού και το αναγνωριστικό PL-Power Line) και παρέχουν μία απλή και οικονομικά αποδοτική λύση όσον αφορά τον έλεγχο απλών οικιακών συστημάτων. Αποτελούν ιδανική λύση για συσκευές ήχου, φωτισμού, θέρμανσης/ψύξης αλλά και ασφάλειας αφού μπορούν να επικοινωνούν πολύ εύκολα μέσω συνεχούς ή εναλλασσόμενου ρεύματος.

Για την εφαρμογή που υλοποιήθηκε στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν δύο είδη τέτοιων πομποδεκτών ο PL 3120 και ο PL 3150. Ο PL 3120 λειτουργεί είτε στα 6.5536MHz (A-band) είτε στα 10.0MHz (C-band). Για εφαρμογές οι οποίες απαιτούν περισσότερη μνήμη χρησιμοποιείται συνήθως ο PL 3150 είτε στα 6.5536MHz (A-band) είτε στα 10.0MHz (C-band). Για τα A-band, C-band γίνεται ανάλυση στην επόμενη ενότητα.

Στο σχήμα που ακολουθεί παρατηρούμε κάποια γενικά χαρακτηριστικά για τους συγκεκριμένους πομποδέκτες.

Product Name	Model Number	Maximum Input Clock	EEPROM	RAM	ROM	External Memory Interface	IC Package
PL 3120- E4T10	15310-1000B	10 MHz	4 Kbytes	2 Kbytes	24 Kbytes	No	38 TSSOP
PL 3150-L10	15320-960B	10 MHz	0.5 Kbytes	2 Kbytes	N/A	Yes	64 LQFP

Σχήμα 2.2: Χαρακτηριστικά πομποδεκτών

Η ενότητα που ακολουθεί περιγράφει αναλυτικά την επικοινωνία των Smart Transceivers μέσω της γραμμής μεταφοράς του ηλεκτρικού ρεύματος ενώ στο κεφάλαιο που ακολουθεί δίνεται μια πλήρης περιγραφή των στοιχείων της πλατφόρμας LONWORKS (Neuron Chip, Smart Transceivers, Network Tools, Network Management, Network Interfaces).

2.3 Σηματοδοσία μέσω γραμμών μεταφοράς ηλεκτρικής ενέργειας

Η τεχνολογία σηματοδοσίας η οποία χρησιμοποιείται στους PL Smart Transceivers αναπτύχθηκε και βελτιστοποιήθηκε μέσω μιας σειράς δοκιμών οι οποίες διήρκησαν πάνω από δέκα χρόνια. Πάνω από 20 εκατομμύρια πομποδέκτες αυτής της φιλοσοφίας έχουν αναπτυχθεί, σε ένα μεγάλο εύρος βιομηχανικών και οικιστικών εφαρμογών παγκοσμίως. Στοιχεία όπως η σηματοδοσία BPSK περιορισμένου εύρους ζώνης, η διπλή φέρουσα συχνότητα, προσαρμοστικός φορέας και δεδομένα συσχέτισης, η διόρθωση παλμικού θορύβου αλλά και γενικότερα η διόρθωση σφαλμάτων στη μετάδοση, δίνουν στους πομποδέκτες αυτούς το στοιχείο της αξιοπιστίας απέναντι σε κάθε λογής πηγή θορύβου.

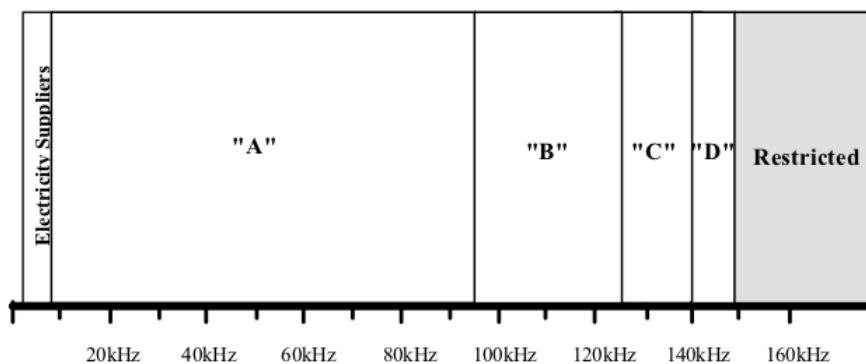
Λειτουργία διπλής φέρουσας συχνότητας

Οι PL Smart Transceivers χρησιμοποιούν μια τεχνολογία διπλής φέρουσας συχνότητας για να παράσχουν την καλύτερη δυνατή αξιοπιστία όσον αφορά την επικοινωνία έναντι των πηγών παραγωγής θορύβου. Στην περίπτωση την οποία έχουμε επιβεβαιωμένα μηνύματα, τα πακέτα των μηνυμάτων αρχικά μεταδίδονται στην πρωτογενή συχνότητα και αν η επιβεβαίωση δεν φτάσει στον αποστολέα τότε το πακέτο επαναμεταδίδεται στη δευτερεύουσα συχνότητα. Στην περίπτωση των μη-επιβεβαιωμένων μηνυμάτων, τα πακέτα εναλλάσσονται μεταξύ της πρωτογενούς και της δευτερογενούς συχνότητας. Οι συσκευές επιτρέπεται να μεταδίδουν μόνο όταν ο ανιχνευτής χρήσης της συχνότητας (band-in-use detector) διαπιστώσει, ότι η συχνότητα είναι ελεύθερη για τουλάχιστον 85msec.

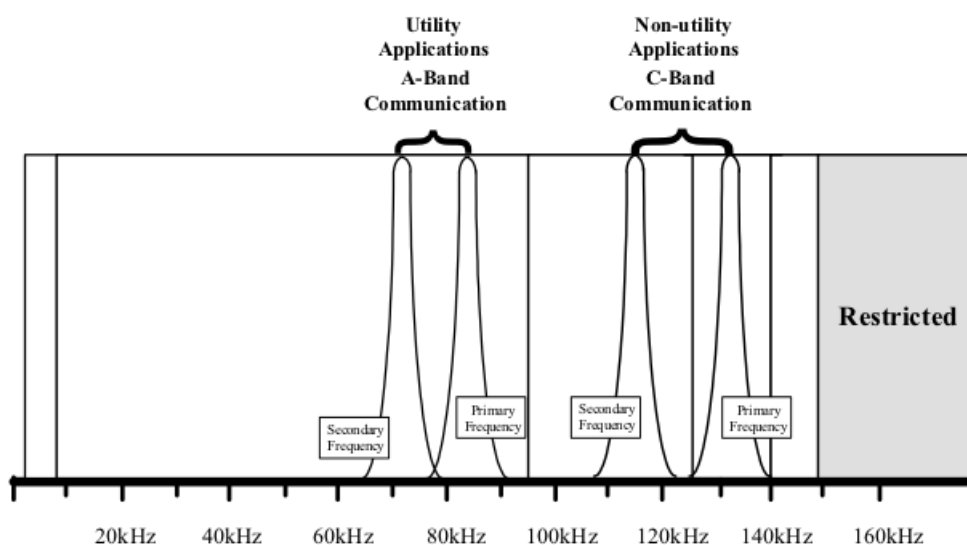
Οι PL Smart Transceivers έχουν σχεδιαστεί ώστε να ανταποκρίνονται σωστά σε κάποιους διεθνείς κανόνες μεταξύ των οποίων είναι και το πρωτόκολλο επικοινωνίας CENELEC EN50065-1. Αυτό το πρωτόκολλο πρόσβασης επιτρέπει σε συσκευές από διάφορους κατασκευαστές να λειτουργούν πάνω στο ίδιο κανάλι. Συνήθως είναι απενεργοποιημένο βοηθώντας στην αύξηση της απόδοσης του συστήματος. Το πρωτόκολλο αυτό είναι πλήρως υλοποιημένο από τους πομποδέκτες αυτούς εξαλείφοντας την ανάγκη, οι χρήστες να δημιουργούν περίπλοκους αλγορίθμους πρόσβασης. Έτσι οι PL Smart Transceivers μπορούν να λειτουργούν είτε στο εύρος CENELEC (A-Band) είτε στο εύρος κατανάλωσης (C-Band). Ο χρήστης επιλέγει αν θα ενεργοποιήσει ή όχι το CENELEC με την επιλογή του κατάλληλου καναλιού (Το πρωτόκολλο πρέπει να ενεργοποιείται μόνο σε χώρες οι οποίες ακολουθούν τις απαιτήσεις του CENELEC και είναι οι περισσότερες χώρες της E.E).

Στη παρούσα διπλωματική το πρωτόκολλο αυτό παρέμεινε ανενεργό για λόγους καλύτερης απόδοσης αλλά και βασιζόμενοι στο γεγονός ότι σε οικιστικές εφαρμογές χρησιμοποιείται πάντα το C-band (αλλά και μη γνωρίζοντας τι ισχύει στην Ελλάδα απο πλευράς κανονισμών του πρωτοκόλλου). Έτσι επιλέχθηκε το PL-20N ως κανάλι επικοινωνίας.

Στο σχήμα 2.3 που ακολουθεί αποτυπώνονται οι περιορισμοί συχνότητας οι οποίοι απορρέουν από το πρωτόκολλο CENELEC (διαθέσιμο εύρος συχνοτήτων), ενώ στο σχήμα 2.4 αποτυπώνεται η πρωτογενής και δευτερογενής συχνότητα μέσα στο εύρος των διαθέσιμων συχνοτήτων (λειτουργία διπλής φέρουσας συχνότητας).



Σχήμα 2.3: Γενική σχεδίαση του εύρους συχνοτήτων



Σχήμα 2.4: Λειτουργία διπλής φέρουσας συχνότητας

Διόρθωση λαθών

Πολλές πηγές θορύβου παρεμβαίνουν στην σηματοδότηση που πραγματοποιείται μέσω των γραμμών μεταφοράς του ρεύματος μεταβάλλοντας ή διαφθείροντας τα πακέτα που περιέχουν δεδομένα. Οι PL πομποδέκτες χρησιμοποιούν έναν υψηλής απόδοσης αλγόριθμο για διόρθωση λαθών, ο οποίος ονομάζεται FEC (Forward Error Correction). Ο αλγόριθμος αυτός λειτουργεί παράλληλα με ένα μηχανισμό για κυκλικό έλεγχο πλεονασμού (CRC-cyclical redundancy check) με σκοπό να ξεπεραστούν τα λάθη στα πακέτα δεδομένων.

Ισχυρός ενισχυτής εξόδου

Η εξωτερική αλλά και υψηλής απόδοσης, σχεδίαση του ενισχυτή, η οποία αναπτύχθηκε με σκοπό τη χρήση της πάνω στους PL Smart Transceivers παρέχει μια αντίσταση εξόδου της τάξης του 1Ω και μια 1Ar-p ικανότητα οδήγησης υψηλών εξόδων σε χαμηλής αντίστασης κυκλώματα. Παράλληλα βοηθάει στην διατήρηση των εξαιρετικά χαμηλών επιπέδων της παραμόρφωσης του σήματος. Για εφαρμογές οι οποίες απαιτούν ακόμα υψηλότερη ισχύ εξόδου, είναι διαθέσιμη μια υψηλότερης ισχύος σχεδίαση η οποία παρέχει ρεύμα εξόδου της τάξης των 2Ar-p.

Μεγάλο δυναμικό εύρος

Το δυναμικό εύρος σχετίζεται με την ευαισθησία του δέκτη. Οι PL Smart Transceivers έχουν ένα δυναμικό εύρος μεγαλύτερο από 80db. Σε μια ήρεμη γραμμή οι πομποδέκτες μπορούν να λαμβάνουν σήματα τα οποία έχουν εξασθενήσει έως και 10.000 φορές.

Χαμηλή κατανάλωση ρεύματος

Οι πομποδέκτες και τα συνδεδεμένα σε αυτούς, κυκλώματα ενισχυτών ισχύος, τροφοδοτούνται με +8.5 έως +18VDC (VA) και +5VDC (VDD5). Η διαχείριση της ισχύος είναι πολύ χρήσιμο ζήτημα ειδικά για προϊόντα χαμηλού κόστους όπως πρίζες, διακόπτες και λαμπτήρες. Η πολύ χαμηλή κατανάλωση της τάξης των 350μΑ από την τροφοδοσία Va και των 9mA από την Vdd5 μειώνει το μέγεθος και το κόστος της τροφοδοσίας. Οι PL Smart Transceivers επικοινωνούν (μεταδίδουν ή λαμβάνουν) με ρυθμό 5.4kbps (C-band) ή 3.6kbps (A-band) , ο οποίος αντιστοιχεί σε ένα ρύθμο 20 και 13 πακέτων ανά δευτερόλεπτο (ρυθμαπόδοση), αντίστοιχα στις δύο ζώνες συχνοτήτων.

Ο αναγνώστης μπορεί να ανατρέξει στην πηγή [4], για επιπλέον πληροφορίες.

Κεφάλαιο 3

Η Πλατφόρμα LONWORKS

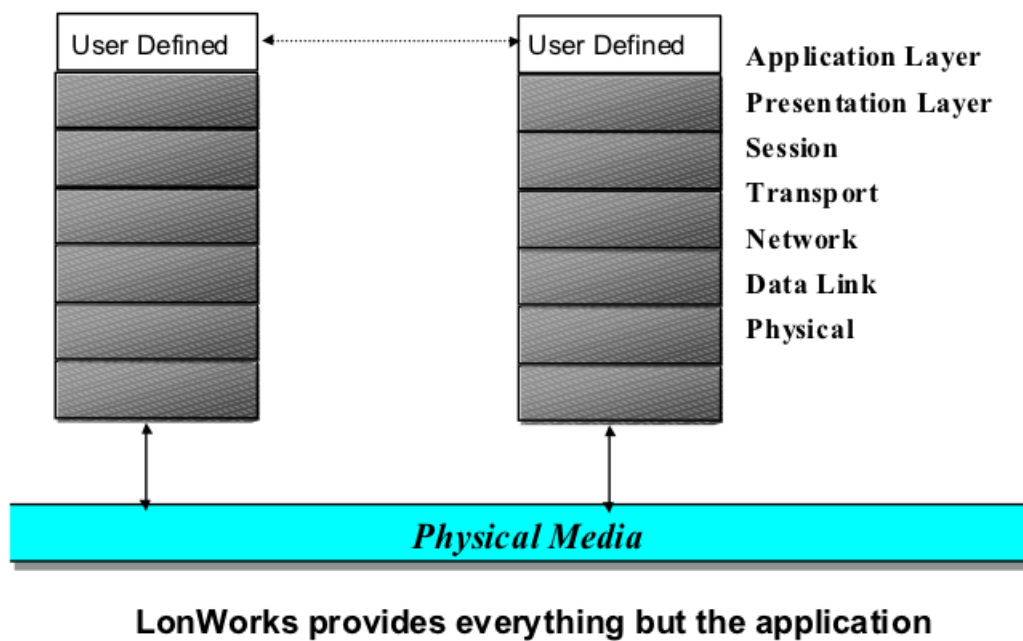
3.1 Επισκόπηση

Η Echelon Corporation επινόησε το LONWORKS πρωτόκολλο. Χιλιάδες κατασκευαστές συστημάτων ελέγχου χρησιμοποιούν αυτό το σύστημα, αν και μερικοί από αυτούς δεν έχουν κάνει προτεραιότητά τους την υλοποίηση συστημάτων τα οποία βασίζονται σε ένα κύριο πρότυπο αλλά και σε ένα πρότυπο διαχείρισης του δικτύου.

Η Echelon άρχισε τη δημιουργία αυτής της πλατφόρμας ανάπτυξης το 1988. Δημιούργησε μια οικονομικά αποδοτική μέθοδο η οποία επιτρέπει σε πολύ απλές συσκευές ελέγχου να επικοινωνούν μεταξύ τους χωρίς καμία ιδιαίτερη δυσκολία. Η εταιρεία κατάλαβε ότι δημιουργώντας απλά ένα πρωτόκολλο δεν μπορούσε να καταστήσει επαρκή την επίτευξη του στόχου ενός πλήρως λειτουργικού συστήματος ελέγχου. Ήταν απαραίτητο να φτιαχτεί ένα σύστημα πλήρως αποδοτικό σε όλους τους τομείς, μία μέθοδος ουσιαστικά μέσω της οποίας το πρωτόκολλο θα είχε τη δυνατότητα να προμηθεύει το σύστημα με όλα τα απαραίτητα εργαλεία ανάπτυξης και προϊόντα δικτύωσης.

Ο πρωταρχικός στόχος ήταν να καταστεί εύκολη και αποδοτική η υλοποίηση ενός ανοικτού συστήματος ελέγχου. Η Echelon ανέπτυξε το σύστημα LONWORKS πιστεύοντας πως υπήρχαν τρία θεμελιώδη ζητήματα τα οποία έπρεπε να αντιμετωπιστούν. Καταρχάς ένα βελτιστοποιημένο πρωτόκολλο για τα δίκτυα ελέγχου, το οποίο όμως θα ήταν ικανό να δουλεύει πάνω σε διαφορετικούς τύπους ελέγχου. Δεύτερον θα έπρεπε να αντιμετωπίσει το κόστος ανάπτυξης αυτού του πρωτοκόλλου και τρίτον η δημιουργία του, δεν θα έπρεπε να καταστρέψει τη διαλειτουργικότητα.

Η Echelon με σκοπό να αντιμετωπίσει αυτά τα ζητήματα δημιούργησε μια πλήρη πλατφόρμα για σχεδίαση, δημιουργία και εγκατάσταση ευφυών συσκευών ελέγχου. Προμήθευσε τους χρήστες με όλα τα στοιχεία, τα οποία ήταν απαραίτητα για τη δημιουργία συστημάτων ελέγχου.



Σχήμα 3.1: Μια άποψη της λειτουργίας του LONWORKS συστήματος βάσει των επιπέδων του πρωτοκόλλου

Το κεφάλαιο αυτό περιγράφει τα ακόλουθα στοιχεία τα οποία συνθέτουν την πλατφόρμα LONWORKS :

- Neuron Chip και Έξυπνοι πομποδέκτες (Smart Transceivers)
- Εργαλεία ανάπτυξης
- Δρομολογητές
- Διασυνδέσεις δικτύου
- Διαχείριση της πλατφόρμας
- Εργαλεία δικτύου

3.1.1 Το Neuron chip

Η Echelon έχοντας σαν σκοπό της την ανάπτυξη ενός πλήρους λειτουργικού συστήματος ελέγχου δημιούργησε τον πυρήνα Neuron. Ο πυρήνας αυτός πήρε το όνομά του, από τις ομοιότητες που υπάρχουν μεταξύ των δικτύων ελέγχου και του ανθρώπινου εγκεφάλου. Για τον προγραμματιστή η ομορφιά του Neuron chip βρίσκεται στην πληρότητά του. Το ενσωματωμένο πρωτόκολλο επικοινωνίας καθώς και οι επεξεργαστές του, απομακρύνουν την ανάγκη για κάποια επιπλέον προγραμματιστική ανάπτυξη πάνω σε αυτά τα στοιχεία. Το Neuron chip παρέχει τα πρώτα 6 επίπεδα του πρωτοκόλλου ISO/OSI το οποίο είναι ένα μοντέλο αναφοράς του πρωτόκολλου που χρησιμοποιείται στα LONWORKS δίκτυα. Μόνο το επίπεδο της εφαρμογής χρειάζεται να υλοποιηθεί από τους χρήστες.

Οι περισσότερες LONWORKS συσκευές επωφελούνται από τις λειτουργίες του Neuron chip και το χρησιμοποιούν ως τον κεντρικό επεξεργαστή ελέγχου. Το Neuron chip περιλαμβάνει τρεις επεξεργαστές, οι οποίοι προσφέρουν πολλές δυνατότητες επικοινωνίας. Οι κατασκευαστές των συσκευών παρέχουν κώδικα ο οποίος τρέχει πάνω στο Neuron chip καθώς επίσης και συσκευές εισόδου/εξόδου οι οποίες μπορούν να συνδεθούν με το τσιπ αυτό.

Το Neuron chip ουσιαστικά είναι ένα ολοκληρωμένο σύστημα επικοινωνίας πάνω σε μια πλακέτα (system-on-chip), με μνήμη ανάγνωσης -εγγραφής καθώς και μνήμη μόνο ανάγνωσης (RAM & ROM). Η μνήμη ROM περιέχει το πρωτόκολλο LONWORKS και κάποιες χρήσιμες λειτουργίες εισόδου/εξόδου και η μνήμη RAM τον κώδικα της εφαρμογής.

Ο κώδικας των εφαρμογών γράφεται σε Neuron C (υπάρχει αναλυτική περιγραφή για τη γλώσσα σε επόμενο κεφάλαιο της εργασίας).

Ως μια σύντομη αναφορά για τη γλώσσα θα μπορούσαμε να πούμε ότι στηρίζεται στην ANSI C με τις ακόλουθες όμως πολύ σημαντικές προεκτάσεις :

- Την πρόταση *when* η οποία εισάγει την έννοια του γεγονότος καθορίζοντας επίσης την σειρά εκτέλεσης των λειτουργιών (συναρτήσεων) του κώδικα.
- 37 επιπλέον τύπους δεδομένων (35 I/O objects και 2 timer objects)
- Ένα πλήρη μηχανισμό για αποστολή και λήψη μηνυμάτων μεταξύ των συσκευών (μηνύματα μέσω μεταβλητών δικτύου αλλά και άλλα είδη μηνυμάτων)

Είναι μια γλώσσα εύκολη όσον αφορά την εκμάθησή της. Χρησιμοποιεί ένα διαφορετικό προγραμματιστικό μοντέλο βασισμένο σε γεγονότα (events). Με άλλα λόγια τα γεγονότα

τα οποία συμβαίνουν σε ένα δίκτυο καθορίζουν και τη μετέπειτα συμπεριφορά του. Δηλαδή τα γεγονότα είναι αυτά που οδηγούν το δίκτυο. Αυτό σημαίνει ότι τα LONWORKS δίκτυα δεν είναι τόσο φορτωμένα από πλευράς δεδομένων όσο άλλα δίκτυα.

3.1.1.1 Πομποδέκτες

Στο πρώτο κεφάλαιο αναφέραμε τους πομποδέκτες στη γενική τους μορφή ως στοιχείο μιας συσκευής ενός δικτύου ελέγχου. Στο δεύτερο κεφάλαιο παρουσιάσαμε κάποια γενικά χαρακτηριστικά τους με σκοπό την ανάλυση της σηματοδότησης μέσω των γραμμών του ρεύματος, δηλαδή του τρόπου με τον οποίο επικοινωνούν εβρισκόμενοι πάνω σε ένα δίκτυο LONWORKS. Κάποια επιπλέον χαρακτηριστικά :

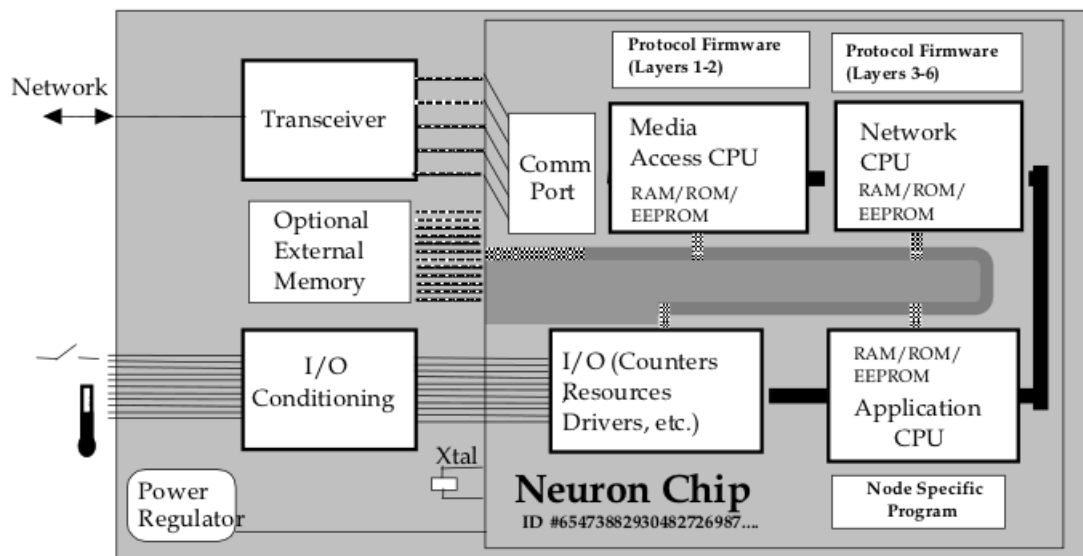
- Κάθε συσκευή δικτύου περιέχει ένα πομποδέκτη. Οι πομποδέκτες παρέχουν μια φυσική διεπαφή μεταξύ μιας LONWORKS συσκευής και του LONWORKS δικτύου.
- Οι πομποδέκτες απλοποιούν την ανάπτυξη των LONWORKS συσκευών και είναι διαθέσιμοι για μια μεγάλη ποικιλία καναλιών και τοπολογιών.
- Προϊόντα με διαφορετικούς τύπους πομποδεκτών μπορούν να αλληλεπιδρούν μόνο με τη χρήση ενός δρομολογητή.

Η τεχνική ανάλυση των πομποδεκτών αλλά και όλα τα άλλα στοιχεία που αφορούν τη χρησιμότητά τους στην εκπόνηση της διπλωματικής περιγράφονται στο κεφάλαιο 7, το οποίο αφορά στην παρουσίαση του LONWORKS συστήματος από πλευράς υλικού.

3.1.2 Εργαλεία ανάπτυξης

Η Echelon παρέχει ένα μεγάλο εύρος εργαλείων ανάπτυξης LONWORKS συσκευών. Ακολουθεί μια λίστα με τα διαθέσιμα εργαλεία ανάπτυξης :

- Mini FX Evaluation Kit
- NodeBuilder FX Development Tool
- ShortStack Developer's Kit
- FTXL Developer's Kit



Σχήμα 3.2: Το εσωτερικό μιας τυπικής συσκευής ελέγχου LONWORKS

Στη δική μας εργασία χρησιμοποιήθηκε το Mini FX Evaluation Kit για σηματοδοσία μέσω της γραμμής μεταφοράς του ρεύματος.

Τα εργαλεία ανάπτυξης τυπικά περιλαμβάνουν ένα περιβάλλον για την ανάπτυξη και την αποσφαλμάτωση του κώδικα της εφαρμογής, ένα διαχειριστή του δικτύου για την ρύθμιση των συσκευών καθώς και ένα αναλυτή πρωτοκόλλου (protocol analyzer), ο οποίος εξασφαλίζει την επαρκή ικανότητα του δικτύου. Τα εργαλεία ανάπτυξης καθιστούν εύκολη και ανέξοδη για τους κατασκευαστές, τη σχεδίαση μεμονωμένων LONWORKS συσκευών.

3.1.3 Δρομολογητές

Η υποστήριξη πολλών μέσων επικοινωνίας (καναλιών) επιτυγχάνεται μέσω των *δρομολογητών*. Οι δρομολογητές μπορούν να χρησιμοποιηθούν επίσης για τον έλεγχο της ροής των δεδομένων στο δίκτυο με στόχο την αύξηση της ρυθμαπόδοσης (throughput) του δικτύου. Τα εργαλεία δικτύου βασίζόμενα στην τοπολογία του δικτύου, ρυθμίζουν αυτόματα τους δρομολογητές με σκοπό την εύκολη εγκατάστασή τους στο δίκτυο.

Οι δρομολογητές δίνουν τη δυνατότητα σε ένα δίκτυο peer-to-peer να υποστηρίξει πολλούς τύπους καναλιών όπως επίσης και ένα μεγάλο αριθμό συσκευών (περίπου 10.000). Κάθε δρομολογητής έχει δυο πλευρές, η καθεμία από τις οποίες έχει ένα δικό της κατάλληλο πομποδέκτη, ο οποίος συνδέεται με τα κανάλια στα οποία ο δρομολογητής είναι

συνδεδεμένος. Η λειτουργία των δρομολογητών είναι καθορισμένη μέσα σε ένα δίκτυο και είναι η μεταφορά των πακέτων από ένα κόμβο σε ένα άλλο. Όμως οι δρομολογητές δεν είναι απαραίτητο να μεταφέρουν όλα τα πακέτα γιατί από τη στιγμή κατά την οποία ρυθμίζονται από ένα εργαλείο δικτύου αποκτούν αρκετή γνώση για το σύστημα. Έτσι μπορούν να μπλοκάρουν τα πακέτα τα οποία δεν έχουν διευθύνσεις είτε του αποστολέα είτε του παραλήπτη. Χρησιμοποιώντας ένα άλλο τύπο δρομολογητή, ο οποίος ονομάζεται IP-852, οι LONWORKS δρομολογητές επεκτείνονται και σε δίκτυα ευρείας περιοχής (wide-area) όπως είναι το Internet.

Η Echelon παρέχει δρομολογητές, οι οποίοι μπορούν να συνδεθούν σε διαφορετικούς τύπους καναλιών συνεστραμμένου ζεύγους (twisted pair) όπως επίσης και τον τύπο του δρομολογητή που αναφέρθηκε πρωτίτερα (IP-852) ο οποίος δρομολογεί τα bits από ένα κανάλι συνεστραμμένου ζεύγους σε ένα IP δίκτυο όπως το Internet ή σε ένα εικονικό ιδιωτικό δίκτυο (Virtual Private Network- VPN).

Ο πλήρης κατάλογος των διαθέσιμων δρομολογητών, οι οποίοι παρέχονται από την Echelon βρίσκεται στο "www.echelon.com/products". Ακολουθεί μια λίστα με τους συνηθέστερα χρησιμοποιούμενους δρομολογητές :

- i.LON SmartServer– Δρομολογεί bits μεταξύ ενός καναλιού IP-852 και ενός καναλιού συνεστραμμένου ζεύγους
- i.LON 600
- MPR-50 Multi-Port Router– Δρομολογεί bits συνδέοντας ουσιαστικά πέντε κανάλια μεταξύ τους
- LonPoint Router– Δρομολογεί bits μεταξύ δύο καναλιών συνεστραμμένου ζεύγους

3.1.4 Διασυνδέσεις δικτύου– Network Interfaces

Μια διασύνδεση δικτύου είναι ουσιαστικά μια συσκευή σύνδεσης ή μια κάρτα, η οποία χρησιμοποιείται για να συνδέσει τον υπολογιστή με ένα LONWORKS δίκτυο. Η συσκευή αυτή δεν τρέχει κάποια εφαρμογή από μόνη της, απλά χρησιμοποιείται για να δώσει μια εικόνα του δικτύου (πόσα πακέτα και κάθε πόσο αυτά μεταδίδονται).

Ακολουθεί μια λίστα με τις συνηθέστερα χρησιμοποιούμενες συσκευές διασύνδεσης :

- U10/U20 USB Network Interface– Μια μικρή USB συσκευή η οποία συνδέεται σε μια θύρα του υπολογιστή. Στη δική μας εργασία χρησιμοποιήθηκε το U20 USB

interface για την παρακολούθηση του δικτύου. Αναλυτικότερη περιγραφή δίνεται σε επόμενο κεφάλαιο

- i.LON SmartServer– Ένα χειριστήριο το οποίο μπορεί να χρησιμοποιηθεί για σύνδεση με ένα IP δίκτυο
- i.LON 600– Είναι ένας δρομολογητής ο οποίος μπορεί να χρησιμοποιηθεί και ως συσκευή διασύνδεσης με μια IP σύνδεση σε ένα ηλεκτρονικό υπολογιστή

3.1.5 Διαχείριση της πλατφόρμας

Τα δίκτυα LONWORKS μπορούν να κατηγοριοποιηθούν ανάλογα με τη μέθοδο που χρησιμοποιούν για να εκτελέσουν τη λειτουργία της εγκατάστασής τους. Υπάρχουν δύο είδη δικτύων : τα *διαχειριζόμενα* (managed) και τα *αυτο-εγκαθιστώμενα* (self-installed) δίκτυα. Ένα διαχειριζόμενο δίκτυο είναι ένα δίκτυο στο οποίο ένας διακομιστής (server) διαχείρισης δικτύου χρησιμοποιείται για την εγκατάσταση του δικτύου. Αυτός ο διακομιστής μπορεί να είναι μέρος ενός λειτουργικού συστήματος του δικτύου (περιγράφεται αργότερα στο κεφάλαιο) ή μπορεί να είναι κομμάτι ενός Internet server όπως ο SmartServer ο οποίος αναφέρθηκε προωτέρα. Ο χρήστης μπορεί να χρησιμοποιήσει ένα εργαλείο δικτύου ώστε να μπορεί να αλληλεπιδρά με το διακομιστή καθώς και να ορίσει το πως οι συσκευές θα πρέπει να ρυθμιστούν και να επικοινωνούν. Ένα τέτοιο εργαλείο ονομάζεται *εργαλείο διαχείρισης δικτύου* και περιγράφεται λίγο αργότερα στο κεφάλαιο. Παρόλο που ένα εργαλείο διαχείρισης δικτύου και ένας διακομιστής χρησιμοποιούνται για να καθορίσουν την σωστή επικοινωνία μέσα στο δίκτυο, δεν είναι αναγκαία η παρουσία τους καθ' όλη τη διάρκεια της λειτουργίας του. Χρειάζονται μόνο όταν απαιτούνται κάποιες συγκεκριμένες αλλαγές στη ρύθμιση του δικτύου.

Σε ένα διαχειριζόμενο δίκτυο, τα εργαλεία διαχείρισης του δικτύου και ο διακομιστής κατανέμουν τους πόρους του δικτύου. Ο διακομιστής είναι ενήμερος για την τοπολογία του δικτύου και μπορεί να ρυθμίζει τις συσκευές με στόχο τη βέλτιστη απόδοσή τους εντός των περιορισμών της τοπολογίας.

Η εναλλακτική λύση σε ένα διαχειριζόμενο δίκτυο είναι ένα αυτο-εγκαθιστώμενο δίκτυο. Σε αυτά τα δίκτυα δεν υπάρχει κάποιο κεντρικό εργαλείο ούτε κάποιος κεντρικός διακομιστής που να διαχειρίζεται ολόκληρο το σύστημα. Αντ' αυτού κάθε συσκευή περιέχει κομμάτια κώδικα τα οποία αντικαθιστούν τη λειτουργία (ή μέρος της λειτουργίας) του κεντρικού διακομιστή διαχείρισης του δικτύου. Αυτό έχει σαν αποτέλεσμα την δημιουργία

ενός δίκτυου, το οποίο δεν απαιτεί πλέον την παρουσία κάποιου εργαλείου ή διακομιστή για να μπορέσει να ρυθμιστεί σωστά.

Η εγκατάσταση του δικτύου περιλαμβάνει τα ακόλουθα στάδια :

- Εχχώρηση λογικών διευθύνσεων σε όλες τις συσκευές και στις ομάδες των συσκευών
- Ένωση των μεταβλητών δικτύου (Network variables) με σκοπό να δημιουργηθούν λογικές συνδέσεις μεταξύ των συσκευών.
- Ρύθμιση των παραμέτρων του πρωτοκόλλου ελέγχου του δικτύου για κάθε συσκευή που συμμετέχει στο δίκτυο, με σκοπό την επιθυμητή απόδοση. Αυτές οι ρυθμίσεις περιλαμβάνουν το ρυθμό μετάδοσης του καναλιού καθώς και υπηρεσίες για την αναγνώριση και προτεραιότητα των μηνυμάτων.

Η εγκατάσταση του δικτύου μπορεί να γίνει μια αρκετά περίπλοκη διαδικασία και η πολυπλοκότητα αυτή κρύβεται πίσω από τις λύσεις που υπάρχουν για την διαχείριση του δικτύου. Στις διαχειριζόμενες εφαρμογές η διαδικασία σχεδίασης του δικτύου μπορεί να είναι πολύ σύντομη, απλά χρησιμοποιώντας ένα σχεδιαστικό πρόγραμμα και ενώνοντας τις εισόδους – εξόδους των εφαρμογών ώστε να μπορούν να επικοινωνούν. Το εργαλείο διαχείρισης του δικτύου αντιστοιχίζει αυτόματα λογικές διευθύνσεις, ενώνει τις μεταβλητές δικτύου όπως αυτές σχεδιάστηκαν στο γραφικό εργαλείο και ρυθμίζει τις παραμέτρους ελέγχου του δικτύου. Για τα αυτο-εγκαθιστώμενα δίκτυα, η σχεδίαση του δικτύου είναι τόσο απλή όσο το πάτημα ενός κουμπιού σε καθεμία από ένα ζευγάρι συσκευών ώστε να δημιουργηθεί η σύνδεση και να επιτευχθεί η μεταξύ τους επικοινωνία.

Ένα δίκτυο μπορεί να ξεκινήσει να λειτουργεί ώντας ένα αυτο-εγκαθιστώμενο δίκτυο χρησιμοποιώντας τη μηχανή ISI (ISI protocol) και όσο το μέγεθος και η πολυπλοκότητα αυξάνουν, να μετατραπεί σε ένα διαχειριζόμενο δίκτυο. Επίσης η μετατροπή αυτή μπορεί να αξιοποιήσει την επιπρόσθετη αξιοπιστία η οποία λαμβάνεται μέσω των διαχειριζόμενων δικτύων (λόγω της παρουσίας του διακομιστή και των εργαλείων διαχείρισης του δικτύου).

3.1.5.1 Το λειτουργικό σύστημα του δικτύου

Για να διασφαλίσει τη διαλειτουργικότητα μεταξύ των εργαλείων δικτύου και των εφαρμογών, η πλατφόρμα LONWORKS δημιούργησε ένα λειτουργικό σύστημα για τα διαχειριζόμενα δίκτυα το οποίο ονομάζεται LNS Network Operating System. Το LNS παρέχει μια πλατφόρμα, η οποία υποστηρίζει διαλειτουργικές εφαρμογές ικανές να τρέξουν πάνω στην πλατφόρμα LONWORKS. Τα εργαλεία που υποστηρίζει χρησιμοποιούνται για το σχεδιασμό, την εγκατάσταση, τη ρύθμιση, τη λειτουργία και τη συντήρηση των LONWORKS δικτύων. Επίσης υποστηρίζει ένα μοντέλο πελάτη/διακομιστή έτσι ώστε πολλαπλές εφαρμογές να μπορούν να είναι ενεργές στο δίκτυο την ίδια χρονική στιγμή, επιτρέποντας παράλληλα στους χρήστες να εγκαθιστούν συσκευές και να προβαίνουν ταυτόχρονα και σε τυχόν επισκευές του δικτύου.

3.1.5.2 ISI

Κάθε συσκευή σε ένα αυτο-εγκαθιστώμενο δίκτυο είναι υπεύθυνη για τη ρύθμισή της καθώς δεν βασίζεται σε κάποιο διακομιστή διαχείρισης δικτύου (network management server) για να συντονίσει το δίκτυο. Επειδή λοιπόν κάθε συσκευή είναι υπεύθυνη για την αυτορύθμισή της πρέπει να υπάρχει ένα κοινό πρωτόκολλο το οποίο να διασφαλίζει τη συμβατότητα μεταξύ συσκευών, οι οποίες καθορίζουν μόνες τους τη λειτουργία τους. Το πρωτόκολλο αυτό το οποίο υλοποιεί την αυτο-εγκατάσταση συσκευών πάνω στη LONWORKS πλατφόρμα ονομάζεται ISI πρωτόκολλο και αποτελεί τη βάση για την υλοποίηση αυτής της εργασίας. Το ISI πρωτόκολλο μπορεί να χρησιμοποιηθεί για δίκτυα χωρητικότητας μέχρι 200 συσκευών ενώ παράλληλα επιτρέπει τη μεταξύ τους επικοινωνία. Μεγαλύτερα και πιο πολύπλοκα δίκτυα μπορούν είτε να εγκατασταθούν και να λειτουργήσουν σαν διαχειριζόμενα δίκτυα είτε να διασπαστούν σε μικρότερα υποσυστήματα με όριο τις 200 συσκευές, πληρώνοντας έτσι τις προϋποθέσεις της τοπολογίας του ISI και τους περιορισμούς σύνδεσης. Συσκευές οι οποίες συμμορφώνονται με τις απαιτήσεις του ISI ονομάζονται ISI συσκευές. Ένας διακομιστής που ονομάζεται LonBridge Server μπορεί να χρησιμοποιηθεί για τη δημιουργία σύνδεσης μεταξύ ενός IP δικτύου και ενός δικτύου ISI συσκευών και είναι διαθέσιμος τόσο για Windows όσο και για συστήματα Linux.

Το πρωτόκολλο ISI ερευνήθηκε σε βάθος σε αυτήν την εργασία. Η λεπτομερής ανάλυσή του καθώς και η εφαρμογή του στο σύστημα που υλοποιήθηκε παρουσιάζονται πολύ αναλυτικά σε επόμενα κεφάλαια.

3.1.6 Εργαλεία δικτύου

Τα εργαλεία δικτύου είναι εφαρμογές λογισμικού οι οποίες λειτουργούν στην κορυφή του λειτουργικού συστήματος των δικτύων με σκοπό τον σχεδιασμό, τη ρύθμιση αλλά και τη συντήρησή τους. Πολλά εργαλεία μπορούν να χρησιμοποιηθούν για την περάτωση αυτών των λειτουργιών, αλλά αυτά που χρησιμοποιούνται περισσότερο είναι :

- Network Integration Tools. Παρέχουν τις απαραίτητες λειτουργίες για την εγκατάσταση και τη διαμόρφωση του δικτύου.
- Network Diagnostic Tools. Είναι εργαλεία ειδικού σκοπού και αφορούν την παρατήρηση, ανάλυση και διάγνωση των προβλημάτων που αντιμετωπίζει το δίκτυο.
- HMI Development Tools. Είναι εργαλεία ανάπτυξης HMI εφαρμογών. Οι HMI εφαρμογές χρησιμοποιούνται για την δημιουργία διεπαφών ανθρώπου-μηχανής.
- I/O Servers. Διακομιστές γενικού σκοπού, οι οποίοι παρέχουν πρόσβαση στα LONWORKS δίκτυα.

Τα εργαλεία δικτύου τα οποία βασίζονται στο λειτουργικό σύστημα LNS, ενισχύουν τη διαλειτουργικότητα. Αυτό σημαίνει ότι μπορούν να λειτουργούν την ίδια χρονική στιγμή, στο ίδιο ακριβώς δίκτυο και να διατηρούν μια συνεκτική εικόνα των συσκευών του δικτύου. Η προσφορά της Echelon στον τομέα των εργαλείων δικτύου περιλαμβάνει το LonMaker Integration Tool και το LonScanner™ Protocol Analyzer .

3.1.6.1 LonMaker Integration Tool

Το εργαλείο αυτό είναι ένα πακέτο λογισμικού το οποίο χρησιμοποιείται για τη σχεδίαση, την εγκατάσταση και την συντήρηση ενός δικτύου LONWORKS. Βασισμένο στο LNS το εργαλείο αυτό συνδυάζει μια αρχιτεκτονική πελάτη-διακομιστή (client/server) με το περιβάλλον εργασίας του Microsoft Visio. Το αποτέλεσμα είναι ένα εργαλείο αρκετά εξελιγμένο για τη σχεδίαση δικτύων ελέγχου. Το LonMaker Integration Tool συμμορφώνεται με τις απαιτήσεις του προτύπου LNS , το οποίο επιτρέπει στους κατασκευαστές LONWORKS συσκευών να παρέχουν κατάλληλες εφαρμογές. Αυτές οι εφαρμογές ξεκινούν αυτόματα όταν το LonMaker επιλέξει την συσκευή που θα απορροφήσει την συγκεκριμένη εφαρμογή.

Στους χρήστες παρέχεται ένα αρκετά οικείο περιβάλλον CAD, το οποίο περιλαμβάνει μια σειρά από σχήματα ικανά να χρησιμοποιηθούν από τους χρήστες για την σχεδίαση του δικτύου. Η σχεδίαση είναι πάρα πολύ απλή και περιλαμβάνει σχήματα για διάφορα στοιχεία των δικτύων (κανάλια, προφίλ συσκευών), αλλά και τη δημιουργία υποσυστημάτων (δίκτυο μέσα σε δίκτυο).

Ο χρόνος εγκατάστασης ενός δικτύου μπορεί να ελαχιστοποιηθεί και αυτό εξαρτάται από την ικανότητα του εκάστοτε εργαλείου δικτύου να διαμορφώνει σωστά και γρήγορα τις συσκευές. Οι συσκευές μπορούν να αναγνωριστούν και να εγκατασταθούν πολύ εύκολα στο δίκτυο. Αυτό επιτυγχάνεται μέσω του Neuron ID , του service pin ή και μέσω άλλων εντολών. Τέλος το LonMaker είναι ένα εργαλείο το οποίο επιδέχεται πολλές επεκτάσεις και βελτιώσεις όσον αφορά τη δομή του, ενώ παράλληλα έχει και τη δυνατότητα να καλύπτει ολόκληρο τον κύκλο ζωής του δικτύου.

3.1.6.2 LonScanner Protocol Analyzer

Ο αναλυτής πρωτόκολλου LonScanner είναι ένα πακέτο λογισμικού το οποίο παρέχει διαγνωστικά εργαλεία για το δίκτυο με σκοπό την παρατήρηση, τη διάγνωση και την ανάλυση της συμπεριφοράς του. Μπορεί να χρησιμοποιηθεί για τη συλλογή και την αποθήκευση όλων των πακέτων της πληροφορίας σε κάποιο LONWORKS κανάλι. Τα πακέτα αποθηκεύονται σε αρχεία καταγραφής, τα οποία μπορούν αργότερα να προβληθούν και να αναλυθούν. Αυτό το εργαλείο είναι ένα εξελιγμένο εργαλείο ανάλυσης και μπορεί επίσης να εξετάσει οποιοδήποτε πακέτο τη στιγμή που λαμβάνεται αλλά και να συνδέσει σε αυτό άλλα σχετιζόμενα πακέτα, με σκοπό βέβαια την βοήθεια του χρήστη στην κατανόηση του τρόπου λειτουργίας του δικτύου.

Έχει τη δυνατότητα να παράσχει αναλυτικό κείμενο για το κάθε πακέτο αλλά και μια περιγραφή της υπηρεσίας του πρωτοκόλλου (LONWORKS) που χρησιμοποιήθηκε για τη μετάδοση του πακέτου στο δίκτυο. Τέλος παρέχει ένα εργαλείο για τον έλεγχο και τη συλλογή στατιστικών δεδομένων όσον αφορά τη κυκλοφορία των πακέτων στο δίκτυο. Εξετάζει τη συμπεριφορά του δικτύου αφού μέσα στα δεδομένα της στατιστικής περιλαμβάνονται αποτελέσματα μετρήσεων για λάθη στα πακέτα, μετρήσεις για τα συνολικά πακέτα, δηλαδή για το φορτίο του δικτύου.

Κεφάλαιο 4

Το πρωτόκολλο LonTalk

4.1 Επισκόπηση

Η δημιουργία της πλατφόρμας LONWORKS βασίστηκε σε ένα πρωτόκολλο που ονομάζεται ISO/IEC 14908-1 Control Network Protocol (CNP). Υπάρχουν πολλές υλοποιήσεις αυτού του πρωτοκόλλου. Μια από αυτές τις υλοποιήσεις είναι και η υλοποίηση της Echelon, η οποία ονομάζεται πρωτόκολλο LonTalk ANSI/EIA 709.1 Control Networking Standard. Το πρωτόκολλο αυτό είναι η καρδιά του LONWORKS συστήματος παρέχοντας μια σειρά υπηρεσιών επικοινωνίας για την εφαρμογή η οποία αναπτύσσεται σε μια συσκευή. Αυτές οι υπηρεσίες δίνουν την δυνατότητα στη συσκευή να στέλνει αλλά και να δέχεται μηνύματα από άλλες συσκευές του δικτύου χωρίς να χρειάζεται επιπλέον γνώση της τοπολογίας του. Το πρωτόκολλο αυτό παρέχει επιβεβαίωση (acknowledgement) των μηνυμάτων από άκρο σε άκρο καθώς και πολλές χρήσιμες υπηρεσίες. Οι υπηρεσίες διαχείρισης δικτύου τις οποίες προσφέρει, επιτρέπουν σε πολλές περιπτώσεις την αναδιάρθρωση του δικτύου αλλά και την διάγνωση τυχόν προβλημάτων που επηρεάζουν τη γενικότερη λειτουργία του.

Το πρωτόκολλο LonTalk είναι ένα πολυεπίπεδο πρωτόκολλο επικοινωνίας, βασισμένο σε μια πολιτική μετάδοσης της πληροφορίας σε μορφή μικρών πακέτων από δεδομένα. Όπως τα πρωτόκολλα του Internet (Ethernet) έτσι και το συγκεκριμένο αυτό πρωτόκολλο ακολουθεί την πολυεπίπεδη δομή του OSI (Open Systems Interconnect), το οποίο είναι ένα μοντέλο αναφοράς. Με τη διαφορά όμως ότι το πρωτόκολλο LonTalk έχει σχεδιαστεί και για κάποιες άλλες συγκεκριμένες απαιτήσεις των δικτύων ελέγχου και όχι μόνο για μετάδοση δεδομένων.

Τα επτά επίπεδα του OSI με τα οποία συμμορφώνεται και το πρωτόκολλο LonTalk (παρέχοντας συγκεκριμένες υπηρεσίες) παρουσιάζονται στους δύο πίνακες που ακολουθούν. Στον πίνακα 4.1 αποτυπώνονται τα πρώτα 3 επίπεδα του πρωτοκόλλου OSI.

Αριθμός	Επίπεδο πρωτοκόλλου	Σκοπός	Υπηρεσίες
1	Φυσικό επίπεδο	Προδιαγραφές επικοινωνίας	κανάλια επικοινωνίας (twisted pair, power line, radio frequency...).
2	Ζεύξης δεδομένων	Πρόσβαση στο κανάλι	κυκλικός έλεγχος λαθών (CRC), προτεραιότητα μηνυμάτων, αποφυγή συγκρούσεων.
3	Δικτύου	Παράδοση μηνυμάτων	Δρομολόγηση πακέτων, διευθυνσιοδότηση.

Πίνακας 4.1: Τα τρία πρώτα επίπεδα του μοντέλου αναφοράς OSI

Το μοντέλο του πρωτοκόλλου OSI χρησιμοποιείται ως σημείο αναφοράς για τη σύγκριση των χαρακτηριστικών αλλά και της λειτουργίας των διαφόρων πρωτοκόλλων επικοινωνίας. Το βασικό χαρακτηριστικό του είναι η διασύνδεση μεταξύ των επιπέδων, η οποία υπαγορεύει τις προδιαγραφές της αλληλεπίδρασής τους. Συνήθως τα επίπεδα του πρωτοκόλλου είναι αυστηρά διαχωρισμένα μεταξύ τους: αξιοποιούν τις υπηρεσίες του κατώτερου επιπέδου τους και προσφέρουν υπηρεσίες στο ανώτερό τους, αλλά το καθένα δεν παρεμβαίνει στις λειτουργίες του άλλου—πιθανόν να μη γνωρίζει καν γι' αυτές.

Αυτός ο λογικός διαχωρισμός των επιπέδων διευκολύνει πολύ τη μελέτη της συμπεριφοράς των πρωτοκόλλων και επιτρέπει τη σχεδίαση πολύπλοκων και αξιόπιστων στοιβών

πρωτοκόλλων. Μία συγκεκριμένη υλοποίηση του μοντέλου, με καθορισμένα πρωτόκολλα για κάθε επίπεδο, ονομάζεται στοίβα πρωτοκόλλων ή απλά στοίβα.

Στον πίνακα 4.2 αποτυπώνονται τα 4 τελευταία επίπεδα του πρωτοκόλλου.

Αριθμός	Επίπεδο πρωτοκόλλου	Σκοπός	Υπηρεσίες
4	Μεταφοράς	Αξιοπιστία από άκρο σε άκρο	Έλεγχος για διπλότυπα μηνύματα, επιβεβαίωση μηνυμάτων από άκρο σε άκρο, αλληλουχία πακέτων.
5	Συνόδου	Έλεγχος	Αίτηση-Απόκριση, Αναγνώριση μηνυμάτων.
6	Παρουσίασης	Ερμηνεία δεδομένων	μεταβλητές δικτύου, μηνύματα εφαρμογών.
7	Εφαρμογής	Συμβατότητα εφαρμογής	Υπηρεσίες δικτύου, διάφοροι τύποι δεδομένων και άλλων αντικειμένων, ιδιότητες διαμόρφωσης, μεταφορά αρχείων.

Πίνακας 4.2: Τα τέσσερα τελευταία επίπεδα του μοντέλου αναφοράς OSI

Στη συνέχεια του κεφαλαίου δίνεται μια περιγραφή των διαφόρων επιπέδων του μοντέλου OSI καθώς και μια περιγραφή των υπηρεσιών που προσφέρει το πρωτόκολλο LonTalk σε κάθε επίπεδο.

4.2 Φυσικό επίπεδο

Το επίπεδο αυτό καθορίζει τη μετάδοση μιας σειράς από bits πάνω σε ένα κανάλι επικοινωνίας. Το κανάλι όπως έχουμε πει είναι ένα μέσο μεταφοράς πακέτων. Το φυσικό επίπεδο εξασφαλίζει ότι όταν σταλεί ένα 1-bit από μία συσκευή-αποστολέα θα ληφθεί ως 1-bit από μια συσκευή-παραλήπτη. Το πρωτόκολλο LonTalk είναι ανεξάρτητο του καναλιού και για αυτό το λόγο μια LONWORKS συσκευή μπορεί να συνδεθεί σε οποιονδήποτε πομποδέκτη, ο οποίος δημιουργεί τη σύνδεση στο κανάλι. Υπάρχουν πομποδέκτες διαθέσιμοι για πολλά είδη καναλιών (twisted pair, link power, power line, radio frequency (RF), fiberoptic). Η φυσική μορφή ενός καναλιού εξαρτάται από το μέσο που χρησιμοποιεί όπως για παράδειγμα, ένα κανάλι συνεστραμμένου ζεύγους χρησιμοποιεί ένα καλώδιο συνεστραμμένου ζεύγους ή ένα κανάλι ραδιοσυχνότητας χρησιμοποιεί μια φέρουσα ραδιοσυχνότητα ή μια γραμμή ισχύος ένα συγκεκριμένο εύρος συχνοτήτων. Ένα δίκτυο LONWORKS μπορεί να αποτελείται από πολλά είδη καναλιών τα οποία επικοινωνούν μέσω δρομολογητών, οι οποίοι δρομολογούν τα πακέτα μεταξύ των καναλιών. Ο ρυθμός μετάδοσης των καναλιών εξαρτάται από το μέσο αλλά και τη σχεδίαση του πομποδέκτη. Πολλοί πομποδέκτες, με διαφορετικό ρυθμό μετάδοσης, μπορούν να σχεδιαστούν για ένα κανάλι, με σκοπό την αύξηση της ρυθμικόδοσης αλλά και τη μείωση του κόστους μιας συσκευής.

4.2.1 Τύποι καναλιών

Ο πίνακας 4.3 της σελίδας 50 παρουσιάζει μια λίστα με τα συνηθέστερα κανάλια επικοινωνίας τα οποία χρησιμοποιούνται στα LONWORKS δίκτυα.

4.2.1.1 PL-20

Το πιο εύκολο κανάλι όσον αφορά την εγκατάσταση, σε οικιστικά αλλά και γενικότερα σε κτιριακά συστήματα είναι αυτό που μπορεί να επικοινωνεί με την ήδη υπάρχουσα καλωδίωση της μεταφοράς ηλεκτρικού ρεύματος (Power Line wiring). Οι συσκευές οι οποίες επικοινωνούν με αυτό τον τρόπο μπορούν απλά να συνδεθούν σε μια πρίζα χωρίς να είναι απαραίτητη κάποια περαιτέρω καλωδίωση.

Οι πομποδέκτες οι οποίοι χρησιμοποιούν αυτό το κανάλι, προσφέρουν την πιο ισχυρή αλλά και την πιο οικονομικά αποδοτική λύση επικοινωνίας. Η Echelon προσφέρει αυτούς τους PL-20 πομποδέκτες, οι οποίοι αποτελούν ένα πλήρες σύστημα με σκοπό την παροχή ενός

Όνομα	ID	Μέσο	Ρυθμός μετάδοσης
IP-852	154	IP-852 Tunneling	N/A
PL-20A	15	CENELEC A-band Power Line	2613bps
PL-20C	16	CENELEC C-band Power Line w/access protocol	156.3kbps
PL-20N	17	CENELEC C-band Power Line w/o access protocol	156.3kbps
TP/FT-10	4	Free Topology Twisted Pair	78.13kbps
TP/XF-1250	3	Transformer-I	1.25Mbps

Πίνακας 4.3: Τύποι καναλιών LONWORKS

συστήματος αξιόπιστης σηματοδοσίας περιορισμένου εύρους ζώνης (BPSK), αλλά και ενός πυρήνα Neuron που να μπορεί να τρέχει με επιτυχία τις διάφορες εφαρμογές.

Οι πομποδέκτες αυτοί χρησιμοποιούν BPSK σηματοδοσία σε ένα εύρος συχνοτήτων 115kHz–132kHz.



Σχήμα 4.1: PL-20 φέρουσες συχνότητες

Όπως αναφέρθηκε και στην ενότητα της σηματοδοσίας μέσω γραμμών μεταφοράς οι PL-20 transceivers εφαρμόζουν ψηφιακή επεξεργασία σήματος σε συνδυασμό με την χρήση κάποιων αλγορίθμων διόρθωσης των επιπέδων θορύβου αλλά και γενικότερα των λαθών που προκύπτουν στη μετάδοση. Αυτά τα χαρακτηριστικά έχουν μετατρέψει τους πομποδέκτες αυτούς σε πλήρως αξιόπιστες συσκευές επικοινωνίας.

Κλείνοντας την υποενότητα αυτή θα ήθελα να κάνω μια απλή αναφορά στους τρόπους με τους οποίους μπορεί να γίνει η σύνδεση των πομποδεκτών με το κανάλι επικοινωνίας PL-20. Αυτή η σύνδεση ονομάζεται *προσαρμογή ή ζεύξη (coupling)* του πομποδέκτη στη γραμμή του ρεύματος.



Σχήμα 4.2: Πομποδέκτες PL3120 και PL3150

Υπάρχουν δύο τέτοιες μέθοδοι προσαρμογής : *line-to-neutral* (γραμμής –ουδέτερου) και *line-to-earth* (γραμμής –γης) προσαρμογή. Τα κανάλια PL-20C(Line-Neutral), PL-20N(Line-Neutral), and PL-20A(Line-Neutral), προσδιορίζουν μια σύζευξη κυκλωμάτων τα οποία μεταδίδουν και λαμβάνουν σήματα επικοινωνίας μέσω των αγωγών του ρεύματος γραμμής –ουδέτερου. Τα κανάλια PL-20C και PL-20N μπορούν να χρησιμοποιήσουν σύζευξη line-to-earth, δηλαδή μετάδοση και λήψη σημάτων μέσω των αγωγών γραμμής –γης. Ακολουθεί ένας πίνακας με κάποιες προδιαγραφές (παραμέτρους) των διαφόρων καναλιών PL-20.

Παράμετρος	PL-20(L-N)	PL-20(L-E)	PL-20A(L-N)
Τεχνική σύζευξης	Line-to-neutral	Line-to-earth	Line-to-neutral
Ρυθμός μετάδοσης	5kbps	5kbps	3600bps
Σηματοδοσία	BPSK	BPSK	BPSK
Έυρος συχνοτήτων	125kHz–140kHz	125kHz–140kHz	70kHz–95kHz
Αντίσταση εξόδου	Z=6W (μέγιστο)	Z=8, 15W (μέγιστο) για 120,240V	Z=1.1W (μέγιστο)70kHz–95kHz
Αντίσταση εισόδου	100W (ελάχιστο)	100W (ελάχιστο)	500W (ελάχιστο)70kHz–95kHz

Πίνακας 4.4: Προδιαγραφές καναλιών PL-20

Η μέθοδος σύζευξης που χρησιμοποιείται στην δική μας εφαρμογή αναλύεται λεπτομερώς, με όλες τις παραμέτρους που τη συνοδεύουν, στο δεύτερο μέρος της εργασίας στο οποίο γίνεται η πλήρης ανάλυση του υλικού.

4.3 Επίπεδο ζεύξης δεδομένων

Το επίπεδο ζεύξης δεδομένων καθορίζει τις μεθόδους πρόσβασης στο μέσο (κανάλι) αλλά και την αποκωδικοποίηση των δεδομένων με σκοπό την εξασφάλιση της αποτελεσματικής χρήσης του καναλιού. Η σειρά των bits του φυσικού επιπέδου διαχωρίζεται σε πλαίσια δεδομένων. Το επίπεδο αυτό καθορίζει πότε μια συσκευή μπορεί να μεταδίδει πλαίσια από δεδομένα και πότε μια συσκευή- αποδέκτης μπορεί να λαμβάνει αυτά τα πλαίσια δεδομένων. Επίσης ορίζεται ένας μηχανισμός προτεραιότητας, με σκοπό την ασφαλή παράδοση πολύ σημαντικών μηνυμάτων. Προτού μια συσκευή μεταδώσει, περιμένει ώστε το κανάλι να καταστεί αδρανές. Επειδή όμως πολλές συσκευές μπορεί να περιμένουν ταυτόχρονα ώστε το κανάλι να καταστεί αδρανές για να μεταδώσουν, κάθε συσκευή περιμένει ένα τυχαίο χρονικό διάστημα προτού μεταδώσει. Ένα μοναδικό χαρακτηριστικό του πρωτοκόλλου LonTalk είναι ότι ο αριθμός των χρονικών περιόδων (slots) αυξάνεται, όσο αυξάνεται και το φορτίο του δικτύου. Αυτό το χαρακτηριστικό γνώρισμα του πρωτοκόλλου, δίνει τη δυνατότητα για αξιόπιστη και σταθερή απόδοση του δικτύου, ακόμα και σε δίκτυα τα οποία έχουν φορτωθεί πολύ.

4.3.1 Πρόσβαση στο μέσο

Το πρωτόκολλο LonTalk χρησιμοποιεί ένα αλγόριθμο (MAC) για την πρόσβαση στο μέσο, ο οποίος επιτρέπει σε ένα κανάλι να λειτουργεί κοντά στο μέγιστο της χωρητικότητάς του. Άλλοι MAC αλγόριθμοι τείνουν να υποβαθμίζουν την ικανότητα του καναλιού, λόγω των συγκρούσεων μεταξύ των πακέτων οι οποίες καταναλώνουν το διαθέσιμο εύρος ζώνης συχνοτήτων. Οι συχνότερα χρησιμοποιούμενοι αλγόριθμοι πρόσβασης είναι οι : *token ring*, *token bus* και *CSMA*.

Ο αλγόριθμος που χρησιμοποιείται από το πρωτόκολλο LonTalk ανήκει στην οικογένεια αλγορίθμων του CSMA.

4.3.1.1 Αλγόριθμος πρόσβασης CSMA

Η οικογένεια των αλγορίθμων CSMA είναι καλύτερα προσαρμοσμένη στις απαιτήσεις των δικτύων ελέγχου από ότι οι άλλοι MAC αλγόριθμοι. Απαιτούν το κανάλι να είναι αδρανές έτσι ώστε οι συσκευές του δικτύου να μπορούν να μεταδίδουν. Όμως όταν το κανάλι

καταστεί αδρανές, οι αλγόριθμοι δεν συμπεριφέρονται με τον ίδιο τρόπο. Αυτό έχει σαν αποτέλεσμα διαφορετική συμπεριφορά του δικτύου, κάτω από συνθήκες μεγάλου φόρτου δεδομένων.

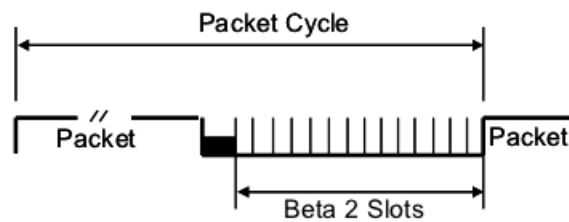
Μερικοί CSMA αλγόριθμοι χρησιμοποιούν διακριτά χρονικά διαστήματα (slots) για να αποκτήσουν πρόσβαση στο κανάλι. Με τον περιορισμό της πρόσβασης μιας συσκευής στο κανάλι κάποιες συγκεκριμένες χρονικές περιόδους, μειώνεται η πιθανότητα σύγκρουσης δύο ή περισσότερων πακέτων. Η περισσότερο συχνά εφαρμοζόμενη μέθοδος περιορισμού πρόσβασης στο κανάλι υλοποιείται από τον αλγόριθμο *p-persistent CSMA*. Σε αυτόν τον αλγόριθμο, όταν μια συσκευή έχει να στείλει κάποιο μήνυμα, το κάνει σε μια τυχαία χρονική στιγμή με πιθανότητα p .

Τα δίκτυα ελέγχου μπορεί να αποτελούνται από χιλιάδες συσκευές και πολλά κανάλια. Εξαιτίας των διαφορετικών χαρακτηριστικών που διαθέτουν τα κανάλια αλλά και της ανάγκης να υποστηριχθούν μεγάλες αποστάσεις στα δίκτυα, τα LONWORKS δίκτυα πρέπει να είναι σε θέση να υποστηρίξουν κανάλια υψηλής αλλά και χαμηλής ταχύτητας. Το πρωτόκολλο πρόσβασης CSMA/CD συμπεριφέρεται μη-αποδοτικά σε περιόδους υψηλού φορτίου στο δίκτυο και γι αυτό το λόγο δεν χρησιμοποιείται σε εφαρμογές ελέγχου.

Το P-persistent CSMA συμπεριφέρεται πολύ αποδοτικά για μικρές τιμές της πιθανότητας p . Η οικογένεια των CSMA αλγορίθμων, δεν απαιτεί κάποια τοπολογία δικτύου σε μορφή δακτυλίου, συγχρονισμό ή επαναδιαμόρφωση αλλά αντίθετα επιτρέπει την προσθήκη νέων συσκευών στο δίκτυο αλλά και την αποχώρηση από αυτό ήδη υπάρχοντων συσκευών. Υποστηρίζει πολλές συσκευές και είναι πολύ φθηνή η υλοποίησή του αλγορίθμου αυτό πάνω το υλικό καθιστώντας τον πλήρως αποδοτικό.

Το πρωτόκολλο LonTalk χρησιμοποιεί μια παραλλαγή του P-persistent CSMA, το predictive p-persistent CSMA. Όπως στο P-persistent CSMA, οι LONWORKS συσκευές παρεμβαίνουν με τυχαίο τρόπο στο κανάλι. Έτσι αποφεύγεται η αναπόφευκτη σύγκρουση πακέτων όταν δύο ή περισσότερες συσκευές περιμένουν ώστε το κανάλι να καταστεί αδρανές, για να μεταδώσουν. Αν ο χρόνος στον οποίο περιμένουν έχει την ίδια διάρκεια τότε έχουμε συγκρούσεις. Προσπαθώντας λοιπόν να δημιουργήσουμε μια τυχαία καθυστέρηση πρόσβασης στο κανάλι αποφεύγουμε σε μεγάλο βαθμό τις συγκρούσεις. Αυτές οι τυχαίες καθυστερήσεις ονομάζονται beta 2 slots και αποτυπώνονται στο παρακάτω σχήμα.

Στο P-persistent CSMA όταν μια συσκευή ήθελε να μεταδώσει κάποιο πακέτο, το έκανε σε μια τυχαία χρονική στιγμή (slot) με πιθανότητα p . Όμως ο αλγόριθμος αυτός (predictive p-persistent CSMA) έχει το εξής επιπλέον χαρακτηριστικό : Η πιθανότητα p



Σχήμα 4.3: Beta 2 Slots

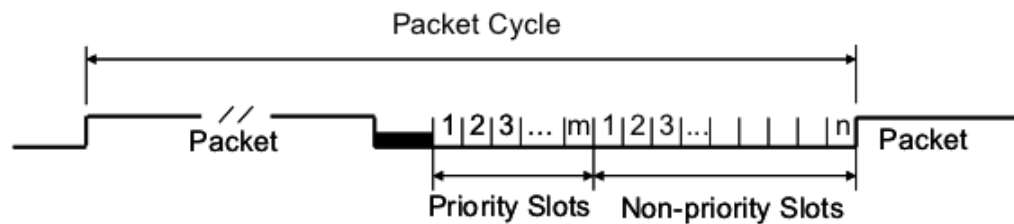
μπορεί να προσαρμόζεται δυναμικά στο φορτίο του δικτύου. Όταν το δίκτυο είναι αδρανές, οι συσκευές μπορούν να δημιουργήσουν 16 τυχαία beta 2 slots. Έτσι η μέση καθυστέρηση σε ένα αδρανές δίκτυο είναι πλάτους 8 slots. Όταν αυξάνεται το εκτιμώμενο φορτίο του δικτύου, ο αριθμός των slots μεγαλώνει κατά ένα παράγοντα n ο οποίος κυμαίνεται μεταξύ 1 και 63. Έτσι ο μεγαλύτερος αριθμός beta 2 slots είναι $63 \cdot 18 = 1008$.

Στο πρωτόκολλο LonTalk ο παράγοντας n ονομάζεται εκτιμώμενο *backlog* (καθυστέρηση). Αυτή η μέθοδος εκτίμησης της καθυστέρησης αλλά και αλλά και η δυναμική προσαρμογή της πρόσβασης στο κανάλι επιτρέπει στο LonTalk να διατηρεί ένα μικρό αριθμό beta 2 slots σε περιόδους μικρού φορτίου, ενώ παράλληλα διατηρεί το πλεονέκτημα της κατοχής πολλών beta 2 slots σε περιόδους μεγάλης φόρτισης του καναλιού με αποτέλεσμα την αποφυγή των συγκρούσεων.

4.3.1.2 Προτεραιότητα

Μια βασική απαίτηση των δικτύων ελέγχου είναι η έγκαιρη απόκριση στην αποστολή μηνυμάτων τα οποία είναι σε προτεραιότητα. Το πρωτόκολλο LonTalk διαθέτει ένα μηχανισμό προτεραιότητας με σκοπό την βελτίωση του χρόνου απόκρισης στα κρίσιμα πακέτα. Το πρωτόκολλο επιτρέπει στο εργαλείο διαχείρισης του δικτύου που χρησιμοποιείται να δεσμεύσει ένα σταθερό αριθμό slots τα οποία όπως είδαμε ονομάζονται beta 2 slots, ως τα slots προτεραιότητας, τα οποία δίνονται στις συσκευές προτεραιότητας. Ο αριθμός αυτών των slots μπορεί να είναι από 0 έως 127. Το εργαλείο διαχείρισης του δικτύου το οποίο αναθέτει αυτά τα χρονικά διαστήματα σε μεμονωμένες συσκευές, διασφαλίζει ότι σε μία και μόνο μία συσκευή έχει ανατεθεί το συγκεκριμένο slot προτεραιότητας πάνω στο κανάλι.

Το σχήμα 4.4 αποτυπώνει τη μορφή των priority και non-priority slots, τα οποία ελέγχονται από το εργαλείο διαχείρισης που χρησιμοποιεί το δίκτυο.



Σχήμα 4.4: Slots προτεραιότητας και μη-προτεραιότητας

4.3.2 Ρυθμαπόδοση καναλιών

Η ρυθμαπόδοση του καναλιού εξαρτάται από το ρυθμό μετάδοσης των bits, από τα χαρακτηριστικά του πομποδέκτη, το μέσο μέγεθος των πακέτων καθώς και από την χρήση των επιβεβαιώσεων και των προτεραιοτήτων στο κανάλι. Το μέσο μέγεθος του πακέτου είναι μεταξύ 10 και 16 bytes και εξαρτάται από το μέγεθος του πεδίου των δεδομένων αλλά και από το μέγεθος του πεδίου της διευθυνσιοδότησης. Το μέγιστο μέγεθος πακέτου είναι 249 bytes.

Σε περιπτώσεις στις οποίες έχουμε χαμηλό ρυθμό μετάδοσης ή μεγάλα σε μέγεθος πακέτα, η ρυθμαπόδοση (throughput) του καναλιού περιορίζεται από το χρόνο μετάδοσης των πακέτων αλλά και από τη μέση καθυστέρηση πρόσβασης στο κανάλι.

Σε μεγαλύτερους ρυθμούς μετάδοσης με μικρότερα σε μέγεθος πακέτα, η ισχύς επεξεργασίας πακέτων του πυρήνα Neuron περιορίζει την απόδοση του καναλιού.

Ο πίνακας που ακολουθεί αποτυπώνει κατα προσέγγιση την ρυθμαπόδοση του δικτύου συναρτήσει του ρυθμού μετάδοσης και του μεγέθους των πακέτων.

Ρυθμός Μετάδοσης	Μέγιστος αριθμός πακέτων/ sec	Σταθερός αριθμός πακέτων/ sec
4.883	25	20
9.766	45	35
19.531	110	85
39.063	225	180
78.125	400	320

Πίνακας 4.5: Ρυθμαπόδοση καναλιού με 12-byte πακέτα

4.4 Επίπεδο δικτύου

Το επίπεδο δικτύου καθορίζει με ποιο τρόπο τα πακέτα δρομολογούνται από μία συσκευή-αποστολέα σε μία ή περισσότερες συσκευές –παραλήπτες. Καθορίζει την ονοματοδοσία και τη διευθυνσιοδότηση των συσκευών, ώστε να εξασφαλιστεί η σωστή παράδοση των πακέτων. Τα μηνύματα είναι δυνατό να απευθύνονται σε μία συσκευή, σε μια ομάδα συσκευών, αλλά και σε όλες τις συσκευές ενός δικτύου. Η διευθυνσιοδότηση ομάδων συσκευών συντελεί στην μείωση της κυκλοφορίας των πακέτων μέσα στο δίκτυο αφού υποστηρίζει την παράδοση ενός μεμονωμένου μηνύματος σε πολλαπλές συσκευές. Το επίπεδο αυτό καθορίζει επίσης τη δρομολόγηση των πακέτων από μία ή περισσότερες συσκευές σε συσκευές οι οποίες λειτουργούν πάνω σε διαφορετικά κανάλια επικοινωνίας.

Οι διευθύνσεις σχηματίζονται χρησιμοποιώντας μια ιεραρχική δομή, η οποία υποστηρίζει τη χρήση δρομολογητών. Οι δρομολογητές αυτοί χρησιμοποιούνται για φιλτράρουν τα μηνύματα, βασισμένοι βέβαια πάντα στις διευθύνσεις προορισμού των μηνυμάτων αυτών. Υποστηρίζοντας δρομολογητές, το πρωτόκολλο LonTalk βοηθά και στη δημιουργία μεγάλων δικτύων με χιλιάδες συσκευές

4.4.1 Τύποι διευθύνσεων

Η διεύθυνση καθορίζει μοναδικά μια συσκευή ή μια ομάδα από συσκευές σε ένα δίκτυο. Μπορεί να εκχωρηθεί σε μια συσκευή αλλά μετά τη δημιουργία της συσκευής είναι δυνατό

Ρυθμός Μετάδοσης	Μέγιστος αριθμός πακέτων/ sec	Σταθερός αριθμός πακέτων/ sec
4.883	7	5
9.766	13	10
19.531	25	20
39.063	50	40
78.125	100	80

Πίνακας 4.6: Ρυθμαπόδοση καναλιού με 64-byte πακέτα

να τροποποιηθεί. Στη συνέχεια παρουσιάζονται οι διαθέσιμοι τύποι διευθύνσεων σε ένα δίκτυο LONWORKS.

4.4.1.1 Φυσική διεύθυνση

Κάθε συσκευή τεχνολογίας LONWORKS περιλαμβάνει ένα μοναδικό 48-bit αναγνωριστικό το οποίο όπως έχουμε δει ονομάζεται Neuron ID. Αυτό το Neuron ID μπορεί να χρησιμοποιηθεί σαν μια διεύθυνση, η οποία εκχωρείται σε μια συσκευή και δεν τροποποιείται καθ' όλη τη διάρκεια ζωής της. Δεν χρησιμοποιείται όμως ως η μοναδική μορφή διευθυνσιοδότησης, επειδή υποστηρίζει μόνο επικοινωνία ένα-προς-ένα αλλά και επειδή προκαλεί επιπλοκές στην αντικατάσταση συσκευών.

Για την απλοποίηση της δρομολόγησης των πακέτων, το πρωτόκολλο LONWORKS προσφέρει μια ιεραρχική μορφή διευθυνσιοδότησης χρησιμοποιώντας τρία είδη διευθύνσεων : κόμβου (*node*) , υποδικτύου (*subnet*) και τομέα (*domain*) . Για την περαιτέρω διευκόλυνση της διευθυνσιοδότησης διάσπαρτων συσκευών, το πρωτόκολλο ορίζει μια άλλη κατηγορία διευθύνσεων οι οποίες ονομάζονται διευθύνσεις ομάδων (*groups*). Η χρήση δυναμικά καταχωρίσεων διευθύνσεων σε αντίθεση με την χρήση σταθερών τύπων διευθύνσεων απλοποιεί σημαντικά την αντικατάσταση των συσκευών σε ένα δίκτυο. Η συσκευή αντικατάστασης λαμβάνει την ίδια διεύθυνση με την συσκευή την οποία αντικαθιστά.

4.4.1.2 Διεύθυνση τομέα

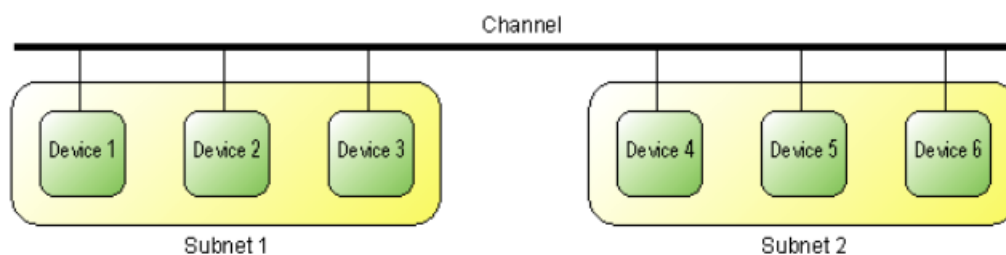
Ένας τομέας (domain) είναι μια συλλογή από συσκευές σε ένα ή περισσότερα κανάλια. Η επικοινωνία είναι δυνατή μόνο μεταξύ συσκευών στο ίδιο domain. Ως εκ τούτου ένα domain σχηματίζει ένα εικονικό δίκτυο. Πολλά domains μπορούν να καταλάβουν ίδια κανάλια, με αποτέλεσμα να βοηθούν στην αποφυγή παρεμβολών μεταξύ συσκευών σε διαφορετικά δίκτυα.

Για παράδειγμα μπορεί να έχουμε δύο γειτονικά κτίρια τα οποία χρησιμοποιούν συσκευές με ενσωματωμένους power line πομποδέκτες, οι οποίοι να λειτουργούν στις ίδιες συχνότητες. Για την αποφυγή παρεμβολών θα πρέπει οι συσκευές να ρυθμιστούν κατάλληλα, ώστε να ανήκουν σε διαφορετικά domains. Ένας τομέας είναι δυνατό να αναγνωριστεί μέσω του domain ID. Το domain ID μπορεί να είναι από 0 έως 6 bytes.

4.4.1.3 Διεύθυνση υποδικτύου

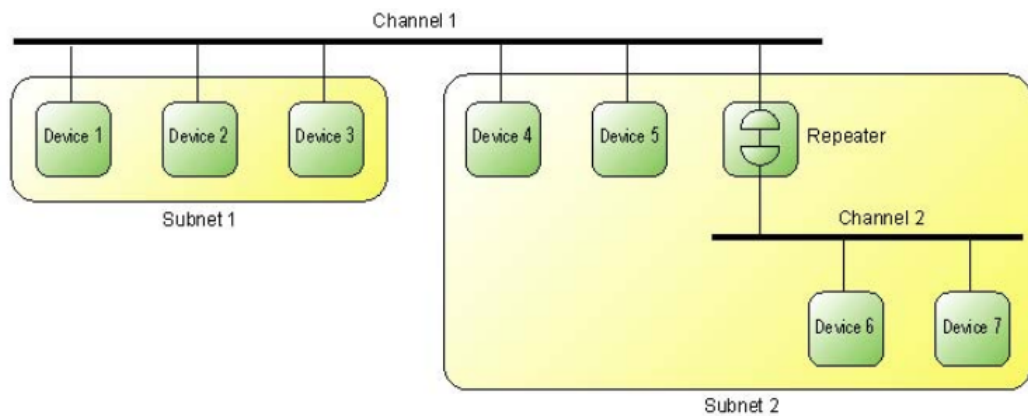
Ένα υποδίκτυο είναι μια λογική συλλογή, μέχρι 127, συσκευών οι οποίες βρίσκονται σε ένα τομέα. Μέχρι 255 υποδίκτυα είναι δυνατό να οριστούν μέσα σε ένα τομέα. Όλες οι συσκευές μέσα σε ένα υποδίκτυο πρέπει να βρίσκονται στο ίδιο τμήμα (segment).

Το παρακάτω σχήμα απεικονίζει έξι συσκευές εγκατεστημένες σε δύο υποδίκτυα. Αυτές οι συσκευές είναι συνδεδεμένες στο ίδιο κανάλι.

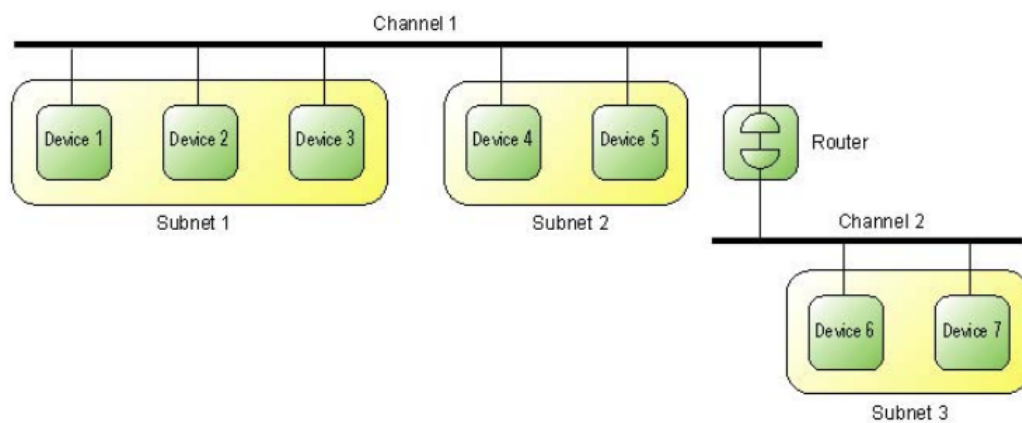


Σχήμα 4.5: Δύο υποδίκτυα στο ίδιο κανάλι

Το σχήμα 4.6 απεικονίζει επτά συσκευές εγκατεστημένες σε δύο υποδίκτυα. Αυτές οι συσκευές είναι εγκατεστημένες σε δύο κανάλια επικοινωνίας, τα οποία βρίσκονται σε κοινό τμήμα (segment) λόγω της παρουσίας του επαναλήπτη. Στο σχήμα 4.7 απεικονίζονται τρία υποδίκτυα και δύο ξεχωριστοί τομείς.



Σχήμα 4.6: Δύο υποδίκτυα σε κοινό segment



Σχήμα 4.7: Τρία υποδίκτυα σε δύο segments

Ένα υποδίκτυο είναι δυνατό να αναγνωριστεί μέσω του subnet ID.

4.4.1.4 Διεύθυνση κόμβου

Κάθε συσκευή μέσα σε ένα υποδίκτυο έχει ένα μοναδικό node ID, το οποίο μπορεί να είναι μέχρι και 7 bits. Άρα είναι δυνατό να υπάρχουν μέχρι 127 συσκευές σε ένα υποδίκτυο αλλά και ένα μέγιστο της τάξης των 32,385 συσκευών (255 subnets x 127 devices per subnet) ανά τομέα.

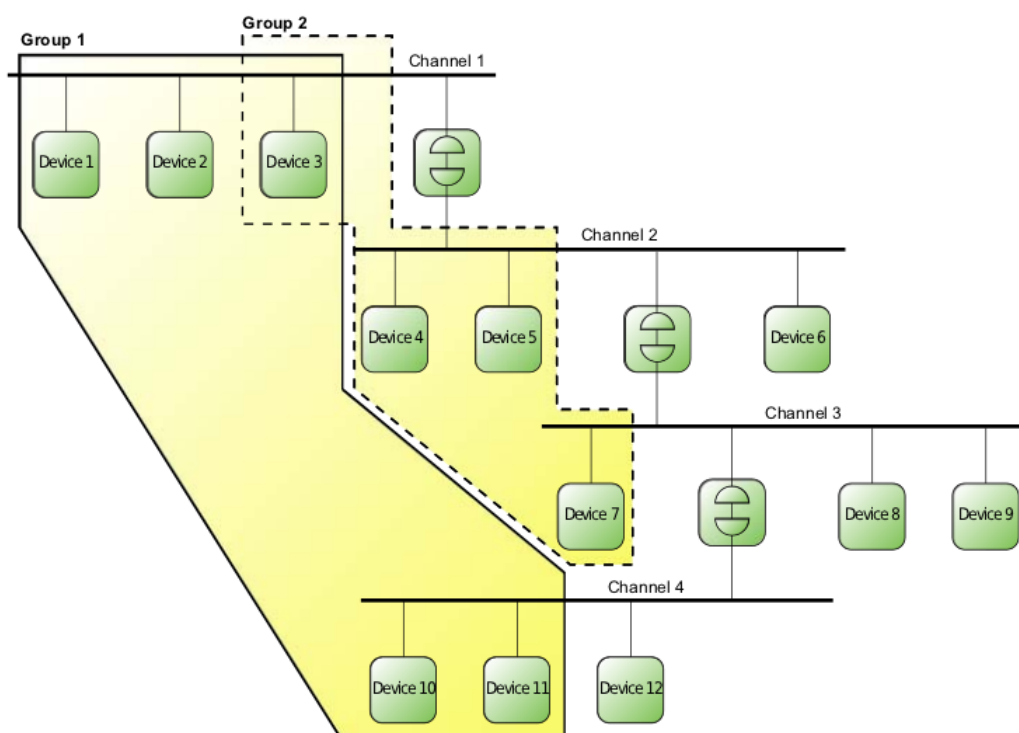
Κάθε συσκευή μπορεί να αναγνωριστεί μοναδικά σε ένα δίκτυο από το συνδυασμό των τριών IDs (domain ID, subnet ID, and node ID).

4.4.1.5 Ομάδες συσκευών

Μια ομάδα συσκευών (group) είναι μια λογική συλλογή από συσκευές μέσα σε ένα τομέα. Σε αντίθεση με το υποδίκτυο, εδώ οι συσκευές ομαδοποιούνται μαζί χωρίς να λαμβάνεται υπόψη η φυσική τους θέση μέσα στον τομέα.

Οι ομάδες είναι ένας πολύ αποτελεσματικός τρόπος να χρησιμοποιεί κανείς το εύρος ζώνης του δικτύου για μηνύματα τα οποία απευθύνονται σε πολλαπλές συσκευές (δηλαδή για συνδέσεις ένα - προς - πολλά).

Το παρακατω σχήμα απεικονίζει ένα δίκτυο με δύο ομάδες συσκευών. Η ομάδα 1 αποτελείται από τις συσκευές 1, 2, 3, 10, 11, 12. Η ομάδα 2 αποτελείται από τις 3, 4, 5, 7. Η συσκευή 3 ανήκει και στις 2 ομάδες.



Σχήμα 4.8: Ομάδες συσκευών

Μια ομάδα αναγνωρίζεται από έναν αριθμό ομάδας (1-byte), οπότε ένας τομέας μπορεί να περιέχει μέχρι 255 ομάδες. Το σχήμα 4.9 παρουσιάζει τα όρια του πρωτοκόλλου LonTalk, όσον αφορά το πλήθος των συσκευών σε ένα δίκτυο LONWORKS.

• Devices in a subnet	127
• Subnets in a domain	255
• Devices in a domain	32,385
• Domains in a network	2⁴⁸
• Maximum devices in system	32Kx2⁴⁸
• Members in a group	
♦ Unacknowledged or Repeated	No Limit
♦ Acknowledged or Request Response	63
• Groups in a domain	255
• Channels in a network	No Limit

Σχήμα 4.9: Επιτρεπτό πλήθος συσκευών σε ένα δίκτυο LONWORKS

4.5 Επίπεδο μεταφοράς

Το επίπεδο αυτό εξασφαλίζει την αξιόπιστη παράδοση των μηνυμάτων. Τα μηνύματα μπορούν να ανταλλάσσονται χρησιμοποιώντας μια υπηρεσία επιβεβαιώσεων, σύμφωνα με την οποία η συσκευή η οποία στέλνει το πακέτο περιμένει την επιβεβαίωση από τον παραλήπτη ότι το πακέτο έχει φτάσει αλλιώς το επαναμεταδίδει. Το επίπεδο αυτό καθορίζει επίσης τον τρόπο με τον οποίο εντοπίζονται αλλά και διαγραφόνται τα διπλότυπα πακέτα, τα οποία δημιουργούνται αν ένα μήνυμα έχει επαναμεταδοθεί λόγω χαμένης επιβεβαίωσης. Μηνύματα τα οποία δεν απαιτούν την αξιοπιστία αυτής της υπηρεσίας, μπορούν να χρησιμοποιήσουν άλλες υπηρεσίες όπως είναι τα επαναλαμβανόμενα μηνύματα ή τα ανεπιβεβαίωτα μηνύματα, για να μεταδωθούν χωρίς να χρειάζεται να περιμένουν για κάποια επιβεβαίωση.

4.5.1 Υπηρεσίες μηνυμάτων

Το πρωτόκολλο LonTalk προσφέρει τέσσερις βασικές υπηρεσίες μηνυμάτων : μηνύματα με επιβεβαίωση, χωρίς επιβεβαίωση, επαναλαμβανόμενα, αίτηση-απόκριση. Τις τρεις πρώτες κατηγορίες υπηρεσιών τις διαχειρίζεται το επίπεδο μεταφοράς ενώ την τελευταία το επίπεδο συνόδου.

- Η πιο αξιόπιστη υπηρεσία μετάδοσης μηνυμάτων είναι η μετάδοση μηνυμάτων με επιβεβαίωση, στην οποία ένα μήνυμα στέλνεται σε μια συσκευή ή σε μια ομάδα συσκευών και μια σειρά από μεμονωμένες επιβεβαιώσεις αναμένονται από κάθε δέκτη. Αν η επιβεβαίωση από κάποιον δέκτη δεν φτάσει, ο αποστολέας επιχειρεί πάλι την

μετάδοση. Ο αριθμός των επαναλήψεων αλλά και το χρονικό διάστημα στο οποίο ο αποστολέας περιμένει την επιβεβαίωση (timeout), δεν έχουν σταθερή τιμή.

- Μια εξίσου αξιόπιστη υπηρεσία είναι τα επαναλαμβανόμενα μηνύματα ή αλλιώς επαναλαμβανόμενα μηνύματα χωρίς επιβεβαίωση, στην οποία ένα μήνυμα στέλνεται σε μια συσκευή ή ομάδα συσκευών και δεν αναμένεται καμία απάντηση. Η υπηρεσία αυτή τυπικά χρησιμοποιείται όταν το μήνυμα μεταδίδεται σε περισσότερες από μια συσκευές, επειδή ο αριθμός των πακέτων που απαιτείται για να επαναληφθεί η μετάδοση είναι μικρότερος από τον αριθμό των πακέτων που απαιτείται για τις επιβεβαιώσεις. Η υπηρεσία αυτή πρέπει να χρησιμοποιείται αντί της υπηρεσίας με τις επιβεβαιώσεις μηνυμάτων όποτε το μέγεθος της ομάδας των συσκευών είναι μεγαλύτερο από το συνολικό αριθμό επαναλήψεων.
- Η λιγότερο αξιόπιστη υπηρεσία παράδοσης μηνυμάτων είναι η παράδοση μηνυμάτων χωρίς επιβεβαίωση, όπου το μήνυμα στέλνεται μόνο μια φορά στη συσκευή και δεν αναμένεται κάποια επιβεβαίωση. Αυτή η υπηρεσία τυπικά χρησιμοποιείται όταν τα δεδομένα στέλνονται περιοδικά από μια συσκευή σε μια άλλη, χωρίς να υπάρχει ο φόβος της απώλειάς τους. Αυτή η υπηρεσία χρησιμοποιείται στην δική μας εφαρμογή, όπου ο αισθητήρας θερμοκρασίας παίρνει περιοδικά τιμές της θερμοκρασίας από ένα χώρο. Ο κίνδυνος να χαθεί μια ενημερωμένη τιμή θερμοκρασίας μπορεί να υπάρχει αλλά δεν είναι κρίσιμος, αφού η θερμοκρασία θα ανανεωθεί ξανά στο τέλος ενός επαναλαμβανόμενου μετρητή.

4.6 Επίπεδο συνόδου

Το επίπεδο συνόδου προσθέτει έλεγχο στα δεδομένα τα οποία ανταλλάσσονται από τα χαμηλότερα επίπεδα του πρωτοκόλλου. Υποστηρίζει απομακρυσμένες ενέργειες έτσι ώστε ο πελάτης να είναι σε θέση να υποβάλει μια αίτηση σε ένα απομακρυσμένο διακομιστή και να λάβει μια απάντηση στο αίτημα του. Το επίπεδο συνόδου καθορίζει επίσης και ένα πρωτόκολλο ελέγχου ταυτότητας (authentication), το οποίο επιτρέπει στους δέκτες των μηνυμάτων να διαπιστώσουν αν ο αποστολέας έχει το δικαίωμα να μεταδίδει αυτά τα μηνύματα. Αυτό το πρωτόκολλο χρησιμοποιείται για να αποφευχθεί η μη-εξουσιοδοτημένη πρόσβαση σε συσκευές αλλά και στις εφαρμογές τους.

4.6.1 Αίτηση–Απόκριση

Η υπηρεσία αυτή χρησιμοποιείται όταν ένα μήνυμα στέλνεται σε μια συσκευή ή σε μια ομάδα συσκευών και παράλληλα απαιτούνται μεμονωμένες αποκρίσεις από τους δέκτες. Το εισερχόμενο μήνυμα επεξεργάζεται από την συσκευή λήψης, προτού παραχθεί η απόκριση. Οι αποκρίσεις ενδέχεται να περιλαμβάνουν και δεδομένα, γεγονός το οποίο καθιστά την υπηρεσία αυτή κατάλληλη για την υλοποίηση απομακρυσμένων διαδικασιών ή εφαρμογών πελάτη/διακομιστή. Η ροή των μηνυμάτων σε αυτή την υπηρεσία είναι ίδια με την ροή στην υπηρεσία επιβεβαιωμένων μηνυμάτων, με τη μόνη διαφορά ότι εδώ αντί επιβεβαίωσης στέλνεται απόκριση. Τα εργαλεία διαχείρισης του δικτύου πρέπει να λαμβάνουν υπόψη τον επιπλέον χρόνο επεξεργασίας που απαιτείται έτσι ώστε η εφαρμογή να παράξει την απόκριση.

4.6.2 Υπηρεσία ελέγχου ταυτότητας

Όταν χρησιμοποιούνται μηνύματα ελέγχου ταυτότητας (authenticated messages), ο αποδέκτης αυτών των μηνυμάτων είναι σε θέση να διαπιστώσει αν ο αποστολέας είναι εξουσιοδοτημένος να στείλει αυτά τα μηνύματα. Για παράδειγμα με τη χρησιμοποίηση ελέγχου ταυτότητας, μια ηλεκτρονική συσκευή κλειδώματος μπορεί να επαληθεύσει ότι μια αίτηση δεν προέρχεται από κάποιον που προσπαθεί να μπει στο σύστημα ενώ δεν έχει δικαίωμα πρόσβασης.

Ο έλεγχος ταυτότητας υλοποιείται με τη διανομή στις συσκευές του δικτύου 48-bit κλειδιών (ένα ανα τομέα), την στιγμή της εγκατάστασης του δικτύου. Για να γίνει δεκτό ένα μήνυμα ελέγχου ταυτότητας από τον παραλήπτη, πρέπει και ο αποστολέας αλλά και ο παραλήπτης να διαθέτουν το ίδιο κλειδί. Αυτό το κλειδί είναι διαφορετικό από το Neuron ID της συσκευής.

Το πρωτόκολλο ελέγχου ταυτότητας είναι πάντα ενεργοποιημένο σε κάθε LONWORKS συσκευή και κάθε τομέας σε κάθε συσκευή έχει ένα κλειδί ελέγχου. Δοθέντος ενός μηνύματος, εναπόκειται στον αποστολέα του να ξεκινήσει τη διαδικασία ελέγχου ταυτότητας, με το να θέσει το authentication bit στο format του πακέτου ίσο με 1. Ο αποστολέας δεν πρέπει να ξεκινήσει τη διαδικασία ελέγχου ταυτότητας εκτός αν αυτή απαιτείται από το δέκτη. Αυτό γίνεται γιατί αυτές οι διαδικασίες ελέγχου ταυτότητας καταναλώνουν διπλάσιο εύρος ζώνης σε αντίθεση με τις υπόλοιπες διαδικασίες ελέγχου.

4.7 Επίπεδο παρουσίασης

Το επίπεδο παρουσίασης προσθέτει μια δομή στα δεδομένα τα οποία ανταλλάσσονται μεταξύ των χαμηλότερων επιπέδων, μέσω της κωδικοποίησης των μηνυμάτων που πραγματοποιεί. Οι υπηρεσίες του επιπέδου αυτού παρέχονται μέσω του εσωτερικού κώδικα του πυρήνα Neuron, ο οποίος και τρέχει την εκάστοτε εφαρμογή.

4.7.1 Μηνύματα

Στο επίπεδο αυτό τα δεδομένα ανταλλάσσονται μεταξύ των εφαρμογών αφού κωδικοποιηθούν πρώτα σε μηνύματα. Κάθε μήνυμα αποτελείται από ένα κωδικό μηνύματος (1-byte) και ένα κομμάτι δεδομένων (μέχρι 228 bytes) με τη μόνη εξαίρεση να αποτελούν οι μεταβλητές δικτύου οι οποίες έχουν μέγεθος από 1 έως 31 bytes. Ο κωδικός του μηνύματος καθορίζει τον τύπο των δεδομένων που περιέχονται στο μήνυμα. Ο πίνακας 4.7 δίνει τους διαθέσιμους τύπους μηνυμάτων, οι οποίοι υποστηρίζονται από το πρωτόκολλο LonTalk καθώς επίσης και τους κωδικούς των μηνυμάτων για κάθε τύπο.

Τύπος μηνύματος	Κωδικός (hex)	Κωδικός (decimal)
Μηνύματα εφαρμογών από το χρήστη	00 – 2F	0 – 47
Μηνύματα εφαρμογών	30 – 3E	48 – 62
Μηνύματα εξωτερικού πλαισίου	40 – 4E	64 – 78
Διαγνωστικά μηνύματα	50 – 5F	80 – 95
Μηνύματα διαχείρισης δικτύου	60 – 7F	96 – 127
Μηνύματα μεταβλητών δικτύου	80 – FF	128 – 255

Πίνακας 4.7: Κωδικοί μηνυμάτων

Οι εφαρμογές τυπικά, ανταλλάσσουν μηνύματα χρησιμοποιώντας μεταβλητές δικτύου (network variables). Οι μεταβλητές δικτύου είναι μια κατηγορία πακέτων με ένα αναγνωριστικό, το οποίο προσδιορίζει τα δεδομένα ως μια τιμή η οποία έχει την ικανότητα να

μεταδίδεται από συσκευή σε συσκευή εύκολα μέσα στο δίκτυο. Η κωδικοποίηση των μεταβλητών δικτύου παρέχεται από ένα πρότυπο μεταβλητών δικτύου, το *SNVT*. Το *SNVT* είναι ένα πρότυπο κωδικοποίησης των δεδομένων, το οποίο καθορίζει τον τρόπο με τον οποίο θα ερμηνεύσουν οι διάφορες εφαρμογές τα δεδομένα που περιέχονται σε μια μεταβλητή δικτύου. Όπως γίνεται φανερό το *SNVT* προωθεί τη διαλειτουργικότητα, εξασφαλίζοντας ότι οι εφαρμογές ερμηνεύουν με ένα κοινό τρόπο τα δεδομένα που ανταλλάσσουν. Για παράδειγμα, δύο συσκευές οι οποίες ανταλλάσσουν μια μεταβλητή δικτύου θερμοκρασίας μπορούν να χρησιμοποιήσουν το *SNVT* για να εξασφαλίσουν ότι χρησιμοποιούν την ίδια μονάδα θερμοκρασίας (βαθμοί Κελσίου ή Fahrenheit).

Τα μηνύματα εφαρμογών είναι πακέτα μηνυμάτων με ένα 6-bit κωδικό μηνύματος. Ο πίνακας που ακολουθεί δίνει τους κωδικούς για τα μηνύματα εφαρμογών.

Τύπος μηνύματος	Κωδικός (hex)	Κωδικός (demical)
Μηνύματα καταγραφής δεδομένων	3C	60
Μηνύματα ISI (αυτοεγκατάστασης)	3D	61
Μηνύματα μεταφοράς αρχείων	3E	62

Πίνακας 4.8: Μηνύματα εφαρμογών

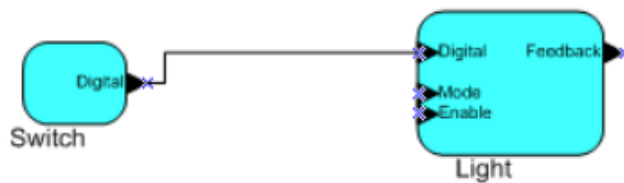
4.7.2 Μεταβλητές δικτύου

Το πρωτόκολλο LonTalk υλοποιεί την καινοτόμο ιδέα των μεταβλητών δικτύου. Οι μεταβλητές δικτύου (network variables) αναπαριστούν μεμονωμένες τιμές δεδομένων, δομές ή και ενώσεις πολλαπλών τιμών. Μια συσκευή μπορεί να έχει πολλές μεταβλητές δικτύου καθεμία από τις οποίες μπορεί να συνδεθεί με μία ή περισσότερες μεταβλητές δικτύου σε μία ή περισσότερες συσκευές στο δίκτυο. Οι μεταβλητές δικτύου μπορούν επίσης να οργανωθούν και σε πίνακες όπου κάθε μεμονωμένο στοιχείο του πίνακα μπορεί να συνδεθεί σε μια μεταβλητή δικτύου άλλης συσκευής.

Κάθε μεταβλητή δικτύου έχει *τύπο*, *κατεύθυνση* και *μέγεθος*. Η κατεύθυνση έχει δύο διακριτές τιμές – είσοδος ή έξοδος –, ανάλογα με το αν η μεταβλητή δικτύου χρησιμοποιείται για τη λήψη ή τη μετάδοση δεδομένων. Ο τύπος της μεταβλητής δικτύου καθορίζει την κωδικοποίηση και τις μονάδες των δεδομένων.

Μεταβλητές δικτύου με τον ίδιο τύπο αλλά με διαφορετική κατεύθυνση μπορούν να συνδεθούν και με αυτόν τον τρόπο να επιτρέψουν στις συσκευές να ανταλλάξουν πληροφορία. Οι συνδέσεις μπορούν να γίνουν μεταξύ μιας μεταβλητής εξόδου και μιας μεταβλητής εισόδου σε διαφορετικές συσκευές, αλλά ακόμα και πάνω στην ίδια συσκευή. Οι συνδέσεις αυτές μπορούν να θεωρηθούν σαν εικονικά καλώδια.

Ένα παράδειγμα σύνδεσης μεταξύ δύο συσκευών είναι το ακόλουθο : μια εφαρμογή σε μια συσκευή φωτισμού η οποία έχει μια μεταβλητή δικτύου εισόδου τύπου switch, και μια εφαρμογή σε μια συσκευή-διακόπτη η οποία έχει μια μεταβλητή δικτύου εξόδου τύπου και αυτή switch. Ένα εργαλείο δικτύου όπως το LonMaker ή το πρωτόκολλο ISI μπορούν να χρησιμοποιηθούν για να συνδέσουν τις δύο συσκευές μεταξύ τους (όπως φαίνεται στο σχήμα 4.10).

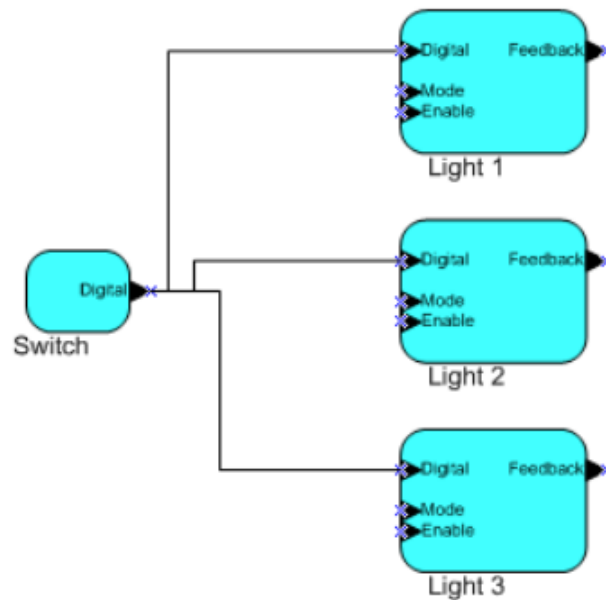


Σχήμα 4.10: Σύνδεση δύο μεταβλητών δικτύου

Η κατεύθυνση ορίζεται από τα μικρά τρίγωνα στο σχήμα. Μια μεταβλητή δικτύου μπορεί να συνδέεται όχι μόνο με μία αλλά και με πολλές άλλες μεταβλητές του ίδιου τύπου και αντίθετης κατεύθυνσης. Ένα παράδειγμα δίνεται στο σχήμα 4.11 στη σελίδα 67.

Η εφαρμογή δεν χρειάζεται να γνωρίζει κάτι για την κατεύθυνση των μεταβλητών δικτύου της. Όταν η εφαρμογή δημιουργεί μια καινούργια τιμή για μια μεταβλητή δικτύου, απλά περνάει την ενημερωμένη αυτή τιμή στο δίκτυο χωρίς να εκθέσει την εσωτερική της λειτουργία. Έτσι έχοντας και τη σύνδεση των μεταβλητών δικτύου η ενημερωμένη τιμή μεταδίδεται εύκολα μέσα στο δίκτυο.

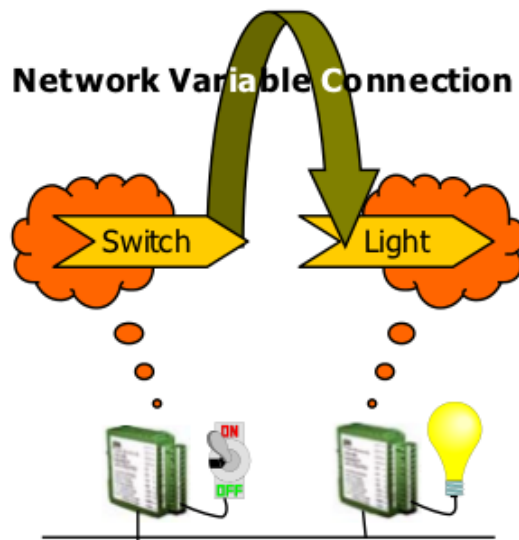
Η σύνδεση μεταξύ των μεταβλητών δικτύου δημιουργείται μέσω μιας διαδικασίας η οποία ονομάζεται "δέσιμο" (binding) και λαμβάνει χώρα τη στιγμή του σχεδιασμού του δικτύου. Το δέσιμο μπορεί να επιτευχθεί μέσω ενός εργαλείου δικτύου όπως το LonMaker ή μέσω του ίδιου του κώδικα της εφαρμογής, μια διαδικασία την οποία περιγράψαμε στο προηγούμενο κεφάλαιο και η οποία χρησιμοποιεί το πρωτόκολλο ISI. Όταν γίνεται η σύνδεση των μεταβλητών δικτύου, το πρωτόκολλο LonTalk είναι ρυθμισμένο κατάλληλα



Σχήμα 4.11: Σύνδεση πολλών μεταβλητών δικτύου

ώστε να γνωρίζει τις λογικές διευθύνσεις των άλλων συσκευών στο δίκτυο, από τις οποίες περιμένει τις τιμές των δικών τους μεταβλητών δικτύου για να γίνει η σύνδεση. Παρόμοια, όταν η υλοποίηση του πρωτοκόλλου λάβει μια ενημερωμένη τιμή για μια μεταβλητή δικτύου εισόδου (input), δίνει τα δεδομένα στο πρόγραμμα της εφαρμογής. Η διαδικασία binding με τη σειρά της δημιουργεί τις λογικές συνδέσεις μεταξύ των μεταβλητών δικτύου εισόδου-εξόδου και έτσι πραγματοποιείται η επικοινωνία.

Το πρωτόκολλο LonTalk μπορεί να υποστηρίξει μέχρι 4096 μεταβλητές δικτύου. Το τυπικό όριο για μια συσκευή Neuron είναι 254 μεταβλητές δικτύου οι οποίες οργανώνονται σε πίνακες στη μνήμη της συσκευής. Οι πίνακες αυτοί περιέχουν καταχωρήσεις οι οποίες αφορούν τα ονόματα, τις διευθύνσεις αλλά και τους δείκτες των μεταβλητών δικτύου. Χρησιμοποιούνται για να συνδέσουν τις διευθύνσεις με τους δείκτες. Οι διευθύνσεις είναι τελείως διαφορετικές σε αυτό το επίπεδο από ότι στο επίπεδο δικτύου, καθώς βοηθούν στην επιλογή της σωστής μεταβλητής δικτύου (network variable selector) και όχι απλά στην επιλογή της σωστής συσκευής (επίπεδο 3). Γι' αυτό το λόγο λοιπόν, οι μεταβλητές



Σχήμα 4.12: Σύνδεση δύο συσκευών

δικτύου οργανώνονται σε πίνακες με δείκτες και διευθύνσεις και βοηθούν στην αποτελεσματική υλοποίηση των εφαρμογών. Πιο αναλυτική παρουσίαση των πινάκων που περιέχουν τις μεταβλητές δικτύου γίνεται στο δεύτερο μέρος της εργασίας, όπου και παρουσιάζεται εκτενέστερα ο κώδικας της εφαρμογής και ορίζονται με ακρίβεια οι πίνακες κατανομής των network variables οι οποίοι χρησιμοποιήθηκαν. Το μοντέλο των μεταβλητών δικτύου απλοποιεί σε μεγάλο βαθμό τον προγραμματισμό πολύπλοκων εφαρμογών καθώς δίνει μια πιο ευέλικτη μορφή στα δεδομένα τα οποία ανταλλάσσονται μεταξύ των συσκευών. Οι προγραμματιστές δεν χρειάζεται να ασχολούνται με λεπτομέρειες όπως, τη διευθυνσιοδότηση του δικτύου ή το μέγεθος των buffers. Στο σύστημα LONWORKS λοιπόν κάθε συσκευή παίρνει τις δικές της αποφάσεις ελέγχου, οι οποίες βασίζονται στις πληροφορίες που συλλέγει από τις άλλες συσκευές του δικτύου.

4.8 Επίπεδο εφαρμογής

Το επίπεδο αυτό καθορίζει ένα πλούσιο σύνολο από υπηρεσίες δικτύου, οι οποίες χρησιμοποιούν τα δεδομένα που ανταλλάσσονται από τα χαμηλότερα επίπεδα. Οι υπηρεσίες του επιπέδου αυτού εξασφαλίζουν ότι συσκευές από διαφορετικούς κατασκευαστές ή προγραμματιστές μπορούν εύκολα να αλληλεπιδρούν μεταξύ τους. Η παρακάτω λίστα περιέχει τις περισσότερες υπηρεσίες του επιπέδου αυτού, καθώς οι υπόλοιπες παρέχονται από την

LONMARK International [19], ένα παγκόσμιο οργανισμό ο οποίος δημιουργήθηκε για να προάγει την αποδοτική και αποτελεσματική υλοποίηση του πρωτοκόλλου LonTalk καθώς και άλλων παρόμοιων πρωτοκόλλων. Οι υπηρεσίες λοιπόν του επιπέδου αυτού ασχολούνται με διάφορα ζητήματα όπως :

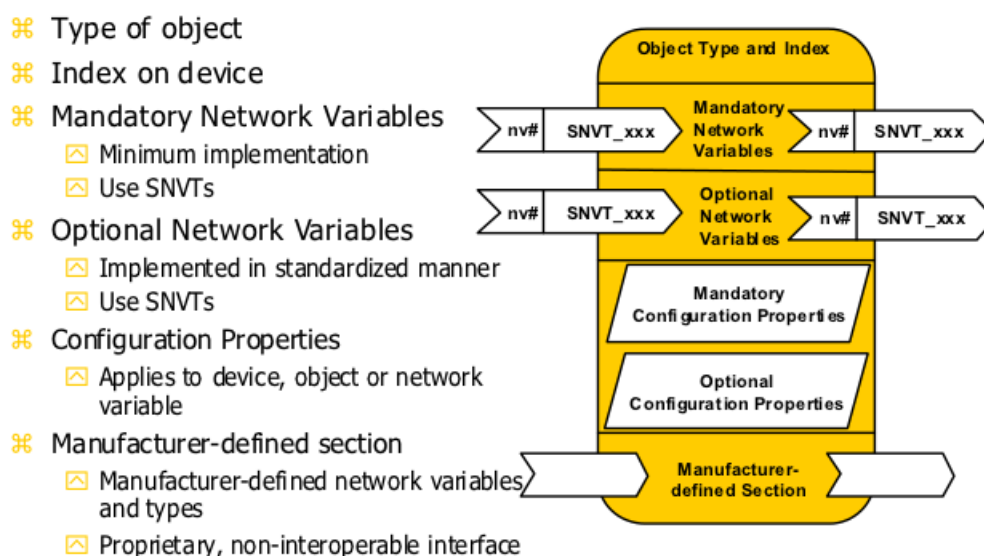
- Ρύθμιση του δικτύου
- Διάγνωση προβλημάτων στο δίκτυο
- Μεταφορά αρχείων
- Ρύθμιση της εφαρμογής
- Διαχείριση της εφαρμογής
- Προδιαγραφές της εφαρμογής
- Χρονοπρογραμματισμός
- Καταγραφή δεδομένων

Κάθε εφαρμογή η οποία βασίζεται στο πρωτόκολλο LonTalk περιλαμβάνει μια σειρά από διεπαφές για μια συσκευή με σκοπό την καταγραφή των λειτουργιών τις οποίες επιτελεί. Κάθε λειτουργία σε αυτό το επίπεδο αναφέρεται ως functional block, ένα σύνολο δηλαδή από μεταβλητές δικτύου και ιδιότητες ρύθμισης (configuration properties). Οι ιδιότητες ρύθμισης είναι στοιχεία δεδομένων όπως ακριβώς και οι μεταβλητές δικτύου και χρησιμοποιούνται για να χαρακτηρίσουν την συμπεριφορά μιας συσκευής. Υπάρχει ποικιλία τέτοιων ιδιοτήτων σε ένα πρωτόκολλο το οποίο ονομάζεται SCPT. Μια εφαρμογή μπορεί να διαχωριστεί σε πολλά functional blocks τα οποία όπως είπαμε εκτελούν μια συγκεκριμένη λειτουργία. Μπορούν να δέχονται δεδομένα από το δίκτυο ή από το υλικό της συσκευής αλλά και να δίνουν δεδομένα στο δίκτυο ή στο υλικό της συσκευής. Κάθε functional block ορίζεται από ένα functional profile το οποίο είναι ένα πρότυπο που παρέχεται από τη LONMARK International, ενώ το functional block είναι η υλοποίηση του προτύπου αυτού. Συσκευές οι οποίες χρησιμοποιούν τα LONMARK functional profiles ή αλλιώς τα LONMARK objects ονομάζονται LONMARK συσκευές. Η LONMARK δημιούργησε αυτά τα functional profiles με σκοπό να αναπτυχθεί η διαλειτουργικότητα μεταξύ των συσκευών οι οποίες συμμετέχουν σε ένα δίκτυο LONWORKS. Δημιούργησε δηλαδή ένα κοινό πρότυπο επικοινωνίας πάνω στο οποίο μπορούν να βασίζονται οι προγραμματιστές για

να σχεδιάζουν εφαρμογές οι οποίες βασίζονται σε θερμοκρασία, πίεση, φωτισμό, ασφάλεια κ.α .

Οι είσοδοι και έξοδοι του functional block αποτελούνται από μεταβλητές δικτύου και ιδιότητες ρύθμισης. Οι μεταβλητές δικτύου παρέχουν λειτουργικά δεδομένα εισόδου και εξόδου και οι ιδιότητες ρύθμισης καθορίζουν τη συμπεριφορά του functional block. Κάθε functional profile καθορίζει υποχρεωτικές και προαιρετικές μεταβλητές δικτύου και ιδιότητες ρύθμισης. Ένα functional block πρέπει πάντα να υλοποιεί τις υποχρεωτικές μεταβλητές δικτύου του functional profile αλλά και τις υποχρεωτικές ιδιότητες ρύθμισης.

Ένα παράδειγμα ενός functional profile φαίνεται στο παρακάτω σχήμα.



Σχήμα 4.13: Functional Profile

Το επίπεδο εφαρμογής προσφέρει την δυνατότητα σε ένα πρόγραμμα να μπορεί εύκολα να ξεχωρίζει από ένα άλλο μέσω του Program ID.

4.8.1 Program ID

Το Program ID είναι ένα 64-bit (16-hex) αναγνωριστικό, το οποίο χρησιμοποιείται για να προσδιορίσει μοναδικά μια εφαρμογή. Συνήθως αποτελείται από 8 ζευγάρια δεκαεξαδικά κωδικοποιημένων ψηφίων. Τα 16 αυτά ψηφία οργανώνονται σε έξι πεδία, τα οποία προσδιορίζουν τον τύπο της συσκευής, τον κατασκευαστή, το functional profile, τον τρόπο χρήσης, τον τύπο του καναλιού και τον αριθμό μοντέλου της συσκευής. Κάθε Program

ID χρησιμοποιεί το ακόλουθο format :

FM:MM:MM:CC:CC:UU:TT:NN

Τα πεδία αυτά περιγράφονται ως εξής :

- F—Format. Μία 4-bit τιμή η οποία καθορίζει τη δομή του program ID. Τα format 8 και 10–15 έχουν δεσμευθεί αποκλειστικά για LonMark συσκευές οι οποίες μπορούν να αλληλεπιδρούν μεταξύ τους. Το format 8 χρησιμοποιείται για να δηλώσει μια πιστοποιημένη LonMark συσκευή. Το format 9 δηλώνει μια συσκευή η οποία δεν έχει συμμορφωθεί με τους κανόνες της LonMark αλλά μπορεί να χρησιμοποιηθεί για ανάπτυξη αλλά και δοκιμές, με σκοπό την εκπλήρωση των LonMark προϋποθέσεων.
- M:MM:MM—Manufacturer. Ένα αναγνωριστικό μεγέθους 20-bit, το οποίο προσδιορίζει τον κατασκευαστή της συσκευής. Το ID αυτό ανατίθεται στον εκάστοτε κατασκευαστή μετά από αίτηση, όταν γίνει μέλος του LONMARK Interoperability Association. Οι κατασκευαστές οι οποίοι δεν έχουν ακόμα ID, μπορούν να χρησιμοποιήσουν το 0 για ανάπτυξη, δοκιμές αλλά και κατασκευή πρωτότυπων εφαρμογών.
- CC:CC—Device Class. Είναι ένα αναγνωριστικό μεγέθους 16-bit το οποίο προσδιορίζει την κατηγορία στην οποία ανήκει η συσκευή. Η κατηγορία αναγνωρίζεται από την κύρια λειτουργία που επιτελεί η συσκευή, ουσιαστικά δηλαδή ποιο functional profile υλοποιεί. Μπορεί να είναι για παράδειγμα μια συσκευή μέτρησης θερμοκρασίας, μέτρησης πίεσης, ελέγχου φωτισμού ή ένας απλός ενεργοποιητής.
- UU—Usage. Ένα αναγνωριστικό μεγέθους 8-bit το οποίο χρησιμοποιείται για να δηλώσει την προβλεπόμενη χρήση της συσκευής. Κάποιες ενδεικτικές τιμές είναι η οικιακή, βιομηχανική και η εμπορική χρήση.
- TT—Channel Type. Ένα αναγνωριστικό μεγέθους 8-bit το οποίο χρησιμοποιείται για να δηλώσει τον τύπο του καναλιού που χρησιμοποιείται ανάλογα και με τις προδιαγραφές του πομποδέκτη που χρησιμοποιείται στη συσκευή.
- NN—Model Number. Ένα αναγνωριστικό μεγέθους 8-bit το οποίο χρησιμοποιείται για να δηλώσει το συγκεκριμένο μοντέλο του προϊόντος. Οι αριθμοί των μοντέλων εκχωρούνται από τον κατασκευαστή του προϊόντος και πρέπει να είναι μοναδικοί για κάθε συσκευή.

4.9 Σύνοψη πρωτοκόλλου

Συνοψίζοντας, η ποικιλία των υπηρεσιών που παρέχονται από το πρωτόκολλο LonTalk επιτρέπει βελτιωμένη αξιοπιστία, ασφάλεια αλλά και βελτιστοποίηση στην χρήση των πόρων του δικτύου.

Τα χαρακτηριστικά αλλά και τα οφέλη, τα οποία παρέχονται από αυτές τις υπηρεσίες περιλαμβάνουν :

- Υποστήριξη ενός μεγάλου πλήθους από μέσα επικοινωνίας (κανάλια) όπως είναι τα καλώδια συνεστραμμένου ζεύγους, τα καλώδια ηλεκτρικής ισχύος (power line) καθώς και η επικοινωνία μέσω IP (Internet access).
- Υποστήριξη δικτύων τα οποία κατασκευάζονται με βάση ένα μείγμα καναλιών επικοινωνίας και κατα συνέπεια γίνονται δίκτυα διαφορετικών ταχυτήτων.
- Υποστήριξη της αποτελεσματικής μεταφοράς μικρών σε μέγεθος μηνυμάτων, βελτιστοποιώντας τη χρήση των πόρων του δικτύου από τις εφαρμογές.
- Υποστήριξη αξιόπιστης επικοινωνίας, συμπεριλαμβανομένης και της άμυνας ενάντια σε συσκευές, οι οποίες κάνουν μη επιτρεπόμενη χρήση του συστήματος.
- Υποστήριξη περαιτέρω αξιοπιστίας του συστήματος, μέσω εξάλειψης των σημείων που δημιουργούν προβλήματα στο σύστημα.
- Υποστήριξη υλοποίησης χαμηλού κόστους για τις συσκευές, τα εργαλεία και τις εφαρμογές του συστήματος.
- Ελαχιστοποίηση του κόστους εγκατάστασης και συντήρησης του συστήματος, με αποτέλεσμα μικρότερο κόστος σε κύκλους ζωής (life-cycle).
- Υποστήριξη χιλιάδων συσκευών σε ένα δίκτυο, παραμένοντας όμως εξίσου αξιόπιστο και για δίκτυα πολύ μικρότερου μεγέθους.
- Υποστήριξη εύκολης και ευέλικτης σύνδεσης μεταξύ των συσκευών.
- Υποστήριξη peer-to-peer επικοινωνίας, με αποτέλεσμα την ευέλικτη χρήση του πρωτοκόλλου σε κεντρικά αλλά και κατακεντρωμένα συστήματα ελέγχου.
- Παροχή ενός αποτελεσματικού μηχανισμού για τη διαλειτουργικότητα των συσκευών, με αποτέλεσμα την εύκολη ανταλλαγή φυσικών ποσοτήτων μεταξύ συσκευών από διαφορετικούς κατασκευαστές.

Ο αναγνώστης μπορεί να ανατρέξει στην πηγή [17], για επιπλέον πληροφορίες.

Κεφάλαιο 5

Το πρωτόκολλο ISI

5.1 Επισκόπηση

Τα δίκτυα ελέγχου κατηγοριοποιούνται ανάλογα με τη μέθοδο που χρησιμοποιούν για την εγκατάστασή τους. Όπως αναφέρθηκε και στο κεφάλαιο 4 υπάρχουν δύο τέτοια είδη δικτύων, τα διαχειριζόμενα και τα αυτοεγκαθιστώμενα δίκτυα. Στα διαχειριζόμενα δίκτυα υπάρχει ένας κεντρικός διακομιστής (server) για την εγκατάσταση του δικτύου. Ο χρήστης τυπικά χρησιμοποιεί ένα εργαλείο το οποίο μπορεί και αλληλεπιδρά με το server (LonMaker Integration Tool). Το εργαλείο αυτό χρησιμοποιείται για την δημιουργία της επικοινωνίας με το server και γενικότερα για την επικοινωνία μεταξύ των συσκευών του δικτύου και δεν χρειάζεται να είναι παρών καθ' όλη τη διάρκεια της λειτουργίας του. Χρειάζεται μόνο όταν είναι ανάγκη να γίνουν κάποιες αλλαγές στη ρύθμιση του δικτύου. Ο server είναι ενήμερος για την τοπολογία του δικτύου και έχει επίσης τη δυνατότητα να ρυθμίζει τις συσκευές για βέλτιστη απόδοση, εντός βέβαια των περιορισμών που επιβάλλονται από την τοπολογία. Η εναλλακτική σε ένα διαχειριζόμενο δίκτυο είναι ένα αυτοεγκαθιστώμενο δίκτυο. Σε ένα τέτοιο δίκτυο δεν υπάρχει κάποιο κεντρικό εργαλείο ή server που να πραγματοποιεί την εγκατάσταση του δικτύου. Αντ' αυτού, κάθε συσκευή περιέχει κώδικα ο οποίος αντικαθιστά μέρος της λειτουργικότητας του server, και έχει ως αποτέλεσμα τη δημιουργία και εγκατάσταση του δικτύου χωρίς τη βοήθεια κάποιου ειδικού εργαλείου.

Δεδομένου λοιπόν ότι κάθε συσκευή είναι υπεύθυνη για τη ρύθμισή της, χρειάζεται ένα κοινό πρότυπο για να εξασφαλίσει ότι οι συσκευές ρυθμίζονται με ένα συμβατό τρόπο. Αυτό το πρότυπο ονομάζεται ISI (Interoperable Self-Installation). Το πρότυπο αυτό

μπορεί να χρησιμοποιηθεί για δίκτυα τα οποία έχουν μέχρι 200 συσκευές. Τα μεγαλύτερα και πιο πολύπλοκα δίκτυα είναι καλύτερο να λειτουργούν σαν διαχειριζόμενα δίκτυα ή να διασπαστούν σε μικρότερα δίκτυα (μέχρι 200 συσκευές), ώστε να πληρούν τις προϋποθέσεις του ISI. Συσκευές οι οποίες συμμορφώνονται με το πρωτόκολλο ISI καλούνται ISI συσκευές. Το πρωτόκολλο αυτό λειτουργεί πάνω από το πρωτόκολλο LonTalk (στο επίπεδο εφαρμογής), το οποίο αναλύθηκε στο προηγούμενο κεφάλαιο.

Το πρωτόκολλο ISI εκτελεί τρεις βασικές λειτουργίες :

1. Domain acquisition
2. Network address assignment
3. Connection management

Η πρώτη εξασφαλίζει ότι οι συσκευές του δικτύου μπορούν να αλληλεπιδρούν μεταξύ τους, χωρίς όμως να παρεμβαίνουν σε άλλα γειτονικά δίκτυα. Η δεύτερη εξασφαλίζει ότι κάθε συσκευή στο δίκτυο έχει μια μοναδική λογική διεύθυνση με αποτέλεσμα το δίκτυο να μπορεί να μετατραπεί εύκολα σε διαχειριζόμενο (managed). Τέλος η τρίτη λειτουργία επιτρέπει στις συσκευές του δικτύου να ανταλλάσσουν πολύ εύκολα τα δεδομένα τους.

5.2 Λειτουργία του πρωτοκόλλου

Υπάρχουν δύο τύποι ISI δικτύων, τα ISI-S δίκτυα τα οποία είναι πολύ απλά και αυτόνομα και τα ISI-DA τα οποία είναι πιο μεγάλα και υποστηρίζουν περισσότερες συσκευές και περίπλοκες τοπολογίες. Τα ISI-S δίκτυα υποστηρίζουν μέχρι 32 συσκευές ενώ τα ISI-DA μέχρι 200. Επίσης υποστηρίζουν δύο τύπους καναλιών, το PL-20 power line και το TP/FT-10 free topology twisted pair.

Το κύριο πλεονέκτημα της χρήσης του ISI για την εγκατάσταση δικτύων είναι ότι εξαλείφει την ανάγκη της χρησιμοποίησης μιας κεντρικής συσκευής διαχείρισης, η οποία αν χαθεί θα προκαλέσει την απώλεια όλης της πληροφορίας του δικτύου. Για να συμμετάσχει σε ένα ISI δίκτυο, μια συσκευή πρέπει να υλοποιήσει κάποια τμήματα του πρωτοκόλλου, τα οποία χρειάζονται χώρο τουλάχιστον 3 Kbytes στη μνήμη της συσκευής. Οι ISI συσκευές έχουν ανάγκη από περιοδική, μεταξύ τους, επικοινωνία για τη συντήρηση του αυτοεγκαθιστώμενου δικτύου. Το ISI περιορίζει την κυκλοφορία των πακέτων στο δίκτυο σε 1 πακέτο/5sec κατά μέσο όρο για δίκτυα power line μειώνοντας το διαθέσιμο εύρος ζώνης.

5.2.1 Αλγόριθμος Fire-and-Forget

Οι λειτουργίες network address assignment και Connection management απαιτούν κατανομή των πόρων του δικτύου. Στην περίπτωση της πρώτης οι πόροι που πρέπει να κατανεμηθούν είναι τα subnet και node ID's, ενώ στη δεύτερη είναι τα network variable selectors. Σε ένα διαχειριζόμενο δίκτυο ο server διαχείρισης του δικτύου κατανέμει τους πόρους του δικτύου και εξασφαλίζει ότι δεν υπάρχει ο παραμικρός κίνδυνος (race conditions). Σε ένα self-installed δίκτυο, κάθε συσκευή πρέπει να κατανήμει τους δικούς της πόρους δικτύου και αυτόματα να αποτρέψει συγκρούσεις και προβλήματα που μπορεί να προκύψουν από διπλό-εκχωρήσεις πόρων.

Το πρωτόκολλο ISI χρησιμοποιεί τον αλγόριθμο fire-and-forget για να κατανήμει τους πόρους και τις διευθύνσεις του δικτύου. Ο αλγόριθμος αυτός ελαχιστοποιεί τις καθυστερήσεις, την ανάγκη για ένα κεντρικό server, αλλά διευκολύνει και τη σωστή ISI λειτουργία σε περιόδους μερικής διακοπής του δικτύου αλλά και σε περιόδους μη-διαθεσιμότητας πολλών εγκατεστημένων συσκευών.

Όταν μια ISI συσκευή χρειάζεται μια νέα διεύθυνση δικτύου, ο ISI κώδικας της συσκευής επιλέγει τυχαία μια διεύθυνση και την μεταδίδει (broadcast) σε όλες τις άλλες συσκευές του δικτύου. Αυτό είναι το fire κομμάτι του αλγορίθμου fire-and-forget. Όταν η συσκευή πραγματοποιήσει αυτή την ενέργεια δεν περιμένει απόκριση από τις άλλες συσκευές. Υποθέτει ότι έχει επιλεγεί η κατάλληλη διεύθυνση και συνεχίζει να τη χρησιμοποιεί. Αυτό είναι το κομμάτι forget του αλγορίθμου. Όλες οι συσκευές παρακολουθούν αυτές τις περιοδικές μεταδόσεις των διευθύνσεων. Αν μια συσκευή λάβει ένα μήνυμα το οποίο δηλώνει σύγκρουση διευθύνσεων τότε αλλάζει τη ρύθμισή της για να εξαλείψει αυτή τη σύγκρουση.

Κανάλι	Ρυθμός πακέτων
Power Line or Power Line plus Free Topology	1 πακέτο/5sec
Free Topology	1 πακέτο/2.5sec

Πίνακας 5.1: Ρυθμός πακέτων σε ένα ISI δίκτυο

Είναι σημαντικό να τονιστεί ότι τα δίκτυα ελέγχου βρίσκονται σε μια διαρκή κατάσταση ρευστότητας. Συσκευές μπορεί να προστίθενται ή να αφαιρούνται, μονάδες μπορεί να τροφοδοτούνται ή να σβήνουν, ή μπορεί επίσης να υπάρχουν διακοπές της λειτουργίας του δικτύου ανά πάσα στιγμή. Χάρη στην περιοδική επικοινωνία, ο αλγόριθμος fire-and-forget

εξασφαλίζει ότι κάθε συσκευή χειρίζεται αποτελεσματικά την οποιαδήποτε νέα κατάσταση. Όπως αναφέρθηκε προηγουμένως, οι συσκευές μεταδίδουν περιοδικά τις διευθύνσεις συσκευής και σύνδεσης έτσι ώστε οι άλλες συσκευές του δικτύου να μπορούν να ανακαταστήσουν τους πόρους που έχουν διπλο-εκχωρηθεί. Κάθε συσκευή μεταδίδει περιοδικά το subnet και node ID της σε slots. Όταν είναι η σειρά μιας συσκευής να μεταδώσει, τότε αρχίζει να μεταδίδει μια σειρά μηνυμάτων. Στην αρχή μεταδίδει το μήνυμα που όπως είπαμε περιέχει πληροφορίες για τα subnet και node ID (DRUM message). Στη συνέχεια ένα μήνυμα για κάθε σύνδεση στην οποία η συσκευή είναι host (CSMI message) και τέλος, ένα μήνυμα υπενθύμισης αν η σύνδεση είναι αυτόματη (CSMR message). Στη σειρά των μηνυμάτων αυτών μπορεί να μπου και διάφορα άλλα, ειδικά για την εφαρμογή, μηνύματα. Όπως έχει αναφερθεί οι εφαρμογές οι οποίες τρέχουν πάνω στις συσκευές χωρίζονται σε πολλά functional blocks, τα οποία ορίζονται μέσω των functional profiles. Αυτά τα functional profiles είναι ένα σύνολο από network variables και configuration properties. Οι εφαρμογές LONWORKS τυπικά, ανταλλάσσουν δεδομένα μέσω των network variables, αφού πρώτα συνδεθούν μέσω της διαδικασίας binding. Η διαδικασία αυτή λαμβάνει χώρα στα διαχειριζόμενα δίκτυα ελέγχου. Για τα self-installed δίκτυα ελέγχου η διαδικασία ονομάζεται enrollment. Και η διαδικασία enrollment αλλά και η διαδικασία binding μπορούν να ρυθμίσουν μια μεταβλητή δικτύου έτσι ώστε να μπορεί να εμφανίζεται σαν μια εικονική μεταβλητή δικτύου. Αυτές οι εικονικές μεταβλητές δικτύου ονομάζονται *aliases*. Το πρωτόκολλο ISI μπορεί να υποστηρίξει μέχρι 254 μεταβλητές δικτύου και 254 *aliases* σε μια συσκευή.

Η επικοινωνία μέσω των network variables είναι αυτό που κάνει τα δίκτυα LONWORKS τεχνολογίας να ξεχωρίζουν. Όταν μια ενημερωμένη τιμή μιας μεταβλητής δικτύου μεταδίδεται στο δίκτυο, περιέχει την τιμή της μεταβλητής αλλά και πληροφορίες διεύθυνσης, ώστε να εντοπιστεί εύκολα η συσκευή στην οποία στέλνεται το update. Οι πληροφορίες διεύθυνσης χρησιμοποιούνται επίσης ώστε να αναγνωριστούν και οι network variables οι οποίες θα λάβουν την ενημερωμένη τιμή. Οι πληροφορίες διεύθυνσης περιέχουν δύο στοιχεία : μια διεύθυνση του επιπέδου 3 (LonTalk) η οποία και πρόκειται να εντοπίσει τη συσκευή που θα λάβει το update, αλλά και μια διεύθυνση επιπέδου 6 (LonTalk), η οποία και ονομάζεται όπως έχει αναφερθεί και πρωτύτερα network variable selector. Η διεύθυνση αυτή βοηθά στον εντοπισμό της κατάλληλης μεταβλητής μέσα στη συσκευή. Είναι ένα 14-bit αναγνωριστικό το οποίο παίρνει τιμές από 0 έως 3FFF hex. Οι selectors οργανώνονται σε πίνακες, ο ένας εκ των οποίων ονομάζεται network variable

configuration table ή address table. Αυτός περιέχει μια καταχώρηση selector για κάθε network variable στη συσκευή και έχει μέγιστο μέγεθος 15 καταχωρήσεων. Ο δεύτερος πίνακας που χρησιμοποιείται ονομάζεται alias table. Αυτός ο πίνακας περιέχει έναν αριθμό από selectors για την κάθε μεταβλητή και έχει μέγιστο μέγεθος 62 καταχωρήσεων.

Τα subnet και node ID's είναι ένα ζευγάρι αναγνωριστικών, τα οποία παρέχουν μια μοναδική διεύθυνση για μια συσκευή μέσα σε ένα δίκτυο. Για τα ISI δίκτυα η τιμή του subnet ID είναι μεταξύ 64-127 για συσκευές TP/FT-10 και μεταξύ 128-191 για τις PL-20 συσκευές. Στα διαχειριζόμενα δίκτυα, τα subnets χρησιμοποιούνται για αποτελεσματική δρομολόγηση των πακέτων. Το node ID παίρνει τιμές από 2 έως 125.

Το group όπως έχει αναφερθεί είναι μια λογική συλλογή συσκευών, η οποία ξεχωρίζει από τις άλλες μέσω του group ID. Το ID αυτό παίρνει τιμές από 0 έως 255 και καθορίζει τον τρόπο χρήσης των συσκευών που συμμετέχουν στο group.

5.2.2 Το μοντέλο σύνδεσης

Οι ISI συνδέσεις δημιουργούνται μέσω μιας συνδεσμολογίας που ονομάζεται *connection assembly*. Η συνδεσμολογία αυτή είναι η διασύνδεση που δημιουργείται στο δίκτυο για ένα μπλοκ λειτουργικότητας, για μια συγκεκριμένη λειτουργία, δηλαδή για ένα functional block. Μια συνδεσμολογία η οποία αναφέρεται σε μια μόνο μεταβλητή δικτύου ονομάζεται single assembly.

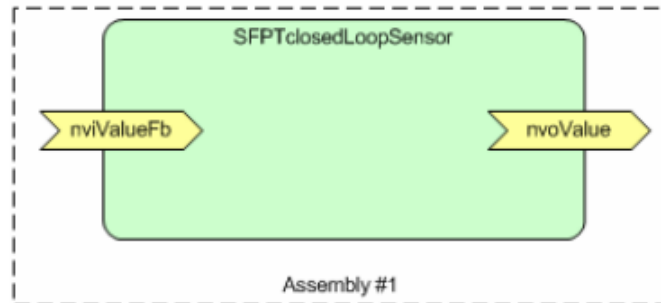


Σχήμα 5.1: Single Assembly

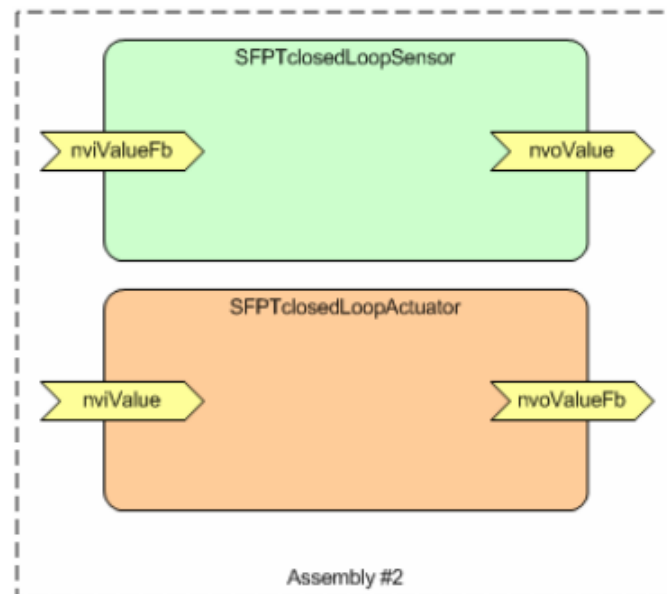
Ουσιαστικά μια συνδεσμολογία σύνδεσης (connection assembly), περιλαμβάνει ένα σύνολο από μεταβλητές δικτύου οι οποίες μπορούν να ενωθούν μαζί σαν μια ενιαία ομάδα, ώστε να εκτελέσουν μια κοινή λειτουργία. Μια συνδεσμολογία η οποία περιλαμβάνει περισσότερες από μια μεταβλητές δικτύου, από ένα functional profile ονομάζεται compound assembly.

Ακολουθούν τα σχήματα 5.2 και 5.3 τα οποία δείχνουν ένα compound assembly, μια συνδεσμολογία δηλαδή η οποία μπορεί να περιλαμβάνει περισσότερα από ένα functional

blocks και πολλαπλές μεταβλητές δικτύου.



Σχήμα 5.2: Compound Assembly with 1 functional block



Σχήμα 5.3: Compound Assembly with 2 functional blocks

Για να προσδιορίσει η εφαρμογή ένα assembly εκχωρεί σε αυτό έναν αριθμό ο οποίος είναι από 0 έως 254. Ο αριθμός αυτός εκχωρείται ακολουθιακά, ξεκινώντας από το 0, σε κάθε μεταβλητή δικτύου που βρίσκεται πάνω στο assembly και έχει οριστεί από κάποιο functional profile.

Κάθε assembly έχει ένα πλάτος, το οποίο ισοδυναμεί με το πλήθος των μεταβλητών δικτύου που περιέχει. Για παράδειγμα η εικόνα 5.1 δείχνει ένα assembly πλάτους 1, ενώ η εικόνα 5.2 ένα compound assembly πλάτους 2. Σε ένα compound assembly μια από τις μεταβλητές δικτύου τίθεται πάντα ως η πρωτεύουσα μεταβλητή δικτύου (primary network

variable).

Για να ανοίξει μια σύνδεση, η συσκευή η οποία ξεκινά την σύνδεση (host), στέλνει ένα μήνυμα σύνδεσης το οποίο περιλαμβάνει συνήθως τις ακόλουθες πληροφορίες : τον τύπο της μεταβλητής δικτύου του assembly που θα συμμετάσχει στη σύνδεση (υπενθυμίζουμε ότι για να γίνει η σύνδεση πρέπει οι μεταβλητές δικτύου να έχουν ίδιο τύπο και διαφορετική κατεύθυνση), τον αριθμό του functional profile που περιέχει τον τύπο της μεταβλητής δικτύου και το πλάτος του assembly. Το μήνυμα σύνδεσης στέλνεται μέσω ενός ISI μηνύματος το οποίο ονομάζεται CSMO. Οι άλλες συσκευές του δικτύου που λαμβάνουν το μήνυμα προσπαθούν να το ερμηνεύσουν και να αποφασίσουν αν θα μετάσχουν ή όχι στη σύνδεση. Για παράδειγμα αν η συσκευή της εικόνας 5.1 στείλει ένα μήνυμα σύνδεσης, αυτό θα πρέπει να περιέχει τον τύπο της μεταβλητής pnoValue, η οποία δεν ακολουθεί κανένα functional profile, αλλά και το πλάτος του assembly το οποίο είναι 1. Στα compound assemblies στα οποία τυχόν χρησιμοποιούνται περισσότερα από ένα functional profiles πρέπει να οριστεί και ένα πεδίο του CSMO που ονομάζεται *variant*. Το variant μπορεί να πάρει τιμές από 1-127. Στη συνέχεια του κεφαλαίου το CSMO μήνυμα σύνδεσης παρουσιάζεται εκτενέστερα.

5.2.3 Διαχείριση συνδέσεων

Όπως αναφέρθηκε και πρωτίτερα η διαδικασία δημιουργίας συνδέσεων με το πρωτόκολλο ISI ονομάζεται *enrollment*. Η διαδικασία αυτή περιλαμβάνει τρεις διακριτές μεθόδους σύνδεσης οι οποίες είναι : manual, automatic και controlled. Η manual enrollment είναι μια πολύ απλή μέθοδος σύνδεσης η οποία χρησιμοποιεί ένα button και ένα LED. Είναι κατάλληλη για συνδέσεις στις οποίες χρειάζεται η παρέμβαση του χρήστη για την επιλογή των συσκευών που θα μετέχουν στη σύνδεση. Η αυτόματη σύνδεση δημιουργεί συνδέσεις χωρίς την παρουσία του χρήστη. Η controlled σύνδεση υποστηρίζει συνδέσεις οι οποίες δημιουργούνται από μια κεντρική συσκευή (connection controller). Μια συσκευή μπορεί να χρησιμοποιεί περισσότερες από μια μεθόδους σύνδεσης.

5.2.3.1 Manual Enrollment

Όπως είπαμε οι συνδέσεις των συσκευών μπορούν να γίνουν χειροκίνητα, με τη βοήθεια του χρήστη. Οι συνδέσεις αυτές πολλές φορές λέγονται και push-button συνδέσεις. Οι συσκευές οι οποίες υποστηρίζουν αυτού του είδους τη σύνδεση, πρέπει να δημιουργήσουν

μια κατάλληλη διεπαφή για τον έλεγχο αυτής της διαδικασίας όπως για παράδειγμα ένα LED για το κάθε assembly το οποίο ονομάζεται Connect Light ή ένα button (Connect button).

Έτσι λοιπόν, από την πλευρά του χρήστη η σύνδεση αυτή θα μπορούσε να στηθεί ως εξής: Ο χρήστης πατάει το Connect button σε μια συσκευή η οποία θα επιθυμούσε να είναι η συσκευή host για τη σύνδεση, η συσκευή θα στείλει το μήνυμα σύνδεσης στις άλλες συσκευές του δικτύου, οι συσκευές αυτές θα αρχίσουν να αναβοσβήνουν το LED σύνδεσής τους για να προσδιορίσουν ότι είναι συσκευές που θα μπορούσαν να μετέχουν στη σύνδεση, ο χρήστης επιλέγει ποια συσκευή θα μετέχει της σύνδεσης πατώντας το Connect button και έτσι το LED σταθεροποιείται για να υποδηλώσει το γεγονός ότι είναι έτοιμη η συσκευή για σύνδεση. Τέλος ο χρήστης επιστρέφει στη συσκευή host και επιβεβαιώνει τη σύνδεση. Τα LEDs και στις δύο συσκευές σβήνουν επιβεβαιώνοντας τη σύνδεση.

5.2.3.2 Controlled Enrollment

Οι συνδέσεις αυτές πραγματοποιούνται με τη βοήθεια και του χρήστη αλλά και μιας κεντρικής συσκευής ελέγχου. Ο χρήστης ξεκινά την σύνδεση αλληλεπιδρώντας με τη συσκευή αυτή, ενώ στη συνέχεια η συσκευή ελέγχου στέλνει αιτήματα (CTRQ message) προς όλες τις συσκευές που πρόκειται να συνδεθούν. Αυτό γίνεται με το πάτημα των Connect buttons στις συσκευές, από το χρήστη. Τα μηνύματα αυτά περιέχουν τις παραμέτρους σύνδεσης. Οι συσκευές που δέχονται αυτά τα μηνύματα απαντούν με ένα CTRP message για το αν μπορούν να υποστηρίξουν τις παραμέτρους του μηνύματος σύνδεσης. Η διαδικασία που ακολουθείται είναι η ίδια με αυτή που ακολουθείται και στην προηγούμενη μέθοδο με τη μόνη διαφορά ότι εδώ τη διαδικασία σύνδεσης την ξεκινά η συσκευή ελέγχου.

5.2.3.3 Automatic Enrollment

Αυτή η μέθοδος της αυτόματης σύνδεσης είναι και η μέθοδος η οποία χρησιμοποιείται στην εργασία αυτή και η οποία έχει σαν σκοπό της να συνδέσει δύο εφαρμογές μεταξύ τους χωρίς την παρουσία του χρήστη. Η αυτόματη σύνδεση πραγματοποιείται με μηνύματα τα οποία μεταδίδονται αυτόματα από τη συσκευή-host στις άλλες συσκευές του δικτύου. Αυτά τα μηνύματα αποστέλλονται εκ νέου σε τακτά χρονικά διαστήματα, επιτρέποντας σε συσκευές να προστίθενται στην ήδη υπάρχουσα σύνδεση αλλά βοηθώντας παράλληλα και τη διατήρηση των ήδη υπάρχουσών συνδέσεων. Οι συσκευές οι οποίες δέχονται τα μηνύματα σύνδεσης μπορούν να κατανοήσουν αν τα μηνύματα αυτά προέρχονται από αυτόματη ή

manual σύνδεση. Οι συσκευές οι οποίες υλοποιούν αυτόματες συνδέσεις, δεν προχωρούν σε σύνδεση πριν από ένα τυχαίοποιημένο χρονικό διάστημα (T_{auto}) της τάξης των 11.30 λεπτών (περίπου) κυρίως για λόγους καλύτερης απόδοσης της σύνδεσης αλλά και αποφυγής προβλημάτων από την αύξηση της ζήτησης της σύνδεσης τη στιγμή της εκκίνησης (power-up).

5.2.4 Μηνύματα ISI

Το πρωτόκολλο ISI ορίζει ένα σύνολο από τυποποιημένα μηνύματα, τα οποία χρησιμοποιούνται για να συντονίσουν την εγκατάσταση των συσκευών σε ένα δίκτυο ISI. Η μηχανή ISI η οποία είναι κομμάτι της βιβλιοθήκης του πρωτοκόλλου, δημιουργεί και επεξεργάζεται αυτόματα σχεδόν όλα τα ISI μηνύματα. Οι βασικές διαδικασίες της ISI engine παρουσιάζονται λίγο αργότερα στο κεφάλαιο. Στη συνέχεια δίνεται μια μικρή περιγραφή των κυριότερων μηνυμάτων που μπορεί να συναντήσει κάποιος, στην προσπάθεια υλοποίησης ενός δικτύου ISI.

- **Device Resource Usage Message (DRUM)**– Αυτό το μήνυμα μεταδίδεται περιοδικά από όλες τις συσκευές του δικτύου. Περιλαμβάνει τη φυσική διεύθυνση (Neuron ID), την λογική διεύθυνση (domain, subnet, node IDs), αλλά και τον τύπο του καναλιού, πάνω στο οποίο η συσκευή μεταδίδει. Επίσης δίνει τη δυνατότητα του εντοπισμού όλων των συσκευών σε ένα δίκτυο ISI, μέσω ενός πίνακα που δημιουργεί (device table), ο οποίος κρατάει πληροφορίες για τις διευθύνσεις των συσκευών στις οποίες το μήνυμα αυτό μεταδίδεται. Η εκτεταμένη έκδοση του μηνύματος αυτού προσθέτει ένα πεδίο το οποίο αφορά στην χρησιμότητα της συσκευής.
- **Connection Status Messages (CSMs)**– Αυτή η ομάδα μηνυμάτων χρησιμοποιείται για τη δημιουργία, συντήρηση και διαγραφή συνδέσεων. Αυτή η κατηγορία περιλαμβάνει πολλούς τύπους μηνυμάτων, συμπεριλαμβανομένων των μηνυμάτων δημιουργίας χειροκίνητης σύνδεσης (CSMO), δημιουργίας αυτόματης σύνδεσης (CSMA, CSMR) αλλά και διαγραφής σύνδεσης (CSMD). Οι συσκευές οι οποίες δέχονται αυτά τα μηνύματα μπορούν να χρησιμοποιήσουν τις πληροφορίες που αυτά μεταφέρουν και να αποφασίσουν αν θα μετέχουν ή όχι στη σύνδεση. Η εκτεταμένη έκδοση των μηνυμάτων αυτών περιέχει κάποια πεδία, τα οποία βοηθούν τη συσκευή να ελέγξει αν η σύνδεση προϋποθέτει την παροχή επιβεβαίωσης (acknowledgements) ή κομματιών

του program ID.

- Timing Guidance Message (TIMG)– Αυτό το μήνυμα μεταδίδεται περιοδικά από τους διακομιστές διευθύνσεων domain. Περιέχει πληροφορίες για το μέγεθος και την τοπολογία του δικτύου. Είναι ένα προαιρετικό μήνυμα, αλλά αρκετά σημαντικό γιατί μπορεί να βοηθήσει τις συσκευές να προγραμματίσουν καλύτερα την αποστολή των μηνυμάτων τους. Έτσι βοηθάει στην αποτελεσματική χρήση του εύρους ζώνης του καναλιού και στην ελαχιστοποίηση της επιβάρυνσης που προκαλεί το πρωτόκολλο ISI.

Θέλοντας να εμβαθύνουμε λίγο περισσότερο στα μηνύματα CSMs, παρουσιάζουμε τα πεδία που χρησιμοποιεί ένα CSMO μήνυμα για να μπορέσει να δημιουργήσει μια σύνδεση εφαρμογών. Έτσι λοιπόν εκτός της κεφαλίδας το μήνυμα αυτό περιέχει και τα ακόλουθα πεδία :

1. Group ID– Καθορίζει τη χρησιμότητα της σύνδεσης.
2. Direction– Καθορίζει την κατεύθυνση της μεταβλητής δικτύου η οποία προσφέρεται για τη σύνδεση (εισόδου, εξόδου, αλλά και τις δύο επιλογές όταν έχουμε πολλαπλές συνδέσεις).
3. Width– Καθορίζει τον αριθμό των μεταβλητών δικτύου σε ένα connection assembly.
4. Profile– Καθορίζει τον αριθμό του functional profile που χρησιμοποιεί το connection assembly.
5. NvType– Καθορίζει τον τύπο της μεταβλητής δικτύου που προσφέρεται για σύνδεση.
6. Variant– Καθορίζει πόσα functional profiles χρησιμοποιούνται.
7. Acknowledged– Καθορίζει αν η σύνδεση απαιτεί επιβεβαίωση.
8. Poll– Δεν χρησιμοποιείται εφόσον έχουμε σύνδεση που βασίζεται σε γεγονότα (events).
9. Scope– Συνήθως είναι 0, για να καθορίσει ότι το functional profile αλλά και η μεταβλητή δικτύου προέρχονται από τα πρότυπα LonMark και SNVT, αντίστοιχα.
10. Application– Περιλαμβάνει τα πρώτα 6-byte του Program ID της συσκευής.

11. Member– Περιλαμβάνει μια άλλη μεταβλητή μέσα στη συσκευή, η οποία και θα χρησιμοποιηθεί για τη σύνδεση.

5.3 Μηχανή ISI

5.3.1 Βασικές διαδικασίες

Στην υποενότητα αυτή περιγράφονται τα βασικά στοιχεία της μηχανής ISI. Στο δεύτερο μέρος της εργασίας, όπου και αναλύεται εκτενώς ο κώδικας της εφαρμογής, ο αναγνώστης μπορεί να διαπιστώσει τη χρησιμότητα της ISI μηχανής για τη δημιουργία αυτοεγκαθιστώμενων δικτύων. Η μηχανή αυτή υλοποιεί τα περισσότερα κομμάτια της λογικής του πρωτοκόλλου ISI, καλώντας τα API (Application programming interface) της μηχανής και αλλάζοντας τις προεπιλεγμένες (default) υλοποιήσεις με αυτές που απαιτεί κάθε φορά η εφαρμογή.

Η μηχανή ISI, η οποία πρέπει να υλοποιείται σε όλες τις εφαρμογές ενός δικτύου, στέλνει και λαμβάνει ISI μηνύματα, αλλά επίσης διαχειρίζεται και τη λειτουργία της συσκευής. Για να δημιουργηθεί μια ISI εφαρμογή πρέπει να ξεκινήσει η ISI μηχανή, να καλείται περιοδικά αλλά και να μπορεί να διαχειρίζεται τα μηνύματα που καταφθάνουν. Στην ουσία χρησιμοποιούνται τέσσερις συναρτήσεις, αφού πρώτα βέβαια γίνει η εισαγωγή της βιβλιοθήκης `isi.h` :

- `IsiStartS()`– χρησιμοποιείται για την εκκίνηση της μηχανής.
- `IsiTickS()`– χρησιμοποιείται για τη συνεχή συντήρηση της μηχανής. Η συνάρτηση αυτή καλεί τη μηχανή 4 φορές το δευτερόλεπτο.
- `IsiApproveMsg()`
`IsiProcessMsgS()`– Οι συναρτήσεις αυτές χρησιμοποιούνται για τον προσδιορισμό και την επεξεργασία των ISI μηνυμάτων.

Επίσης για να μπορέσει να λειτουργήσει η μηχανή, η εφαρμογή ISI πρέπει να συνδεθεί με κάποιες βιβλιοθήκες, καθεμία από τις οποίες έχει τα δικά της μοναδικά χαρακτηριστικά. Η υλοποίηση του πρωτοκόλλου ISI είναι διαχωρισμένη στις εξής βιβλιοθήκες :

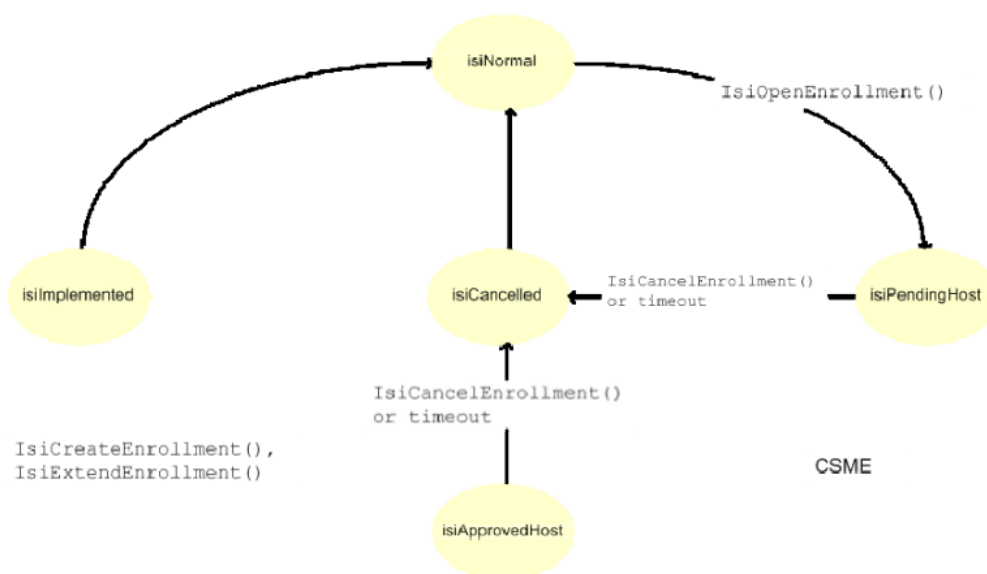
- `IsiFull.lib` – Η βιβλιοθήκη αυτή περιέχει τα περισσότερα κομμάτια της υλοποίησης του πρωτοκόλλου.

- IsiCompactAuto.lib – Η βιβλιοθήκη αυτή περιέχει τις βασικότερες συναρτήσεις για αυτόματες συνδέσεις.
- IsiCompactManual.lib – Η βιβλιοθήκη αυτή περιέχει τις βασικότερες συναρτήσεις για manual και controlled συνδέσεις.
- IsiCompactS.lib – Η βιβλιοθήκη αυτή όπως και οι υπόλοιπες που ακολουθούν περιέχουν χρήσιμες συναρτήσεις για ISI-DA δίκτυα, πιο μεγάλα και πιο πολύπλοκα.
- IsiCompactSHb.lib
- IsiCompactDa.lib
- IsiCompactDaHb.lib
- IsiPL3170.lib

Αφού στηθεί η μηχανή ISI, μπορούν να δημιουργηθούν συνδέσεις μεταξύ των μεταβλητών δικτύου, δύο ή περισσότερων εφαρμογών μέσω της διαδικασίας *enrollment*. Η μηχανή χρησιμοποιεί τις παρακάτω συναρτήσεις για τη δημιουργία συνδέσεων :

- IsiOpenEnrollment(unsigned Assembly)–για manual και controlled συνδέσεις. Μετά την κλήση αυτής της συνάρτησης καλείται η IsiCreateCsmo(unsigned Assembly, IsiCsmoData* pCsmoData) για δημιουργήσει το CSMO μήνυμα σύνδεσης, το οποίο θα σταλεί σε όλες τις συσκευές του δικτύου, γεμίζοντας τα πεδία της δομής IsiCsmoData που παρουσιάστηκε πρωτύτερα στο κεφάλαιο. Για τις controlled συνδέσεις το μήνυμα που στέλνεται, όπως έχει αναφερθεί είναι το CTRQ.
- IsiInitiateAutoEnrollment(const IsiCsmoData* pCsmoData, unsigned Assembly)– για αυτόματες συνδέσεις. Η συνάρτηση αυτή καλείται αφού περάσει πρώτα το random χρονικό διάστημα (T_{auto}) των 11.30 λεπτών (περίπου). Αυτό το χρονικό διάστημα συνδέεται με ένα γεγονός το οποίο ονομάζεται IsiWarm event και έχει να κάνει με τον εσωτερικό κώδικα του πρωτοκόλλου ISI, ο οποίος έχει δημιουργηθεί από την Echelon. Μετά την κλήση της συνάρτησης, η μηχανή στέλνει ένα CSMA μήνυμα σύνδεσης. Το μήνυμα αυτό περιέχει ακριβώς τα ίδια πεδία δεδομένων με το κλασικό CSMO και δημιουργείται και αυτό μέσω της συνάρτησης IsiCreateCsmo(). Η μηχανή ISI επίσης στέλνει περιοδικά ένα μήνυμα το οποίο ονομάζεται CSMR για την συντήρηση της σύνδεσης.

Η μηχανή έχει τη δυνατότητα να παρέχει ανατροφοδότηση (feedback) στο χρήστη, μέσω της συνάρτησης `IsiUpdateUserInterface()`, η οποία προσφέρει ενημέρωση για την κατάσταση των συνδέσεων μέσω των events (signals) που διαχειρίζεται (`isiPending`, `isiNormal`, `isiApproved`, `isiImplemented`, `isiCancelled`, `isiDeleted`, `isiWarm`). Τα events αυτά δίνουν πληροφορίες για την κατάσταση της σύνδεσης. Στο σχήμα που ακολουθεί παρουσιάζεται η τυπική ακολουθία των γεγονότων, για μια συσκευή-host η οποία χρησιμοποιεί manual ή controlled σύνδεση.



Σχήμα 5.4: Ακολουθία των γεγονότων σε manual και controlled συνδέσεις

Μια συσκευή η οποία λαμβάνει πρόσκληση για συμμετοχή σε σύνδεση, μπορεί να καθορίσει τα assemblies τα οποία θα συμμετέχουν στη σύνδεση. Η μηχανή ISI καλεί την συνάρτηση `IsiGetAssembly(const IsiCsmoData* pCsmoData, boolean Auto)` έχοντας υπόψη και το μήνυμα σύνδεσης. Η συνάρτηση αυτή επιστρέφει τους αριθμούς των assemblies που μπορούν να πάρουν μέρος στη σύνδεση. Επίσης μέσω του ορίσματος `Auto` μπορεί να διαχωρίσει αν η σύνδεση που πρόκειται να δημιουργηθεί είναι αυτόματη ή όχι. Τέλος καλείται η συνάρτηση `IsiGetNvIndex()` η οποία και θα προσδιορίσει τους selectors για τις μεταβλητές δικτύου των assemblies που πρόκειται να ενωθούν.

Όταν πρόκειται να δημιουργηθεί μια manual ή controlled σύνδεση, χρησιμοποιούνται οι

συναρτήσεις `IsiCreateEnrollment()` και `IsiExtendEnrollment()`, ενώ η μηχανή ISI στέλνει στη συσκευή που δημιούργησε τη σύνδεση ένα μήνυμα αποδοχής (CSME). Η σύνδεση όμως δεν μπορεί να υλοποιηθεί αν δεν την επιβεβαιώσει η συσκευή-host (με ένα μήνυμα επιβεβαίωσης CSMC). Σε αυτού του είδους τις συνδέσεις όπως έχουμε πει, η παρουσία του χρήστη αλλά και της συσκευής (LEDs, buttons) είναι πολύ σημαντική.

Σε αντίθεση με τις manual και controlled συνδέσεις, την αυτόματη σύνδεση τη χειρίζεται αποκλειστικά και μόνο η μηχανή ISI. Η αυτόματη σύνδεση πραγματοποιείται μόνο με την κλήση της `IsiInitiateAutoEnrollment()` και αφού έχει περάσει το random χρονικό διάστημα που ορίζει ο protocol timer T_{auto} (δεν χρειάζεται η αποδοχή της σύνδεσης από καμία συσκευή).

Ένα ιδιαίτερο χαρακτηριστικό των ISI δικτύων είναι ο ISI πίνακας σύνδεσης. Είναι ένας πίνακας ο οποίος περιέχει την κατάσταση αλλά και άλλες χρήσιμες λεπτομέρειες για τις συνδέσεις στις οποίες μια συσκευή έχει λάβει μέρος. Η ISI μηχανή χρησιμοποιεί κατά κόρον τις πληροφορίες αυτού του πίνακα. Ο πίνακας αυτός υλοποιήθηκε στην δική μας εφαρμογή, ενώ χρησιμοποιήθηκαν διάφορες συναρτήσεις για την προσαρμογή του (`IsiGetConnectionTableSize()`, `IsiGetConnection()`, `IsiSetConnection()`).

Μια τελευταία αλλά πολύ σημαντική λειτουργία της μηχανής βασίζεται στο γεγονός ότι οι ISI εφαρμογές πρέπει να υποστηρίζουν μια ιδιότητα η οποία ονομάζεται `SCPTnwrkCnfg`. Αυτή η ιδιότητα απευθύνεται σε ολόκληρη την εφαρμογή και παρέχει ένα interface στα εργαλεία δικτύου (LonMaker), έτσι ώστε αυτά να είναι σε θέση να απενεργοποιήσουν την διαδικασία του self-installation (και κατ' επέκταση την μηχανή ISI) για τις ISI συσκευές. Αυτή η ιδιότητα βοηθάει στην παρατήρηση της κατάστασης της συσκευής και η default τιμή της είναι `CFG EXTERNAL`. Αυτή η τιμή επιτρέπει στο δίκτυο να μετατρέπεται σε διαχειριζόμενο χωρίς κανένα απολύτως σφάλμα. Όταν σε κατάσταση διαχειριζόμενου δικτύου, η ISI συσκευή μπορεί να γυρίσει σε καθεστώς αυτοεγκατάστασης χρησιμοποιώντας ως τιμή για την `SCPTnwrkCnfg` την `CFG LOCAL`.

Ο αναγνώστης μπορεί να ανατρέξει στις πηγές [5], [6] για επιπλέον πληροφορίες.

Κεφάλαιο 6

Η γλώσσα προγραμματισμού Neuron C

6.1 Επισκόπηση

Η *Neuron C* είναι μια γλώσσα προγραμματισμού η οποία βασίζεται στην ANSI C και έχει σχεδιαστεί από την Echelon αποκλειστικά για τα Neuron Chips και τους Smart Transceivers. Η Neuron C υλοποιεί όλους τους βασικούς τύπους δεδομένων της ANSI C (πίνακες, δομές, ενώσεις, δείκτες, συναρτήσεις), αλλά παρέχει και μοναδικά στοιχεία που αφορούν γενικότερα στις LONWORKS εφαρμογές.

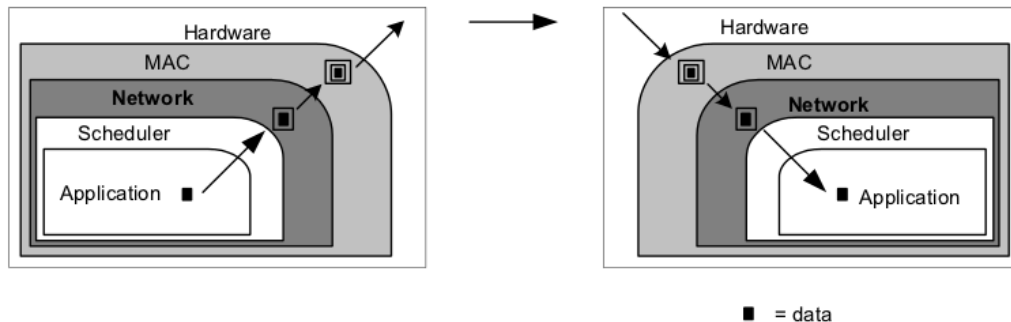
Οι *network variables* είναι ένα από τα στοιχεία αυτά, ώντας τύποι δεδομένων οι οποίοι μοιάζουν πολύ με τις κλασσικές μεταβλητές της ANSI C, με την επιπλέον όμως ικανότητα, αυτή της εύκολης διάδοσης σε όλο μήκος του δικτύου ελέγχου (από την *output network variable* μιας συσκευής στην *input network variable* μιας άλλης συσκευής).

Τα *configuration properties* είναι και αυτά μέρος της διεπαφής της συσκευής και αποτελούν τύπους δεδομένων της Neuron C. Χρησιμοποιούνται για τη ρύθμιση της συμπεριφοράς μιας συσκευής του δικτύου, η οποία πραγματοποιείται μέσω των εργαλείων δικτύου (LonMaker) ή του πρωτοκόλλου ISI.

Η Neuron C παρέχει επίσης και έναν τρόπο οργάνωσης των *network variables* και *configuration properties* με στόχο την εκτέλεση μιας συγκεκριμένης λειτουργίας, ο οποίος ονομάζεται όπως έχουμε δει *functional block*. Αυτά τα *functional blocks* όπως έχουμε επίσης δει ορίζονται από κάποια πρότυπα τα οποία ονομάζονται *functional profiles*.

Η Neuron C έχει σχεδιαστεί για να λειτουργεί στο περιβάλλον το οποίο παρέχεται

από το υλικό και το λογισμικό του συστήματος Neuron. Αυτό το περιβάλλον παρέχει ένα σύστημα προγραμματισμού βασισμένο σε γεγονότα (event-driven), αλλά και μια υπηρεσία μηνυμάτων χαμηλού επιπέδου, η οποία έχει ενσωματωθεί στη Neuron C και παράλληλα με το μοντέλο των network variables οδηγούν σε αποτελεσματικό προγραμματισμό των LONWORKS συσκευών.



Σχήμα 6.1: Αποστολή μηνυμάτων

Ευνοείται επίσης η διαλειτουργικότητα και η αποτελεσματική επικοινωνία μεταξύ των συσκευών. Τα events γίνονται διαχειρίσιμα μέσω κάποιων statements τα οποία ονομάζονται when clauses. Αυτά τα statements υλοποιούν την λειτουργία ενός προγράμματος Neuron, αφού υπάρχει η απουσία της συνάρτησης main() της κλασικής ANSI C.

Άλλος ένας τύπος δεδομένων της Neuron C είναι οι timers. Οι timers μπορούν να δηλωθούν όπως οι μεταβλητές της γλώσσας, ενώ έχουν επίσης και τη δυνατότητα να ενσωματώνουν αποτελεσματικά τα events, το μοντέλο που χρησιμοποιεί ως επί το πλείστον η Neuron C, μαζί με τα network variables.

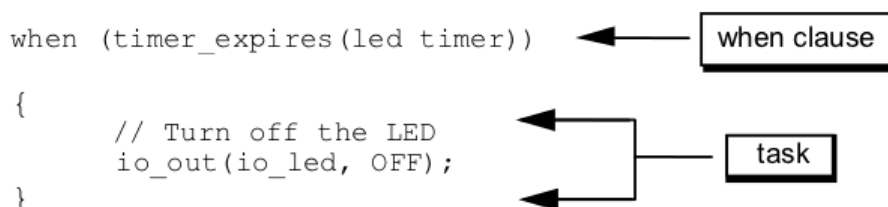
Η Neuron C τέλος παρέχει πολλά αντικείμενα εισόδου/εξόδου, τα οποία μπορούν να θεωρηθούν οδηγοί συσκευών (drivers), ενώ χρησιμοποιούνται και για τη διασύνδεση του προγράμματος της συσκευής (λογισμικό), με το υλικό της (LEDs, buttons, sensors).

Στα προηγούμενα κεφάλαια (4,5), έχει γίνει εκτενής αναφορά στις μεταβλητές δικτύου αλλά και στα configuration properties, όπως επίσης και στον τρόπο με τον οποίο οι μεταβλητές δικτύου μπορούν να συνδεθούν για την αποτελεσματική επικοινωνία των συσκευών ενός δικτύου ελέγχου LONWORKS τεχνολογίας. Σε αυτά τα κεφάλαια επίσης είδαμε και τον τρόπο οργάνωσής τους σε functional blocks. Ο τρόπος με τον οποίο όλα αυτά τα στοιχεία δηλώνονται και χρησιμοποιούνται σε ένα Neuron C πρόγραμμα παρουσιάζεται αναλυτικά στο κεφάλαιο 8, όπου και γίνεται εκτενής ανάλυση του κώδικα των εφαρμογών. Στη συνέχεια του κεφαλαίου αυτού δίνεται μια μικρή περιγραφή για τα υπόλοιπα καινούργια

στοιχεία, τα οποία εισήγαγε η Neuron C στο προγραμματισμό δικτύων ελέγχου.

6.2 Χρονοπρογραμματιστής

Ο χρονοπρογραμματισμός των επιμέρους λειτουργιών του προγράμματος της εφαρμογής είναι βασισμένος στα γεγονότα αλλά και σε μια round-robin λογική : όταν μια συνθήκη που περιέχει ένα event γίνεται αληθής, το σώμα του κώδικα που συνδέεται με αυτή τη συνθήκη εκτελείται, αλλιώς ο χρονοπρογραμματιστής προχωράει στην εξέταση της επόμενης συνθήκης. Όταν φτάσει στο τέλος, έχοντας εξετάσει όλες τις συνθήκες επιστρέφει πάλι στην αρχή (Σχήμα 6.3). Οι συνθήκες ορίζονται μέσα στα when clauses. Το σχήμα που ακολουθεί αποτυπώνει ένα τέτοιο when-clause.



Σχήμα 6.2: When clause

Τα events τα οποία ορίζονται μέσα σε ένα when clause, υπάγονται σε δύο μεγάλες κατηγορίες : τα predefined events και τα events τα οποία ορίζονται από το χρήστη (user-defined events) και μπορεί να αφορούν μια οποιαδήποτε λειτουργία. Η παρακάτω λίστα περιέχει τα πρότυπα predefined events :

- flush_completes
- io_changes, io_update_occurs
- io_in_ready, io_out_ready
- msg_arrives
- msg_completes
- msg_fails, msg_succeeds

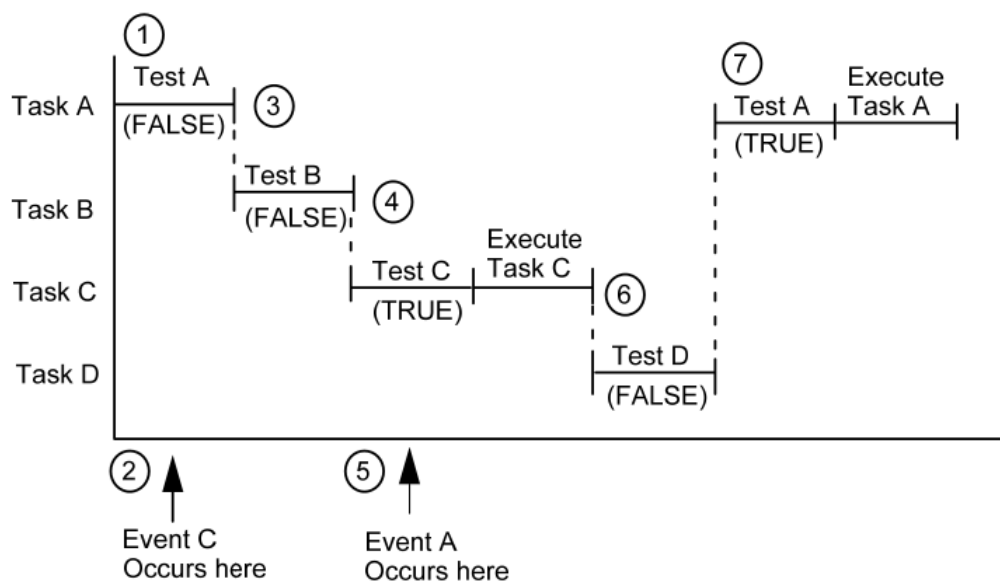
- nv_update_occurs, nv_update_completes
- nv_update_succeeds, nv_update_fails
- resp_arrives
- online, offline
- timer_expires
- wink,reset

Αξίζει να αναφερθούμε σε δύο ιδιαίτερα predefined events : το wink και το reset. Τα events αυτά υλοποιούνται πάντα σε LONWORKS εφαρμογές. Το wink είναι ένα event το οποίο χρησιμεύει στη αναγνώριση μιας συσκευής, μέσω κάποιων ειδικών ενεργειών (flashing leds, audio signals), ανεξάρτητα από το αν η συγκεκριμένη συσκευή μετέχει ή όχι σε ένα δίκτυο. Εκτελεί μια από αυτές τις ενέργειες ως απάντηση σε ένα αίτημα αναγνώρισης από κάποιο χρήστη ή εργαλείο δικτύου. Το reset είναι η γνωστή διαδικασία της επανεκκίνησης μιας συσκευής. Εκτελείται ώντας ένα φυσικό μέρος της εγκατάστασης της συσκευής σε ένα δίκτυο και μόνο όταν η συσκευή βρίσκεται σε κατάσταση ρύθμισης (κατάσταση στην οποία φορτώνεται το πρόγραμμα της εφαρμογής). Επίσης εκτελείται όταν η συσκευή παραμένει απορυθμισμένη λόγω μιας οδηγίας προς τον compiler (`#pragma run_unconfigured`).

Αυτές οι οδηγίες προς τον compiler ονομάζονται *pragmas* και είναι ένα ξεχωριστό χαρακτηριστικό της Neuron C. Χρησιμοποιούνται όταν χρειάζεται να οριστούν ξανά ή να αλλάξουν κάποιες ειδικές παράμετροι ή να μεταβληθεί η κατανομή ειδικών πόρων του συστήματος Neuron. Τέτοιου είδους οδηγίες χρησιμοποιούνται στην παρούσα εργασία και βοηθούν στην κατανομή των μεταβλητών δικτύου σε πίνακες. Η χρησιμότητά τους φαίνεται στο Κεφάλαιο 8, όταν και παρουσιάζεται ο κώδικας της εφαρμογής.

Έστω ότι έχουμε τέσσερα predefined events A, B, C, D τα οποία βρίσκονται εντός κάποιων when-clauses. Το σχήμα 6.3 που ακολουθεί στην επόμενη σελίδα, αποτυπώνει τον τρόπο με τον οποίο μπορεί να λειτουργήσει ένας χρονοπρογραμματιστής βασιζόμενος στις αλλαγές που ενδέχεται να προκύψουν λόγω του διαφορετικού είδους αυτών των predefined events.

Η ροή της λειτουργίας του χρονοπρογραμματιστή του σχήματος 6.3 αποτυπώνεται με βάση τους αριθμούς που βρίσκονται μέσα στους κύκλους και είναι :



Σχήμα 6.3: Παράδειγμα λειτουργίας χρονοπρογραμματιστή

1. Ο χρονοπρογραμματιστής ξεκινάει από το event A. Εφόσον η συνθήκη που το συνοδεύει είναι ψευδής η λειτουργία που ακολουθεί το A δεν εκτελείται.
2. Τη χρονική στιγμή αυτή συμβαίνει το γεγονός C, η συνθήκη γίνεται αληθής και η λειτουργία που τη συνοδεύει εκτελείται.
3. Ο χρονοπρογραμματιστής συνεχίζει στο event B. Εφόσον η συνθήκη του είναι ψευδής, η λειτουργία του δεν εκτελείται.
4. Ο χρονοπρογραμματιστής συνεχίζει στο C. Η λειτουργία του εκτελείται αφού η συνθήκη είναι αληθής.
5. Η συνθήκη του A γίνεται αληθής. Προφανώς πρόκειται για κάποιο ειδικό event, το οποίο μπορεί να διακόπτει τη λειτουργία του χρονοπρογραμματιστή. Μπορεί να είναι για παράδειγμα το `nv_update_occurs`.
6. Ο χρονοπρογραμματιστής συνεχίζει με το event D. Εφόσον η συνθήκη είναι ψευδής, η συνδεδεμένη με αυτό το event λειτουργία αγνοείται.
7. Ο χρονοπρογραμματιστής επιστρέφει στο A. Εφόσον η συνθήκη είναι αληθής, η λειτουργία που το συνοδεύει εκτελείται κανονικά.

Τα περισσότερα εκ των predefined events χρησιμοποιούνται στην υλοποίηση αυτής της εργασίας, μιας και έχουν και το πλεονέκτημα της λιγότερης δυνατής δέσμευσης μνήμης της συσκευής.

6.3 Timers

Δύο τύποι χρονοδιακοπών είναι διαθέσιμοι για τις Neuron C εφαρμογές, οι millisecond και οι second timers. Οι millisecond timers έχουν διάρκεια από 1 έως 64.000 msec, ενώ οι second timers από 1 έως 65.535 sec. Οι timers ορίζονται όπως και οι κλασσικές μεταβλητές. Για παράδειγμα :

```
// start timer with value of 5 sec  
stimer led_timer = 5;
```

Στην εργασία αυτή οι timers συνδυάστηκαν κυρίως με το *timer_expires* event, μια άποψη του οποίου είδαμε στο σχήμα 6.1.

6.4 Αντικείμενα εισόδου/εξόδου

Κάθε Neuron Chip και Smart Transceiver περιέχει μια ποικιλία εγκατεστημένων (built-in) επιλογών διασύνδεσης (interface options), για τη διενέργεια λειτουργιών εισόδου-εξόδου. Πρόκειται για κάποια I/O αντικείμενα τα οποία μπορούν να παρακολουθούν και να ελέγχουν τα 11 από τα 12 pins του Neuron Chip ή του Smart Transceiver (Κεφάλαιο 7) και ονομάζονται IO_0, IO_1,, IO_11. Οι κύριες συναρτήσεις που χρησιμοποιήθηκαν στην παρούσα εργασία για τη διενέργεια λειτουργιών εισόδου/εξόδου είναι οι *io_in()*, *io_out()*. Τα I/O αντικείμενα που χρησιμοποιήθηκαν για να εκτελέσουν λειτουργίες I/O χωρίζονται σε *Direct I/O* μοντέλα, εκ των οποίων χρησιμοποιήθηκαν τα μοντέλα *bit* και *touch*, σε *Serial I/O*, εκ των οποίων χρησιμοποιήθηκε το μοντέλο *bitshift* και σε *Timer/Counter* μοντέλα, εκ των οποίων χρησιμοποιήθηκε το μοντέλο *frequency*. Η δήλωση των I/O objects εκτελεί τις ακόλουθες λειτουργίες :

- Ενημερώνει τον compiler για τον τύπο της λειτουργίας I/O που εκτελείται, αλλά και με ποιο pin του Neuron core συνδέεται αυτή η λειτουργία.
- Συνδέει το όνομα του I/O αντικειμένου με το υλικό της συσκευής.

Ένα παράδειγμα δήλωσης I/O object είναι το ακόλουθο :

```
IO_3 input bit ioProxDetector;
```

Ο αναγνώστης μπορεί να ανατρέξει στις πηγές [7], [3], [16] για επιπλέον πληροφορίες.

Μέρος ΙΙ

ΥΛΟΠΟΙΗΣΗ

Κεφάλαιο 7

Ανάλυση του υλικού

7.1 Επισκόπηση

Το κεφάλαιο αυτό περιγράφει τα επιμέρους κομμάτια της σχεδίασης του υλικού (PL evaluation boards), που χρησιμοποιήθηκε σε αυτή τη διπλωματική εργασία. Τα περισσότερα κομμάτια της σχεδίασης είναι κοινά και στα δύο boards. Για την ολοκλήρωση της μελέτης του υλικού της εργασίας, περιγράφεται επίσης το Mini Gizmo I/O board καθώς και το USB interface.

Οι PL3120 και PL3150 Smart Transceivers ενσωματώνουν ένα πυρήνα επεξεργαστή Neuron και ένα πομποδέκτη (power line). Αποτελούν την κορυφαία λύση για συσκευές ήχου/βίντεο, φωτισμού, ασφάλειας αλλά και ψύξης/θέρμανσης. Ουσιαστικά είναι ένα σύστημα πάνω σε chip (system-on-chip) το οποίο χρησιμοποιείται για να τρέχει εφαρμογές αλλά και για να διαχειρίζεται την επικοινωνία του δικτύου, ενσωματώνοντας μια ποικιλία εσωτερικής και εξωτερικής μνήμης.

Οι Power Line Smart Transceivers μπορούν να επικοινωνήσουν μέσω AC ή DC, συνδυάζοντας μια σειρά από ιδιαίτερα χαρακτηριστικά, τα οποία τους επιτρέπουν να λειτουργούν γρήγορα και αξιόπιστα. Τα ιδιαίτερα αυτά στοιχεία παρουσιάζονται στη λίστα που ακολουθεί :

- Μοναδική διπλή φέρουσα συχνότητα, η οποία αυτόματα επιλέγει μια εναλλακτική συχνότητα για να γίνει η πρωτογενής, όταν αυτή έχει μπλοκαριστεί από το θόρυβο.
- Αλγόριθμος FEC ο οποίος είναι αρκετά αποτελεσματικός στην διόρθωση λαθών που δημιουργούνται από το θόρυβο.

- Προηγμένη ψηφιακή επεξεργασία σήματος και αλγόριθμοι διόρθωσης της παραμόρφωσης.
- Υψηλή έξοδος, σχεδίαση εξωτερικού ενισχυτή για χαμηλή παραμόρφωση, η οποία μπορεί να αποφέρει μέχρι και 1Ar-p σε χαμηλά φορτία αντίστασης, εξαλείφοντας την ανάγκη για δαπανηρούς ζεύκτες σε οικιακές εφαρμογές.

Τα γενικά χαρακτηριστικά των δύο PL Smart Transceivers δίνονται στον πίνακα που ακολουθεί.

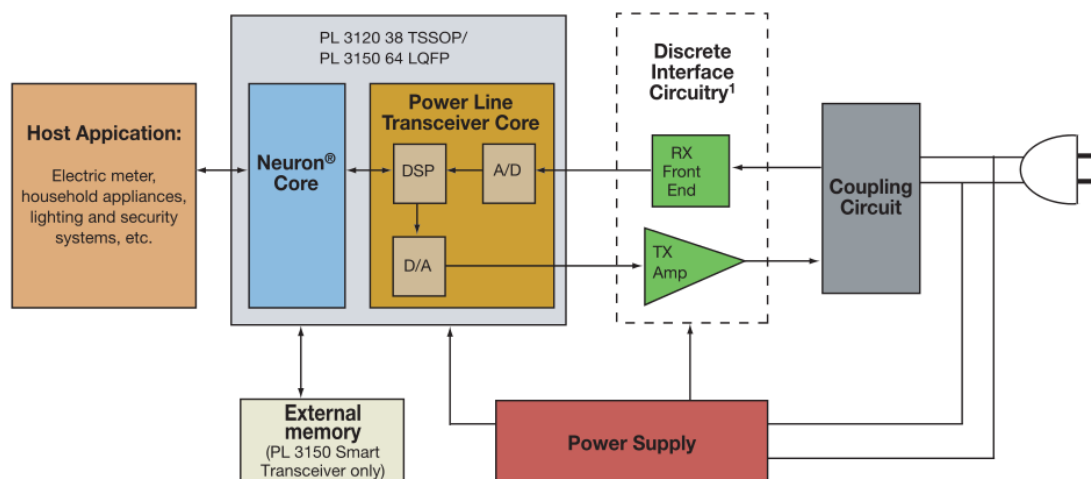
Χαρακτηριστικά	PL3150 Smart Transceiver	PL3120 Smart Transceiver
RAM Bytes	2.048	2.048
ROM Bytes	–	24.576
EEPROM Bytes	512	4096
I/O pins γενικού σκοπού	12	12
16-Bit Timer/Counters	2	2
Interface εξωτερικής μνήμης	NAI	OXI
Πακέτο	64 pin LQFP	38 pin TSSOP

Πίνακας 7.1: Χαρακτηριστικά των PL Smart Transceivers

Τα PL evaluation boards μπορούν να χρησιμοποιηθούν για αρκετές εφαρμογές ανα τον κόσμο, καθώς είναι συμβατές με τους κανονισμούς του πρωτοκόλλου European CENELEC EN50065-1 (Κεφάλαιο 2), αλλά και με άλλους (Industry Canada, Japan MPT, FCC). Οι PL Smart Transceivers υλοποιούν το πρωτόκολλο CENELEC, το οποίο μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί από το χρήστη. Το γεγονός αυτό εξαλείφει την ανάγκη των χρηστών να δημιουργούν πολύπλοκους αλγορίθμους για την πρόσβαση στο μέσο. Οι PL Smart Transceivers όπως έχουμε πει μπορούν να λειτουργούν είτε στο CENELEC utility (A-band) είτε στο general signaling (C-band). Στη δική μας εργασία χρησιμοποιήθηκε το εύρος συχνοτήτων γενικής σηματοδότησης C-band.

7.2 Μπλοκ διάγραμμα των PL 3120 / PL 3150 Power Line Smart Transceivers

Το παρακάτω σχήμα αποτυπώνει το μπλοκ διάγραμμα μιας LonWorks συσκευής, τα επιμέρους κομμάτια δηλαδή της σχεδίασης του υλικού μιας τέτοιας συσκευής.



Σχήμα 7.1: Μπλοκ διάγραμμα των PL3120/PL3150 Smart Transceivers

Στη συνέχεια του κεφαλαίου δίνεται μια αναλυτική περιγραφή των διαφόρων στοιχείων του γενικού αυτού μπλοκ λειτουργικότητας μιας LonWorks συσκευής. Αυτό το μπλοκ αποτελείται από έναν αριθμό πολύ “φθηνών”, εξωτερικών στοιχείων τα οποία ουσιαστικά συνθέτουν ένα κόμβο *LonWorks*.

7.2.1 Πυρήνας Neuron

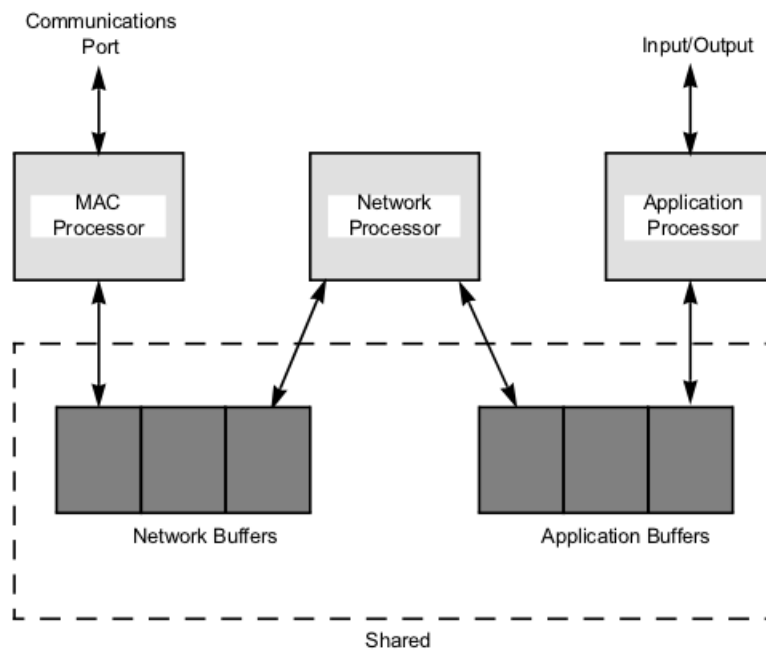
Το τμήμα του πυρήνα Neuron αποτελεί το κυριότερο κομμάτι του συστήματος LonWorks και είναι αυτό που διαχειρίζεται την επικοινωνία και τον έλεγχο των υπολοίπων τμημάτων του μπλοκ διαγράμματος που παρουσιάστηκε νωρίτερα.

Ο πυρήνας Neuron αποτελείται από τρεις επεξεργαστές. Ο επεξεργαστής 1 είναι ο επεξεργαστής του επιπέδου MAC και χειρίζεται τα επίπεδα 1 και 2 της στοίβας του πρωτοκόλλου LonTalk. Περιλαμβάνει την εκτέλεση του αλγορίθμου ελέγχου πρόσβασης στο κανάλι. Ο επεξεργαστής 1 επικοινωνεί με τον επεξεργαστή 2 μέσω των buffers του δικτύου που βρίσκονται στην κοινή μνήμη RAM.

Ο επεξεργαστής 2 είναι ο επεξεργαστής δικτύου, ο οποίος υλοποιεί τα επίπεδα 3 έως 6

του πρωτοκόλλου LonTalk. Χειρίζεται την επεξεργασία των μεταβλητών δικτύου, την επεξεργασία της επικοινωνίας, τη διευθυνσιοδότηση, τον έλεγχο ταυτότητας των μηνυμάτων, τις συναρτήσεις δρομολόγησης και γενικότερα τη διαχείριση του δικτύου. Ο επεξεργαστής αυτός χρησιμοποιεί buffers εφαρμογών (βρίσκονται και αυτοί στην κοινή μνήμη RAM) για την επικοινωνία του με τον επεξεργαστή 3.

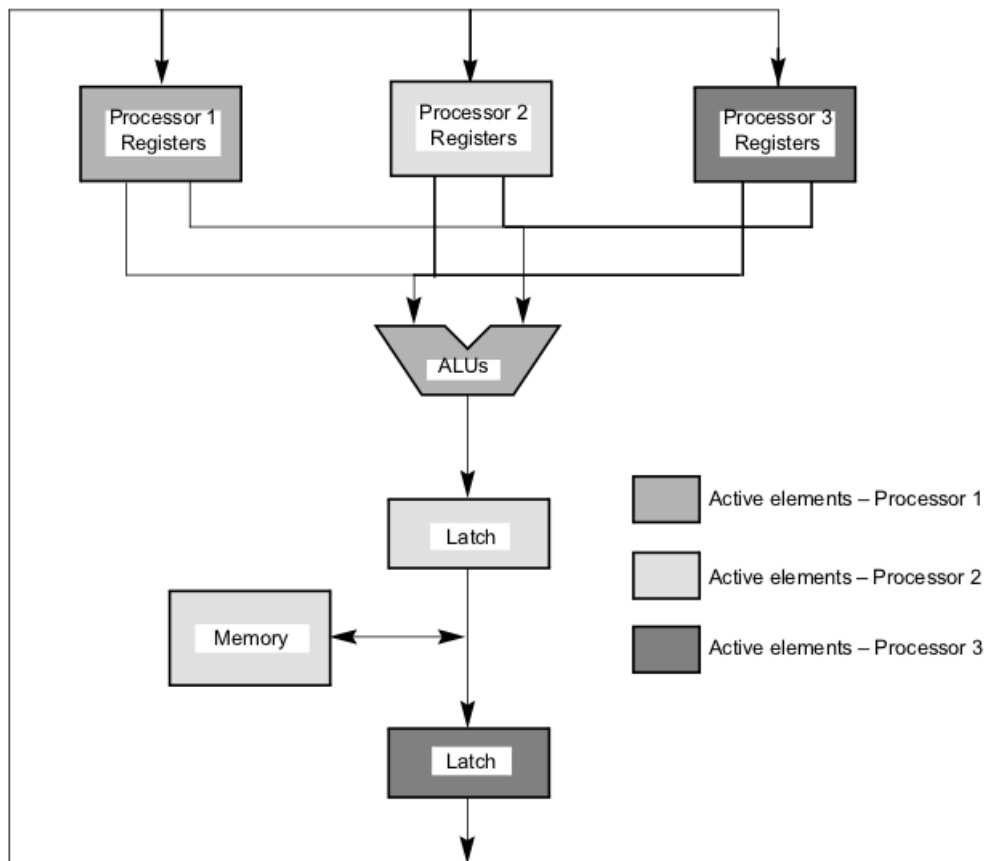
Ο επεξεργαστής 3 είναι ο επεξεργαστής της εφαρμογής. Εκτελεί τον κώδικα που έχει γράψει ο χρήστης μαζί με τις υπηρεσίες του λειτουργικού συστήματος που επίσης καλεί ο χρήστης. Ο κώδικας των εφαρμογών πρέπει να είναι πάντα γραμμένος σε Neuron C, μια γλώσσα προγραμματισμού, τα κυριότερα χαρακτηριστικά της οποίας περιγράφηκαν στο τελευταίο κεφάλαιο του πρώτου μέρους της εργασίας.



Σχήμα 7.2: Μια άποψη της αρχιτεκτονικής του πυρήνα Neuron

Κάθε ένας από αυτούς τους τρεις πανομοιότυπους επεξεργαστές έχει το δικό του σύνολο καταχωρητών, αλλά και οι τρεις μαζί μπορούν να μοιράζονται δεδομένα, ALU's (αριθμητικές και λογικές μονάδες) αλλά και κυκλώματα πρόσβασης στη μνήμη. Κάθε μικρός κύκλος της CPU (minor cycle) αποτελείται από τρεις μικρούς κύκλους ρολογιού ή φάσεις. Κάθε φάση ισούται με δύο κύκλους ρολογιού εισόδου. Οι μικροί κύκλοι (minor cycles) των τριών επεξεργαστών διαχωρίζονται από ένα κύκλο ρολογιού του συστήματος,

έτσι ώστε κάθε επεξεργαστής να μπορεί να προσπελάσει τη μνήμη και την ALU μια φορά σε κάθε κύκλο του instruction (Ανάθεση εντολής). Αυτό επιτρέπει την παράλληλη εκτέλεση και των τριών επεξεργαστών χωρίς να έχουμε χρονοβόρες διακοπές αλλά και μεταγωγές πλαισίων (context switching). Στο σχήμα που ακολουθεί βλέπουμε μια άποψη της δραστηριότητας των τριών επεξεργαστών στη διάρκεια μιας φάσης ενός minor cycle.



Σχήμα 7.3: Λειτουργία των επεξεργαστών κατά τη διάρκεια μιας φάσης ενός minor cycle

7.2.1.1 Μνήμη

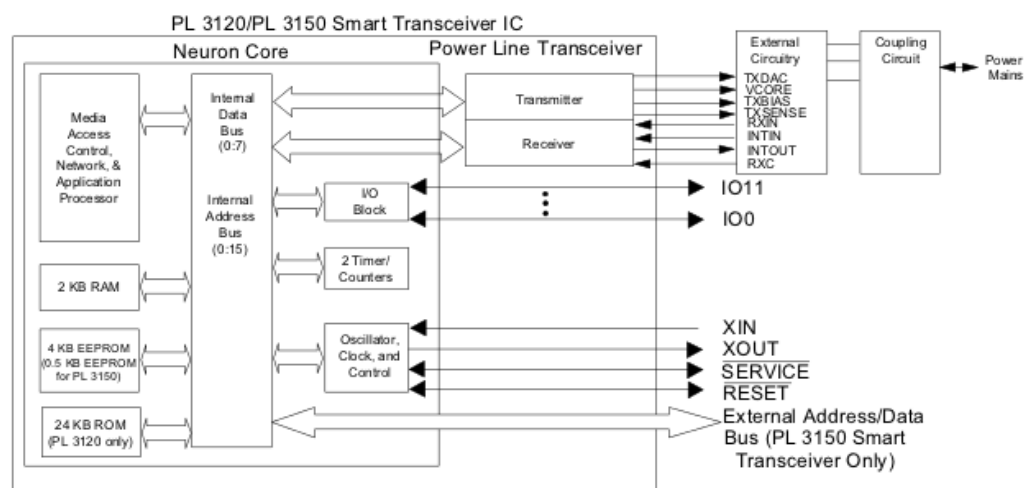
Κατανομή μνήμης για τον PL 3120 Smart Transceiver

- 4.096 bytes εσωτερικής μνήμης EEPROM για την αποθήκευση πληροφοριών που αφορούν την ρύθμιση του δικτύου, τις διευθύνσεις, το Neuron ID καθώς και το κώδικα της εφαρμογής ο οποίος έχει γραφτεί από τον χρήστη.

- 2.048 bytes στατικής μνήμης RAM για την αποθήκευση δεδομένων του συστήματος, network και application buffers.
- 24.576 bytes ROM για την αποθήκευση του εσωτερικού κώδικα του πυρήνα Neuron (προγράμματα και βιβλιοθήκες).

Κατανομή μνήμης για τον PL 3150 Smart Transceiver

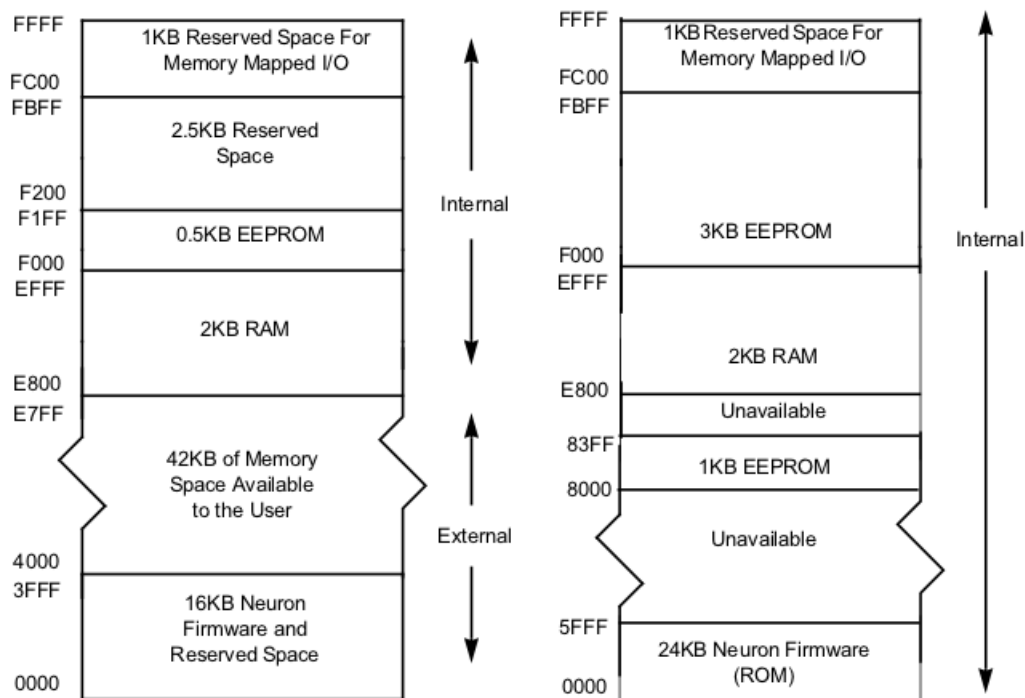
- 512 bytes εσωτερικής μνήμης EEPROM για την αποθήκευση πληροφοριών που αφορούν την ρύθμιση του δικτύου, τις διευθύνσεις, το Neuron ID καθώς και το κώδικα της εφαρμογής ο οποίος έχει γραφτεί από τον χρήστη.
- 2.048 bytes στατικής μνήμης RAM για την αποθήκευση δεδομένων του συστήματος, network και application buffers.
- Ο επεξεργαστής μπορεί να προσπελάσει 59.392 bytes από τα διαθέσιμα 65.536 bytes του χώρου διευθύνσεων, μέσω του interface της εξωτερικής μνήμης που διαθέτει αυτό το board.
- 16.384 bytes εξωτερικής μνήμης από το συνολικό διαθέσιμο των 59.392 bytes, για την αποθήκευση του εσωτερικού κώδικα λειτουργίας του πυρήνα Neuron
- Τα υπόλοιπα bytes της εξωτερικής μνήμης είναι διαθέσιμα για τον κώδικα της εφαρμογής και για τους διάφορους buffers (network και application).



Σχήμα 7.4: Το interface των πυρήνων PL 3120 / PL 3150

Το παραπάνω σχήμα αποτυπώνει το μπλοκ διάγραμμα των πυρήνων PL 3120 και PL 3150.

Το σχήμα που ακολουθεί δείχνει την κατανομή της μνήμης στα δύο boards. Στο αριστερό μέρος είναι ο χάρτης μνήμης του PL 3150 Smart Transceiver ενώ στο δεξιό μέρος είναι ο χάρτης μνήμης του PL 3120 Smart Transceiver.

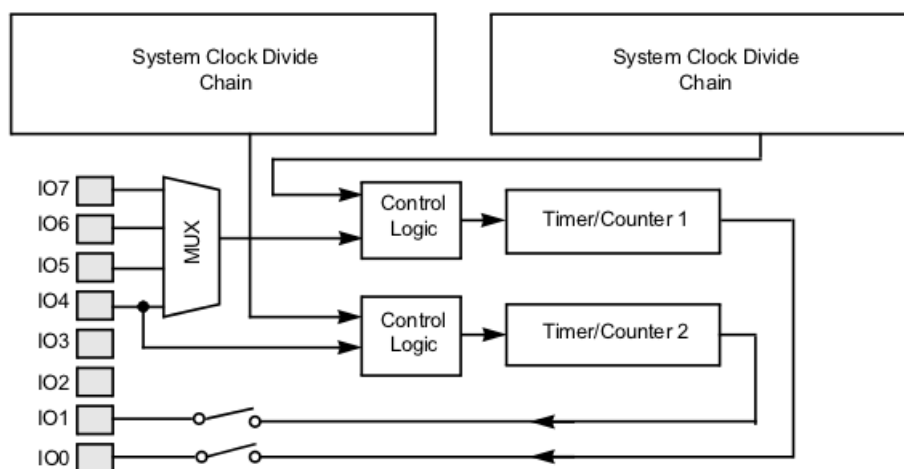


Σχήμα 7.5: Κατανομή μνήμης στα PL 3150 και PL 3120

7.2.1.2 Είσοδος/ Έξοδος

Οι PL Smart Transceivers διαθέτουν 12 αμφίδρομα pins, τα οποία μπορούν να χρησιμοποιηθούν για την παροχή ευέλικτης διασύνδεσης με το εξωτερικό υλικό, αλλά και την παροχή πρόσβασης στους εσωτερικούς timers/counters. Τα pins IO_4 – IO_7 και IO_11 διαθέτουν προγραμματιζόμενες pull-up πηγές ρεύματος, οι οποίες ενεργοποιούνται ή απενεργοποιούνται μέσω κάποιων οδηγιών προς τον compiler (pragmas). Τα pins IO_0 – IO_3 έχουν μια ικανότητα δρομολόγησης του ρεύματος της τάξης των 20 mA @ 0.8 V. Τα υπόλοιπα pins έχουν ικανότητα της τάξης των 1.4 mA @ 0.5 V. Όλα τα pins έχουν TTL επίπεδα εισόδου, ενώ τα pins IO_0 – IO_7 διαθέτουν και μανδαλωτές (latches) χαμηλού επιπέδου ανίχνευσης.

Οι PL Smart Transceivers διαθέτουν και δύο 16-bit timer/counters (εσωτερικά), οι οποίοι έχουν υλοποιηθεί ως καταχωρητές φορτίου ικανοί για εγγραφή από τον επεξεργαστή (16-bit counter), αλλά και ως αναγνώσιμα latch (16-bit timer). Οι καταχωρητές αυτοί προσπελούνται κατά 1-byte τη φορά. Ο πρώτος timer/counter έχει μια είσοδο η οποία κυμαίνεται μεταξύ των pins IO_4 – IO_7 και μια έξοδο στο pin IO_0. Η είσοδος του δεύτερου timer/counter είναι το pin IO_4 και η έξοδος του είναι το pin IO_1. Το σχήμα που ακολουθεί δείχνει τα κυκλώματα των timer/counters.



Σχήμα 7.6: Εσωτερικά κυκλώματα των timer/counters

Οι PL Smart Transceivers χρειάζονται για τη λειτουργία τους ένα εσωτερικό ρολόι των 10MHz το οποίο συντονίζει τις ενέργειες του κυκλώματος. Επίσης παρέχουν δύο σήματα εξόδου (signals), τα PKD και BIU, τα οποία προορίζονται για να οδηγούν χαμηλής –κατανάλωσης διόδους εκπομπής φωτός (LEDs). Τα δύο αυτά σήματα είναι αρκετά σημαντικά για την υλοποίηση και συνδέονται σε διαφορετικά LEDs, τα οποία βρίσκονται πάνω στα PL 3120 και PL 3150 boards ενημερώνοντας το χρήστη.

Το σήμα BIU (Band-In-Use detector) ενεργοποιείται όταν ένα σήμα ξεπεράσει τα $86dBmV_{RMS} * 10^{-3}$ για 4msec, οπουδήποτε στο εύρος συχνοτήτων 131.5kHz – 133.5kHz. Χρησιμοποιείται για να υποδηλώσει τη χρήση του εύρους ζώνης από την εφαρμογή, λειτουργώντας παράλληλα με το πρωτόκολλο CENELEC . Το πρωτόκολλο αυτό όταν είναι ενεργοποιημένο καθορίζει τότε μια συσκευή είναι σε θέση να μεταδώσει ενεργοποιώντας το LED του σήματος BIU.

Το σήμα PKD (Packet Detect) είναι ενεργό όταν ένα πακέτο LonTalk, το οποίο

έχει πάνω του τη διεύθυνση μιας οποιασδήποτε συσκευής, λαμβάνεται από τον PL Smart Transceiver. Το σήμα αυτό έχει μικρότερη ευαισθησία από το BIU καθώς ενεργοποιεί το LED που του έχει δωθεί, όταν ανιχνεύσει σήμα με επίπεδο μικρότερο των $36dBmV_{RMS} * 10^{-3}$. Οι Smart Transceivers διαθέτουν επίσης και το σήμα TXON (οδηγεί και αυτό ένα δικό του LED στο board), το οποίο είναι ενεργό όταν μια οποιαδήποτε συσκευή μεταδίδει πακέτα. Το σήμα αυτό ουσιαστικά χρησιμοποιείται για να ελέγχει την έξοδο του ενισχυτή του πομποδέκτη.

Ουσιαστικά με αυτά τα LEDs είμαστε σε θέση να διαπιστώσουμε πότε ένα πακέτο στέλνεται από μια συσκευή σε μια άλλη, αλλά και πότε ένα πακέτο λαμβάνεται από μια συσκευή. Υπάρχουν βέβαια και άλλα σήματα τα οποία συνδέονται με την μετάδοση και λήψη πακέτων. Όλα αυτά τα σήματα γίνονται διαχειρίσιμα από το εξωτερικό κύκλωμα (εκτός του Neuron interface), το οποίο συνδέεται με τον πομποδέκτη (external circuitry— Σχήμα 7.4).

Υπάρχουν επιπλέον σήματα τα οποία σχετίζονται με το κομμάτι της μετάδοσης του πομποδέκτη και είναι :

- TXDAC: Ψηφιακή σε αναλογική μετάδοση
- TXSENSE: Ανάδραση του κυκλώματος (feedback)
- TXBIAS: Ρύθμιση της μετάδοσης
- VCORE: Σήμα διακοπής μετάδοσης

Υπάρχουν επίσης και επιπλέον σήματα, τα οποία σχετίζονται με το κομμάτι της λήψης πακέτων ή μηνυμάτων (receiver):

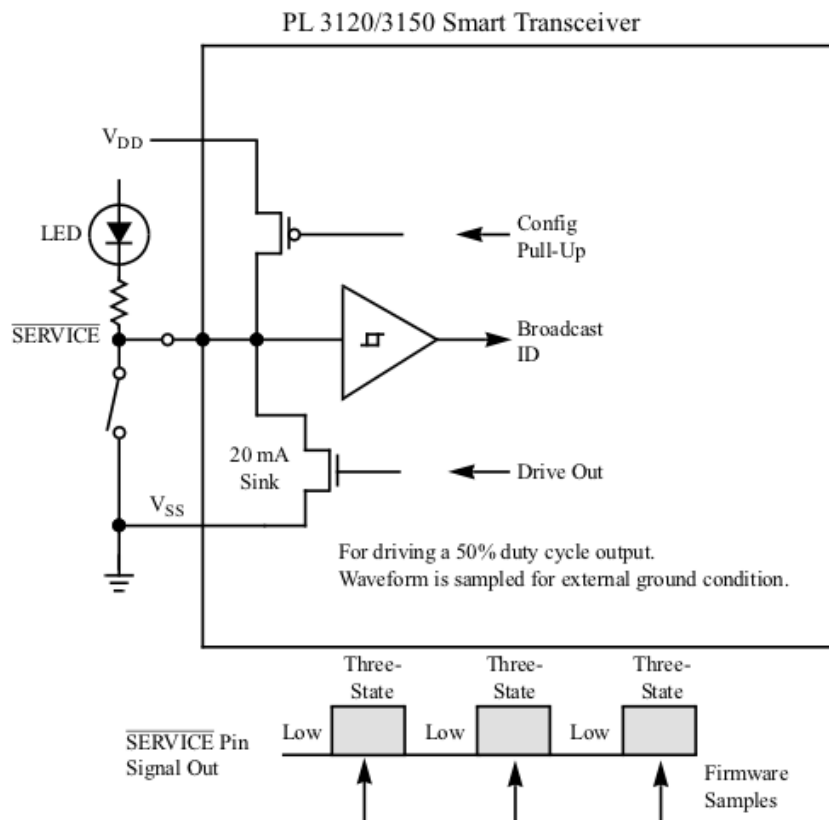
- RXIN: Είσοδος δεδομένων
- RXC: Προσαρμοσμένα δεδομένα εισόδου (Analog- to - digital)
- INTOUT, INTIN : Σήματα για την φάση της προσαρμογής των δεδομένων.

Μια πολύ σημαντική λειτουργία, την οποία ενσωματώνουν τα PL evaluation boards, είναι η λειτουργία reset. Η λειτουργία αυτή παίζει ένα αρκετά σημαντικό ρόλο όσον αφορά τα ακόλουθα ζητήματα:

- Εξασφαλίζει την σωστή αρχικοποίηση του PL Smart Transceiver μέσω της τάσης V_{dd5} .

- Εξασφαλίζει την κατάλληλη αξιοποίηση του PL Smart Transceiver όταν σταθεροποιηθεί η V_{DD5} .
- Ανάκτηση του προγράμματος της εφαρμογής.
- Εξασφαλίζει σωστό κλείσιμο της συσκευής.
- Βοηθάει στην προστασία της εσωτερικής μνήμης EEPROM από την κατάρρευση.

Η λειτουργία reset μπορεί να πραγματοποιηθεί είτε μέσω του προγράμματος της εφαρμογής (Neuron C κώδικας), είτε μέσω του reset pin το οποίο και βρίσκεται πάνω στα boards. Μια ακόμα χρήσιμη λειτουργία του PL evaluation board είναι η λειτουργία service μέσω του service pin. Το κουμπί αυτό χρησιμοποιείται για την μετάδοση του Neuron ID αλλά και του ID της εφαρμογής στο δίκτυο. Το τυπικό κύκλωμα της λειτουργίας του service button δίνεται στο ακόλουθο σχήμα.

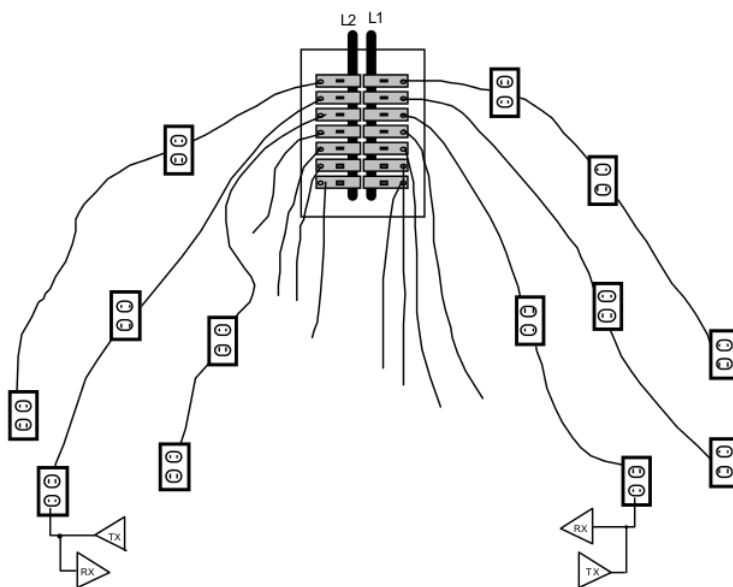


Σχήμα 7.7: Τυπικό κύκλωμα της λειτουργίας service

Σε αυτή την ενότητα δόθηκε η περιγραφή της αρχιτεκτονικής και της λειτουργίας του Neuron πυρήνα, μαζί με τη λειτουργία του πομποδέκτη (Smart Transceiver). Δόθηκε επίσης μια περιγραφή του εξωτερικού κυκλώματος (discrete interface circuit), το οποίο συνδέεται με τον πομποδέκτη και περιλαμβάνει τον ενισχυτή της μετάδοσης αλλά και το δέκτη (μαζί με σήματα που τους συνοδεύουν).

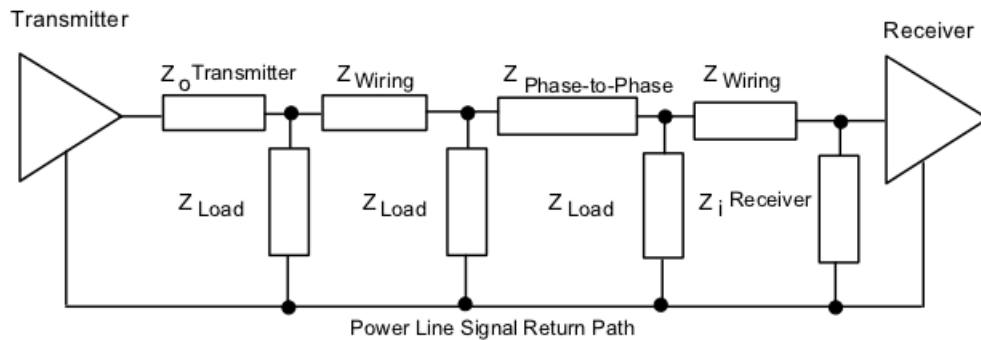
7.2.2 Κύκλωμα ζεύξης

Στην ενότητα αυτή περιγράφεται το κύκλωμα ζεύξης (coupling circuit–Σχήμα 7.1), άλλο ένα κομμάτι του λειτουργικού μπλοκ ενός κόμβου LonWorks. Το κύκλωμα ζεύξης υλοποιεί τη διασύνδεση των PL 3120 και PL 3150 Smart Transceivers με τις γραμμές μεταφοράς του ηλεκτρικού ρεύματος. Οι PL Smart Transceivers χρησιμοποιούν προηγμένες τεχνικές ψηφιακής επεξεργασίας σήματος, σε συνδυασμό με έναν ενισχυτή μετάδοσης χαμηλής αντίστασης εξόδου, αλλά και με ένα πολύ μεγάλο δυναμικό εύρος (> 80db) δέκτη. Τα στοιχεία αυτά βοηθούν στο να ξεπερασθεί η εξασθένηση του σήματος (attenuation), αλλά και ο θόρυβος ο οποίος δημιουργείται και εμποδίζει την επικοινωνία. Η εξασθένηση ενός σήματος ορίζεται ως ο λόγος του επιπέδου του σήματος στην έξοδο του πομπού (transmitter), προς την τιμή του επιπέδου του σήματος στην είσοδο του δέκτη (receiver). Το σχήμα που ακολουθεί δείχνει την πορεία που μπορεί να ακολουθήσει ένα σήμα ξεκινώντας από μια πρίζα και καταλήγοντας σε μια άλλη μέσω της καλωδίωσης ενός κτιρίου.



Σχήμα 7.8: Πιθανή διαδρομή σήματος σε ένα PL δίκτυο

Υπάρχουν πολλές πηγές εξασθένησης του σήματος σε ένα δίκτυο που χρησιμοποιεί την καλωδίωση του ρεύματος ως βασικό κανάλι όπως για παράδειγμα οι πρίζες του προηγούμενου σχήματος, οι οποίες μπορούν να παράγουν θόρυβο. Το μοντέλο εξασθένησης σε ένα κανάλι PL παρουσιάζεται στο επόμενο σχήμα.

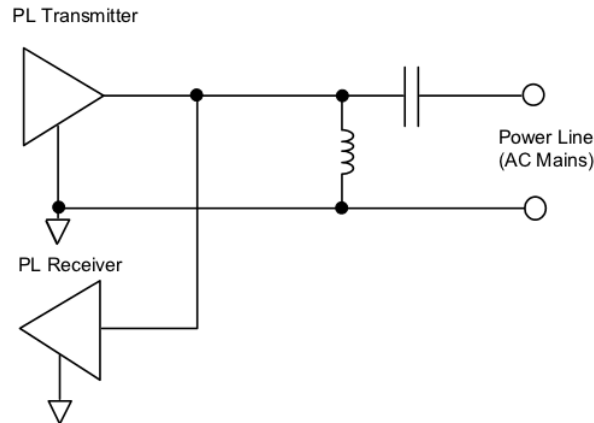


Σχήμα 7.9: Μοντέλο εξασθένησης

Στις συσκευές επικοινωνίας LonWorks, η μετάδοση αλλά και η λήψη δεδομένων πραγματοποιούνται μέσω της ίδιας φυσικής γραμμής. Το κύκλωμα ζεύξης προσπαθεί να διαχωρίσει την μετάδοση από τη λήψη, ελαχιστοποιώντας παράλληλα το θόρυβο. Η εισαγωγή ενός σήματος επικοινωνίας στο δίκτυο ισχύος (power mains), επιτυγχάνεται με τη χωρητική σύζευξη της εξόδου του πομποδέκτη στο δίκτυο αυτό. Εκτός από αυτό χρειάζεται και ένα πηνίο ή ένας μετασχηματιστής. Το πηνίο και ο πυκνωτής λειτουργούν σαν ένα υψιπερατό (high-pass) φίλτρο για το σήμα επικοινωνίας του πομποδέκτη. Το φίλτρο αυτό εξασθενεί το μεγάλο AC σήμα, ενώ παράλληλα αφήνει να διέλθει το σήμα επικοινωνίας του πομποδέκτη. Η τιμή του πυκνωτή πρέπει να είναι αρκετά μεγάλη, έτσι ώστε η αντίστασή του στις συχνότητες επικοινωνίας να είναι χαμηλή, αλλά και αρκετά μικρή έτσι ώστε να έχει υψηλή αντίσταση στη συχνότητα του ηλεκτρικού ρεύματος. Η αντίσταση του πυκνωτή μπορεί να θεωρηθεί μέρος της αντίστασης εξόδου του πομπού (Z_o Transmitter– Σχήμα 7.9). Διατηρώντας την αντίσταση του πυκνωτή χαμηλή ελαχιστοποιούμε την απώλεια σήματος, η οποία προκαλείται από τον διαιρέτη τάσης που σχηματίζεται μεταξύ της αντίστασης εξόδου του ενισχυτή και του φορτίου του ρεύματος (Z_{load}).

Η αντίσταση του πηνίου επιλέγεται να είναι αρκετά υψηλή στις συχνότητες επικοινωνίας των PL Smart Transceivers και θεωρείται μέρος της Z_i Receiver– Σχήμα 7.9. Διατηρώντας σε υψηλά επίπεδα την αντίσταση του πηνίου ελαχιστοποιούμε την απώλεια σήματος στο δέκτη. Το βασικό κύκλωμα ζεύξης με το πηνίο και το πυκνωτή φαίνεται στο παρακάτω

σχήμα.



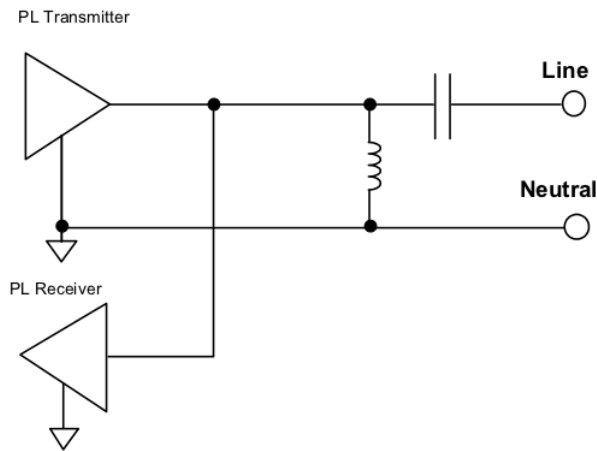
Σχήμα 7.10: Μοντέλο κυκλώματος ζεύξης

Ένας πολύ σημαντικός παράγοντας ο οποίος επηρεάζει τον τύπο του κυκλώματος ζεύξης που χρησιμοποιείται στις εφαρμογές είναι ο τρόπος (στυλ) καλωδίωσης του συστήματος διανομής ισχύος, στο οποίο θα συνδεθεί το κύκλωμα αυτό. Έτσι δημιουργούνται δύο μέθοδοι ζεύξης : η μέθοδος line-to-earth και η line-to-neutral. Στη δική μας εργασία χρησιμοποιείται η δεύτερη μέθοδος ζεύξης. Σύμφωνα με αυτή τη μέθοδο το σήμα επικοινωνίας εισάγεται στη γραμμή (line), ενώ το ουδέτερο καλώδιο χρησιμοποιείται ως το “μονοπάτι” επιστροφής του σήματος. Δηλαδή το κύκλωμα σύζευξης είναι συνδεδεμένο μεταξύ γραμμής –ουδετέρου. Η μέθοδος αυτή καθίσταται ιδανική για οικιστικές εφαρμογές.

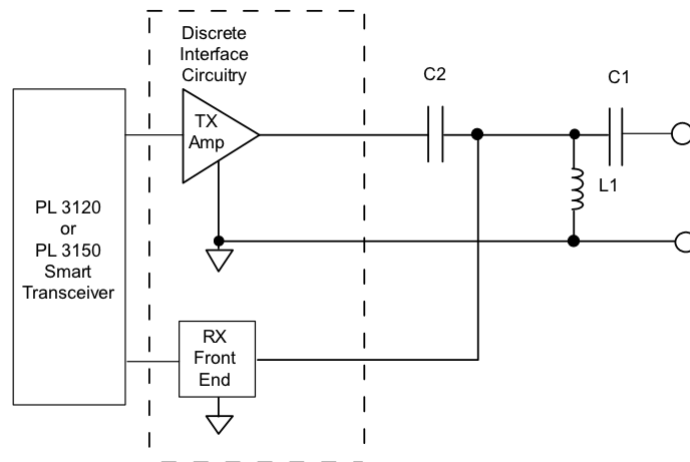
Το μοντέλο που παρουσιάστηκε στο σχήμα 7.10 είναι το γενικό μοντέλο ζεύξης, το οποίο χρησιμοποιεί τη μέθοδο line-to-neutral. Ξεκινώντας την ανάλυση της μεθόδου αλλά και τη σταδιακή δημιουργία ενός πλήρους λειτουργικού κυκλώματος ζεύξης, μπορούμε να προσθέσουμε σε αυτό το μοντέλο και ένα δεύτερο πυκνωτή, έτσι ώστε να αποφευχθεί η μείωση της τάσης (shorting DC voltage) του ενισχυτή μετάδοσης, από το πηνίο που ήδη υπάρχει στο κύκλωμα ζεύξης.

Στο πρώτο σχήμα που ακολουθεί βλέπουμε το γενικό μοντέλο του κυκλώματος ζεύξης, με τη χρήση της line-to-neutral μεθόδου. Αποτυπώνονται οι δύο γραμμές, μέσα στις οποίες τοποθετείται το κύκλωμα, όπως επίσης ο πυκνωτής και το πηνίο.

Στο σχήμα 7.12 αποτυπώνεται η βελτίωση του προηγούμενου γενικού μοντέλου με την προσθήκη του πυκνωτή C2. Με αυτόν τον τρόπο το κύκλωμα γίνεται περισσότερο πρακτικό και λειτουργικό.



Σχήμα 7.11: Μοντέλο κυκλώματος ζεύξης Line-to-Neutral

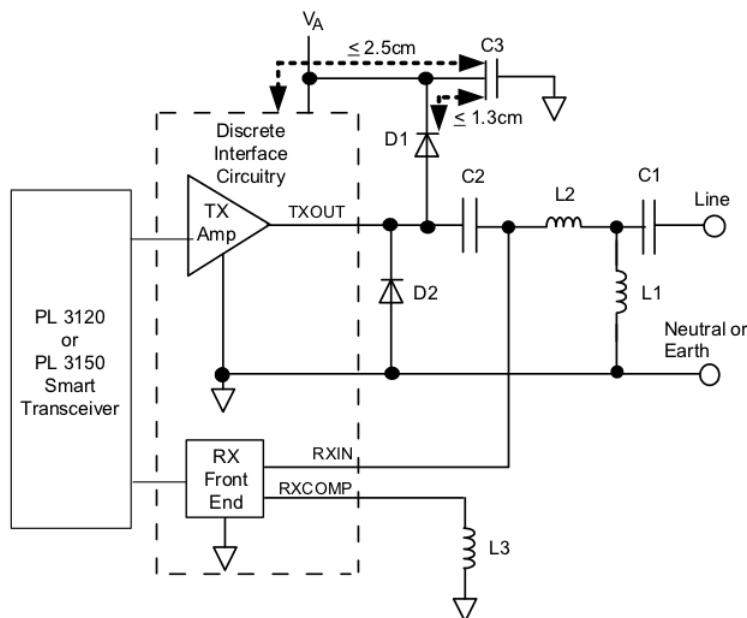


Σχήμα 7.12: Προσθήκη πυκνωτή C2 στο αρχικό κύκλωμα ζεύξης

Ο συνδυασμός των πυκνωτών C1,C2 έχει χαμηλή αντίσταση στις συχνότητες επικοινωνίας των PL Smart Transceivers και μάλιστα είναι της τάξης του 1ohm. Αυτό πρέπει να συμβαίνει για λόγους αντιστάθμισης, επειδή η αντίσταση του φορτίου της γραμμής ισχύος είναι και αυτή 1ohm αλλά και επειδή η αντίσταση εξόδου του ενισχυτή του PL Smart Transceiver είναι μικρότερη από 1ohm.

Συνεχίζοντας την ανάλυση της μεθόδου line-to-neutral του κυκλώματος ζεύξης, ένας πολύ απλός και οικονομικά αποδοτικός τρόπος επίτευξης χαμηλής αντίστασης για τη μετάδοση είναι η προσθήκη ενός πηνίου L2. Το πηνίο αυτό σχηματίζει ένα κύκλωμα σε σειρά, με τους C1,C2 και η τιμή του επιλέγεται έτσι ώστε να ελαχιστοποιείται το κόστος των δύο πυκνωτών. Υπάρχουν όμως και κάποιες τελικές προσθήκες, οι οποίες κάνουν το κύκλωμα

ζεύξης πλήρως λειτουργικό. Το τελικό αυτό κύκλωμα ζεύξης παρουσιάζεται στο σχήμα που ακολουθεί.



Σχήμα 7.13: Πλήρως λειτουργικό line-to-neutral κύκλωμα ζεύξης

Το πηνίο L3 συνδέεται με το κύκλωμα φιλτραρίσματος του δέκτη και η DC αντίστασή του μπορεί να φτάσει μέχρι και τα 55ohm. Οι δύο διόδους D1 και D2 είναι συνδεδεμένες από τον πομπό στον ενισχυτή, με σκοπό να προστατεύσουν τις εισόδους του PL Smart Transceiver από σήματα μεγαλύτερα των 18V. Ο πυκνωτής C3 έχει προστεθεί για να τονίσει το γεγονός ότι αποτελεί ένα αναπόσπαστο κομμάτι του κυκλώματος, έχοντας σαν κύρια λειτουργία του την προστασία της τροφοδοσίας (μέσω της τάσης V_A), από overshoot, όταν θετικής πολικότητας κύματα, εκφορτίζονται μέσω της διόδου D1 και έτσι η διάδος οδηγεί μεγάλο ρεύμα, το οποίο μπορεί όμως να επηρεάσει διάφορα στοιχεία του κυκλώματος τα οποία συνδέονται με την V_A και να τα καταστρέψει. Σε περιπτώσεις κατά τις οποίες αναπτύσσονται μεγάλα μαγνητικά πεδία, είναι πιθανό ένα ή περισσότερα από τα πηνία του κυκλώματος ζεύξης να τα απορροφήσουν και να τα οδηγήσουν στο δίκτυο του ρεύματος.

Για να αποφευχθεί η εξασθένηση των σημάτων που λαμβάνονται στο δέκτη, το επίπεδο αντίστασης του ενισχυτή μετάδοσης ορίζεται περίπου στα 500ohms όταν δεν μεταδίδει. Η αντίσταση λήψης είναι μεγαλύτερη των 250ohms και για το A αλλά και για το C-band.

7.2.3 Τροφοδοσία

Αυτό το τμήμα του κόμβου (Power Supply block–Σχήμα 7.1) διαχειρίζεται την τροφοδοσία της LonWorks συσκευής. Είναι σημαντικό να εξασφαλίσουμε ότι η τροφοδοσία δεν περιορίζει τη συνολική απόδοση της επικοινωνίας των PL Smart Transceivers. Επειδή η είσοδος της τροφοδοσίας είναι συνδεδεμένη απευθείας στο κανάλι, έχει τη δυνατότητα να εξασθενεί το μεταδιδόμενο σήμα αλλά και να εισάγει θόρυβο στην είσοδο του δέκτη. Ομοίως, οι έξοδοι της τροφοδοσίας, V_A και V_{DD5} , έχουν την ικανότητα να εισάγουν θόρυβο στον πομποδέκτη.

Όπως ειπώθηκε και στην προηγούμενη ενότητα, η εξασθένηση ενός power line σήματος επικοινωνίας είναι ένας καθοριστικός παράγοντας, ο οποίος επηρεάζει τη συνολική απόδοση του συστήματος. Επομένως η επιλογή του κατάλληλου τροφοδοτικού είναι ζωτικής σημασίας για ένα σύστημα. Για παράδειγμα, τροφοδοτικά τα οποία έχουν χαμηλή αντίσταση εισόδου στις συχνότητες επικοινωνίας, απαιτούν την προσθήκη ενός πηνίου σε σειρά με την είσοδο του τροφοδοτικού.

Τα τροφοδοτικά μπορούν να εισάγουν θόρυβο και στις εισόδους αλλά και στις εξόδους τους. Ο θόρυβος που λαμβάνει χώρα στην είσοδο και πάνω στην γραμμή AC, μπορεί να μειώσει την απόδοση της επικοινωνίας. Ομοίως, ο θόρυβος στην έξοδο της τροφοδοσίας, μπορεί να εισαχθεί μέσα στον πομποδέκτη και να μειώσει την απόδοση του συστήματος. Λόγω κάποιων κανονιστικών περιορισμών, είναι σημαντικό να βεβαιωθούμε ότι ο θόρυβος βρίσκεται μέσα στα επιτρεπτά όρια. Γι' αυτό το λόγο κάποια τροφοδοτικά απαιτούν και τη χρήση φίλτρων εισόδου.

Όταν ο PL Smart Transceiver βρίσκεται σε receive mode το επιτρεπόμενο εύρος της τάσης τροφοδοσίας V_A (λαμβάνεται από το ηλεκτρικό δίκτυο) είναι 8.5–18V. Όταν όμως μεταδίδει ένα μήνυμα, η τροφοδοσία αυτή πρέπει να είναι μεταξύ 10.8–12.6V. Το ελάχιστο της V_A , είναι δυνατό να χαμηλώσει στα 8.5V, από τα 10.8V όταν συμμετέχουν η θερμοκρασία, η ανοχή αλλά και το φορτίο τροφοδοσίας για τη μετάδοση (συμπεριλαμβανομένου και ενός ρεύματος μετάδοσης της τάξης των 250mA). Το γεγονός αυτό προσφέρει επαρκή περιθώρια, όταν ο ενισχυτής μετάδοσης οδηγεί μια χαμηλής αντίστασης γραμμή και η τάση εξόδου είναι λίγο μικρότερη των 7V_{pp}. Όμως κάθε σχεδίαση τροφοδοσίας η οποία εκμεταλλεύεται το ελάχιστο των 8.5V, πρέπει να διατηρεί και το όριο των > 10.8V, υπό κανονικές συνθήκες γραμμής, συμπεριλαμβανομένων και των συνθηκών θερμοκρασίας του χώρου. Το στοιχείο αυτό επιτρέπει στον ενισχυτή μετάδοσης να οδηγήσει ένα 7V_{pp} σήμα, σε μια γραμμή με συνθήκες χαμηλότερου φορτίου.

Η επέκταση του μεγίστου της V_A , πέραν των 12.6V και μέχρι τα 18V, εξαρτάται από το αν πληρούνται οι θερμικές απαιτήσεις του ενισχυτή της μετάδοσης. Η θερμοκρασία του ενισχυτή εξαρτάται από το πόσο συχνά μπορεί μια συσκευή να μεταδίδει, από την θερμοκρασία του περιβάλλοντος του ενισχυτή και από την τάση τροφοδοσίας. Αν εκφράσουμε το πόσο συχνά μια συσκευή είναι σε θέση να μεταδίδει με βάση τους μέγιστους κύκλους μετάδοσης, τότε οι θερμικές απαιτήσεις του ενισχυτή πληρούνται ακολουθώντας την εξής φόρμουλα :

$$V_{ATXAVE} < (150 - T_{AMAX}) / (8 * D_{MAX})$$

όπου,

V_{ATXAVE} : Μέση τάση τροφοδοσίας V_A καθ' όλη τη διάρκεια της μετάδοσης.

T_{AMAX} : Μέγιστη θερμοκρασία στο εσωτερικό της συσκευής.

D_{MAX} : Μέγιστος αριθμός κύκλων μετάδοσης.

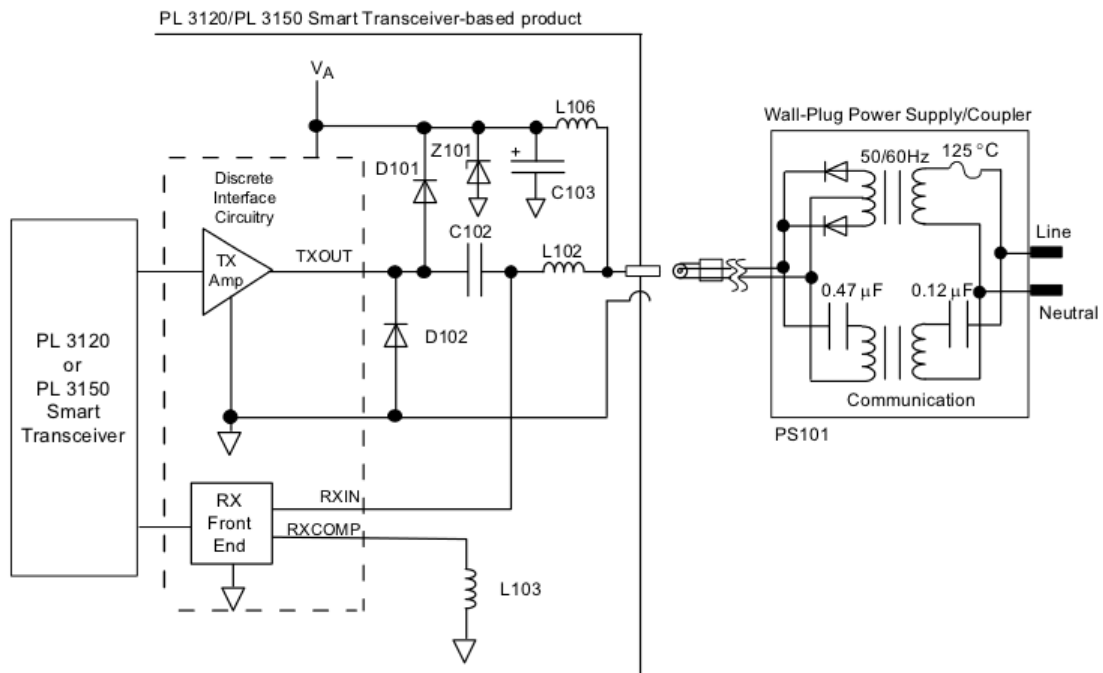
Το μέγιστο όσον αφορά τους κύκλους μετάδοσης είναι 64%, περίπου δηλαδή μια μετάδοση με περίοδο 92.2msec.

$T_{AMAX}(^{\circ}C)$	$D_{MAX}(\%)$	$V_{ATXAVE}(V)$
85	64	12.7
85	45	18
70	64	15.6
70	55	18

Πίνακας 7.2: Χαρακτηριστικές τιμές για την προσέγγιση της μέσης τάσης τροφοδοσίας

Στη δική μας εργασία χρησιμοποιήθηκε μια πολύ βολική και αποδοτική τροφοδοσία γραμμικού τύπου, η οποία συνδυάζει ένα wall-plug (50/60Hz) με ένα κύκλωμα ζεύξης line-to-neutral, το οποίο αναλύθηκε στην προηγούμενη ενότητα. Αυτή η γραμμικού τύπου τροφοδοσία δεν φορτώνει τη γραμμή ισχύος αλλά ούτε και παράγει σημαντικό θόρυβο, ενώ παρέχεται από την Tamura Corporation. Έχοντας ονομαστική τάση γραμμής AC, ο συγκεκριμένος τύπος τροφοδοσίας παρέχει τάση 10.8V–17V, ενώ στη χειρότερη περίπτωση ανοχής τάσης της γραμμής AC, 10%, η τάση τροφοδοσίας V_A είναι από 8.5V έως 18V. Και στις δύο περιπτώσεις το ρεύμα εξόδου είναι 360mADC και το επιτρεπόμενο εύρος θερμοκρασιών κυμαίνεται από 0°C έως 40°C. Ο 50/60Hz μετασχηματιστής τροφοδοσίας στο

εσωτερικό του τροφοδοτικού αυτού, παρέχει ισχύ στον PL Smart Transceiver, ενώ ο μετασχηματιστής επικοινωνίας δημιουργεί τη ζεύξη του σήματος επικοινωνίας με τη συσκευή (η οποία βασίζεται στον PL Smart Transceiver), μέσω ενός ζευγαριού καλωδίων χαμηλής τάσης. Στο εσωτερικό του PL Smart Transceiver τα σήματα ισχύος και επικοινωνίας διαχωρίζονται με σκοπό να επιτελέσουν τα καθήκοντά τους. Το πηνίο L106 χρησιμοποιείται για να εμποδίσει τη χαμηλή αντίσταση του πυκνωτή C103 να βραχυκυκλώσει το σήμα επικοινωνίας. Ο σκοπός της διόδου zener Z101 είναι να εμποδίσει την τάση τροφοδοσίας V_A από το να ξεπεράσει το όριο των 18V, σε συνθήκες χαμηλού φορτίου πάνω στην AC γραμμή. Το πηνίο L102 έχει προστεθεί για να εξασφαλίσει τη σταθερότητα του ενισχυτή μετάδοσης με το μετασχηματιστή επικοινωνίας, ο οποίος βρίσκεται έξω από το κύκλωμα του PL Smart Transceiver.



Σχήμα 7.14: Line-to-neutral κύκλωμα ζεύξης συνδεδεμένο με τροφοδοσία γραμμικού τύπου (wall-plug Tamura Corporation)

Το σχήμα 7.14 αποτυπώνει τη σύνδεση του κυκλώματος ζεύξης, το οποίο χρησιμοποιεί τη μέθοδο line-to-neutral, με το κύκλωμα της τροφοδοσίας (πως είναι κατά προσέγγιση το εσωτερικό του Wall-Plug Power Supply/Coupler).

Στο σχήμα που ακολουθεί παρουσιάζεται ένας πίνακας με τα όρια για τις τιμές των πηνίων, πυκνωτών και διόδων του wall-plug coupling circuit. Όλα τα σχήματα και οι

τιμές των στοιχείων έχουν βελτιστοποιηθεί όσον αφορά το ελάχιστο δυνατό κόστος και αποτυπώνουν τη μορφή του κυκλώματος ζεύξης και τροφοδοσίας στη γενική μορφή τους.

Όλη η ανάλυση η οποία προηγήθηκε σε αυτήν την ενότητα και αφορούσε τη μελέτη της τροφοδοσίας, αλλά και η ανάλυση του κυκλώματος ζεύξης της ενότητας 7.2.2 βασίστηκαν στους τρόπους με τους οποίους μπορεί ένας PL Smart Transceiver να συνδεθεί στο δίκτυο ηλεκτρικής ισχύος για την επίτευξη αποδοτικής επικοινωνίας. Τα σχήματα τα οποία παρουσιάστηκαν, η αξιοποίηση πυκνωτών, διόδων και πηνίων βοηθούν στην κατανόηση του τρόπου σύνδεσης των ιδιαίτερων αυτών πομποδεκτών στο δίκτυο του ηλεκτρικού ρεύματος.

Comp	Value	Required Specifications	Example Vendor/Part#
	C-band		C-band
C102	1.0 μ F	$\pm 10\%$, ≥ 50 VDC, metallized polyester	AVX/ BF074D0105K
C103	≥ 820 μ F	$\pm 20\%$, ≥ 25 VDC, aluminum electrolytic, $\leq 0.3\Omega$ ESR @100kHz	Nichicon/ UHE1E821MPD
D101	1A	Reverse breakdown ≥ 50 VDC, forward voltage ≤ 1.3 V @1A/25C, surge current ≥ 30 A for 8.3ms, reverse recovery ≤ 200 ns, reverse current $\leq 100\mu$ A @100C, typical capacitance ≤ 40 pF @4V	Vishay General Semi/ 1N4935
D102	1A	Reverse breakdown ≥ 50 VDC, forward voltage ≤ 1.0 V @1A/25C, surge current ≥ 30 A for 8.3ms, reverse recovery ≤ 25 ns, reverse current $\leq 100\mu$ A @100C, typical capacitance ≤ 40 pF @4V	Fairchild/ ES1B
Z101	17V	$\pm 5\%$, 5W, Zener	IN5354B
L102	Bead	$\leq 0.5\Omega$ @100kHz, $\geq 20\Omega$ @10MHz, $I_{max} \geq 2$ A	Steward/ HI1206P121R-00
L103	820 μ H	$\pm 10\%$, $I_{max} \geq 30$ mA, $R_{DC} \leq 55\Omega$, 1kHz \leq test frequency ≤ 400 kHz	ACT/ DD821K/ or CTC Coils Limited/ CH Series
L106	220 μ H	$\pm 10\%$, $I_{max} \geq 500$ mA, $R_{DC} \leq 1.0\Omega$	Taiyo Yuden/ LHL08TB221K/ or CTC Coils Limited/ CH Series
PS101	120VAC	120VAC $\pm 10\%$, 60Hz, 12VDC, 400mA	Tamura (1)/ 425A12400P
	230VAC	230VAC $\pm 10\%$, 50Hz, 12VDC, 400mA	Tamura (1)/ 425F12400P

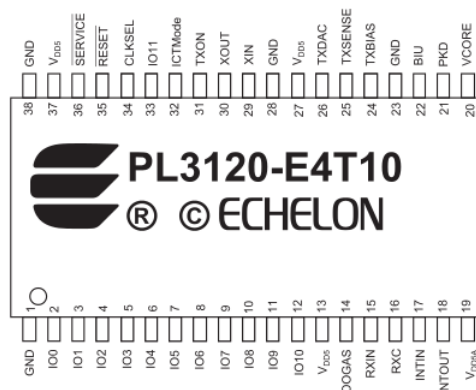
Σχήμα 7.15: Τιμές των στοιχείων του wall-plug coupling circuit

7.2.4 Κυκλωματικά διαγράμματα

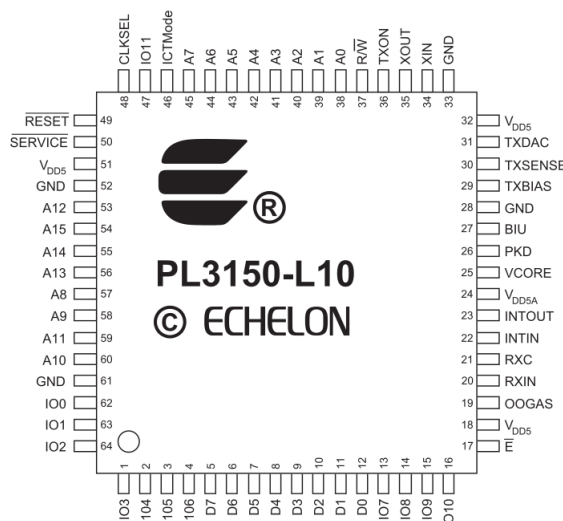
Στην προηγούμενη ενότητα αναλύθηκε ο γενικός τρόπος σχεδίασης της σύνδεσης των PL Smart Transceivers στο ηλεκτρικό δίκτυο. Στην ενότητα αυτή παρουσιάζονται τα

κυκλωματικά διαγράμματα διαφόρων σημείων της υλοποίησης των PL Smart Transceivers . Όλα αυτά τα διαγράμματα είναι μέρος του Power Line Smart Transceiver Development Support Kit (PL DSK version 2.0) της Echelon. Αυτό το support kit ήταν μέρος του υλικού που μας δόθηκε για την εργασία μας. Τα σχήματα που ακολουθούν δίνουν τον πραγματικό τρόπο σύνδεσης των στοιχείων των PL Smart Transceivers μεταξύ τους, καθώς και τις τιμές των διαφόρων πηνίων, πυκνωτών και αντιστάσεων.

Στα δύο επόμενα σχήματα αποτυπώνονται οι PL 3120/ PL 3150 Smart Transceivers μαζί με τα pins για τις εισόδους και εξόδους των I/O objects, αλλά και τα pins των διαφόρων σημάτων μετάδοσης και λήψης.

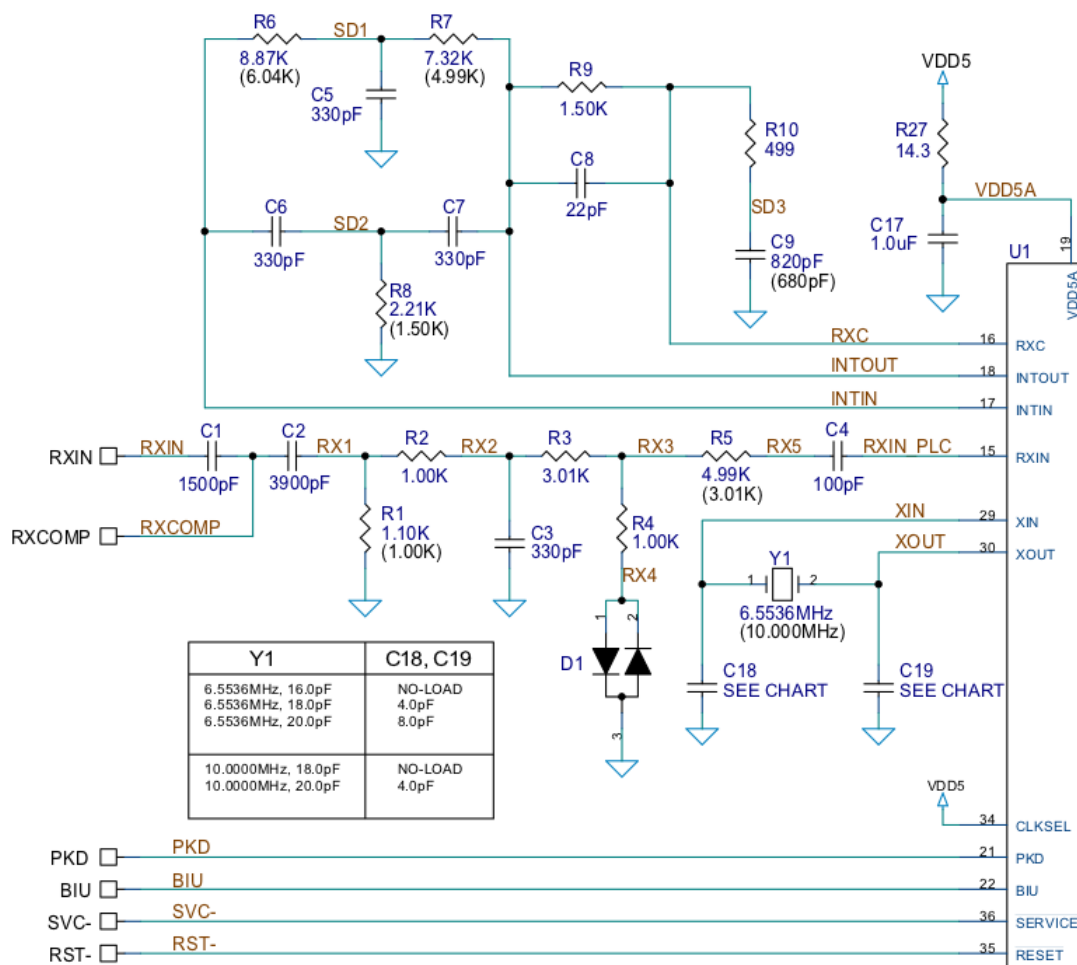


Σχήμα 7.16: PL 3120 Smart Transceiver



Σχήμα 7.17: PL 3150 Smart Transceiver

Στο σχήμα που ακολουθεί φαίνεται η υλοποίηση του εσωτερικού ρολογιού των 10MHz, μέσω των pins XIN, XOUT του PL Smart Transceiver, καθώς και το κυκλωματικό διάγραμμα του σήματος RXIN, το οποίο και αφορά την είσοδο των δεδομένων στον δέκτη αλλά και των σημάτων RXC, INTIN, INTOUT που αφορούν επίσης τη λήψη δεδομένων. Επίσης αποτυπώνονται τα σήματα BIU, PKD, Service, Reset καθώς και η τροφοδοσία V_{dd5} του πομποδέκτη.



Σχήμα 7.18: Εσωτερικό ρολόι 10MHz και σήματα δέκτη

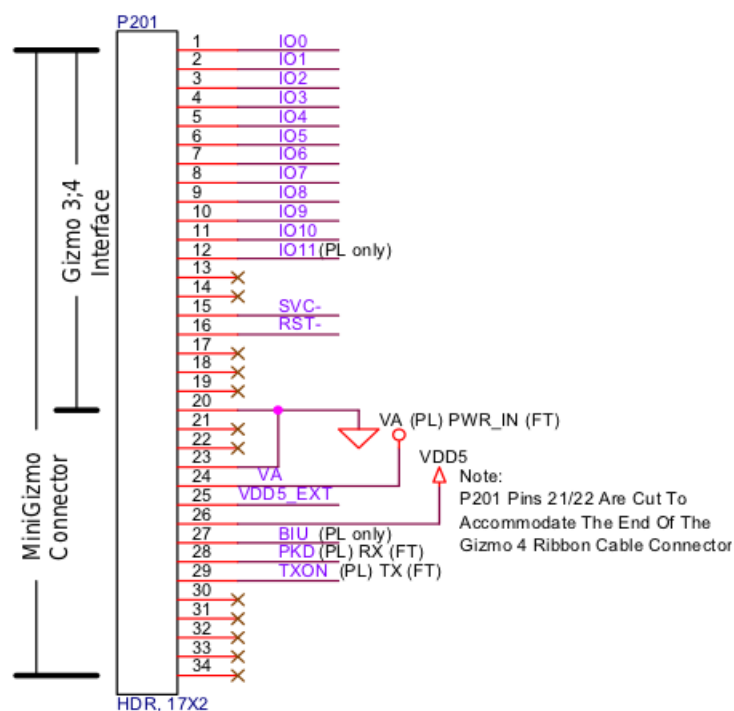
Στο σχήμα που ακολουθεί αποτυπώνεται το κομμάτι της μετάδοσης (transmission) του PL Smart Transceiver. Σε αυτό το κυκλωματικό διάγραμμα φαίνονται τα σήματα που χρησιμοποιούνται από τον PL Smart Transceiver για να υποδηλώσουν την έναρξη, τη ρύθμιση ή και τη διακοπή της μετάδοσης δεδομένων ή μηνυμάτων. Αποτυπώνονται τα σήματα TXON, TXDAC, VCORE, TXBIAS, TXSENSE όπως επίσης και η τάση τροφοδοσίας

αφενός για λόγους συντομίας και απλούστευσης και αφετέρου διότι τα βασικά κυκλώματα μαζί με τις τιμές των στοιχείων τους είναι τα ίδια και στα δύο boards. Το μόνο στοιχείο που αλλάζουν στο PL 3150 board είναι η θέση των pins πάνω στο chip και κάποια στοιχεία που αφορούν στην τάση τροφοδοσίας V_{dd5} , του PL 3150 Smart Transceiver.

7.3 Mini Gizmo I/O board

Έχοντας παρουσιάσει και αναλύσει όλο το μπλοκ διάγραμμα του σχήματος 7.1, εκτός βέβαια από την εφαρμογή η οποία και παρουσιάζεται στο επόμενο κεφάλαιο, προχωράμε στην παρουσίαση και των υπολοίπων στοιχείων του υλικού, που χρησιμοποιήθηκε στην εργασία μας. Το μπλοκ διάγραμμα του σχήματος 7.1 μας έδειξε ουσιαστικά από ποια κομμάτια λειτουργικότητας αποτελείται μια συσκευή LONWORKS τεχνολογίας. Αυτές οι συσκευές (κόμβοι) όμως, συνδέονται μέσω καλωδίων και με ένα test board, το οποίο και ονομάζεται Mini Gizmo I/O board.

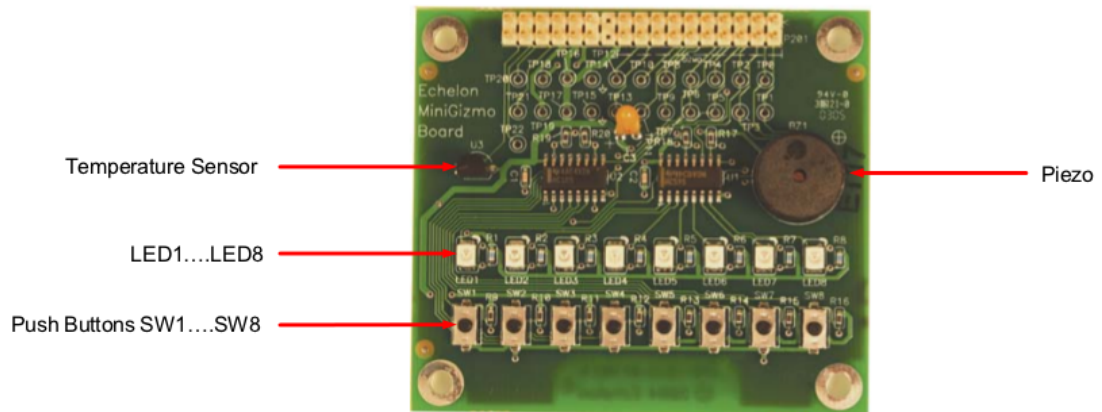
Το interface των καλωδίων σύνδεσης παρουσιάζεται στο επόμενο σχήμα.



Σχήμα 7.20: I/O connector

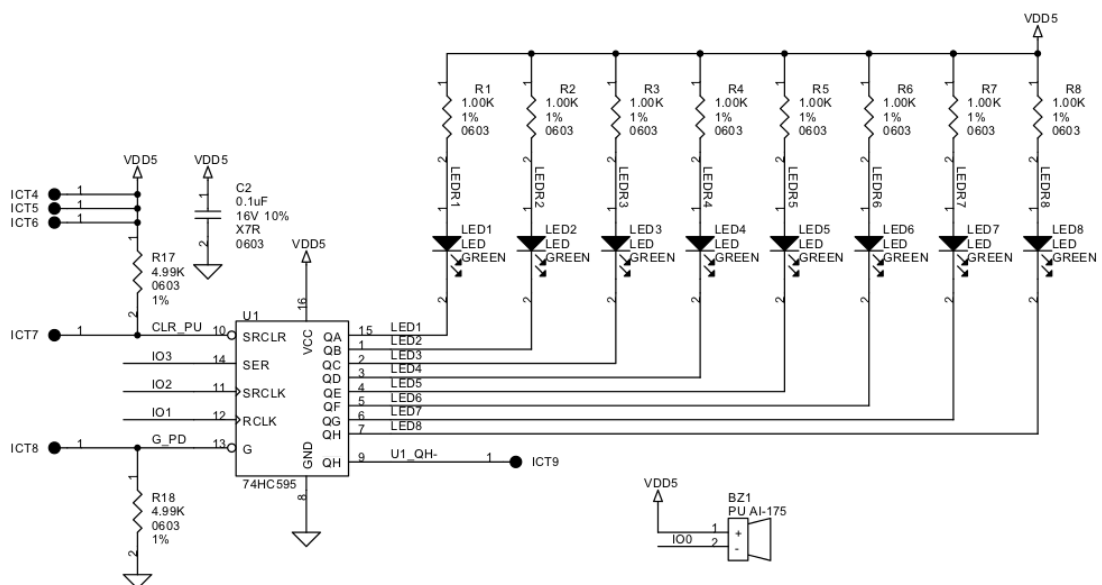
Κάθε πλακέτα Mini Gizmo περιλαμβάνει 8 push buttons, 8 LEDs, έναν αισθητήρα

θερμοκρασίας και ένα piezo buzzer για την παραγωγή ηχητικών σημάτων, στοιχεία τα οποία φαίνονται στο σχήμα 7.21.



Σχήμα 7.21: Mini Gizmo

Τα LEDs και το piezzo buzzer αποτυπώνονται κυκλωματικά στο παρακάτω σχήμα.



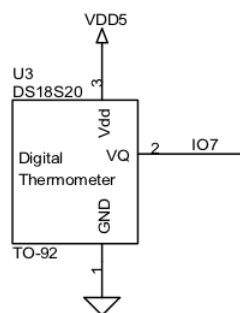
Σχήμα 7.22: Κυκλωματική αναπαράσταση LEDs και Piezzo Buzzer.

Τα LEDs όπως βλέπουμε συνδέονται σε ένα καταχωρητή ολίσθησης (shift register) παράλληλης εισόδου-σειριακής εξόδου, ο οποίος ονομάζεται 74HC595. Για την είσοδο των δεδομένων στο μανδαλωτή (latch) χρησιμοποιούμε το IO_1, με το ρολόι να δίνεται μέσω του pin IO_2.

Το piezzo buzzer, το οποίο χρησιμοποιείται για την παροχή ηχητικής ανάδρασης, συνδέεται με τον PL Smart Transceiver στο pin IO_0. Χρησιμοποιεί το μοντέλο εξόδου frequency και έχει μια συχνότητα συντονισμού κοντά στα 4000Hz, με αποτέλεσμα η έξοδος να ακούγεται καλύτερα κοντά σε αυτή την τιμή.

Τα push buttons SW1 – SW8, συνδέονται και αυτά σε ένα 8-bit καταχωρητή ολίσθησης παράλληλης εισόδου-σειριακής εξόδου (74HC165). Για να μπορούμε να γράφουμε στα buttons ή να διαβάζουμε από αυτά μέσω της εφαρμογής, χρησιμοποιούμε το pin IO_5 για τα δεδομένα και το pin IO_4 για το ρολόι, από τον PL Smart Transceiver.

Ο αισθητήρας θερμοκρασίας βασίζεται σε ένα ψηφιακό θερμόμετρο τεχνολογίας Dallas DS18S20. Χρησιμοποιεί το μοντέλο εισόδου/εξόδου touch και συνδέεται στον Smart Transceiver μέσω του pin IO_7.



Σχήμα 7.23: Αισθητήρας θερμοκρασίας Dallas DS18S20

7.4 Σύνδεση δικτύου ελέγχου με τον υπολογιστή

Για τη σύνδεση του υπολογιστή με το δίκτυο LONWORKS, χρησιμοποιήθηκε ένα χαμηλού κόστους και υψηλής απόδοσης U20 USB interface για κανάλια PL-20C/N. Το interface αυτό χρησιμοποιήθηκε στην εργασία μας κυρίως για λόγους εποπτείας και παρακολούθησης, των μηνυμάτων που στέλνονται από και προς το δίκτυο.

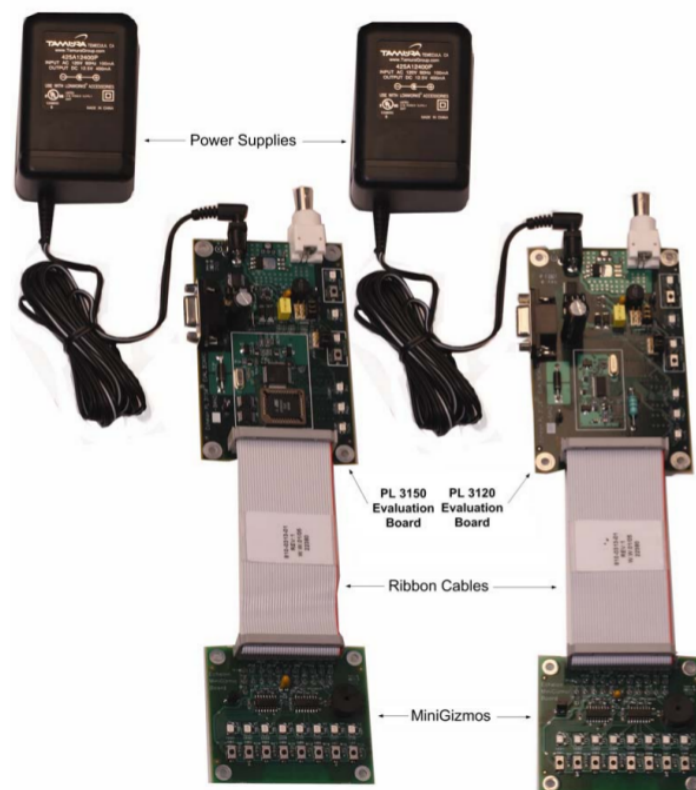
Πρέπει να τονιστεί ότι το USB interface συνδέεται και αυτό στο κανάλι PL-20 (ηλεκτρικό δίκτυο) μέσω τροφοδοσίας, αλλά μπορεί και αντλεί ισχύ περίπου 5V και από τον υπολογιστή μέσω USB (η ισχύς αυτή επαρκεί για την τροφοδοσία των περισσότερων εσωτερικών κυκλωμάτων του interface). Το U20 USB network interface διαθέτει τρία LEDs για τον προσδιορισμό της κατάστασης του δικτύου. Το SVC LED, το οποίο αναβοσβήνει κατά τη διάρκεια φόρτωσης του προγράμματος της εφαρμογής στον κόμβο LONWORKS,

το TX LED, το οποίο και ανάβει όταν ένα μήνυμα στέλνεται μέσω του USB interface στο δίκτυο και το RX LED το οποίο ανάβει όταν το USB interface λάβει ένα μήνυμα από το δίκτυο.



Σχήμα 7.24: Διεπαφή δικτύου U20 USB

Το ολοκληρωμένο σύστημα των PL 3120/PL 3150 Evaluation boards αποτυπώνεται στο σχήμα 7.25.



Σχήμα 7.25: Ολοκληρωμένη άποψη του υλικού

Οι πηγές [18],[15],[14],[9],[11],[13],[12],[10] δίνουν επιπλέον πληροφορίες για το υλικό.

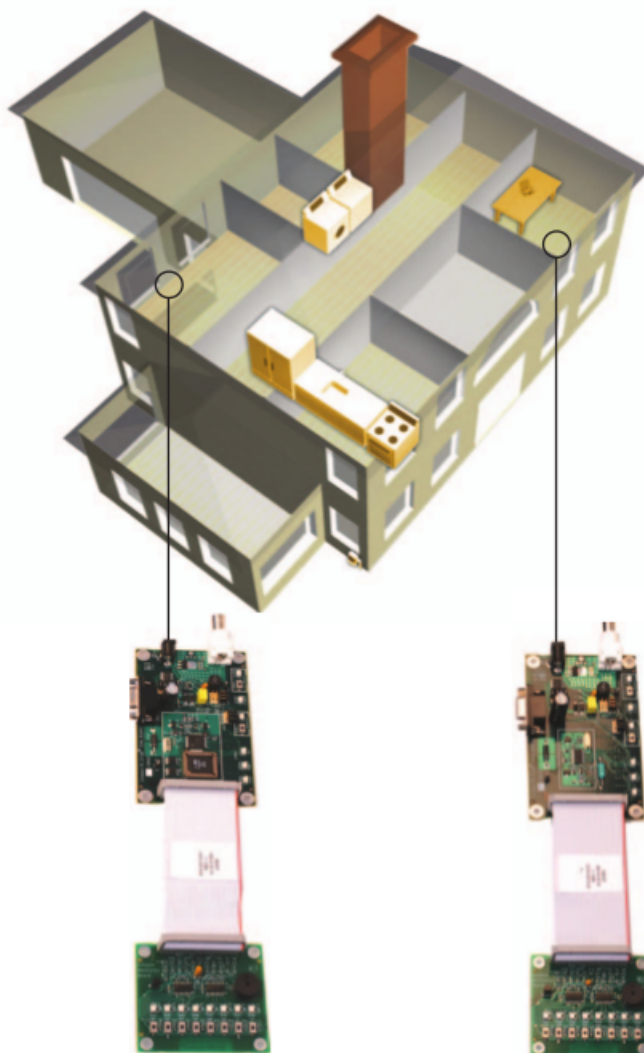
Κεφάλαιο 8

Ανάλυση της εφαρμογής

8.1 Εισαγωγή

Στο πρώτο μέρος της εργασίας και συγκεκριμένα στο πρώτο κεφάλαιο παρουσιάσαμε τα δίκτυα ελέγχου από μια γενική σκοπιά, ενώ στα δύο επόμενα κεφάλαια συνεχίσαμε με την παρουσίαση των δικτύων LONWORKS τεχνολογίας αλλά και την παρουσίαση της πλατφόρμας την οποία χρησιμοποιούν αυτά τα δίκτυα για τον έλεγχο διαφόρων συστημάτων. Στη συνέχεια του πρώτου μέρους της εργασίας παρουσιάστηκαν δύο πολύ σημαντικά πρωτόκολλα επικοινωνίας. Αρχικά, το πρωτόκολλο LonTalk το οποίο παρέχει το σύνολο των κανόνων επικοινωνίας αλλά και τη μορφοποίηση (format) των μηνυμάτων, τα οποία ανταλλάσσονται μεταξύ των συσκευών LONWORKS. Το πρωτόκολλο αυτό σχηματίζει μια ιεραρχία για τη λειτουργικότητα μιας συσκευής, ξεκινώντας από το φυσικό επίπεδο, στο οποίο επιλέγεται το κανάλι επικοινωνίας και φτάνοντας στο έβδομο επίπεδο (επίπεδο εφαρμογής), στο οποίο βρίσκεται ο κώδικας λειτουργίας της συσκευής. Οι συσκευές οι οποίες χρησιμοποιήθηκαν σε αυτή την εργασία χρησιμοποίησαν ως μέσο επικοινωνίας την γραμμή ηλεκτρικής ισχύος (PL-20). Η σηματοδότηση μέσω του καναλιού PL-20 παρουσιάστηκε αναλυτικά στο δεύτερο κεφάλαιο. Τα στοιχεία από τα οποία αποτελούνται αυτές οι συσκευές και αφορούν το υλικό τους (πομποδέκτες, Neuron επεξεργαστές, κυκλώματα ζεύξης, τροφοδοσία, Mini Gizmo, U20 USB interface) αναλύθηκαν διεξοδικά στο πρώτο κεφάλαιο του δεύτερου μέρους της εργασίας. Το δεύτερο και πιο σημαντικό πρωτόκολλο το οποίο και χρειάστηκε να υλοποιηθεί για τις ανάγκες αυτής της εργασίας είναι το πρωτόκολλο ISI. Το πρωτόκολλο αυτό υλοποιείται πάνω στο έβδομο επίπεδο του πρωτόκολλου LonTalk και είναι απαραίτητο για τη σύνδεση των εφαρμογών, οι οποίες τρέχουν πάνω στις

συσκευές, αλλά και κατ' επέκταση για την αξιόπιστη επικοινωνία τους μέσω των network variables. Κάθε συσκευή περιέχει κώδικα ISI ο οποίος και διαμορφώνει τη σύνδεση μέσω του μοντέλου σύνδεσης (μηνύματα ISI), το οποίο και αναλύθηκε στο κεφάλαιο 5. Αυτό το μοντέλο σύνδεσης χρησιμοποιείται για την δημιουργία δικτύων ελέγχου με συσκευές, οι οποίες αλληλεπιδρούν εύκολα αλλά και ενσωματώνονται στο δίκτυο χωρίς τη βοήθεια κάποιου εργαλείου δικτύου όπως το LonMaker.



Σχήμα 8.1: Τοποθέτηση των LONWORKS συσκευών σε ένα κτίριο

Το σχήμα 8.1 παρουσιάζει την τοποθέτηση δύο συσκευών, οι οποίες βασίζονται στους PL 3120/ PL 3150 Smart Transceivers, σε δύο τυχαία σημεία ενός κτιρίου με στόχο την

αξιόπιστη και γρήγορη επικοινωνία τους για την εκτέλεση μιας εφαρμογής.

Ο κύριος στόχος της εργασίας μας βασίζεται ακριβώς στο προηγούμενο σχήμα και είναι η ανάπτυξη ενός δικτύου ελέγχου μεταξύ των PL 3120 και PL 3150 evaluation boards, μέσω της καλωδίωσης του ηλεκτρικού ρεύματος ενός κτιρίου αλλά και μέσω των πρωτοκόλλων LonTalk και ISI. Γι' αυτό το λόγο αναπτύχθηκαν δύο εφαρμογές (Digital Temperature Alarm Sensor, Monitor), μία για το κάθε board.

8.2 Digital Temperature Alarm Sensor

Η πρώτη εφαρμογή (συσκευή) ονομάζεται Digital Temperature Alarm Sensor και δημιουργήθηκε για το PL 3150 Evaluation board. Η εφαρμογή αυτή υλοποιήθηκε με βάση τη γλώσσα προγραμματισμού Neuron C, τα βασικά στοιχεία της οποίας παρουσιάστηκαν στο κεφάλαιο 6.

Ο digital temperature alarm sensor δημιουργήθηκε έχοντας σαν στόχο να διατηρήσει τη θερμοκρασία ενός χώρου κοντά στους 25 °C. Για τις ανάγκες της εφαρμογής και ειδικότερα για τη σύνδεσή της με την εφαρμογή του PL 3120 Smart Transceiver (Monitor), χρησιμοποιήθηκε το functional profile SFPTswitch και η μεταβλητή δικτύου nvoSwitch.

```
network output SNVT_switch nvoSwitch ;
fblock SFPTswitch { nvoSwitch implements nvoSwitch ; }
fbSwitch external_name(" Switch" );
```

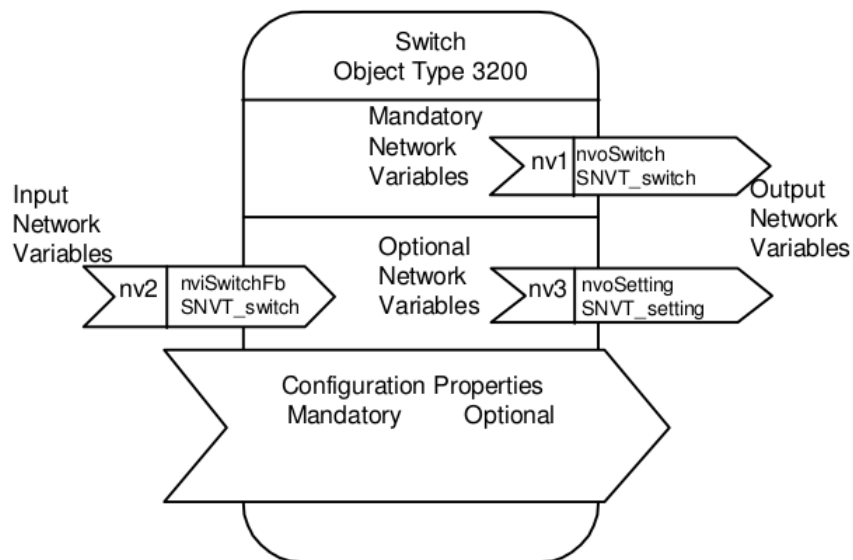
Η μεταβλητή αυτή δικτύου, nvoSwitch, υλοποιεί την μεταβλητή δικτύου του functional profile SFPTswitch, όπως φαίνεται στο κομμάτι του κώδικα που προηγήθηκε.¹ Το SFPTswitch όπως δημιουργήθηκε από την LonMark, αποτυπώνεται στο σχήμα της σελίδας 125.

8.2.1 Σύνδεση ISI

Στη συνέχεια ξεκινάει τη λειτουργία της η μηχανή ISI, για τη σύνδεση των δύο εφαρμογών μεταξύ τους, πρώτα από την πλευρά του digital temperature alarm sensor. Αρχικά προστέθηκε στον κώδικα της εφαρμογής η βιβλιοθήκη < isi.h >, η οποία και περιλαμβάνει βασικές συναρτήσεις για την ενεργοποίηση αλλά και την συντήρηση της μηχανής ISI. Στη

¹Ο ολοκληρωμένος κώδικας του Digital Temperature Alarm Sensor παρουσιάζεται στο τέλος αυτού του κεφαλαίου.

συνέχεια υλοποιήθηκε η διαδικασία reset, η οποία μαζί με τη χρήση της ιδιότητας SCPT-nwrkCnfg nciNetConfig, βοήθησαν στην εκκίνηση της μηχανής. Η συνάρτηση η οποία ουσιαστικά χρησιμοποιήθηκε για την εκκίνηση της μηχανής ISI είναι η IsiStartS(). Η



Σχήμα 8.2: Functional profile SFPTswitch

ιδιότητα nciNetConfig, όπως έχουμε πει χρησιμοποιείται από τα εργαλεία δικτύου για την μετάβαση από ένα αυτο-εγκαθιστώμενο σε ένα διαχειριζόμενο δίκτυο (Χρειάστηκε η προσθήκη της βιβλιοθήκης < snvt_cfg.h >).

```

device_properties { nciNetConfig };
eeprom SCPTnwrkCnfg oldNetConfig = CFG_NUL;
when(reset) {
    SCPTnwrkCnfg networkConfig; networkConfig = oldNetConfig;
    if (networkConfig == CFG_NUL){ ::nciNetConfig = CFG_LOCAL;
}

    oldNetConfig = ::nciNetConfig;
    if (::nciNetConfig == CFG_LOCAL){
        if (networkConfig == CFG_EXTERNAL){
            IsiReturnToFactoryDefaults ();}
        scaled_delay(31745UL);
    }
}

```

```
IsiStartS (isiFlagNone); } }
```

Στο προηγούμενο τμήμα κώδικα παρουσιάστηκαν τα βασικά σημεία της διαδικασίας reset². Η εφαρμογή υλοποιεί και την διαδικασία wink η οποία χρησιμοποιείται για την αναγνώριση της συσκευής αυτής στο δίκτυο, μέσω των LEDs και του Piezzo Buzzer.

Η μηχανή ISI πρέπει να καλείται περιοδικά, ούτως ώστε να εκπέμπει κατά τακτά χρονικά διαστήματα τη διεύθυνση της συσκευής (subnet ID, node ID), με σκοπό τη δημιουργία σύνδεσης με άλλες συσκευές. Για αυτό το λόγο χρησιμοποιήθηκε η συνάρτηση IsiTickS(), σε συνδυασμό με τη χρήση ενός timer, ο οποίος και καλεί την ρουτίνα αυτή κάθε 250ms (4 φορές το δευτερόλεπτο).

Για την επεξεργασία των ISI μηνυμάτων (CSMs, DRUM) που δέχεται η συσκευή μας, χρησιμοποιήθηκαν οι συναρτήσεις IsiApproveMsg() και IsiProcessMsgS().

```
mtimer repeating isiTimer = 250;
when(timer_expires(isiTimer))
{ IsiTickS(); }
when(msg_arrives) {
    if (IsiApproveMsg()) {
        if (IsiProcessMsgS()) { ; }}
    else { ; }}
```

Επιπλέον, η εφαρμογή αυτή, όσον αφορά το κομμάτι της σύνδεσής της με την εφαρμογή που τρέχει στον PL 3120 Smart Transceiver περιλαμβάνει και τις ακόλουθες, πολύ σημαντικές παραμέτρους:

- Κλήση της συνάρτησης IsiGetAssembly(), η οποία χρησιμοποιείται για την αναγνώριση του CSMA μηνύματος σύνδεσης. Χρησιμοποιούμε αυτό το μήνυμα διότι δημιουργούμε αυτόματη σύνδεση μεταξύ των δύο συσκευών κυρίως για λόγους εξάλειψης της πολυπλοκότητας. Το μήνυμα αυτό προέρχεται από την εφαρμογή Monitor, η οποία τρέχει στο PL 3150 Smart Transceiver και εκκινεί την σύνδεση. Όταν ο digital temperature alarm sensor δεχτεί αυτό το μήνυμα, προσπαθεί να αναγνωρίσει τις παραμέτρους της σύνδεσης αλλά και να καθορίσει ποια assemblies θα συμμετέχουν στη σύνδεση, επιστρέφοντας τον αριθμό του assembly. Κάνουμε έλεγχο της παραμέτρου Auto για να πετύχουμε την αυτόματη σύνδεση, αλλά και έλεγχο

²Η default τιμή της nciNetConfig είναι CFG_EXTERNAL.

των πεδίων profile, direction και width του CSMA. Αυτό που μας ενδιαφέρει ουσιαστικά είναι να αναγνωρίσει η συσκευή μας το functional profile το οποίο υλοποιείται στην εφαρμογή Monitor, αλλά και την κατεύθυνση της μεταβλητής δικτύου η οποία προσφέρεται για σύνδεση. Όπως έχουμε πει, η κατεύθυνση μιας μεταβλητής δικτύου πρέπει να είναι αντίθετη από αυτήν της μεταβλητής δικτύου αυτής της συσκευής για να γίνει η σύνδεση (και φυσικά να έχουν ίδιο τύπο). Η μεταβλητή δικτύου που προσφέρεται για σύνδεση είναι μεταβλητή εισόδου ίδιου τύπου (switch=95u) και διαφορετικής κατεύθυνσης με την nvoSwitch του digital temperature alarm sensor. Ο αριθμός του assembly (nvoSwitch) που μπορεί να συμμετέχει στη σύνδεση επιστρέφεται μέσω του global δείκτη της μεταβλητής δικτύου στη συνάρτηση αυτή αλλά και στη συνάρτηση IsiGetNvIndex(). Η συνάρτηση IsiGetNvIndex() ουσιαστικά βοηθά στον προσδιορισμό της ένωσης των μεταβλητών δικτύου, αφού επιστρέφει τον δείκτη της μεταβλητής που πρόκειται να συνδεθεί.

```

unsigned IsiGetAssembly(const IsiCsmoData *pCsmo, boolean
Auto){
    if (Auto && pCsmo->Profile == 4 && pCsmo->Direction ==
isiDirectionInput && pCsmo->Width== 1) {
        return nvoSwitch::global_index;
    }
    return ISI_NO_INDEX;
}
unsigned IsiGetNvIndex(unsigned Assembly, unsigned Offset){
    return nvoSwitch::global_index;
#pragma ignore_notused Offset
#pragma ignore_notused Assembly
}

```

- Κλήση της συνάρτησης IsiGetNextAssembly, η οποία και υποδηλώνει ότι δεν υπάρχει κανένα άλλο assembly, το οποίο να μπορεί να συνδεθεί με την εφαρμογή monitor. Αυτό γίνεται μέσω της τιμής ISI_NO_ASSEMBLY, η οποία ορίζεται ως η τιμή επιστροφής της συνάρτησης.
- Διαμόρφωση του πίνακα σύνδεσης στην εσωτερική μνήμη (eeprom) του Neuron chip.

Ο πίνακας αυτός περιέχει πληροφορίες για τις συνδέσεις στις οποίες έχει λάβει μέρος η συσκευή. Σε αυτή τη συσκευή ο πίνακας σύνδεσης έχει μόνο μία καταχώρηση για τη σύνδεση του digital temperature alarm sensor με το monitor.

```
eeprom IsiConnection MyConnectionTable;
unsigned IsiGetConnectionTableSize(void) {
    return 1u; }
const IsiConnection* IsiGetConnection(unsigned Index){
    return &MyConnectionTable;
    #pragma ignore_notused Index }
void IsiSetConnection(IsiConnection* pConnection ,
unsigned Index){
    MyConnectionTable = *pConnection;
    #pragma ignore_notused Index }
```

Με τη ρύθμιση αυτών των παραμέτρων, η συσκευή του digital temperature alarm sensor, αποδέχεται το μήνυμα σύνδεσης CSMA από την εφαρμογή monitor του PL 3150 Smart Transceiver, με αποτέλεσμα την επιτυχή πραγματοποίηση της σύνδεσης. Η σύνδεση από την πλευρά του monitor παρουσιάζεται στην υποενότητα που αφορά την εφαρμογή monitor.

8.2.2 Λειτουργία

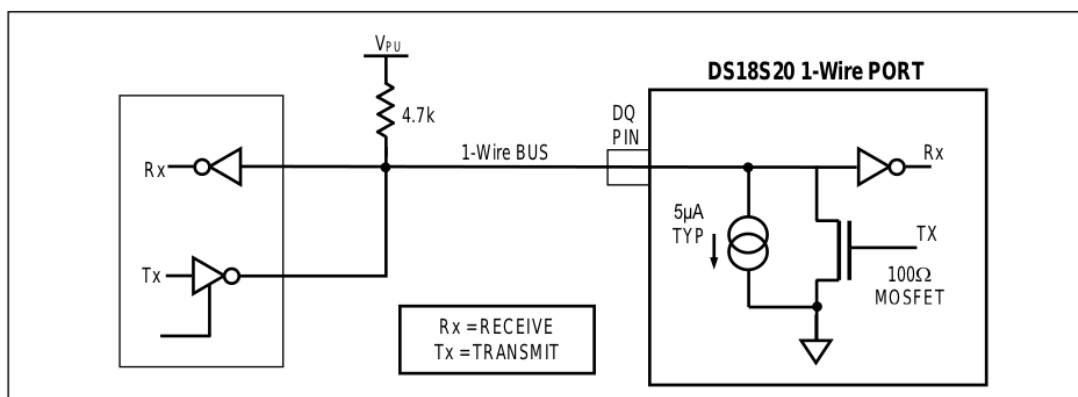
Μετά την υλοποίηση του πρωτοκόλλου ISI με τη βοήθεια της μηχανής ISI και την πραγματοποίηση ουσιαστικά της σύνδεσης από την πλευρά αυτής της συσκευής, η εφαρμογή τρέχει έναν ελεγκτή. Ο ελεγκτής είναι μια συνάρτηση η οποία εκτελεί το κυρίως μέρος της εφαρμογής. Στόχος του όπως είπαμε είναι η διατήρηση της θερμοκρασίας του χώρου γύρω από τους 25 °C, προσομοιώνοντας στην ουσία τη λειτουργία ενός θερμοστάτη.

Ορίζουμε μια μεταβλητή getTemp, τύπου SNVT_temp_p (με ακρίβεια 0.01), η οποία χρησιμοποιείται για την λήψη δειγμάτων της θερμοκρασίας του χώρου.

8.2.2.1 Λήψη δειγμάτων θερμοκρασίας

Η λήψη των δειγμάτων θερμοκρασίας πραγματοποιείται μέσω της συνάρτησης GetTemperature().

Όπως αναφέραμε στο κεφάλαιο της ανάλυσης του υλικού, το Mini Gizmo board περιλαμβάνει ένα Dallas DS18S20 αισθητήρα θερμοκρασίας. Ο αισθητήρας αυτός είναι συνδεδεμένος στον PL 3150 Smart Transceiver μέσω του μοντέλου εισόδου/εξόδου touch στο pin IO_7. Η συνάρτηση GetTemperature χρησιμοποιεί το πρωτόκολλο 1-wire για να εξασφαλίσει δύο τιμές από τον αισθητήρα. Ο αισθητήρας έχει μια 9-bit έξοδο και επικοινωνεί με τον πομποδέκτη μέσω ενός διαύλου, που απορρέει από το πρωτόκολλο 1-wire και ονομάζεται 1-wire bus. Ο δίαυλος αυτός χρησιμοποιείται τόσο για την σηματοδότηση όσο και για την τροφοδοσία του αισθητήρα. Επιπλέον, ο αισθητήρας έχει ακρίβεια 0.5 σε ένα εύρος θερμοκρασιών $-10^{\circ}C - +85^{\circ}C$. Τη στιγμή που τροφοδοτείται, παραμένει σε μια αδρανή κατάσταση χαμηλής ενέργειας ενώ για να αρχίσει η μέτρηση θερμοκρασιών αλλά και η μετατροπή από την αναλογική στην ψηφιακή μέτρηση, ο PL Smart Transceiver πρέπει να δώσει μια εντολή convert. Μετά την μετατροπή τα δεδομένα θερμοκρασίας αποθηκεύονται σε ένα 2-byte καταχωρητή θερμοκρασίας και ο αισθητήρας επιστρέφει στην αδρανή του κατάσταση. Ο δίαυλος επικοινωνίας 1-wire απαιτεί και την ενεργοποίηση ειδικών αντιστάσεων (pullup resistors) στο pin IO_7 περίπου 5kΩ. Η ενεργοποίηση αυτή πραγματοποιείται με μια οδηγία προς τον compiler, #pragma enable_io_pullups.

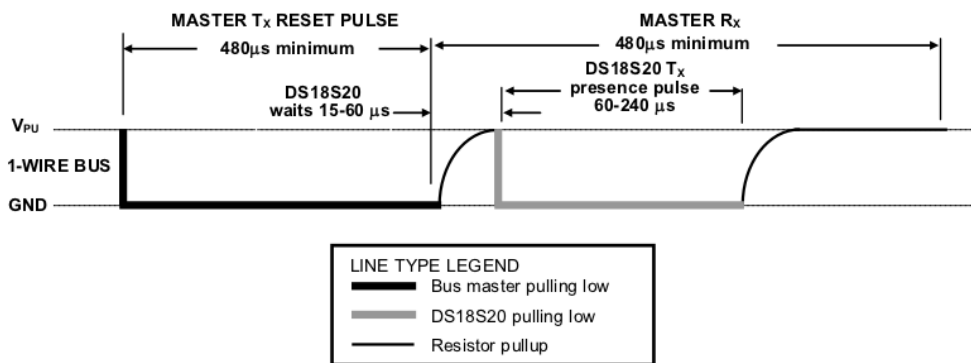


Σχήμα 8.3: Διάγραμμα σύνδεσης αισθητήρα με πομποδέκτη

Συνολικά η προσπέλαση του αισθητήρα DS18S20 περιλαμβάνει τρία βήματα, τα οποία είναι βασικά σημεία του πρωτοκόλλου 1-wire :

- Αρχικοποίηση. Το βήμα αυτό περιλαμβάνει ένα παλμό reset ο οποίος παράγεται από τον PL Smart Transceiver, ακολουθούμενος από ένα presence παλμό ο οποίος μεταδίδεται από τον αισθητήρα DS18S20 ως απάντηση στον παλμό reset. Ο παλμός presence επιτρέπει στον πομποδέκτη να γνωρίζει ποιοι αισθητήρες είναι πάνω στο

δίαυλο και είναι έτοιμοι να λειτουργήσουν. Για την δημιουργία του παλμού reset χρησιμοποιήθηκε η συνάρτηση touch_reset() του interface touch, η οποία παίρνει ως όρισμα το αντικείμενο IO_7 touch ioThermometer το οποίο έχουμε ορίσει και επιστρέφει 0 όταν ανιχνευθεί κάποιος παλμός presence από τον αισθητήρα (δεν επιστρέφει αν δεν ανιχνευθεί παλμός). Ο PL Smart Transceiver εκπέμπει ένα παλμό reset, βάζοντας τον δίαυλο σε κατάσταση χαμηλής ισχύος για περίπου 480μsec. Στη συνέχεια ώντας σε receive mode, ελευθερώνει το δίαυλο και ενεργοποιούνται οι pullup resistors. Ο αισθητήρας περιμένει από 15μsec έως 60μsec τοποθετώντας ξανά τον δίαυλο σε κατάσταση χαμηλής ισχύος για χρόνο 60μsec έως 240μsec και μεταδίδει τον παλμό presence.

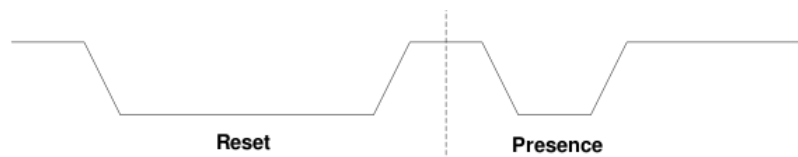


Σχήμα 8.4: Χρονικό διάγραμμα μετάδοσης παλμών

- Εντολή ROM. Αφότου ο πομποδέκτης αναγνωρίσει ότι υπάρχει κάποιος αισθητήρας στο δίαυλο, δίνει μια εντολή ROM (συγκεκριμένα SKIP_ROM). Αυτή η εντολή δίνεται μέσω της συνάρτησης touch_byte, η οποία και αποδίδει στον αισθητήρα μια διεύθυνση την οποία χρησιμοποιεί για την επικοινωνία του με τον πομποδέκτη.
- Λειτουργικές εντολές. Αφότου χρησιμοποιηθεί η εντολή SKIP_ROM για την διευθυνσιοδότηση του αισθητήρα, ο πομποδέκτης δίνει λειτουργικές εντολές ανάγνωσης, εγγραφής και μετατροπής (convert). Αρχικά διαβάζει ακολουθιακά, μέσω της εντολής READ 0xBEu, τα περιεχόμενα της μνήμης του αισθητήρα ξεκινώντας από το λιγότερο σημαντικό bit του byte 0 και συνεχίζοντας μέχρι και το 9ο byte και τα αποθηκεύει, όπως είπαμε σε ένα καταχωρητή 2-byte (μέσω της συνάρτησης touch_byte). Ο πομποδέκτης αμέσως μετά την εντολή ανάγνωσης δημιουργεί slots ανάγνωσης των δεδομένων (με διάρκεια 60μsec), κατά τη διάρκεια των οποίων, μπορεί ο αισθητήρας

να παράσχει τα δεδομένα του. Ο πομποδέκτης μπορεί οποιαδήποτε στιγμή να διακόψει την ανάγνωση από τη μνήμη, μέσω της διαδικασίας reset. Επειδή αυτές οι τιμές που διαβάζει ο πομποδέκτης έχουν ακρίβεια 0.5, η συνάρτηση τις μετατρέπει κατάλληλα, ούτως ώστε να πληρούν την ακρίβεια των 0.01 του SNVT_temp_p, του τύπου των μεταβλητών δικτύου, ο οποίος χρησιμοποιείται για την επιστροφή των δειγμάτων θερμοκρασίας. Για την ανάγνωση των δεδομένων το δεύτερο όρισμα της συνάρτησης touch_byte πρέπει να είναι πάντα 0xFF. Μετά την ανάγνωση των δεδομένων ακολουθεί η A-D μετατροπή, ενώ στη συνέχεια ο αισθητήρας επιστρέφει στην αδρανή κατάσταση χαμηλής ισχύος.

Αν κάποιο από αυτά τα βήματα δεν ακολουθηθεί, τότε ο αισθητήρας δεν ανταποκρίνεται. Ουσιαστικά το μοντέλο που χρησιμοποιείται για την επικοινωνία, είναι ένα μοντέλο master-slave στο οποίο master είναι ο πομποδέκτης και slave ο αισθητήρας Dallas DS19S20, ο οποίος λαμβάνει τις εντολές και τους παλμούς. Η διάρκεια των παλμών reset και presence είναι διαφορετική.



Σχήμα 8.5: Παλμοί reset και presence

Η συνάρτηση GetTemperature() αποτυπώνεται στο παρακάτω κομμάτι κώδικα.

```
IO_7 touch ioThermometer;
SNVT_temp_p GetTemperature(void){
    union { SNVT_temp_p snvtTempP;
            unsigned Bytes[2];} CurrentTemperature;
    CurrentTemperature.snvtTempP = 327671;
    if (touch_reset(ioThermometer)) {
        (void) touch_byte(ioThermometer, DS18S20_SKIP_ROM);
        (void) touch_byte(ioThermometer, DS18S20_READ);
        CurrentTemperature.Bytes[1]=
            touch_byte(ioThermometer, 0xFFu);
        CurrentTemperature.Bytes[0]=
            touch_byte(ioThermometer, 0xFFu);
```

```

if ( touch_reset (ioThermometer)) {
    CurrentTemperature.snvtTempP *= 501;
    (void) touch_byte (ioThermometer , DS18S20_SKIP_ROM);
    (void) touch_byte (ioThermometer , DS18S20_CONVERT);
} else {
    CurrentTemperature.snvtTempP = 327671; }
}
return CurrentTemperature.snvtTempP; }

```

Με αυτό τον τρόπο παίρνουμε δείγματα της θερμοκρασίας του χώρου από τον αισθητήρα. Η επόμενη υποενότητα αναφέρεται στην διαχείριση των υπολοίπων στοιχείων του Mini Gizmo board τα οποία χρησιμοποιήθηκαν από τον ελεγκτή για την ανάπτυξη της εφαρμογής του temperature alarm sensor.

Ο αναγνώστης μπορεί να ανατρέξει στην πηγή [1] για επιπλέον πληροφορίες.

8.2.2.2 Διαχείριση του Piezo Buzzer και των LEDs

Η εφαρμογή χειρίζεται το Piezzo Buzzer του Mini Gizmo, μέσω του μοντέλου εισόδου-εξόδου frequency. Η συσκευή αυτή ηχητικής ανάδρασης (Piezzo Buzzer) συνδέεται με τον κώδικά μας, μέσω ενός αντικειμένου I/O της εξής μορφής :

```
IO_0 output frequency clock(3) invert ioPiezzo = 1;
```

Το μοντέλο frequency χρησιμοποιείται για την παροχή ηχητικής ανάδρασης, παράγοντας ένα συνεχόμενο τετραγωνικό σήμα. Το Piezzo Buzzer, όπως έχουμε δει, συνδέεται με τον PL Smart Transceiver στο pin IO_0. Ουσιαστικά γίνεται μια ρύθμιση του timer/counter 1, τον οποίο είχαμε αναφέρει στο προηγούμενο κεφάλαιο, ώστε να είναι δυνατή η παραγωγή του τετραγωνικού σήματος. Το σήμα που παράγεται έχει περίοδο, η οποία υπολογίζεται από τον τύπο :

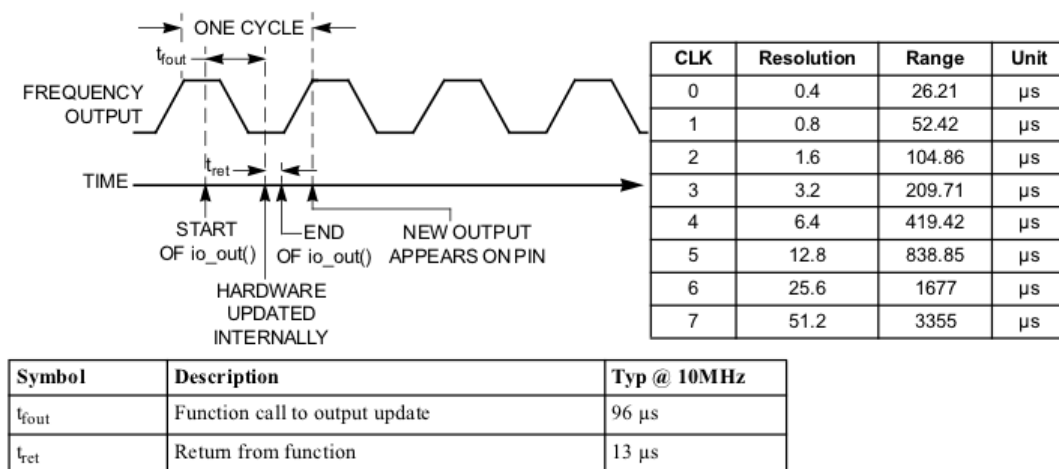
$$period(ns) = \frac{output_value * 4000 * 2^{(clock)}}{input_clock}$$

Για να μπορέσουμε να παράγουμε ένα τέτοιο ηχητικό σήμα χρησιμοποιήσαμε την παρακάτω συνάρτηση :

```
io_out (ioPiezzo ,100);
```

Η συνάρτηση αυτή γράφει στο αντικείμενο ioPiezzo την τιμή 100, παράγοντας ένα σήμα της τάξης των 3.125kHz, με την περίοδο του σήματος, έχοντας output=100, clock=3 και input clock=10MHz να είναι 0.32sec. Για να χρησιμοποιήσουμε την συνάρτηση io_out() χρειάστηκε η σύνδεση του προγράμματός μας με την βιβλιοθήκη < io.types.h >.

Το σχήμα που ακολουθεί παρουσιάζει ένα χρονικό διάγραμμα της συνάρτησης io_out (όσον αφορά την τιμή εξόδου της), καθώς και τις τιμές ακρίβειας όλων των διαθέσιμων τύπων ρολογιού (Εμείς χρησιμοποιούμε το ρολόι 3).



Σχήμα 8.6: Διάγραμμα της συνάρτησης io_out για το μοντέλο εισόδου-εξόδου frequency

Ο digital temperature alarm sensor διαχειρίζεται τα LEDs του Mini Gizmo I/O board μέσω της συνάρτησης SetLEDs(). Η συνάρτηση αυτή ορίζει ένα αντικείμενο εισόδου-εξόδου ioLeds, χρησιμοποιώντας το μοντέλο bitshift (μεταφορά δεδομένων σε shift registers) στο pin IO_2 του PL Smart Transceiver. Χρησιμοποιεί επίσης και το μοντέλο εισόδου-εξόδου bit για την δημιουργία ενός I/O αντικειμένου, ioLEDLd, το οποίο χρησιμοποιείται για τον “καθαρισμό” των D Flip-Flops του καταχωρητή.

```

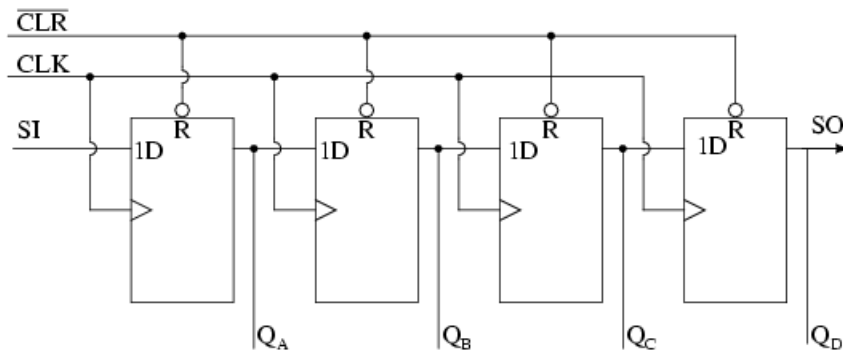
IO_1 output bit ioLEDLd = 1;
IO_2 output bitshift numbits(8) ioLEDs;
void SetLEDs(unsigned LEDs, unsigned Mask) {
    LEDs |= PreviousLEDs & ~Mask;
    PreviousLEDs = LEDs;
    io_out(ioLEDs, ~LEDs);
    io_out(ioLEDLd, 0);
}

```

```
io_out(ioLEDLd, 1);
```

```
}
```

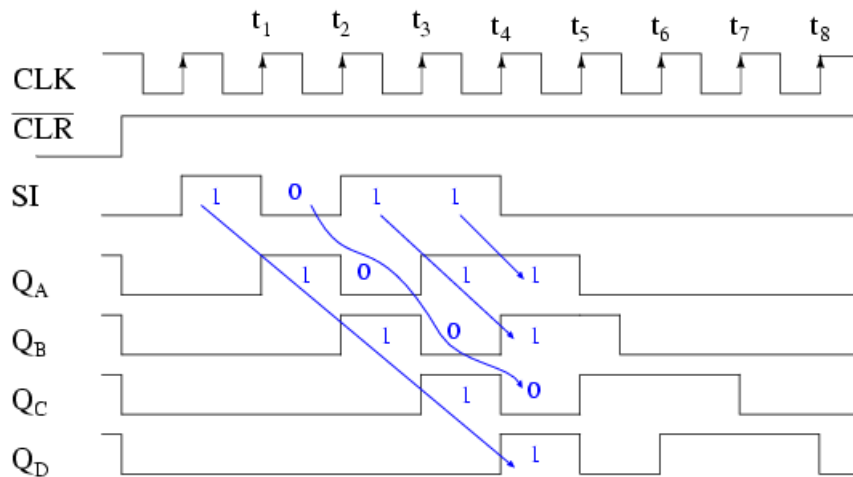
Όπως έχουμε δει, η πλακέτα Mini Gizmo περιλαμβάνει 8 LEDs, τα οποία είναι συνδεδεμένα στον PL Smart Transceiver μέσω ενός καταχωρητή ολίσθησης (serial-in/parallel-out). Ο καταχωρητής αυτός διαθέτει 8 bits όσα και τα LEDs της πλακέτας. Για να μπορέσει να ελέγξει ο PL Smart Transceiver ένα οποιοδήποτε LED, η συνάρτηση πρέπει να καθορίσει και την κατάσταση των υπολοίπων LEDs, μέσω του καταχωρητή. Το σχήμα που ακολουθεί αποτυπώνει έναν απλό καταχωρητή ολίσθησης τεσσάρων επιπέδων.



Σχήμα 8.7: Καταχωρητής ολίσθησης με 4 επίπεδα

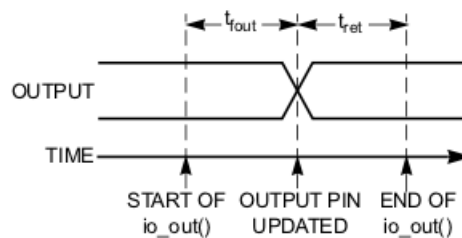
Γίνεται σαφές από το σχήμα ότι στον καταχωρητή αυτό έχουμε μία είσοδο και n-bits εξόδους. Στη δική μας εφαρμογή έχουμε 8 εξόδους, μία για κάθε LED. Η είσοδος είναι μια ενιαία σειρά από bits η οποία μπαίνει στον καταχωρητή όπως δείχνει η διαδικασία ολίσθησης του σχήματος 8.8 της επόμενης σελίδας. (Είσοδος =1011).

Οι τιμές 0 και 1 είναι οι δύο διακριτές τιμές, τις οποίες μπορεί να πάρει ένα LED (0=OFF, 1=ON). Στην ουσία δηλαδή με την τιμή 1011 για παράδειγμα, θέλουμε να θέσουμε όλα τα LEDs αναμμένα και μόνο το τρίτο LED να κρατηθεί κλειστό. Παρατηρούμε ότι, όταν η τιμή 1011 μπει στον καταχωρητή βγαίνει ανεστραμμένη (δεν κρατείται κλειστό το LED που θέλουμε). Γι' αυτό το λόγο για να μπορέσουμε να ανάβουμε το σωστό LED, αντιστρέφουμε την τιμή της εισόδου πριν αυτή μπει για επεξεργασία στον καταχωρητή ολίσθησης. Η επιθυμητή τιμή αποδίδεται ακολουθιακά, 1 bit κάθε κύκλο ρολογιού, σε κάθε LED. Δηλαδή μετά από 8 κύκλους ρολογιού, τα 8 bits του ορίσματος LEDs θα αποδοθούν στα LEDs. Η απόδοση της τιμής εισόδου στα LEDs γίνεται μέσω του αντικειμένου ioLEDs, αλλά και μέσω της συνάρτησης io_out η οποία και μεταδίδει (γράφει) την τιμή



Σχήμα 8.8: Διαδικασία ολίσθησης των bits

αυτή στα LEDs, μέσω του μοντέλου bitshift. Το μοντέλο bit καθαρίζει τα D Flip-Flops, για να είναι σε θέση να δεχτούν κάποια άλλη τιμή αργότερα. Η διάρκεια της συνάρτησης `io_out()`, παίρνοντας ως όρισμα το `ioLEDLd`, φαίνεται στο επόμενο σχήμα.



Symbol	Description	Typ @ 10MHz
$t_{f_{out}}$	Function call to update IO3 – IO5, IO11 All others	69 μ s 60 μ s
t_{ret}	Return from function IO0 – IO11	5 μ s

Σχήμα 8.9: Χρόνος λειτουργίας της `io_out()` με το μοντέλο bit

Χρησιμοποιούμε επίσης και μια τεχνική για τον καλύτερο και ασφαλέστερο προσδιορισμό της κατάστασης των LEDs, η οποία ονομάζεται *Masking*.

```
LEDs |= PreviousLEDs & ~Mask;
```

Η διαδικασία αυτή λαμβάνει χώρα προτού μπει η ανεστραμμένη σειρά των bits στον καταχωρητή ολίσθησης. Η τεχνική αυτή χρησιμοποιεί τις λειτουργίες bitwise AND (&) και bitwise OR (|), οι οποίες συγκρίνουν όλα τα ζευγάρια των bits, δύο δεκαεξαδικών αριθμών. Χρησιμοποιείται μια μάσκα, ένα σταθερό μοτίβο από bits, τα οποία βοηθούν στην εξάλειψη ή στη διατήρηση ενός άλλου μοτίβου από bits. Η τεχνική αυτή :

- Απομονώνει τα bits εκείνα τα οποία χρειάζεται η εφαρμογή, καθορίζοντας επακριβώς την επιθυμητή κατάσταση των LEDs.
- Βοηθά στην ελαχιστοποίηση των λαθών.
- Απαιτεί λίγους πόρους για τη λειτουργία της.

8.2.2.3 Ελεγκτής

Ο ελεγκτής, όπως αναφέρθηκε, είναι μια συνάρτηση η οποία καθορίζει την λειτουργία της εφαρμογής μετά τη σύνδεσή της μέσω ISI, με την εφαρμογή Monitor. Ουσιαστικά δημιουργήθηκε για να ελέγξει το κατά πόσο ο κύριος στόχος της εργασίας, η αυτόματη σύνδεση των δύο εφαρμογών μέσω ISI, λειτουργεί ικανοποιητικά.

Παίρνοντας δείγματα θερμοκρασίας μέσω της συνάρτησης GetTemperature() κάθε 10 δευτερόλεπτα, μέσω του ορίσμου ενός timer και σε συνδυασμό με το predefined event timer_expires, ο ελεγκτής προσπαθεί να προσωμοιάσει τη λειτουργία ενός θερμοστάτη.

```
stimer repeating led_timer = 10;
when( timer_expires( led_timer ) )
{
    Controller ();
}
```

Ο στόχος αυτού του εικονικού θερμοστάτη είναι η διατήρηση της θερμοκρασίας γύρω από τους $25^{\circ}C$. Γι' αυτό το λόγο γύρω από αυτή τη θερμοκρασία έχει δημιουργηθεί μια προστατευτική ζώνη 2 βαθμών κελσίου, ένα guardband. Όταν το δείγμα θερμοκρασίας από τον αισθητήρα βρεθεί μέσα σε αυτό το guardband ($23^{\circ}C - 27^{\circ}C$), ο ελεγκτής δεν προβαίνει σε καμία ενέργεια, κρατώντας κλειστό τον θερμοστάτη. Η κατάσταση στην οποία ο θερμοστάτης βρίσκεται κλειστός, αποτυπώνεται μέσω της τιμής HVAC-OFF της μεταβλητής δικτύου SNVT_hvac_status Status. Για τη χρήση της μεταβλητής αυτής χρησιμοποιήθηκε η βιβλιοθήκη < snvt_hv.h >. Όλα τα LEDs στο board PL 3120 παραμένουν κλειστά

και το Piezzo buzzer δεν παράγει κανένα ηχητικό σήμα. Ουσιαστικά ο θερμοστάτης δεν είναι κλειστός, αφού συνεχίζει να παίρνει δείγματα θερμοκρασίας κάθε 10 δευτερόλεπτα. Με την έννοια κλειστός, θεωρούμε την κατάσταση εκείνη, κατά την οποία ο θερμοστάτης δεν βρίσκεται σε HEAT mode αλλά ούτε και σε COOL mode. Η τιμή της μεταβλητής δικτύου, η οποία χρησιμοποιείται για την σύνδεση και είναι η primary network variable της εφαρμογής (nvoSwitch), παίρνει σε αυτήν την περίπτωση την τιμή -1. Η τιμή αυτή στέλνεται μέσω του δικτύου στην εφαρμογή monitor.³ Η μεταβλητή nvoSwitch είναι μια δομή με τα εξής πεδία :

```
typedef struct {  
    unsigned value;  
    signed state;  
} SNVT_switch;
```

```
nvoSwitch.state = -1;  
SetLEDs(0x00, 0xFFu);  
Status.mode=HVAC_OFF;
```

Όταν η θερμοκρασία βρεθεί πάνω από τους $27^{\circ}C$ (επιθυμητή τιμή + guardband), η εφαρμογή θέτει τον θερμοστάτη σε COOL mode. Σε αυτή την περίπτωση ανάβει το LED 2 στο PL 3120 Evaluation board, για να υποδείξει ότι ο θερμοστάτης έχει αρχίσει να ψύχει το χώρο, ώστε να φέρει τη θερμοκρασία του κοντά στο όριο των 27 βαθμών κελσίου. Παράλληλα παράγεται από τον Piezzo Buzzer ένα σήμα της τάξης των 3.125 KHz, που υποδεικνύει την αλλαγή της κατάστασης. Σε αυτή την περίπτωση η τιμή της nvoSwitch γίνεται 1 και μεταδίδεται στο monitor.

```
nvoSwitch.state = 1;  
SetLEDs(0x02, 0xFFu);  
delay(27000);  
SetLEDs(0x00, 0xFFu);  
io_out(ioPiezzo, 100);  
delay(6900);  
io_out(ioPiezzo, 0);
```

³Η τιμή αυτή ενημερώνεται κάθε 10 δευτερόλεπτα, το monitor ελέγχει την ορθότητα της τιμής αυτής και προβαίνει σε ενέργειες για την επιβεβαίωση της σωστής λειτουργίας της σύνδεσης

```
Status.mode=HVAC.COOL;
```

Τέλος στην περίπτωση κατά την οποία, το δείγμα της θερμοκρασίας από την επιστροφή της συνάρτησης `GetTemperature()` είναι μικρότερο των $23^{\circ}C$ (επιθυμητή τιμή - guard-band), η εφαρμογή θέτει τον θερμοστάτη σε HEAT mode. Σε αυτή την περίπτωση η εφαρμογή οδηγεί το LED 1 στο να ανάψει, για να υποδείξει ότι ο θερμοστάτης έχει αρχίσει να θερμαίνει το χώρο, ώστε να φέρει τη θερμοκρασία κοντά στο guardband. Όπως συνέβη και στην προηγούμενη περίπτωση, παράγεται ένα σήμα 3.125KHz από τον Piezzo Buzzer, το οποίο υποδεικνύει αλλαγή κατάστασης. Σε αυτήν την περίπτωση η μεταδιδόμενη τιμή της `nvoSwitch` γίνεται 0.

```
nvoSwitch.state =0;
SetLEDs(0x01,0xFFu);
delay(27000);
SetLEDs(0x00, 0xFFu);
io_out(ioPiezzo,100);
delay(6900);
io_out(ioPiezzo, 0);
Status.mode=HVAC.HEAT;
```

Θέλοντας να δώσουμε πληροφορία στο χρήστη για το μέγεθος των δειγμάτων θερμοκρασίας τα οποία λαμβάνονται κάθε 10 δευτερόλεπτα, εκτός του εύρους $[23^{\circ}C - 27^{\circ}C]$ ο ελεγκτής χρησιμοποιεί παράλληλα με τα LEDs 1,2 και τα υπόλοιπα 6 LEDs του PL 3120. Με αυτόν τον τρόπο δίνεται μια ολοκληρωμένη άποψη της θερμοκρασίας του χώρου (Πίνακας 8.1)

LED	COOL mode	HEAT mode
3	$[30^{\circ}C - 32^{\circ}C]$	$[20^{\circ}C - 18^{\circ}C]$
4	$(32^{\circ}C - 34^{\circ}C)$	$(18^{\circ}C - 16^{\circ}C)$
5	$(34^{\circ}C - 36^{\circ}C)$	$(16^{\circ}C - 14^{\circ}C)$
6	$(36^{\circ}C - 38^{\circ}C)$	$(14^{\circ}C - 12^{\circ}C)$
7	$(38^{\circ}C - 40^{\circ}C)$	$(12^{\circ}C - 10^{\circ}C)$
8	$(40^{\circ}C - 42^{\circ}C)$	$(10^{\circ}C - 8^{\circ}C)$

Πίνακας 8.1: Αποτύπωση ολοκληρωμένου εύρους θερμοκρασιών πάνω στα LEDs

Με την αυτόματη εναλλαγή των καταστάσεων (HEAT-OFF-COOL), τη χρήση δηλαδή αυτομάτου ελέγχου χωρίς την παρουσία του χρήστη στο σύστημα, μπορούμε να εξοικονομήσουμε ενέργεια. Και αυτό γιατί το εικονικό σύστημα του θερμοστάτη ενεργοποιείται ή απενεργοποιείται αυτόματα, ανάλογα με τα δείγματα θερμοκρασίας που λαμβάνει. Για να διαπιστώσουμε τη σωστή εναλλαγή των καταστάσεων από τις οποίες διέρχεται ο θερμοστάτης χρησιμοποιήσαμε ένα σεσουάρ μαλλιών για την προσομοίωση της θέρμανσης και της ψύξης του χώρου. Ο χρόνος δειγματοληψίας των 10 δευτερολέπτων ο οποίος χρησιμοποιήθηκε, ήταν αρκετός για να φέρει τη θερμοκρασία, μετά την θέρμανση ή την ψύξη του αισθητήρα, στα επιθυμητά επίπεδα (μέσα στο guardband). Ο χρόνος αυτός των 10 δευτερολέπτων χρησιμοποιήθηκε για να αποφευχθούν περιττές εναλλαγές on/off στην κατάσταση του θερμοστάτη κάθε λίγα δευτερόλεπτα, γεγονός το οποίο μεταξύ άλλων θα είχε σαν αποτέλεσμα, σε ένα πραγματικό σύστημα, την άσκοπη δαπάνη ενέργειας.⁴ Γίνεται επίσης σαφές ότι σε ένα πραγματικό σύστημα χρειάζεται ένας χρόνος δειγματοληψίας της τάξης των λεπτών.

Ο ελεγκτής ο οποίος υλοποιήθηκε στην εργασία αυτή είναι ένα είδος feedback ελεγκτή, ο οποίος ονομάζεται on-off controller. Ο ελεγκτής αυτός παίρνει ως είσοδο δείγματα θερμοκρασίας από τον αισθητήρα και βγάζει στην έξοδο την επιθυμητή, ανάλογα με την τιμή της θερμοκρασίας, κατάσταση του θερμοστάτη. Βασίζεται δηλαδή, στην μέτρηση της μεταβλητής ελέγχου (θερμοκρασία), προσαρμόζοντας ανάλογα την έξοδό του.

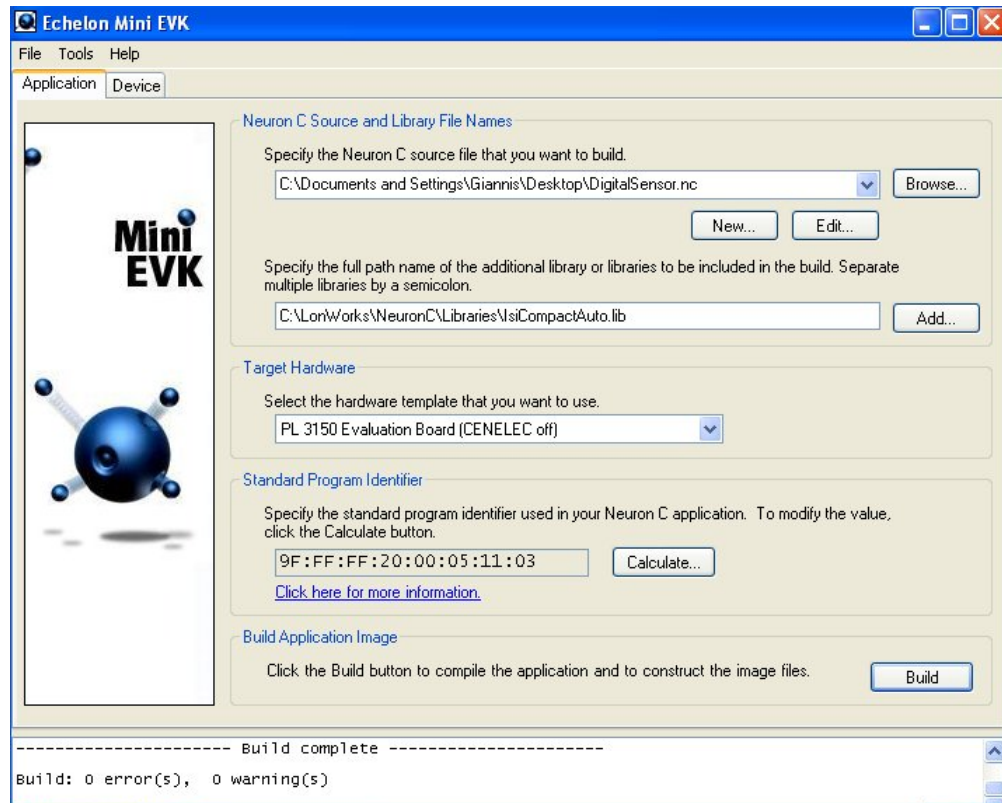
Για την φόρτωση της εφαρμογής του digital temperature alarm sensor στο PL 3150 Evaluation board χρησιμοποιήθηκε το πρόγραμμα Echelon Mini EVK. Στο πρόγραμμα αυτό ακολουθήθηκαν τα εξής βήματα:

- Επιλογή του κώδικα της εφαρμογής για το PL 3150.
- Σύνδεση του κώδικα της εφαρμογής με την βιβλιοθήκη IsiCompactAuto, η οποία περιλαμβάνει βασικές διαδικασίες και συναρτήσεις για τη ρύθμιση των παραμέτρων της αυτόματης σύνδεσης μεταξύ των δύο εφαρμογών.
- Επιλογή ενός μοναδικού Program ID (9F:FF:FF:20:00:05:11:03). Όπως έχουμε πει κάθε εφαρμογή πρέπει να διατηρεί ένα δικό της μοναδικό ID, κυρίως ανάλογα με το functional profile το οποίο υλοποιεί.
- Επιλογή του Neuron Chip για τη φόρτωση του κώδικα της εφαρμογής.

⁴Οι πολλές on/off εναλλαγές στην κατάσταση του θερμοστάτη ίσως δοκιμάσουν και την αντοχή των εσωτερικών κυκλωμάτων του.

- Κατασκευή του image αρχείου, το οποίο φορτώνεται πάνω στο PL 3150 evaluation board και σύνδεση του board με το Mini EVK μέσω του service pin.

Το σχήμα που ακολουθεί αποτυπώνει τα παραπάνω βήματα, τη διαδικασία ουσιαστικά για την κατασκευή του αρχείου image.



Σχήμα 8.10: Διαδικασία φόρτωσης του κώδικα της εφαρμογής στο PL3150 Neuron Chip

Αφότου δημιουργηθεί το image αρχείο, φορτώνεται στο PL 3150 και η εφαρμογή του digital temperature alarm sensor αρχίζει τη λειτουργία της.

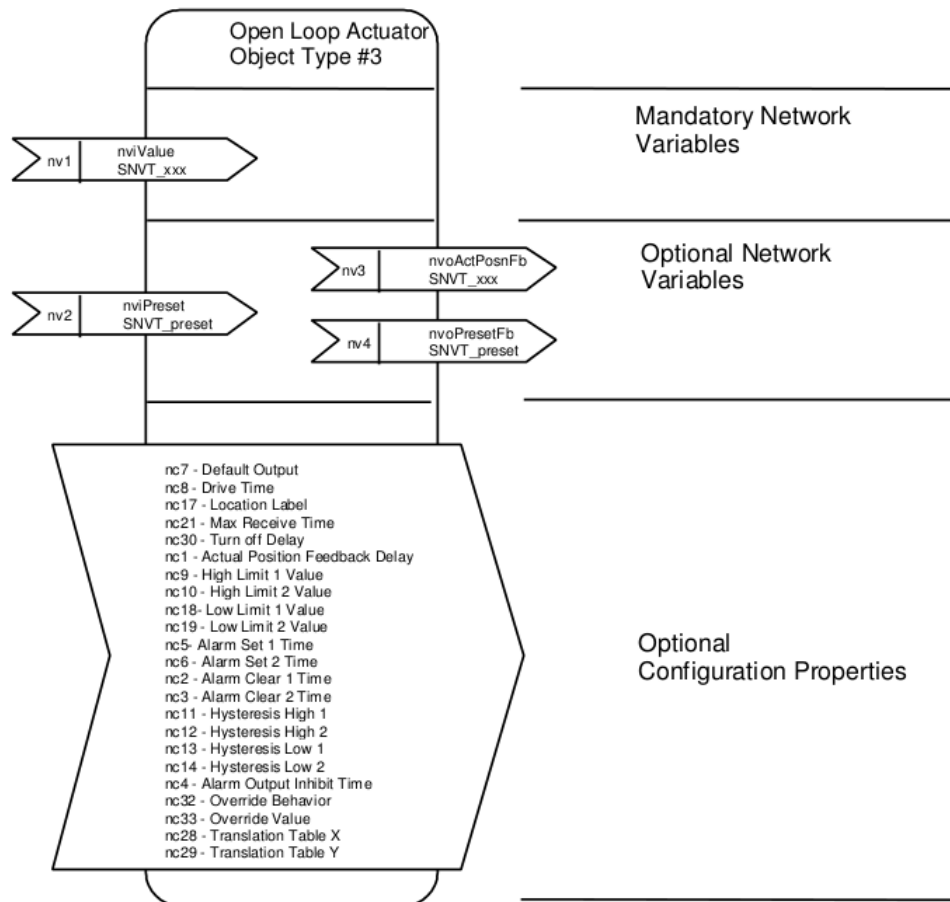
8.3 Monitor

Η εφαρμογή Monitor δημιουργήθηκε για τις ανάγκες του PL 3120 Evaluation board. Συνδέεται με τον digital temperature alarm sensor με σκοπό την ανταλλαγή δεδομένων, μέσω της ISI σύνδεσης των κυρίων μεταβλητών δικτύου τους (primary network variables). Η υλοποίηση της εφαρμογής αυτής, όσον αφορά τα περισσότερα σημεία της είναι η ίδια με την εφαρμογή της προηγούμενης ενότητας. Τα σημεία που τη διαφοροποιούν έχουν να κάνουν

με το κομμάτι της ISI σύνδεσης, αλλά και με το κομμάτι που αφορά την λειτουργικότητα της συσκευής μετά την επίτευξη της σύνδεσης.

8.3.1 Σύνδεση ISI

Στην εφαρμογή αυτή υλοποιήσαμε το functional profile Open Loop Actuator.



Σχήμα 8.11: Functional profile: Open Loop Actuator

Ορίσαμε επίσης και μια μεταβλητή δικτύου εισόδου τύπου `SNVT_switch`, την `nviLight`. Η μεταβλητή αυτή υλοποιεί την μεταβλητή-μέλος `nviValue` του functional profile και χρησιμοποιείται για την σύνδεση του Monitor με την εφαρμογή του digital temperature alarm sensor.

```
network input SNVT_switch nviLight;
fblock SFPTopenLoopActuator {
    nviLight implements nviValue;
```

```
} fbSwitch external_name("Light");
```

Η ISI σύνδεση από την πλευρά του Monitor, χρησιμοποιεί και αυτή την μηχανή ISI. Ακολουθήθηκαν ακριβώς τα ίδια βήματα για την εκκίνηση και τη συντήρηση της μηχανής, με αυτά της εφαρμογής της προηγούμενης υποενότητας, αλλά και ακριβώς τα ίδια βήματα όσον αφορά την διαδικασία για την επεξεργασία των ISI μηνυμάτων που δέχεται η συσκευή.

Ωστόσο στην εφαρμογή αυτή, γίνεται η δημιουργία της αυτόματης σύνδεσης (Auto Enrollment) μεταξύ των δύο εφαρμογών. Για να αρχίσει η διαδικασία της αυτόματης σύνδεσης θα πρέπει πρώτα να έχει περάσει ένα απαιτούμενο χρονικό διάστημα. Το χρονικό αυτό διάστημα είναι τυχαίο και ισούται περίπου με 11.30 λεπτά. Είναι μέρος του εσωτερικού κώδικα της μηχανής ISI και συνδέεται με το event `IsiWarm`. Πρόκειται για έναν timer του πρωτοκόλλου ISI, ο οποίος έχει ως σκοπό του την αποφυγή επικίνδυνων καταστάσεων (race conditions) για την εφαρμογή. Αφότου περάσει αυτό το τυχαίο χρονικό διάστημα, η μηχανή είναι έτοιμη να ξεκινήσει την αυτόματη σύνδεση.

```
void IsiUpdateUserInterface(IsiEvent Event, unsigned
Parameter) {
    if (Event == isiWarm && !IsiIsConnected(nviLight::global_index
)) {
        IsiInitiateAutoEnrollment(&MyCsmoData, nviLight::global_index
);
    }
    #pragma ignore_notused Parameter
}
```

Καλεί την συνάρτηση `IsiInitiateAutoEnrollment()` η οποία δημιουργεί την αυτόματη σύνδεση. Εφόσον γίνει η κλήση της συνάρτησης η υλοποίηση της σύνδεσης από την πλευρά του Monitor έχει υλοποιηθεί. Η συνάρτηση αυτή παίρνει ως όρισμα ένα δείκτη στη δομή του μηνύματος CSMA, το οποίο στέλνεται στην εφαρμογή του digital temperature alarm sensor με σκοπό τη σύνδεση τους, αλλά και το `index` της κύριας μεταβλητής δικτύου της συσκευής. Η δεικτοδότηση της `nviLight` χρησιμοποιείται ως ο αριθμός του assembly το οποίο θα πάρει μέρος στη σύνδεση.

Ακολούθως, η μηχανή ISI στέλνει το μήνυμα σύνδεσης CSMA, το οποίο περιέχει τις ακόλουθες πληροφορίες για την σύνδεση.

```
static const IsiCsmoData MyCsmoData =
```

```

{ ISLDEFAULT_GROUP, isiDirectionInput , 1, 4, 95u, 0,
  {
    0, 0, isiScopeStandard ,{0x00, 0x00, 0x00, 0x00, 0x00, 0x00},1
  }
};

```

Η μεταβλητή σύνδεσης είναι μεταβλητή δικτύου εισόδου τύπου SNVT_switch (95u) με width=1, ενώ το functional profile το οποίο και θα πρέπει να αναγνωρίσει ο digital temperature alarm sensor για να γίνει η σύνδεση είναι το Open Loop Actuator.

Για να κατασκευαστεί το μήνυμα αυτό χρησιμοποιήθηκε η συνάρτηση IsiCreateCsmo(). Επίσης με τη βοήθεια της συνάρτησης αυτής, το μήνυμα σύνδεσης στέλνεται περιοδικά και σε άλλες συσκευές (CSMR), οι οποίες μπορούν να λάβουν μέρος στη σύνδεση. Όπως αναφέρθηκε και στο κεφάλαιο 5, το μήνυμα σύνδεσης CSMA περιέχει ακριβώς τα ίδια πεδία με το κλασσικό CSMO, το οποίο χρησιμοποιείται για manual ή controlled συνδέσεις.



Σχήμα 8.12: Μήνυμα σύνδεσης CSMO/A

8.3.2 Λειτουργία

Η λειτουργία της εφαρμογής αυτής, πραγματοποιείται ουσιαστικά μέσω ενός when statement. Η εφαρμογή αυτή δημιουργήθηκε έχοντας σαν σκοπό της την παρακολούθηση της συσκευής του digital temperature alarm sensor. Λειτουργεί ουσιαστικά σαν μια συσκευή ενημέρωσης του χρήστη, αποτυπώνοντας τις αλλαγές των καταστάσεων της συσκευής του εικονικού θερμοστάτη στα LEDs και στο piezzo buzzer της πλακέτας Mini Gizmo I/O της εφαρμογής αυτής.

Αντανακλά οποιαδήποτε αλλαγή συμβεί στην πλακέτα της πρώτης συσκευής, λαμβάνοντας κάθε φορά την ενημερωμένη τιμή της κύριας μεταβλητής δικτύου του digital temperature alarm sensor. Ουσιαστικά, η πρώτη συσκευή λαμβάνει δείγματα θερμοκρασίας του περιβάλλοντος, καθορίζοντας παράλληλα την κατάστασή της (HEAT, OFF , COOL) κάθε δέκα δευτερόλεπτα και η δεύτερη συσκευή μη έχοντας τον μετρητή των δέκα δευτερολέπτων αποτυπώνει την κατάσταση της πρώτης συσκευής. Αυτό βέβαια γίνεται εφικτό

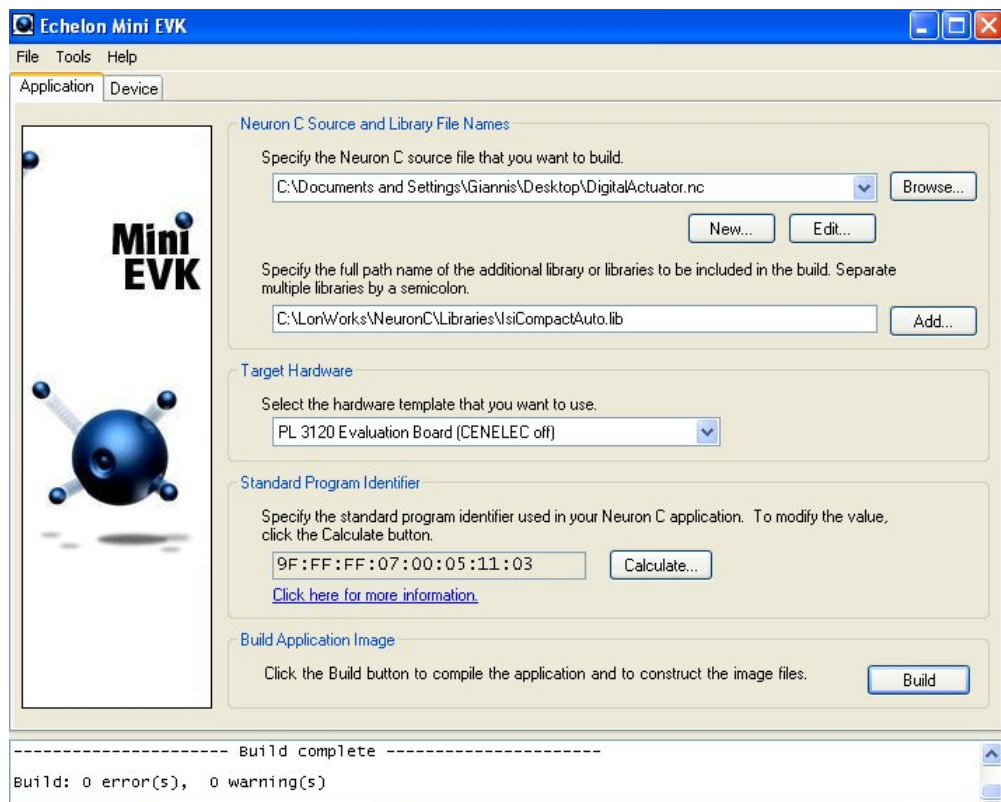
μέσω της ISI σύνδεσης.

```
when(nv_update_occurs(nviLight)){
    unsigned LEDs;
    LEDs=0;
    if(nviLight.state==1){
        LEDs=0x02;
        SetLEDs(LEDs,0xFFu);
        delay(27000);
        SetLEDs(0x00, 0xFFu);
        io_out(ioPiezzo, 50);
        delay(6900);
        io_out(ioPiezzo, 0); }
    if(nviLight.state==0){
        LEDs=0x01;
        SetLEDs(LEDs,0xFFu);
        delay(27000);
        SetLEDs(0x00, 0xFFu);
        io_out(ioPiezzo, 50);
        delay(6900);
        io_out(ioPiezzo, 0); }
    if(nviLight.state==-1){
        LEDs=0x00;
        SetLEDs(LEDs,0xFFu); } }
```

Από τη στιγμή της τροφοδοσίας της και μετά, η συσκευή Monitor περιμένει ουσιαστικά τη στιγμή κατά την οποία θα μεταβληθεί η τιμή της nviLight μέσω του event update_occurs της Neuron C. Ανάλογα με την τιμή αυτή, το Monitor ενεργοποιεί το LED 1 στην περίπτωση του HEAT mode στην συσκευή του digital temperature alarm sensor ή το LED 2 στην περίπτωση του COOL mode (ακριβώς ό,τι συμβαίνει στην πρώτη εφαρμογή). Στην περίπτωση κατά την οποία η nviLight λάβει την τιμή -1, από την nvoSwitch του digital temperature alarm sensor μέσω της σύνδεσης, δεν ενεργοποιεί κανένα LED αφού ο εικονικός θερμοστάτης της πρώτης εφαρμογής έχει διαπιστώσει ότι η θερμοκρασία βρίσκεται εντός των επιθυμητών ορίων. Στις περιπτώσεις των HEAT και COOL mode παράγεται επίσης ένα ηχητικό σήμα από το buzzer, το πλάτος της συχνότητας του οποίου

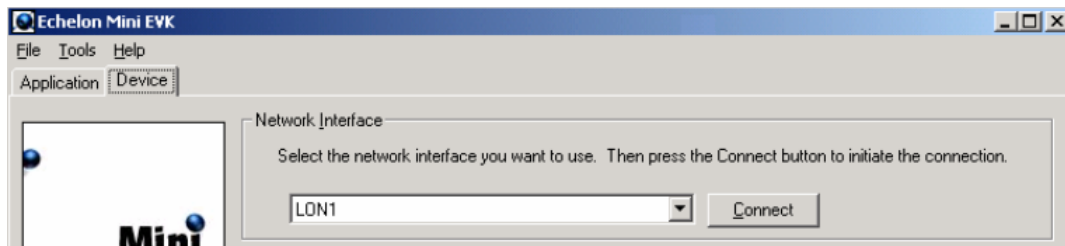
ισούται με το μίσο του πλάτους του σήματος που παράγεται στην εφαρμογή του εικονικού θερμοστάτη.

Για την φόρτωση της εφαρμογής στο PL 3120 Evaluation board χρησιμοποιήθηκε πάλι το πρόγραμμα Mini EVK.



Σχήμα 8.13: Διαδικασία φόρτωσης του κώδικα της εφαρμογής στο PL3120 Neuron Chip

Επιλέγουμε επίσης το network interface που θέλουμε να χρησιμοποιήσουμε στην εφαρμογή μας και τέλος συνδέουμε τον κώδικα μας με το PL3120 Evaluation board για να ξεκινήσει η λειτουργία του.



Σχήμα 8.14: Επιλογή του network interface

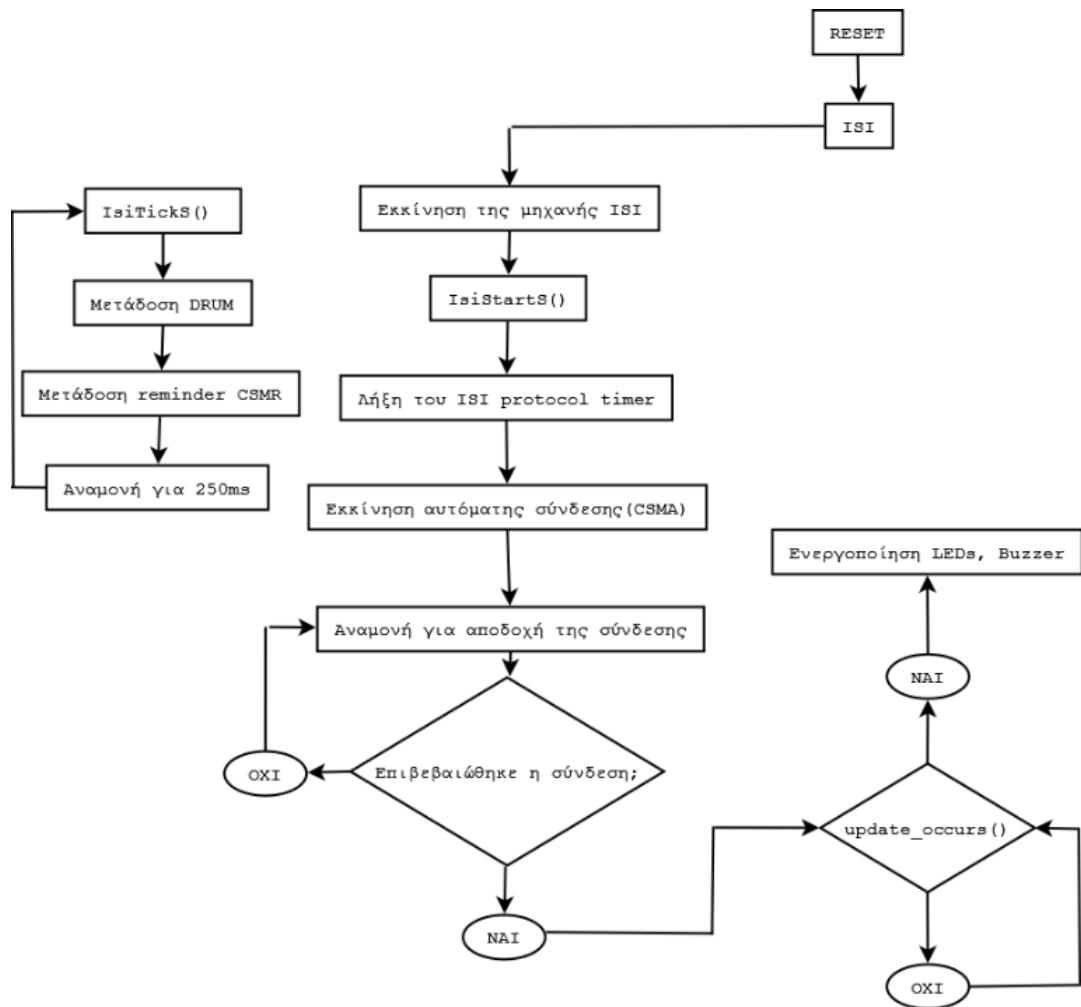


Σχήμα 8.15: Σύνδεση του Monitor με το PL3120 Neuron Chip

8.4 Διαγράμματα ροής

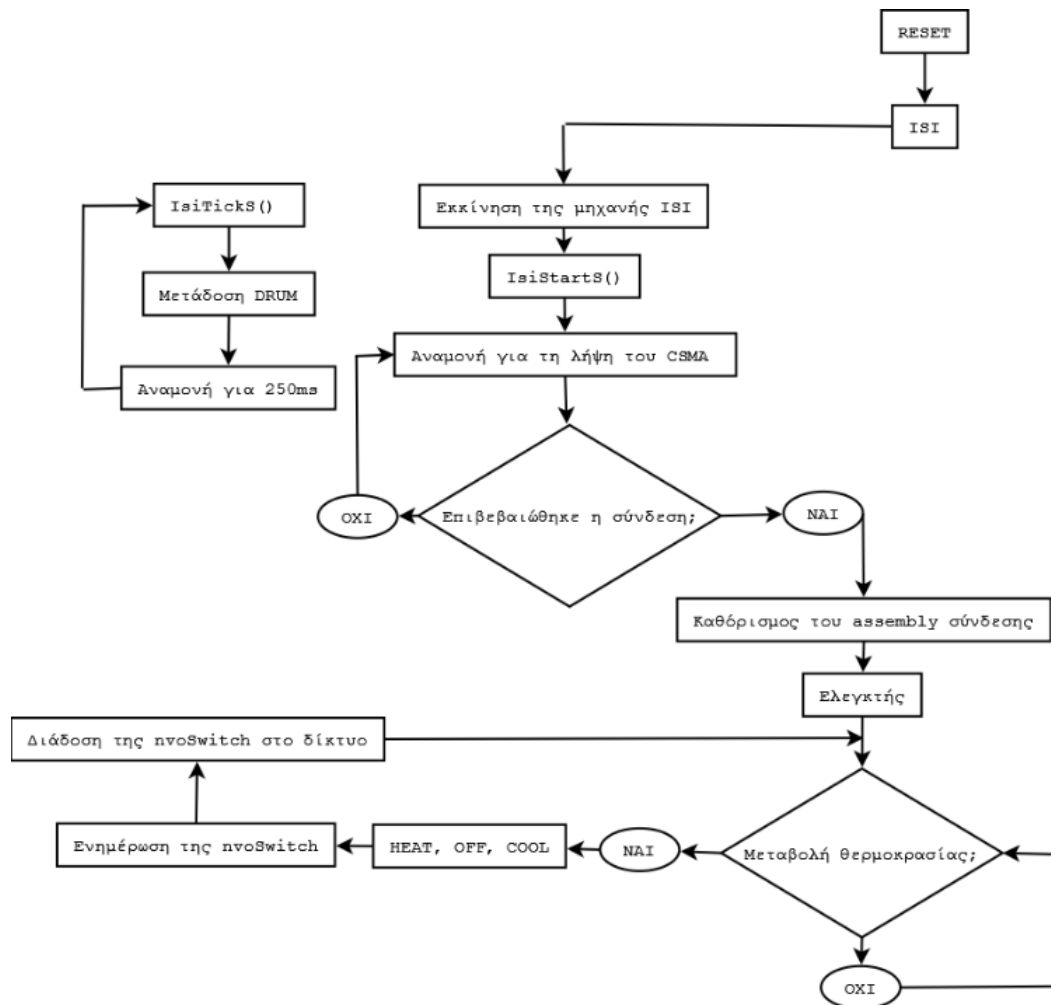
Στην ενότητα αυτή παρουσιάζονται δύο διαγράμματα ροής, τα οποία αφορούν στη λειτουργία των δύο εφαρμογών της εργασίας μας.

Το παρακάτω σχήμα αποτυπώνει το διάγραμμα ροής του Monitor.



Σχήμα 8.16: Διάγραμμα ροής για το Monitor

Το σχήμα που ακολουθεί αποτυπώνει το διάγραμμα ροής για τον digital temperature alarm sensor.



Σχήμα 8.17: Διάγραμμα ροής για τον digital temperature alarm sensor

8.5 Πηγαίοι κωδικες

Στην ενότητα αυτή παρουσιάζονται οι πηγαίοι κώδικες Neuron C των δύο εφαρμογών της εργασίας μας.

MONITOR

```

#pragma enable_sd_nv_names
#pragma num_alias_table_entries 2

#include <snvt_cfg.h>

```

```

#include <isi.h>
#include <control.h>
#include <io_types.h>

IO_0 output frequency clock(3) invert ioPiezzo = 1;
IO_1 output bit ioLEDLd = 1;
IO_2 output bitshift numbits(8) ioLEDs;

unsigned PreviousLEDs = 0;
void SetLEDs(unsigned LEDs, unsigned Mask) {
    LEDs |= PreviousLEDs & ~Mask;
    PreviousLEDs = LEDs;
    io_out(ioLEDs, ~LEDs);
    io_out(ioLEDLd, 0);
    io_out(ioLEDLd, 1);
}

network input SNVT_switch nviLight;

network input SCPTnwrkCnfg cp cp_info(reset_required)
    nciNetConfig = CFG_EXTERNAL;

device_properties {
    nciNetConfig
};

fblock SFPTopenLoopActuator
{
    nviLight implements nviValue;

} fbSwitch external_name("Light");

```

```

when(wink)
{
    unsigned Index, StoredPreviousLeds;
    StoredPreviousLeds = PreviousLEDs;
    for (Index = 0; Index != 255u; ++Index)
    {
        SetLEDs(Index, 0xFFu);
        delay(100);
    }
    SetLEDs(StoredPreviousLeds, 0xFF);
    io_out(ioPiezzo, 100);
    delay(1200);
    io_out(ioPiezzo, 0);
}

```

```

eeprom SCPTnwrkCnfg oldNetConfig = CFG_NUL;

```

```

when(reset) {
    SCPTnwrkCnfg networkConfig;
    SetLEDs(0x99u, 0xFFu);
    networkConfig = oldNetConfig;
    if (networkConfig == CFG_NUL)
    {
        ::nciNetConfig = CFG_LOCAL;
    }
    oldNetConfig = ::nciNetConfig;

    if (::nciNetConfig == CFG_LOCAL)
    {
        if (networkConfig == CFG_EXTERNAL)
        {

            IsiReturnToFactoryDefaults();

```

```

    }

    scaled_delay(31745UL);
    IsiStartS(isiFlagNone);
}

SetLEDs(0x00, 0xFFu);
io_out(ioPiezzo, 100);
delay(1200);
io_out(ioPiezzo, 0);

}

when(msg_arrives) {
    if (IsiApproveMsg()) {
        if (IsiProcessMsgS()) {
            // TODO: process unprocessed ISI messages here (if any)
            ;
        }
    } else {
        // TODO: process other application messages here (if any)
        ;
    }
}

when(nv_update_occurs(nviLight))
{
    unsigned LEDs;
    LEDs=0;

    if(nviLight.state==1)
    {

```

```

    LEDs=0x02;
    SetLEDs(LEDs,0xFFu);
    delay(27000);
    SetLEDs(0x00, 0xFFu);
    io_out(ioPiezzo, 50);
    delay(6900);
    io_out(ioPiezzo, 0);
}
if(nviLight.state==0)
{
    LEDs=0x01;
    SetLEDs(LEDs,0xFFu);
    delay(27000);
    SetLEDs(0x00, 0xFFu);
    io_out(ioPiezzo, 50);
    delay(6900);
    io_out(ioPiezzo, 0);
}
if(nviLight.state==-1)
{
    LEDs=0x00;
    SetLEDs(LEDs,0xFFu);
}
}

mtimer repeating isiTimer = 250;
#define SERVICE_PIN_ACTIVATION (10u*ISL_TICKS_PER_SECOND)
unsigned ServicePinActivation;

when(timer_expires(isiTimer))
{
    IsiTickS();
    if (service_pin_state()) {

```



```

        ++ServicePinActivation;
        if (ServicePinActivation > SERVICE_PIN_ACTIVATION) {
            oldNetConfig = ::nciNetConfig = CFG_LOCAL;
            IsiReturnToFactoryDefaults();
        }
    } else {
        ServicePinActivation = 0;
    }
}

static const IsiCsmoData MyCsmoData =
    { ISLDEFAULT_GROUP, isiDirectionInput, 1, 4, 95u, 0,
      {
        0, 0, isiScopeStandard, {0x00, 0x00, 0x00, 0x00, 0x00, 0x00}, 1
      }
    };

void IsiUpdateUserInterface(IsiEvent Event, unsigned Parameter) {
    if (Event == isiWarm && !IsiIsConnected(nviLight::global_index))
    {

        IsiInitiateAutoEnrollment(&MyCsmoData, nviLight::global_index);

    }
#pragma ignore_notused Parameter
}

void IsiCreateCsmo(unsigned Assembly, IsiCsmoData* pCsmoData) {
    memcpy(pCsmoData, &MyCsmoData, sizeof(IsiCsmoData));
#pragma ignore_notused Assembly
}

```

```

unsigned IsiGetAssembly(const IsiCsmoData* pCsmoData, boolean Auto) {
    return ISLNO_ASSEMBLY;
#pragma ignore_notused pCsmoData
#pragma ignore_notused Auto
}

unsigned IsiGetNextAssembly(const IsiCsmoData* pCsmoData, boolean Auto,
unsigned Assembly) {
    return ISLNO_ASSEMBLY;
#pragma ignore_notused pCsmoData
#pragma ignore_notused Auto
#pragma ignore_notused Assembly
}

unsigned IsiGetNvIndex(unsigned Assembly, unsigned Offset) {
    return nviLight::global_index;
#pragma ignore_notused Offset
#pragma ignore_notused Assembly
}

}

eeprom IsiConnection MyConnectionTable;

unsigned IsiGetConnectionTableSize(void) {
    return 1u;
}

const IsiConnection* IsiGetConnection(unsigned Index) {
    return &MyConnectionTable;
#pragma ignore_notused Index
}

void IsiSetConnection(IsiConnection* pConnection, unsigned Index) {

```

```

        MyConnectionTable = *pConnection;
#pragma ignore_notused Index
    }

#ifdef _DEBUG
#   ifndef _MINIKIT
void IsiSetDomain(domain_struct* pDomain, unsigned Index) {
    ;
#       pragma ignore_notused pDomain
#       pragma ignore_notused Index
    }
#   pragma ignore_notused IsiSetDomain
#   endif
#endif

```

Ακολουθεί ο κώδικας για την εφαρμογή του digital temperature alarm sensor.

DIGITAL TEMPERATURE ALARM SENSOR

```

#pragma enable_sd_nv_names
#pragma num_alias_table_entries 2
#pragma enable_io_pullups

#include <isi.h>
#include <control.h>
#include <io_types.h>
#include <snvt_cfg.h>
#include <snvt_hv.h>

IO_0 output frequency clock(3) invert ioPiezzo = 1;
IO_1 output bit ioLEDLd = 1;
IO_2 output bitshift numbits(8) ioLEDs;

```

```

IO_7 touch ioThermometer;

#define DS18S20_SKIP_ROM 0xCCu
#define DS18S20_CONVERT 0x44u
#define DS18S20_READ 0xBEu

unsigned PreviousLEDs = 0;

void SetLEDs(unsigned LEDs, unsigned Mask) {
    LEDs |= PreviousLEDs & ~Mask;
    PreviousLEDs = LEDs;
    io_out(ioLEDs, ~LEDs);
    io_out(ioLEDLd, 0);
    io_out(ioLEDLd, 1);
}

SNVT_temp_p GetTemperature(void) {
    union {
        SNVT_temp_p snvtTempP;
        unsigned Bytes[2];
    } CurrentTemperature;

    CurrentTemperature.snvtTempP = 327671;

    if (touch_reset(ioThermometer)) {
        (void) touch_byte(ioThermometer, DS18S20_SKIP_ROM);
        (void) touch_byte(ioThermometer, DS18S20_READ);
        CurrentTemperature.Bytes[1]
            = touch_byte(ioThermometer, 0xFFu);
        CurrentTemperature.Bytes[0]
            = touch_byte(ioThermometer, 0xFFu);
        if (touch_reset(ioThermometer)) {

```

```

        CurrentTemperature.snvtTempP *= 501;

        (void) touch_byte(ioThermometer, DS18S20_SKIP_ROM);
        (void) touch_byte(ioThermometer, DS18S20_CONVERT);
    } else {
        CurrentTemperature.snvtTempP = 327671;
    }
}
return CurrentTemperature.snvtTempP;
}

```

```
network output SNVT_switch nvoSwitch;
```

```

fblock SFPTswitch
{
    nvoSwitch implements nvoSwitch;

} fbSwitch external_name("Switch");

```

```

when(wink)
{
    unsigned Index, StoredPreviousLeds;
    StoredPreviousLeds = PreviousLEDS;
    for (Index = 0; Index != 255u; ++Index)
    {
        SetLEDS(Index, 0xFFu);
        delay(100);
    }
    SetLEDS(StoredPreviousLeds, 0xFF);
    io_out(ioPiezzo, 100);
    delay(1200);
}

```

```

        io_out(ioPiezzo, 0);
    }

network input SCPTnwrkCnfg cp cp_info(reset_required)
nciNetConfig = CFG_EXTERNAL;

device_properties {
    nciNetConfig
};

eeprom SCPTnwrkCnfg oldNetConfig = CFG_NUL;

when(reset) {
    SCPTnwrkCnfg networkConfig;
    SetLEDs(0x99u, 0xFFu);
    networkConfig = oldNetConfig;
    if (networkConfig == CFG_NUL)
    {
        ::nciNetConfig = CFG_LOCAL;
    }
    oldNetConfig = ::nciNetConfig;

    if (::nciNetConfig == CFG_LOCAL)
    {
        if (networkConfig == CFG_EXTERNAL)
        {

            IsiReturnToFactoryDefaults();

        }

        scaled_delay(31745UL);
        IsiStartS(isiFlagNone);
    }
}

```

```

    }

    SetLEDs(0x00, 0xFFu);
    io_out(ioPiezzo, 100);
    delay(1200);
    io_out(ioPiezzo, 0);

}

when(msg_arrives) {
    if (IsiApproveMsg()) {
        if (IsiProcessMsgS()) {
            // TODO: process unprocessed ISI messages here (if any)
            ;
        }
    } else {
        // TODO: process other application messages here (if any)
        ;
    }
}

network input SNVT_temp_p nviLimit = 2500ul;
network input SNVT_temp_p nviOffset=200ul;
network input SNVT_hvac_mode nviApplicMode;

void Controller(void)
{
    SNVT_temp_p getTemp;
    SNVT_hvac_status Status;
    getTemp = GetTemperature();
}

```

```

if(getTemp>nviLimit + nviOffset)
{
    nvoSwitch.state =1;
    SetLEDs(0x02,0xFFu);
    delay(27000);
    SetLEDs(0x00, 0xFFu);
    io_out(ioPiezzo, 100);
    delay(6900);
    io_out(ioPiezzo, 0);

    Status.mode=HVAC.COOL;
    nviApplicMode=HVAC.COOL;

}

if (getTemp < nviLimit - nviOffset)
{
    nvoSwitch.state =0;
    SetLEDs(0x01,0xFFu);
    delay(27000);
    SetLEDs(0x00, 0xFFu);
    io_out(ioPiezzo,100);
    delay(6900);
    io_out(ioPiezzo, 0);

    Status.mode=HVAC.HEAT;
    nviApplicMode=HVAC.HEAT;

}

if(getTemp>=nviLimit - nviOffset && getTemp <=

```



```

nviLimit + nviOffset )
{

    if(getTemp>nviLimit){

        nvoSwitch.state =-1;
        SetLEDs(0x00,0xFFu);
        Status.mode=HVAC.OFF;
        nviApplicMode=HVAC.OFF;
    }

    if(getTemp<nviLimit){

        nvoSwitch.state =-1;
        SetLEDs(0x00,0xFFu);
        Status.mode=HVAC.OFF;
        nviApplicMode=HVAC.OFF;
    }

    if(getTemp==nviLimit)
    {
        nvoSwitch.state =-1;
        SetLEDs(0x00,0xFFu);
        Status.mode=HVAC.OFF;
        nviApplicMode=HVAC.OFF;
    }

}

```

```

if((getTemp<=2000ul && getTemp>=1800ul) ||
    (getTemp<=3200ul && getTemp>=3000ul) )
{
    SetLEDs(0x4,0xFFu);
    SetLEDs(0x00, 0xFFu);
}

```

```

if((getTemp<1800ul && getTemp>=1600ul) ||
    (getTemp<=3400ul && getTemp>3200ul) )
{
    SetLEDs(0x8,0xFFu);
    SetLEDs(0x00, 0xFFu);
}

```

```

if((getTemp<1600ul && getTemp>=1400ul) ||
    (getTemp<=3600ul && getTemp>3400ul) )
{
    SetLEDs(0x10,0xFFu);
    SetLEDs(0x00, 0xFFu);
}

```

```

if((getTemp<1400ul && getTemp>=1200ul) ||
    (getTemp<=3800ul && getTemp>3600ul) )
{
    SetLEDs(0x20,0xFFu);
    SetLEDs(0x00, 0xFFu);
}

```

```

if((getTemp<1200ul && getTemp>=1000ul) ||
    (getTemp<=4000ul && getTemp>3800ul) )
{
    SetLEDs(0x40,0xFFu);
    SetLEDs(0x00, 0xFFu);
}

```

```

    }

    if ((getTemp < 1000ul && getTemp >= 800ul) ||
        (getTemp <= 4200ul && getTemp > 4000ul) )
    {
        SetLEDs(0x80, 0xFFu);
        SetLEDs(0x00, 0xFFu);
    }
}

when(timer_expires(led_timer))
{

    Controller();
}

mtimer repeating isiTimer = 250;
#define SERVICE_PIN_ACTIVATION (10u*ISL_TICKS_PER_SECOND)
unsigned ServicePinActivation;

when(timer_expires(isiTimer))
{
    IsiTickS();
    if (service_pin_state()) {
        ++ServicePinActivation;
        if (ServicePinActivation > SERVICE_PIN_ACTIVATION) {
            oldNetConfig = ::nciNetConfig = CFG_LOCAL;
            IsiReturnToFactoryDefaults();
        }
    }
    else {
        ServicePinActivation = 0;
    }
}

```

```

}

unsigned IsiGetAssembly(const IsiCsmoData * pCsmo, boolean Auto) {
    if (Auto &&
        pCsmo->Profile    == 4 &&
        pCsmo->Direction == isiDirectionInput &&
        pCsmo->Width      == 1) {
        return nvoSwitch::global_index;
    }
    return ISL_NO_INDEX;
}

unsigned IsiGetNextAssembly(const IsiCsmoData* pCsmoData, boolean
    Auto, unsigned Assembly) {
    return ISL_NO_ASSEMBLY;
#pragma ignore_notused pCsmoData
#pragma ignore_notused Auto
#pragma ignore_notused Assembly
}

unsigned IsiGetNvIndex(unsigned Assembly, unsigned Offset) {
    return nvoSwitch::global_index;
#pragma ignore_notused Offset
#pragma ignore_notused Assembly

}

eeprom IsiConnection MyConnectionTable;

unsigned IsiGetConnectionTableSize(void) {
    return 1u;
}

```

```

const IsiConnection* IsiGetConnection(unsigned Index) {
    return &MyConnectionTable;
#pragma ignore_notused Index
}

void IsiSetConnection(IsiConnection* pConnection, unsigned Index) {
    MyConnectionTable = *pConnection;
#pragma ignore_notused Index
}

#ifdef DEBUG
#   ifndef _MINIKIT
void IsiSetDomain(domain_struct* pDomain, unsigned Index) {
    ;
#       pragma ignore_notused pDomain
#       pragma ignore_notused Index
}
#   pragma ignore_notused IsiSetDomain
#   endif
#endif

```

Συμπεράσματα

Ο κύριος στόχος της εργασίας αυτής ήταν ο σχεδιασμός και η ανάπτυξη δύο αυτοεγκαιθιστώμενων συσκευών σε ένα δίκτυο LONWORKS τεχνολογίας. Ο στόχος αυτός επιτεύχθηκε με τη βοήθεια του πρωτοκόλλου ISI, το οποίο υλοποιήθηκε ξεχωριστά σε κάθε συσκευή με σκοπό την δημιουργία της μεταξύ τους αυτόματης σύνδεσης. Μέσω λοιπόν του πρωτοκόλλου ISI αλλά και του πρωτοκόλλου LonTalk το οποίο χρησιμοποιήθηκε για τη μορφοποίηση των ανταλλασσόμενων μηνυμάτων, οι συσκευές οι οποίες δημιουργήθηκαν, μπόρεσαν να επικοινωνήσουν μέσω του δικτύου μεταφοράς ηλεκτρικής ισχύος.

Έτσι λοιπόν η εφαρμογή στο PL3150 Neuron Chip λαμβάνει δείγματα θερμοκρασίας από τον αισθητήρα και προσομοιώνοντας τη λειτουργία ενός θερμοστάτη αποτυπώνει την έξοδό του (HEAT, OFF, COOL), στα LEDs του Neuron chip. Η εφαρμογή στο PL3120 Neuron Chip, λειτουργώντας σαν μια απλή συσκευή παρακολούθησης, αποτυπώνει και αυτή με την σειρά της την έξοδο του θερμοστάτη στα δικά της LEDs, μέσω της ανταλλαγής ενημερωμένων, κάθε 10 δευτερόλεπτα, τιμών της μεταβλητής σύνδεσης, με την εφαρμογή του PL3120 Neuron Chip.

Επίσης στην εργασία αυτή :

- Κάθε συσκευή χρειάζεται να είναι υπεύθυνη για τη δική της ρύθμιση και εγκατάσταση.
- Με την δημιουργία αυτόματης σύνδεσης, η συμμετοχή του χρήστη στο σύστημα είναι μηδαμινή. Αυτό έχει ως αποτέλεσμα την ελαχιστοποίηση της πολυπλοκότητας. Μετά την τροφοδοσία των συσκευών τα πάντα στο δίκτυο λειτουργούν αυτόματα.
- Ο ρυθμός λήψης των τιμών θερμοκρασίας του περιβάλλοντος από τον αισθητήρα είναι ιδανικός. Αυτό βοηθά την συσκευή του εικονικού θερμοστάτη να μην εναλλάσσεται μεταξύ των καταστάσεων on/off πολύ γρήγορα, με αποτέλεσμα την άσκοπη σπατάλη ενέργειας. Γίνεται φανερό ότι σε μεγαλύτερα και πραγματικά συστήματα ο ρυθμός

δειγματοληψίας πρέπει να είναι της τάξης των λεπτών.

- Χωρίς την υλοποίηση του πρωτοκόλλου ISI δεν θα υπήρχε σύνδεση μεταξύ των δύο εφαρμογών. Κάθε συσκευή θα λειτουργούσε αυτόνομα, με αποτέλεσμα η εφαρμογή digital temperature alarm sensor να παίρνει δείγματα θερμοκρασίας, να καθορίζει την έξοδο του θερμοστάτη αλλά να μην μπορεί να μεταδώσει την τιμή της εξόδου στην εφαρμογή Monitor. Στην ουσία δεν θα υπήρχε δίκτυο.

Για την υλοποίηση της εργασίας αλλά και για την κατανόηση της επικοινωνίας των δύο συσκευών, χρειάστηκε η μελέτη διαφόρων στοιχείων του υλικού της πλατφόρμας LONWORKS τα κυριότερα εκ των οποίων ήταν :

- Το Neuron Chip.
- Η πλακέτα Mini Gizmo.
- Το κύκλωμα ζεύξης του Neuron Chip με το κανάλι επικοινωνίας.
- Η τροφοδοσία των δύο συσκευών.

Εκτός του υλικού χρειάστηκε επίσης η μελέτη :

- Της σηματοδοσίας μέσω των γραμμών μεταφοράς ηλεκτρικής ισχύος.
- Των δικτύων LONWORKS τεχνολογίας, όσον αφορά τα βασικά σημεία της υλοποίησής τους.
- Του πρωτοκόλλου LonTalk, όσον αφορά τη δομή του αλλά και τις υπηρεσίες που χρησιμοποιεί για την επικοινωνία των δύο συσκευών.
- Του πρωτοκόλλου ISI, για τη δημιουργία συνδέσεων μεταξύ των μεταβλητών δικτύου των δύο συσκευών.
- Της γλώσσας προγραμματισμού Neuron C για την δημιουργία του κώδικα των εφαρμογών.

Με αυτόν τον συνδυασμό υλικού και λογισμικού επετεύχθη η δημιουργία ενός δικτύου ελέγχου αυτοεγκαθιστώμενων συσκευών.

Μελλοντικός σχεδιασμός

Ακολουθούν κάποιες προεκτάσεις της εργασίας, οι οποίες θα μπορούσαν να υλοποιηθούν στο μέλλον:

- Ανάπτυξη ενός managed δικτύου ελέγχου το οποίο θα εγκαθιστά τις συσκευές του με τη χρήση ενός εργαλείου δικτύου, όπως το LonMaker. Θα ήταν ενδιαφέρον σε αυτή την περίπτωση να μελετηθούν οι διαφορές που προκύπτουν, μεταξύ των managed και των αυτοεγκαθιστώμενων δικτύων (ISI networks), καθώς επίσης και οι διάφορες πηγές τροφοδοσίας οι οποίες μπορούν να συνδεθούν με τις συσκευές αυτές για μεγαλύτερη εξοικονόμηση ενέργειας (Energy storage power supplies).
- Ανάπτυξη δικτύων ελέγχου βασιζόμενα σε μια διαφορετική τεχνολογία, η οποία χρησιμοποιεί ένα συνεστραμμένο ζεύγος καλωδίων για την επικοινωνία των συσκευών του δικτύου. Η τεχνολογία αυτή χρησιμοποιεί τα λεγόμενα FT boards (free topology), καθώς και το λογισμικό Nodebuilder για την ανάπτυξη εφαρμογών.
- Εισαγωγή PID ή fuzzy λογικής στον ελεγκτή του υπάρχοντος συστήματος. Με αυτό τον τρόπο ο ελεγκτής θα είναι σε θέση να διαχειρίζεται με διαφορετικό τρόπο την έξοδο του θερμοστάτη.

Βιβλιογραφία

- [1] Atmel Corporation. *AVR318: Dallas 1-Wire master*. United States of America, 2004.
- [2] Echelon Corporation. *078-0183-01A LonWorks Intro*. United States of America, 1999.
- [3] Echelon Corporation. *NeuronCReferenceGuide.pdf*. United States of America, 2001.
- [4] Echelon Corporation. *005-0154-01C PLDatabook*. United States of America, 2005.
- [5] Echelon Corporation. *078-0299-01F ISI Programmer's Guide V3*. United States of America, 2007.
- [6] Echelon Corporation. *078-0300-01F ISI Protocol Spec V3*. United States of America, 2007.
- [7] Echelon Corporation. *078-0002-02H Neuron C Programmers Guide*. United States of America, 2009.
- [8] Echelon Corporation. *078-0183-01B Intro to LonWorks Rev 2*. United States of America, 2009.
- [9] Echelon Corporation. *078-0296-01B.pdf*. United States of America, 2005.
- [10] Echelon Corporation. *PL Technology Evaluation Kit User's Guide*. United States of America, 2005.
- [11] Echelon Corporation. *Mini EVK Hardware Guide Revision 3*. United States of America, 2006.

- [12] Echelon Corporation. *Mini EVK User's Guide Revision 3*. United States of America, 2006.
- [13] Echelon Corporation. *Power Line Smart Transceiver Development Support Kit User's Guide*. United States of America, 2006.
- [14] Echelon Corporation. *078-0394-01A Mini FX PL Examples Guide*. United States of America, 2009.
- [15] Echelon Corporation. *078-0398-01A Mini FX User's Guide*. United States of America, 2009.
- [16] Echelon Corporation. *078-0402-01B Neuron Tools Errors Guide*. United States of America, 2009.
- [17] Georgios A. Saridakis. *Development of an Intelligent Indoor Environment and Energy Management System for Greenhouses*. Brunel University, September 2005.
- [18] Jorge Juan Bueno Manglano. *Design and Development of two devices based on LonWorks Technologies over Power Line media*. Malardalen University, March 2007.
- [19] www.lonmark.org.