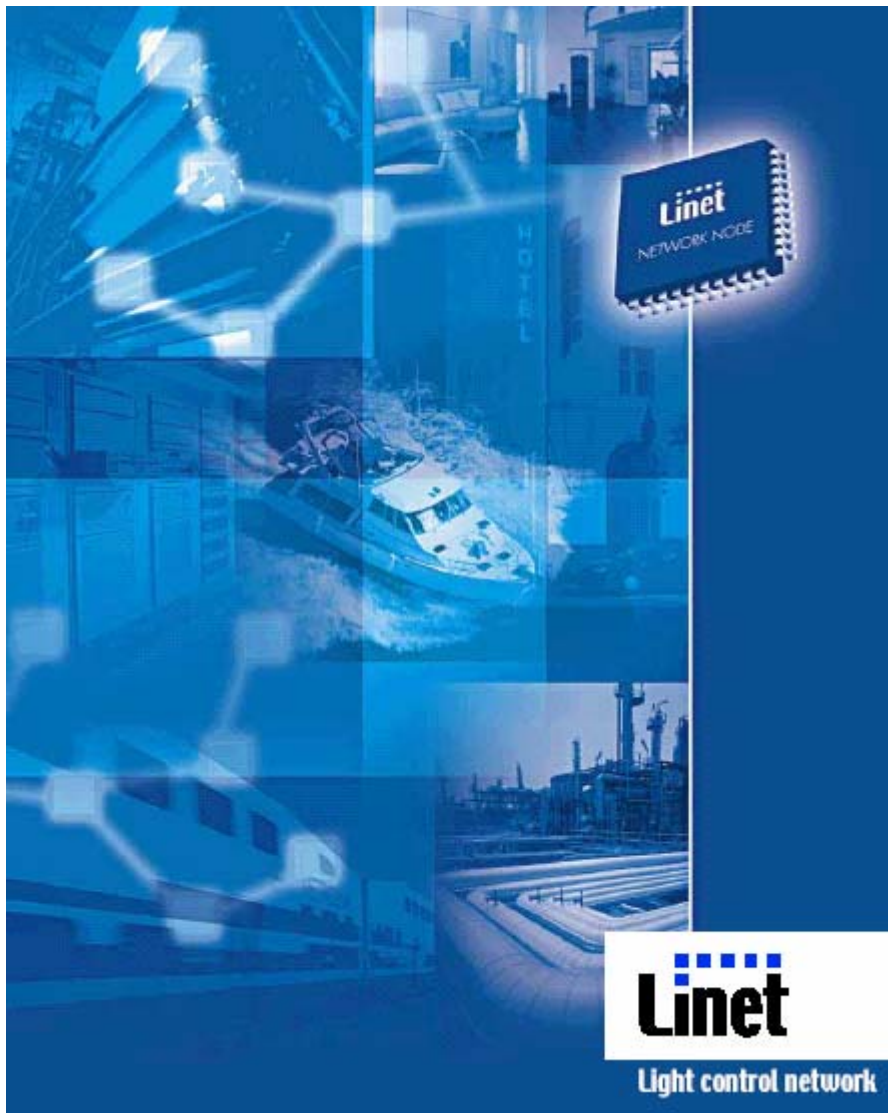


ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΙΑΣ

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΒΙΟΜΗΧΑΝΙΚΟ ΣΥΣΤΗΜΑ ΕΛΕΓΧΟΥ ΤΕΡΜΑΤΙΚΩΝ ΜΟΝΑΔΩΝ ΜΕ ΔΙΚΤΥΟ  
LINET



ΣΠΟΥΔΑΣΤΗΣ  
ΓΑΛΕΡΑΚΗΣ ΕΥΣΤΡΑΤΙΟΣ  
ΕΙΣΗΓΗΤΗΣ  
ΜΗΝΑΔΑΚΗΣ ΙΩΑΝΝΗΣ

ΗΡΑΚΛΕΙΟ 2006

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΚΕΦΑΛΑΙΟ 1

1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ LINET.....	4
1.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	5
1.2.1 Nodes.....	5
1.2.2 Controller.....	6
1.3 ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ.....	7

## ΚΕΦΑΛΑΙΟ 2

2.1 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ NODE LIN02.....	8
2.1.1 Το node σε μορφή ολοκληρωμένου.....	9
2.1.2 Το node σε μορφή hybrid.....	10
2.1.3 Προγραμματιζόμενα bit.....	11
2.2 ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ NODE LIN02.....	15
2.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΦΑΡΜΟΓΩΝ.....	16
2.3.1 Toggle group.....	16
2.3.2 PWM group.....	16
2.3.3 Analog input.....	18
2.3.4 Data groups.....	18
2.4 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ CONTROLLER LIC03.....	21
2.5 ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ CONTROLLER LIC03.....	21
2.6 CONNECTIONS.....	22
2.6.1 Είσοδος ισχύος (J12).....	22
2.6.2 Linet network connector (J3).....	23
2.6.3 Σειριακή διασύνδεση RS232/485 (J4).....	23
2.6.4 Δίαυλος επέκτασης (J1).....	25
2.6.5 Ethernet connector (J7).....	27
2.7 ΜΗΚΟΣ ΔΙΚΤΥΟΥ.....	27
2.8 ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΗ ΣΥΜΒΑΤΟΤΗΤΑ (EMC).....	28

## ΚΕΦΑΛΑΙΟ 3

3.1 ΠΡΟΕΤΟΙΜΑΣΙΑ ΠΡΙΝ ΤΗ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΔΙΚΤΥΟΥ.....	29
3.1.1 Έλεγχος των nodes με dc τάση.....	29
3.1.2 Διαδικασία Lamp test.....	29
3.2 ΑΡΧΙΖΟΝΤΑΣ ΤΗΝ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΔΙΚΤΥΟΥ (CONFIGURATION MODE).....	29
3.2.1 Εγκατάσταση.....	29
3.2.2 Ξεκινώντας την διαμόρφωση.....	30
3.2.3 Ομάδες, διευθύνσεις και λειτουργίες.....	31
3.2.4 Δημιουργώντας ένα Toggle group.....	32
3.2.5 Δημιουργώντας ένα Dimmer group.....	32
3.2.6 Αλλαγές στις ομάδες που έχουν δημιουργηθεί.....	33
3.2.7 Δημιουργώντας Data groups.....	33

3.2.8	Παράδειγμα διαμόρφωσης σε ένα σπίτι.....	33
3.3	ΒΑΣΙΚΕΣ ΡΥΘΜΙΣΕΙΣ.....	39
3.3.1	Αποθήκευση και φόρτωση ρυθμίσεων διαμόρφωσης.....	39
3.3.2	Σβήσιμο των ρυθμίσεων.....	39
3.3.3	Προγραμματιζόμενα bits.....	39
3.3.4	Αρχείο Backup.....	40

## **ΚΕΦΑΛΑΙΟ 4**

4.1	ΕΛΕΓΧΟΣ ΤΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΟ TERMINAL (COMMAND MODE).....	41
4.1.1	Γενική περιγραφή του Command mode.....	41
4.1.2	Εντολές Set και Clear.....	41
4.1.3	Εντολές Step up και Step down.....	42
4.1.4	Εντολές In και Out.....	43
4.1.5	Άλλες εντολές.....	43

## **ΚΕΦΑΛΑΙΟ 5**

5.1	ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΙΑΘΕΣΙΜΩΝ ΟΜΑΔΩΝ.....	46
5.1.1	Βασικές ομάδες – Toggle, Analog input, Dimmer, Data.....	46
5.1.2	Πρόσθετες ομάδες – I/O node, lamp group, call group, AD/state, control.....	49
5.1.3	Δημιουργία ομάδων Master/Slave.....	53

## **ΚΕΦΑΛΑΙΟ 6**

6.1	ΕΠΗΡΟΣΘΕΤΕΣ ΔΥΝΑΤΟΤΗΤΕΣ ΤΟΥ LINET.....	54
6.2	LINET MONITOR.....	54
6.2.1	Κώδικας του Linet Monitor.....	54
6.2.2	Χρήση του Linet Monitor.....	62
6.3	LINET CLIENT.....	64
6.3.1	Κώδικας του Linet Client.....	66
6.3.2	Χρήση του Linet Client.....	71
6.4	TEMP_SENSOR.....	73
6.4.1	Κώδικας του Temp_sensor.....	73
6.4.2	Χρήση του Temp_sensor.....	75

	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	77
--	-------------------	----

	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	78
--	-------------------	----

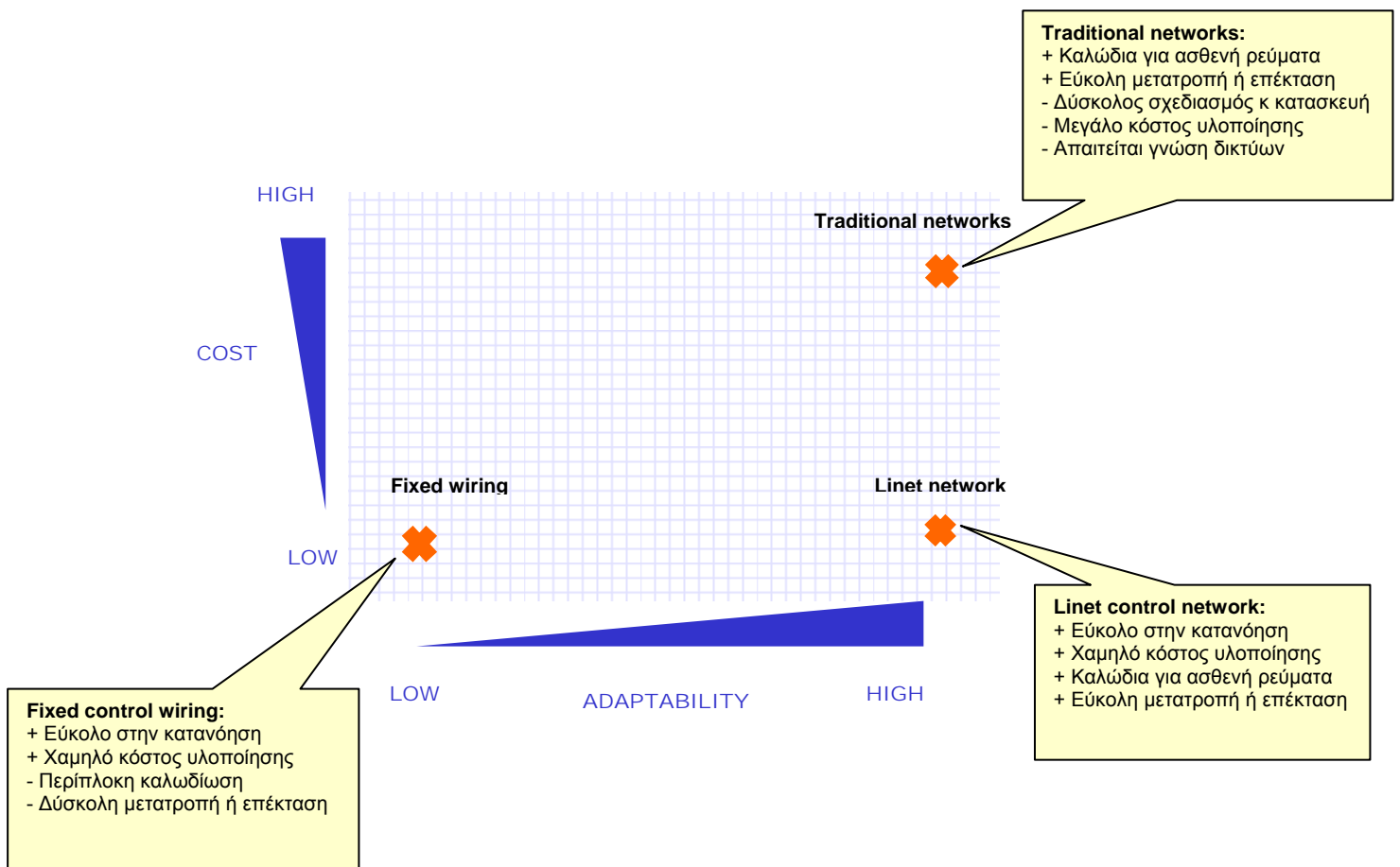
# ΚΕΦΑΛΑΙΟ 1

## 1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ LINET

Το Linet είναι ένα εύκολο, αξιόπιστο και χαμηλού κόστους τοπικό λειτουργικό δίκτυο. Χρησιμοποιείται για τον έλεγχο απλών συσκευών όπως ρελέ, αισθητήρων κλπ χωρίς να χρειάζεται να αναπτυχθούν ειδικές εφαρμογές λογισμικού ή να επενδύσουμε στην δημιουργία άλλων βοηθητικών εξαρτημάτων. Έτσι λοιπόν μια εφαρμογή μπορεί να αναπτυχθεί γρηγορότερα και με μικρότερο κόστος.

Για να χτίσουμε ένα παραδοσιακό σύστημα κατακεντρωμένου ελέγχου μπορούμε να χρησιμοποιήσουμε είτε καλώδια ισχύος (NYM, NYΥ κλπ) με ρελέ, είτε δίκτυα ελέγχου τελευταίας γενιάς. Στην πρώτη περίπτωση η κατασκευή είναι επίπονη και παρουσιάζει δυσκολίες στην συντήρηση καθώς και στην επέκτασή της. Στην δεύτερη οδηγούμαστε σε ακριβά εργαλεία σχεδίασης και πολύπλοκες λύσεις ακόμα και σε απλές εφαρμογές.

Το Linet είναι μεταξύ των δύο αυτών προσεγγίσεων. Είναι ένα νέο σύστημα σχεδιασμένο να εκπληρώνει την ανάγκη για σύνδεση σε δίκτυο απλών συσκευών. Μπορεί εύκολα κάποιος να κατανοήσει την λειτουργία του και να το χειριστεί ενώ προσφέρει την δυναμικότητα και προσαρμοστικότητα των νέων δικτύων.



## 1.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

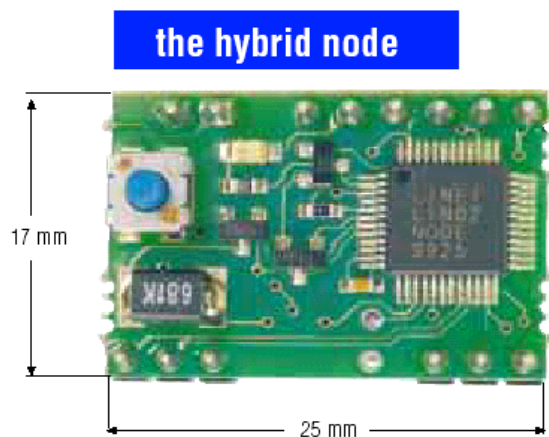
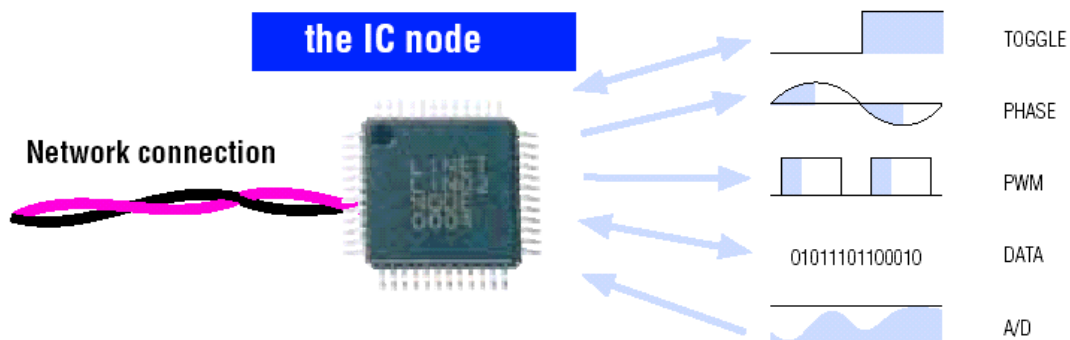
Το Linet αποτελείται από:

- Τους κόμβους (nodes) και
- Την μονάδα ελέγχου (controller)

### 1.2.1 Nodes

Τα nodes είναι μικροί και εύκολοι στην χρήση προσαρμογείς του δικτύου που θέλουμε να ελέγξουμε, με το Linet. Παρέχουν στο σύστημά μας δικτυακή διασύνδεση (network interface), ισχύ (3 mA – 5 V) και έτοιμες I/O λειτουργίες.

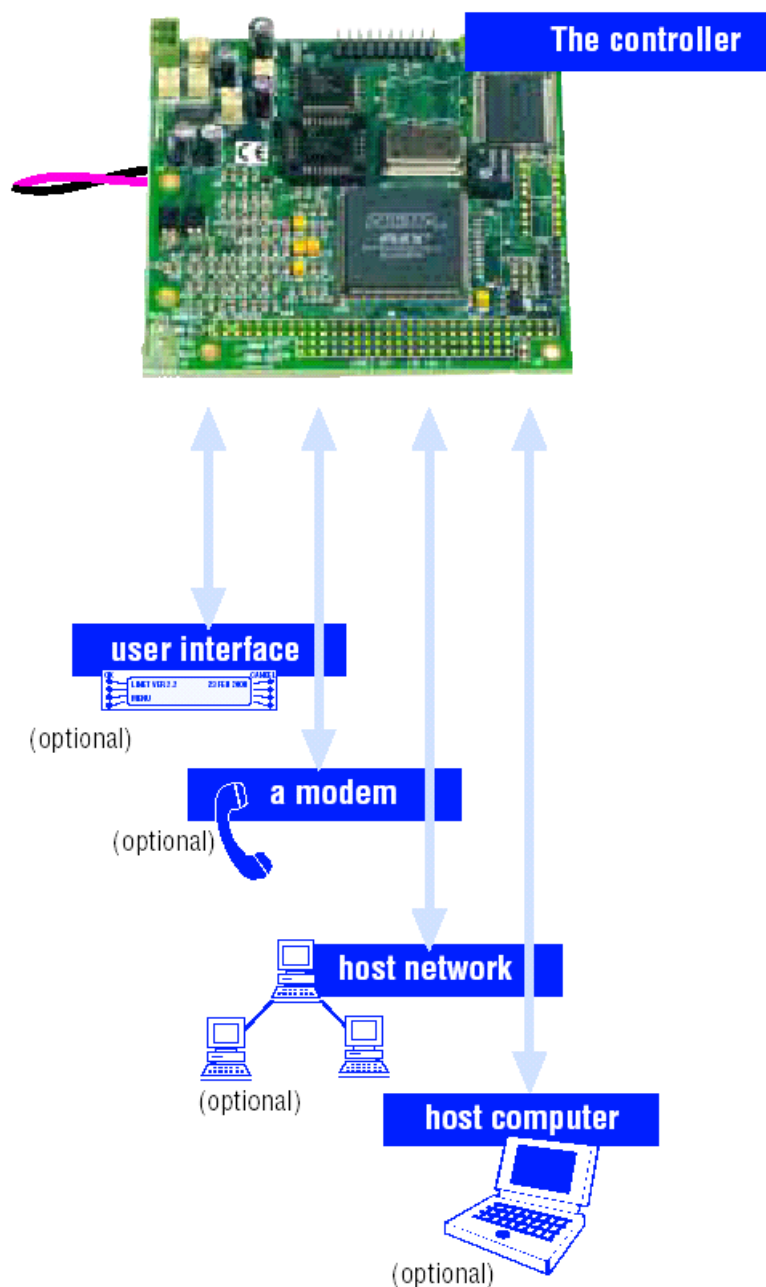
Οι λειτουργίες αυτές είναι συγκεκριμένες και διαθέσιμες σε κάθε node. Επιλέγονται χωριστά για το καθένα κατά την διαδικασία διαμόρφωσης του δικτύου (configuration). Πολλές εφαρμογές μπορούν να ολοκληρωθούν χρησιμοποιώντας μόνο μερικές από αυτές τις λειτουργίες, έτσι είναι έτοιμες χωρίς να χρειαστεί κάποιο software. Αν κάποια εφαρμογή χρειαστεί κάτι παραπάνω το οποίο δεν υποστηρίζεται από τις λειτουργίες του node, η υλοποίηση μπορεί να γίνει με μικροελεγκτή συνδεδεμένο σε κάποιο node. (Μια από τις λειτουργίες είναι και η μεταφορά δεδομένων)



## 1.2.2 Controller

Ο controller είναι μια μονάδα προηγμένης λογικής, η οποία παρέχει ισχύ στα nodes του δικτύου. Είναι ο συνδετικός κρίκος μεταξύ των nodes στο δίκτυο ή μεταξύ των nodes κ ενός κεντρικού υπολογιστή (host). Στο εσωτερικό του τρέχει το λειτουργικό πρόγραμμα το οποίο παρέχει υπηρεσίες διαμόρφωσης του δικτύου (configuration), εξαλείφοντας την ανάγκη για ειδικά εργαλεία ή προγράμματα. Επίσης μπορεί να γίνει αναβάθμιση του λειτουργικού με νεότερες εκδόσεις αν αυτό κριθεί απαραίτητο.

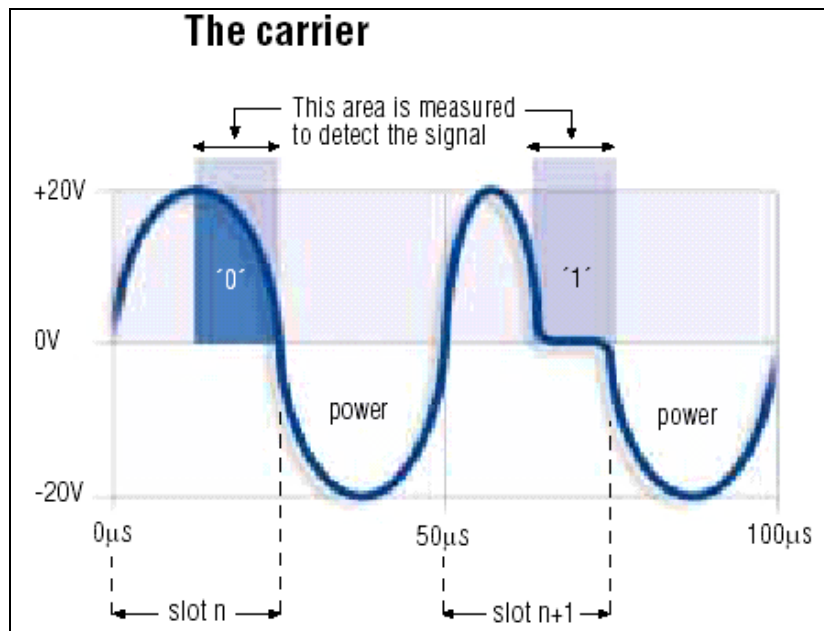
Αφού λοιπόν στήσουμε το δίκτυο μας, το τελευταίο στάδιο που περιλαμβάνει την διαμόρφωση μπορεί να γίνει απλά με έναν Η/Υ που τρέχει κάποιο πρόγραμμα terminal ή με μια LCD οθόνη με μπουτόν που συνδέεται σε μια ειδική θήρα του controller. Με το terminal συνδεόμαστε στον controller και μπαίνουμε στο λειτουργικό του. Από εκεί εύκολα και γρήγορα μπορεί να γίνει ο προγραμματισμός του δικτύου μας.



### 1.3 ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ

Το πρωτόκολλο επικοινωνίας που χρησιμοποιεί το Linet είναι όμοιο με αυτό που χρησιμοποιείται στο σύστημα GSM. Στο GSM όλα τα κινητά τηλέφωνα που είναι συνδεδεμένα στην ίδια βάση, μοιράζονται την ίδια συχνότητα και χρησιμοποιούν το time division protocol (πρωτόκολλο επιμερισμού χρόνου). Ομοίως στο Linet όλα τα nodes μοιράζονται τους ίδιους πόρους (το καλώδιο που τα συνδέει μεταξύ τους) χρησιμοποιώντας το time division protocol και όλες οι πληροφορίες μεταξύ των nodes διακινούνται από τον controller.

Το Linet λειτουργεί στις ακουστικές συχνότητες (20 KHz) και όλα τα σήματα του είναι ημιτονοειδή ώστε να υπάρχει προστασία από τον θόρυβο. Τα δεδομένα και η ισχύς κατανέμονται σε χωριστά κομμάτια της κυματομορφής, έτσι τυχόν αλλαγές στην τροφοδοσία του συστήματος δεν επηρεάζουν την αξιοπιστία των δεδομένων. Στην εικόνα 1 έχουμε την κυματομορφή του σήματος.



Εικόνα 1: Κυματομορφή σήματος

## ΚΕΦΑΛΑΙΟ 2

Έπειτα από αυτή τη γενική περιγραφή για το τι είναι το Linet (τρόπος λειτουργίας και βασικά χαρακτηριστικά), σε αυτό το κεφάλαιο παραθέτω περισσότερες λεπτομέρειες με βάση το σύστημα που επεξεργάστηκα.

### **2.1 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ NODE LIN02**

Όπως αναφέραμε και προηγουμένως τα nodes είναι οι σύνδεσμοι του δικτύου μας με το Linet και έχουν ενσωματωμένες κάποιες λειτουργίες (functions). Ο μέγιστος αριθμός των nodes που μπορούμε να συνδέσουμε στο δίκτυο μας είναι 200 και το μέγιστο μήκος του καλωδίου που θα χρησιμοποιήσουμε δεν πρέπει να ξεπερνάει τα 1000 μέτρα. Το καλώδιο αυτό είναι ένα διπολικό καλώδιο (twisted-pair) το οποίο μεταφέρει πληροφορίες και δεδομένα μεταξύ του controller και των nodes. Η σύνδεση είναι ανεξάρτητη της πολικότητας και της τοπολογίας, επίσης δεν χρειάζονται τερματικές αντιστάσεις.

Τα nodes έχουν προγραμματισμένες κάποιες λειτουργίες τις οποίες μπορούμε να επιλέξουμε κατά το configuration (μια λειτουργία για κάθε node ή ομάδα nodes). Η μόνη παράμετρος που χρειάζεται να προγραμματίσουμε εμείς είναι η δικτυακή διεύθυνση (network address) κάθε node. Η διεύθυνση αυτή είναι ένας ακέραιος από 1 μέχρι 200 και αποθηκεύεται σε μια μικρή μνήμη EEPROM που υπάρχει για αυτό το σκοπό.

Οι βασικότερες λειτουργίες των nodes είναι:

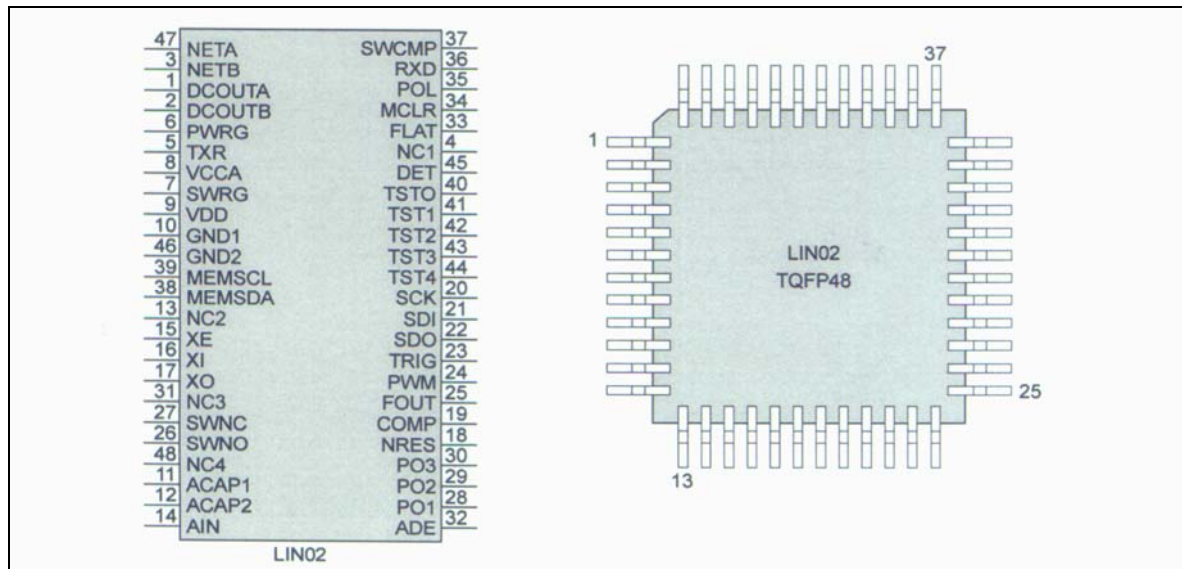
- Διακόπτης εισόδου και εξόδου (on/off)
- Pwm (διαμόρφωση εύρους παλμών) για DC φορτία
- Αναλογική είσοδος
- Σειριακή μετάδοση δεδομένων

Τα nodes είναι διαθέσιμα σε δύο εκδόσεις, στην μια έχουμε μόνο το ολοκληρωμένο κύκλωμα (IC node) σε TQFP48 package, για το οποίο θα χρειαστούμε μερικά ακόμα εξαρτήματα ώστε να λειτουργήσει σωστά. Στην δεύτερη έκδοση (hybrid) έχουμε το ολοκληρωμένο, τα απαραίτητα για την λειτουργία του εξαρτήματα μαζί με ένα ενδεικτικό led και ένα μπουτόν σε DIL20W package. Είναι δηλαδή μια έτοιμη για χρήση plug-and-play κάρτα δικτύου. Στην εργασία μου χρησιμοποίησα hybrid nodes με το ολοκληρωμένο LIN02.



### 2.1.1 Το node σε μορφή ολοκληρωμένου

Στην εικόνα 1 βλέπουμε την διάταξη των Pin του IC-Node και στην συνέχεια ακολουθεί η περιγραφή τους (πίνακας 1).



Εικόνα 1: Διάταξη των Pin του IC-Node

Πίνακας 1: Περιγραφή των Pin του IC-Node

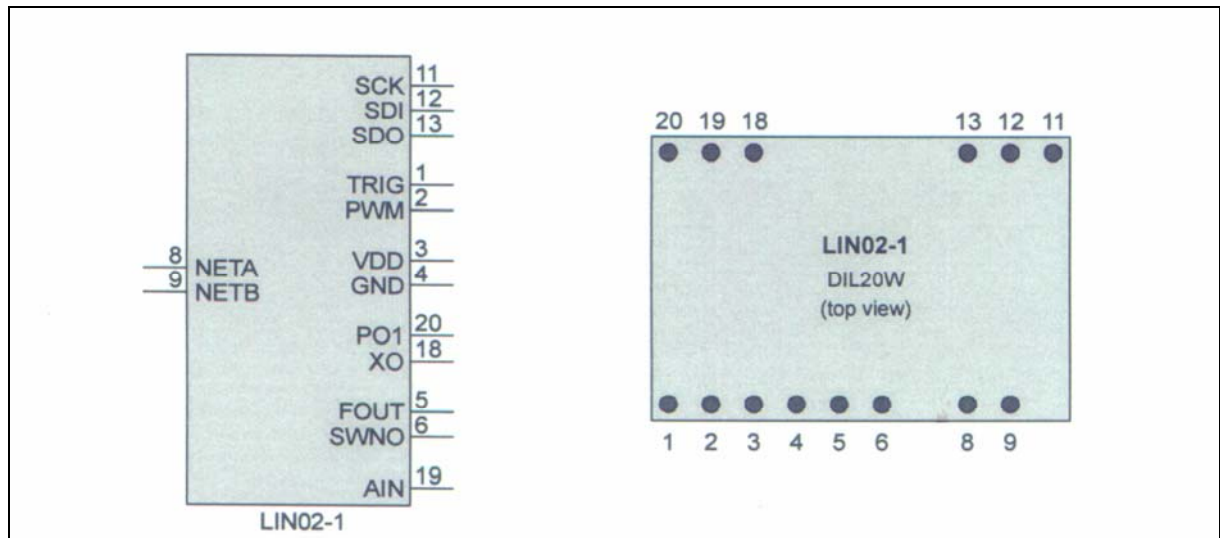
SYMBOL	PIN NUMBER	TYPE	FUNCTION
NETA	47	IN/OUT	Σύνδεση με το δίκτυο (μέσω twisted-pair καλωδίου)
NETB	3	IN/OUT	Σύνδεση με το δίκτυο (μέσω twisted-pair καλωδίου)
SDO	22	OUTPUT	Σειριακή έξοδος δεδομένων
SDI	21	INPUT	Σειριακή είσοδος δεδομένων
SCK	20	OUTPUT	Ρολόι συγχρονισμού σειριακών δεδομένων (clock)
PWM	24	OUTPUT	Διαμόρφωση εύρους παλμών
TRIG	23	INPUT	Σκανδαλισμός για έλεγχο γωνίας φάσης (AC load)
GND1	10	-	Γείωση
GND2	46	-	Γείωση
FOUT	25	OUTPUT	Διακόπτης εξόδου
SWNO	26	INPUT	Διακόπτης εισόδου
XO	17	OUTPUT	Έξοδος ρολογιού
AIN	14	INPUT	Αναλογική είσοδος
PO1	28	OUTPUT	Προγραμματιζόμενη έξοδος 1

SYMBOL	PIN NUMBER	TYPE	FUNCTION
PO2	29	OUTPUT	Προγραμματιζόμενη έξοδος 2
PO3	30	OUTPUT	Προγραμματιζόμενη έξοδος 3
ADE	32	RESERVED	Ενεργοποίηση μετατροπέα A/D (Analog to Digital)
DCOUTA	1	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
DCOUTB	2	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
PWRG	6	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
TXR	5	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
VCCA	8	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
SWRG	7	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
VDD	9	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
MEMSCL	39	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
MEMSDA	38	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
XE	15	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
XI	16	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
SWNC	27	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
ACAP1	11	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
ACAP2	12	INTERNAL	Συνδέεται σε εξωτερικά εξαρτήματα
SWCMP	37	RESERVED	Χωρίς σύνδεση
RXD	36	RESERVED	Χωρίς σύνδεση
POL	35	RESERVED	Χωρίς σύνδεση
MCLR	34	RESERVED	Χωρίς σύνδεση
FLAT	33	RESERVED	Χωρίς σύνδεση
DET	45	RESERVED	Χωρίς σύνδεση
TST0	40	RESERVED	Χωρίς σύνδεση
TST1	41	RESERVED	Χωρίς σύνδεση
TST2	42	RESERVED	Χωρίς σύνδεση
TST3	43	RESERVED	Χωρίς σύνδεση
TST4	44	RESERVED	Χωρίς σύνδεση
COMP	19	RESERVED	Χωρίς σύνδεση
NRES	18	RESERVED	Χωρίς σύνδεση
NC1	4	RESERVED	Χωρίς σύνδεση
NC2	13	RESERVED	Χωρίς σύνδεση
NC3	31	RESERVED	Χωρίς σύνδεση
NC4	48	RESERVED	Χωρίς σύνδεση

### 2.1.2 Το node σε μορφή hybrid

Όπως αναφέραμε και προηγουμένως το hybrid περιέχει το IC-Node (ολοκληρωμένο), τα απαραίτητα για την λειτουργία του εξαρτήματα, ένα ενδεικτικό led και ένα μπουτόν σε DIL20W package.

Στις εικόνες που ακολουθούν έχουμε την διάταξη των Pin του hybrid (εικόνα 2), την περιγραφή τους και τα εξαρτήματα από τα οποία αποτελείται (πίνακας 2), καθώς και το πώς αυτά είναι συνδεδεμένα (εικόνα 3).



Εικόνα 2: Διάταξη των Pin του hybrid

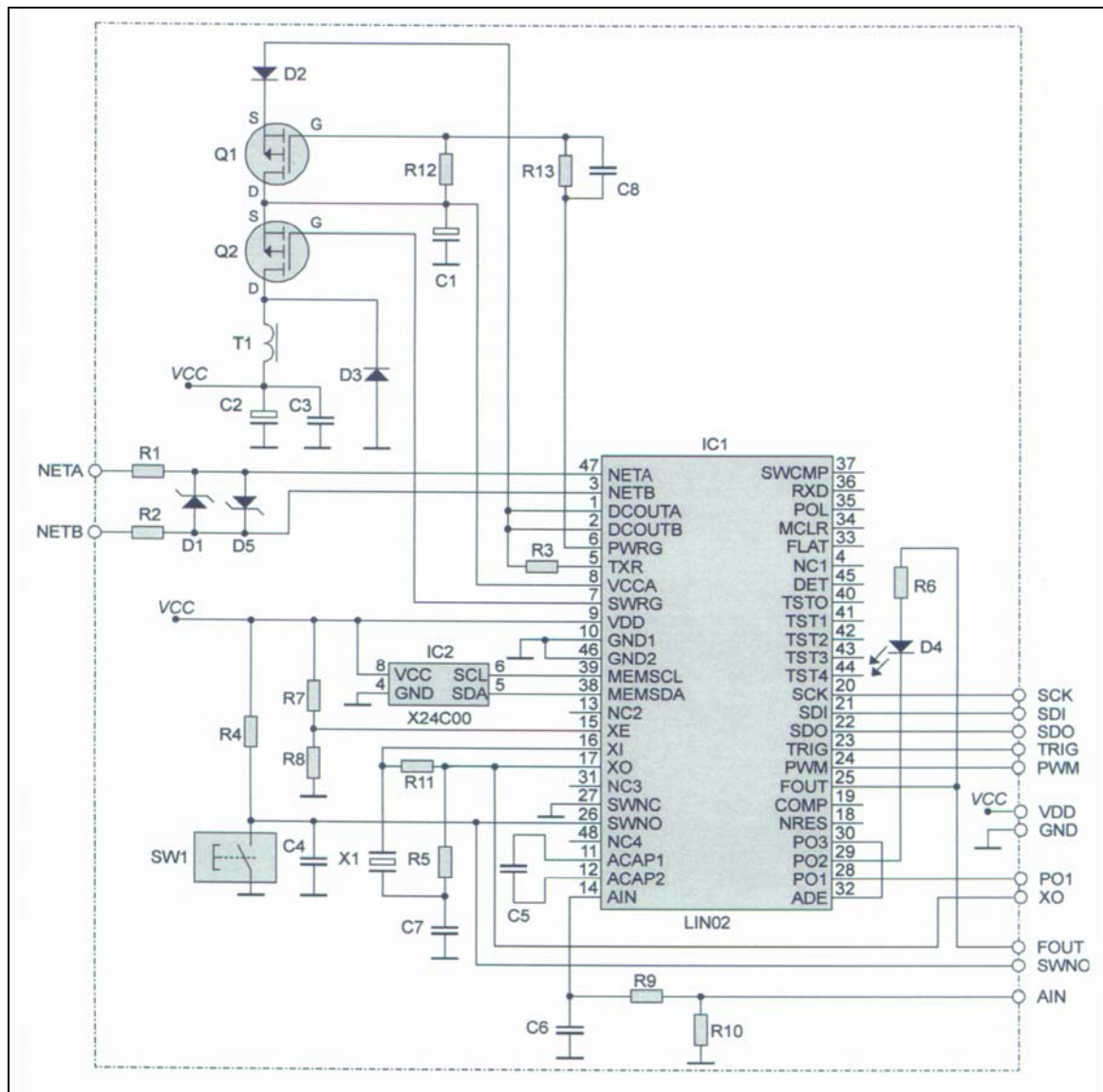
Πίνακας 2: Περιγραφή των Pin του hybrid

SYMBOL	PIN NUMBER	TYPE	FUNCTION
NETA	8	IN/OUT	Σύνδεση με το δίκτυο (μέσω twisted-pair καλωδίου)
NETB	9	IN/OUT	Σύνδεση με το δίκτυο (μέσω twisted-pair καλωδίου)
SDO	11	OUTPUT	Σειριακή έξοδος δεδομένων
SDI	12	INPUT	Σειριακή είσοδος δεδομένων
SCK	13	OUTPUT	Ρολόι συγχρονισμού σειριακών δεδομένων (clock)
PWM	2	OUTPUT	Διαμόρφωση εύρους παλμών
TRIG	1	INPUT	Σκανδαλισμός για έλεγχο γωνίας φάσης (AC load)
+5VOUT	3	-	Θετική πηγή τάσης 5 volt
GND	4	-	Γείωση
FOUT	5	OUTPUT	Διακόπτης εξόδου
SWNO	6	INPUT	Διακόπτης εισόδου
XO	18	OUTPUT	Έξοδος ρολογιού
PO1	20	OUTPUT	Προγραμματιζόμενη έξοδος 1
AIN	19	INPUT	Αναλογική είσοδος

#### Σημείωση:

- Στα Pin NETA και NETB συνδέουμε το καλώδιο του δικτύου μας (twisted pair). Είναι δηλαδή τα Pins που μεταφέρουν τα δεδομένα και την ισχύ μεταξύ των nodes και του controller. Τίποτα άλλο λοιπόν πέρα από το καλώδιο του δικτύου δεν θα πρέπει να συνδέεται εκεί.
- Τα pin SDO, SDI και SCK χρησιμοποιούνται σε εφαρμογές όπου τα nodes στέλνουν δεδομένα στο δίκτυο. Συνήθως με την βοήθεια κάποιου μικροελεγκτή.

- Το Pin PWM χρησιμοποιείται για το έλεγχο με διαμόρφωση εύρους παλμών DC φορτίων. Σε AC φορτία, το Pin TRIG χρησιμοποιείται για τον συγχρονισμό της εξόδου PWM με το φορτίο.
- Τα Pin +5VOUT και GND παρέχουν τάση σε κάποιο βοηθητικό κύκλωμα που ίσως χρησιμοποιήσουμε (π.χ. μικροελεγκτή). Πρέπει όμως να προσέξουμε να μην υπερφορτώσουμε το node.
- Τα pin FOUT και SWNO είναι διακόπτες εξόδου και εισόδου αντίστοιχα. Έτσι το FOUT μπορεί να συνδεθεί στο πηνίο ενός ρελέ ενώ το SWNO μπορεί να συνδεθεί σε διακόπτη, μπουτόν ή στις επαφές ενός ρελέ. Όπως μπορούμε να δούμε στο σχηματικό διάγραμμα του hybrid που ακολουθεί (εικόνα 3), το μπουτόν που υπάρχει σε αυτό συνδέεται παράλληλα με κάθε εξωτερικό διακόπτη που θα συνδέσουμε στο SWNO. Επίσης, το ενδεικτικό led συνδέεται παράλληλα με κάθε φορτίο που θα συνδέσουμε στο FOUT.
- Το Pin XO είναι έξοδος ρολογιού και μπορεί να χρησιμοποιηθεί αντί για ταλαντωτή σε κάποιου μικροελεγκτή.
- Το Pin PO1 είναι προγραμματιζόμενη έξοδος.
- Το Pin AIN είναι η είσοδος του μετατροπέα αναλογικό σε ψηφιακό (A/D converter).



Εικόνα 3: Σχηματικό διάγραμμα hybrid

**Πίνακας 3:** Περιγραφή των εξαρτημάτων της εικόνας 3

<b>ΕΞΑΡΤΗΜΑΤΑ</b>	<b>ΕΙΔΟΣ</b>	<b>ΠΡΟΤΕΙΝΟΜΕΝΟΣ ΚΑΤΑΣΚΕΥΑΣΤΗΣ</b>
IC1	LIN02	Linet
IC2	X24C00	Xicor
Q1, 2	IRLML5103	International Rectifier
D1, 5	SL24	Semtech
D2, 3	BAV99	Any
D4	LED, ultrabright	Any
R1, 2	10R0 resistor	Any
R3	390R resistor	Any
R4, 7, 8	100k resistor	Any
R5	39k resistor	Any
R6	3k3 resistor	Any
R9	10k resistor	Any
R10, 11, 12, 13	1M0 resistor	Any
C1	470nF ceramic/tantalum capacitor	Any
C2	2,2μF ceramic/tantalum capacitor	Any
C3, 4, 6	100nF ceramic capacitor	Any
C5, 8	2,2nF ceramic capacitor	Any
C7	33pF ceramic capacitor	Any
X1	1MHz resonator	Any
T1	680 μH inductor	Any

### 2.1.3 Προγραμματιζόμενα bit

Τα nodes έχουν τρεις προγραμματιζόμενες εξόδους PO1, PO2, PO3 οι οποίες ελέγχονται από τα bit Rb4...Rb1 ενός καταχωρητή. Οι τιμές αυτών των bit είναι αποθηκευμένες σε μια EEPROM που έχουν τα nodes και μπορούν να αλλάξουν μόνο κατά την διαμόρφωση του δικτύου μας.

**Πίνακας 4:** Καταχωρητής διαμόρφωσης (configuration register)

CONFIG. WORD	CONFIGURATION BITS				OUTPUTS			ΣΧΟΛΙΑ
	Rb4	Rb3	Rb2	Rb1	PO3	PO2	PO1	
0	0	0	0	0	0	0	0	Προεπιλογή
1	0	0	0	1	0	0	$\overline{\text{FOUT}}$	
2	0	0	1	0	0	$\overline{\text{FOUT}}$	0	
3	0	0	1	1	0	$\overline{\text{FOUT}}$	$\overline{\text{FOUT}}$	
4	0	1	0	0	1	0	0	AIN
5	0	1	0	1	1	0	$\overline{\text{FOUT}}$	
6	0	1	1	0	1	$\overline{\text{FOUT}}$	0	
7	0	1	1	1	1	$\overline{\text{FOUT}}$	$\overline{\text{FOUT}}$	
8	1	0	0	0	0	0	0	
9	1	0	0	1	0	0	1	
A	1	0	1	0	0	1	0	
B	1	0	1	1	0	1	1	
C	1	1	0	0	1	0	0	
D	1	1	0	1	1	0	1	
E	1	1	1	0	1	1	0	
F	1	1	1	1	1	1	1	

Στο hybrid, το PO1 μπορεί να χρησιμοποιηθεί σαν μια μη πτητική περιφερειακή έξοδος (Rb4=high) ή έξοδος  $\overline{\text{FOUT}}$ . Το PO2 απενεργοποιεί το ενδεικτικό led που υπάρχει στο hybrid όταν γίνει high και τέλος το PO3 ενεργοποιεί τον A/D converter όταν γίνει high.

## 2.2 ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ NODE LIN02

<b>Βασικοί παράμετροι</b>	
Τάση Δικτύου	40 V <sub>pp</sub> (max)
DC test voltage	22 V <sub>DC</sub> (max)
Μέση καταναλισκόμενη ισχύ	4 mW
Μέγιστο ρεύμα εξόδου του IC	3 mA (max)
Τάση εξόδου	5 V
Συχνότητα στο XO pin	1 MHz (approx.)
Θερμοκρασία λειτουργίας του IC	-25...70 °C
Θερμοκρασία λειτουργίας του hybrid	0...70 °C
<b>Λειτουργία Toggle</b>	
Καθυστέρηση μεταφοράς σήματος	50/25/12,5 msec (max)
<b>Λειτουργία Dimmer</b>	
Αριθμός βημάτων (steps)	32
Καθυστέρηση μεταφοράς σήματος, step-up και step down signal	62,5/31,25/15,625 msec (max)
Καθυστέρηση μεταφοράς σήματος, on και off signal	50/25/12,5 msec (max)
Συχνότητα PWM (χωρίς σκανδαλισμό)	100 Hz
<b>Μεταφορά δεδομένων</b>	
Δεδομένα που μεταφέρονται σειριακά (full duplex)	80/160/320 bits/sec
<b>A/D converter</b>	
Ανάλυση	12 bits
Όρια εξόδου	0x000...0xFFE
Όρια τάσης εισόδου	0...+1,25 V
Γραμμικότητα	0,1 %
Μη βαθμολογημένη ακρίβεια	±4 %
Συντελεστής θερμοκρασίας	80 ppm/K

### Σημείωση:

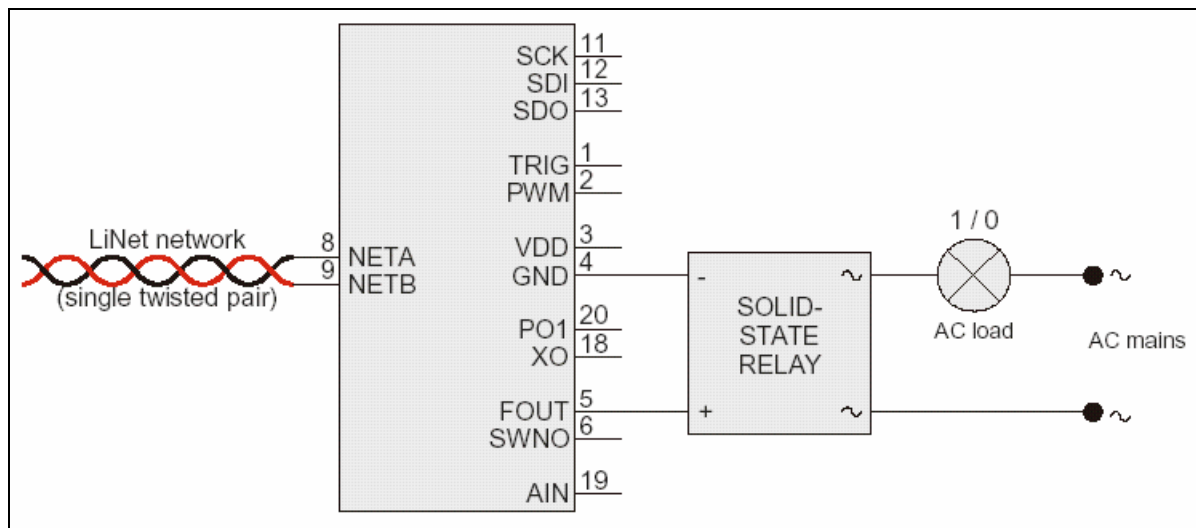
- Το καλώδιο που θα πρέπει να χρησιμοποιούμε είναι ένα απλό διπολικό (twisted pair). Τίποτα άλλο πέρα από τον controller και τα nodes (από 1 μέχρι 200) δεν θα πρέπει να συνδέεται εκεί.
- Θα πρέπει να αποφεύγουμε να υπερφορτώνουμε τα nodes ζητώντας ρεύμα παραπάνω από αυτό που αντέχουν. Ίσως το σύστημα μας να δουλεύει υπερφορτωμένο με μικρό αριθμό από nodes αλλά σίγουρα θα υπάρξουν προβλήματα σε ένα μεγαλύτερο σύστημα. Κάθε hybrid μπορεί να μας δώσει ρεύμα 2 mA, αν όμως απενεργοποιήσουμε το ενδεικτικό led μπορούμε να έχουμε ρεύμα 3 mA.
- Όταν σε κάποια εφαρμογή έχουμε γραμμές ισχύος, τα nodes και οι συσκευές που πρόκειται να χρησιμοποιηθούν θα πρέπει να απομονώνονται γαλβανικά.

## 2.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΦΑΡΜΟΓΩΝ

### 2.3.1 Toggle group

Τα nodes μπορούν να κάνουν έλεγχο σε γραμμές ισχύος χρησιμοποιώντας ρελέ. Μπορούμε λοιπόν να ελέγξουμε κάθε τάση ή ρεύμα με το κατάλληλο ρελέ. Στην εικόνα 4 έχουμε ένα node που ελέγχει ένα AC φορτίο.

Το FOUT του node συνδέεται στο πηνίο του ρελέ και στις επαφές συνδέουμε το φορτίο μας. Μεταξύ των SWNO και GND μπορούμε να συνδέσουμε ένα μπουτόν, έτσι όταν το SWNO ανιχνεύσει rising edge (πατηθεί το μπουτόν) το φορτίο θα αλλάξει κατάσταση (από off θα γίνεται on και αντίστροφα).



Εικόνα 4: Toggle group

### 2.3.2 PWM group

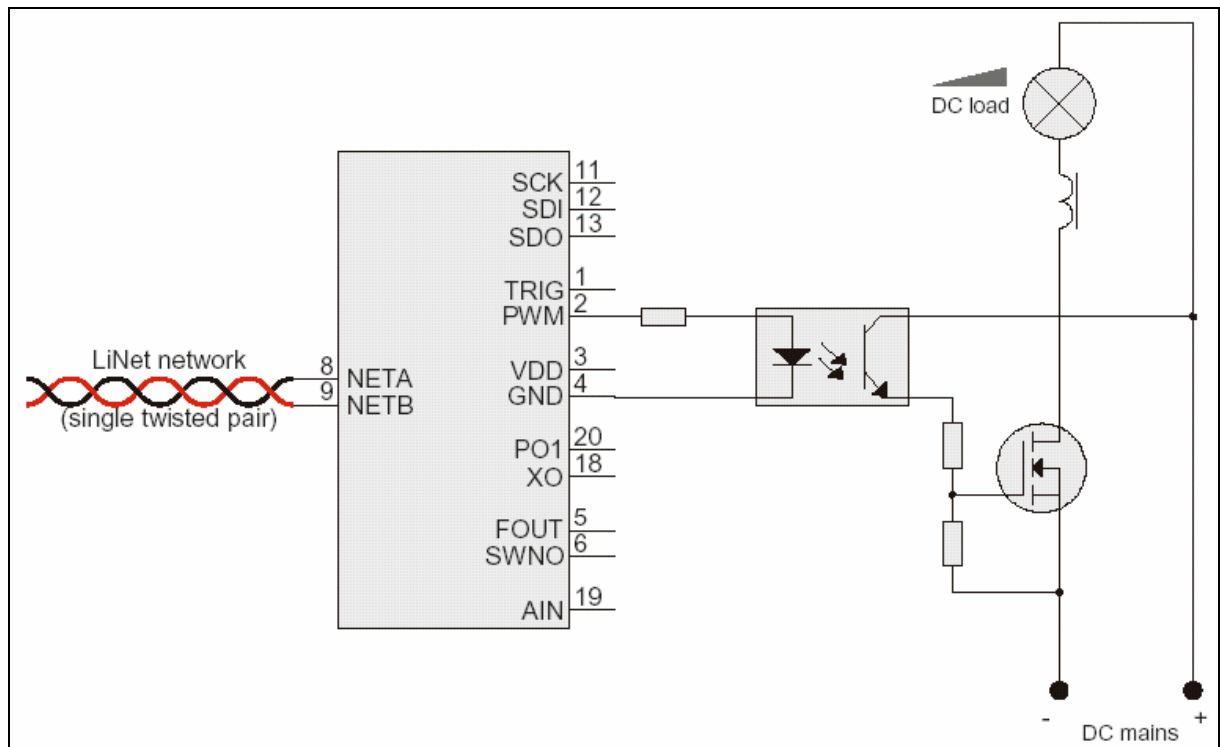
Τα DC φορτία μπορούν να ελεγχθούν με την λειτουργία PWM (διαμόρφωση εύρους παλμών). Για να χρησιμοποιήσουμε την λειτουργία PWM, κατά την διαμόρφωση του συστήματός μας δημιουργούμε ένα Dimmer group.

Όταν το μπουτόν του hybrid (ή κάποιο άλλο εξωτερικό μπουτόν) πατηθεί ελαφρά, το φορτίο αλλάζει κατάσταση. Αν είναι off γίνεται on και λειτουργεί με 50% duty cycle, ενώ αν είναι on γίνεται off. Σε περίπτωση που θέλουμε να μεταβάλλουμε το duty cycle πατάμε το μπουτόν για περισσότερη ώρα, μέχρι την τιμή που θέλουμε.

Η PWM λειτουργία μπορεί να χρησιμοποιηθεί και για έλεγχο γωνίας φάσης AC φορτίων.

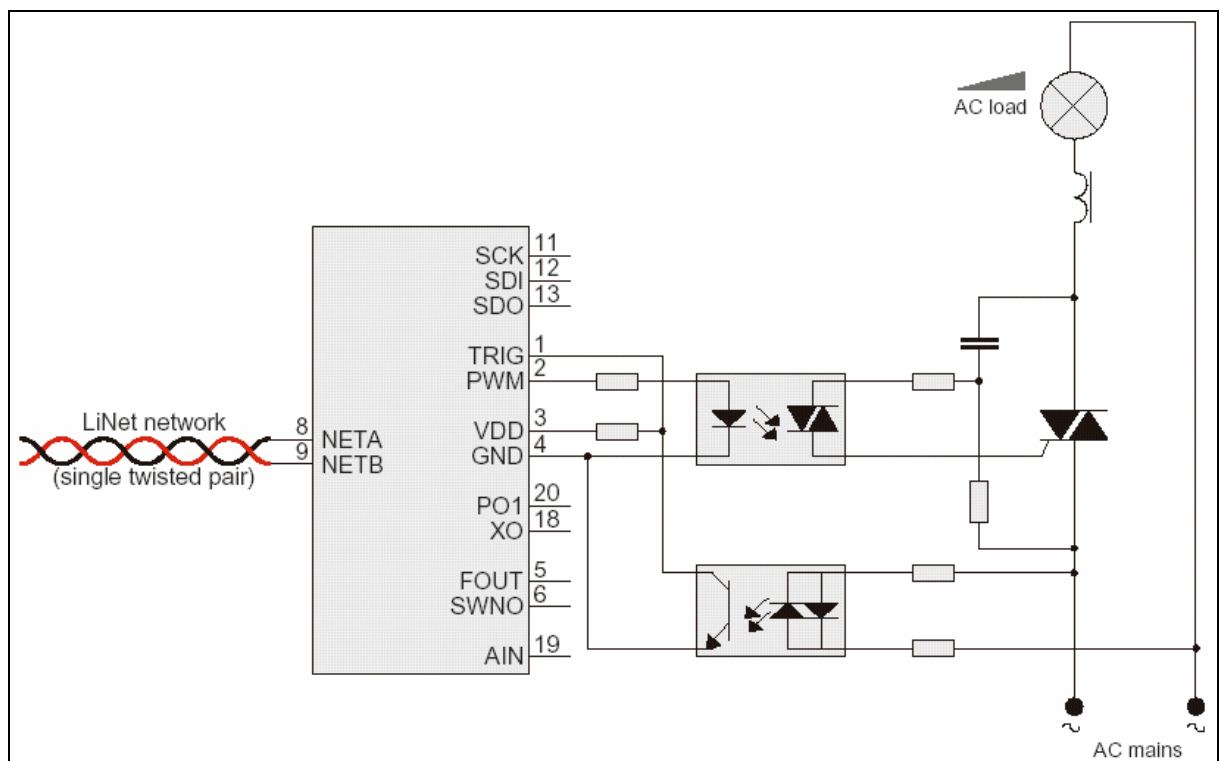
Στην εικόνα 5 έχουμε ένα απλό PWM controller. Το optocoupler (οπτικός απομονωτής) χρησιμοποιείται για να απομονώσει το node από το DC φορτίο και το MOSFET για ενίσχυση.





Εικόνα 5: PWM controller για DC φορτίο

Στην εικόνα 6 έχουμε ένα triac dimmer το οποίο ελέγχεται με node. Το πάνω optocoupler χρησιμοποιείται για να πυροδοτήσει το triac, ενώ το κάτω optocoupler για zero crossing.

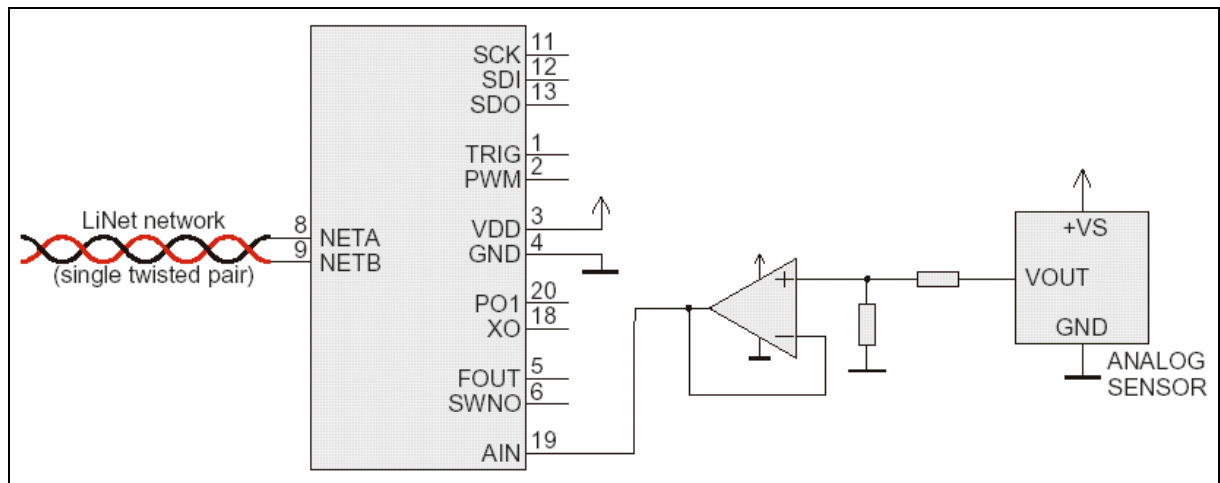


Εικόνα 6: Triac dimmer (AC φορτίο)

### 2.3.3 Analog input

Μια αναλογική τάση στην είσοδο του A/D converter μετατρέπεται σε ψηφιακά δεδομένα 12 bit. Τα δεδομένα αυτά μεταφέρονται σειριακά σε κάποιο άλλο node ή στον controller.

Η λειτουργία αυτή προορίζεται για εφαρμογές που δεν μας ενδιαφέρει και πολύ η ακρίβεια των μετρήσεων, όπως στην μέτρηση θερμοκρασίας ενός δωματίου σπιτιού. Αν όμως θέλουμε μεγαλύτερη ακρίβεια μπορούμε να συνδέσουμε στο node κάποιον άλλο A/D converter, μεγαλύτερης ακρίβειας.



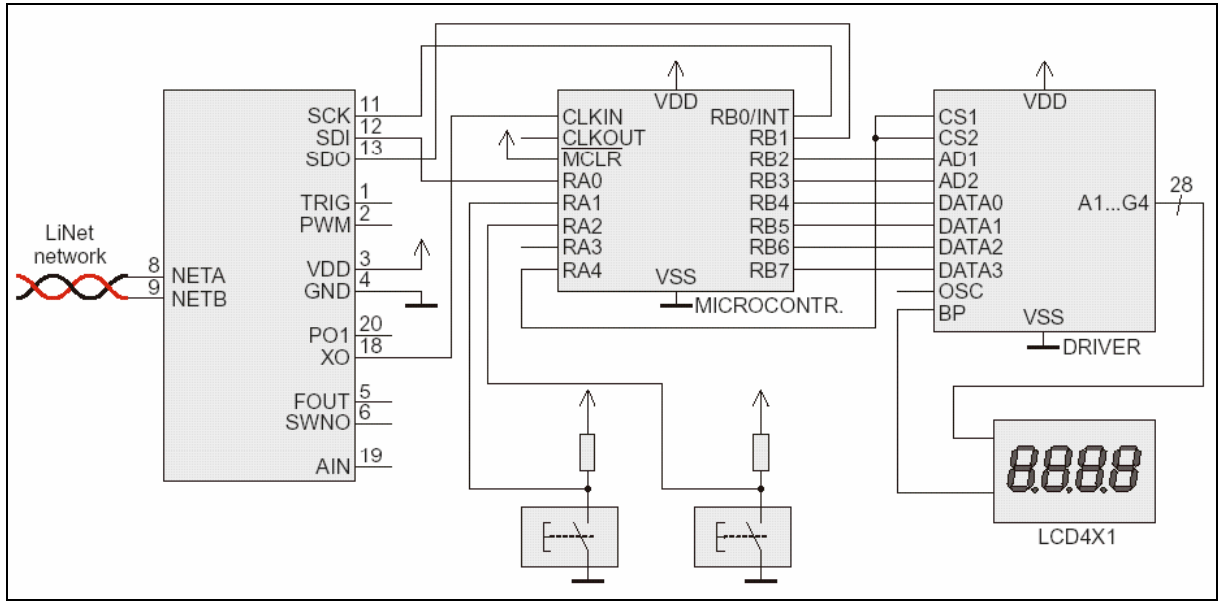
Εικόνα 7: Analog input group

### 2.3.4 Data groups

Όταν σε κάποια εφαρμογή μας χρειάζεται να στέλνουμε δεδομένα μεταξύ των nodes ή μεταξύ nodes και controller τότε χρησιμοποιούμε τα Data groups. Ο ρυθμός μετάδοσης είναι 80 bits/sec, 160 bits/sec, 320 bits/sec ανάλογα με το αν έχουμε συνδέσει 200, 100 ή 50 nodes αντίστοιχα.

Το SCK Pin είναι το ρολόι που συγχρονίζει την μετάδοση των δεδομένων, το SDI Pin είναι η είσοδος και το SDO Pin είναι η έξοδος. Αν στην εφαρμογή μας έχουμε και κάποιο μικροελεγκτή, το node του παρέχει τροφοδοσία και ρολόι (από το XO pin).

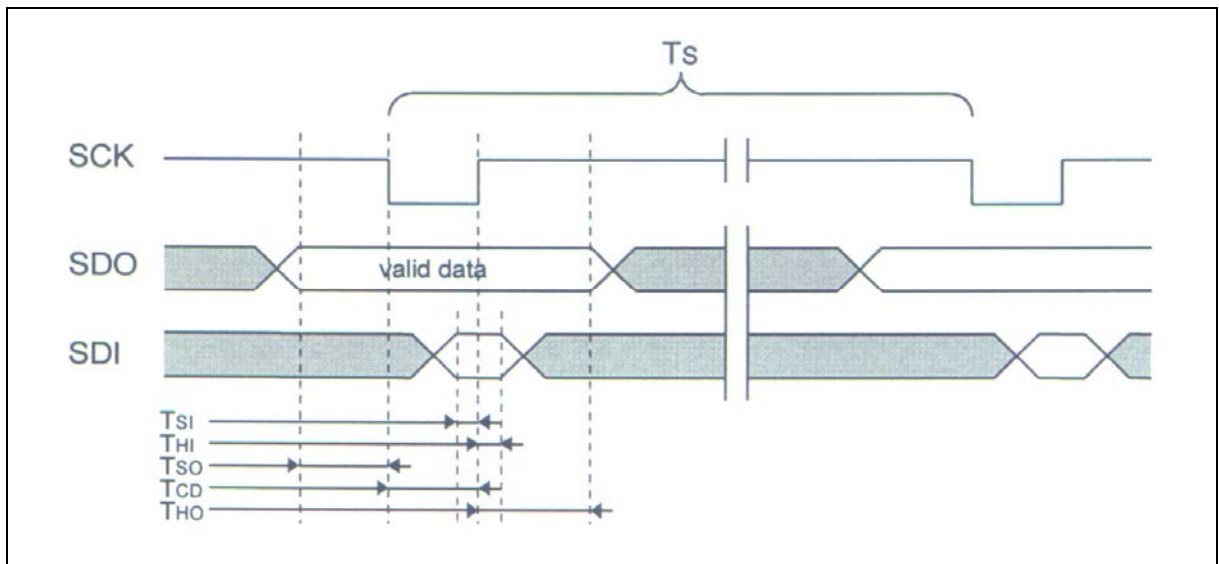
Παρακάτω έχουμε ένα σύστημα θερμοστάτη με την βοήθεια ενός node, μικροελεγκτή, δύο μπουτόν και οθόνη LCD (εικόνα 8). Στην οθόνη βλέπουμε την θερμοκρασία του δωματίου η οποία στέλνεται σειριακά στο node από κάποιο άλλο σημείο του δικτύου. Με τα μπουτόν ρυθμίζουμε την επιθυμητή θερμοκρασία του δωματίου και η τιμή της στέλνεται σειριακά σε κάποιο άλλο σημείο του δικτύου π.χ. για τον έλεγχο του κλιματιστικού.



Εικόνα 8: Data input/output group

Η μετάδοση των δεδομένων συγχρονίζεται από το SCK. Τα εισερχόμενα δεδομένα είναι έγκυρα στο χρονικό διάστημα λίγο πριν, κατά την διάρκεια και λίγο μετά τον συγχρονισμό.

Το node διαβάζει το bit που φεύγει από το SDI την στιγμή που το SCK έχει rising edge. Έτσι η εφαρμογή μας δεν θα πρέπει να μεταβάλει την κατάσταση του SDI κοντά στην χρονική στιγμή που το SCK έχει rising edge.



Εικόνα 9: Διάγραμμα χρονισμού

**Πίνακας 5:** Χρονικά διαστήματα του διαγράμματος χρονισμού (εικόνα 9)

<b>ΣΥΜΒΟΛΟ</b>	<b>ΕΠΕΞΗΓΗΣΗ</b>	<b>ΤΙΜΗ</b>
$T_S$	Περίοδος (εξαρτάται από τον αριθμό των nodes, 200, 100, 50)	12,5/6,25/3,125 msec
$T_{SO}$	Χρονικό διάστημα από έγκυρα δεδομένα στην έξοδο μέχρι το ρολόι να πέσει σε χαμηλή στάθμη	350 $\mu$ sec
$T_{CD}$	Χρονικό διάστημα όπου το ρολόι είναι σε χαμηλή στάθμη	25 $\mu$ sec
$T_{HO}$	Χρονικό διάστημα από υψηλή στάθμη ρολογιού μέχρι έγκυρα δεδομένα στην έξοδο	300 $\mu$ sec
$T_{SI}$	Χρονικό διάστημα από έγκυρη εισαγωγή δεδομένων μέχρι υψηλή στάθμη ρολογιού	10 $\mu$ sec
$T_{HI}$	Χρονικό διάστημα όπου έχουμε έγκυρη εισαγωγή δεδομένων μετά την υψηλή στάθμη ρολογιού	10 $\mu$ sec

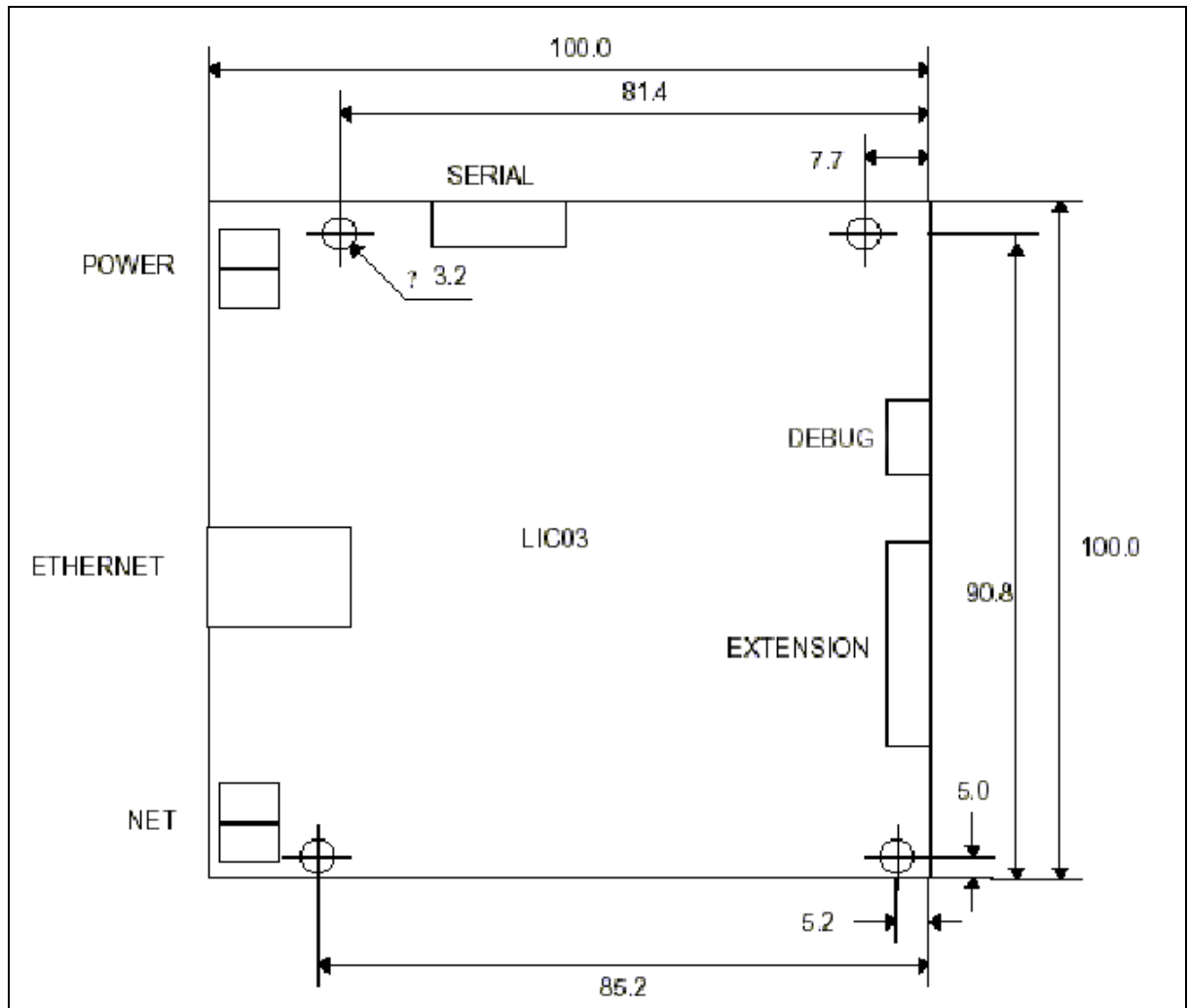
## 2.4 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ CONTROLLER LIC03

Ο controller είναι η καρδιά του δικτύου μας. Παρέχει σε αυτό ισχύ και μας δίνει την δυνατότητα διαμόρφωσης και ελέγχου του. Περιέχει ένα LINET bus driver, μικροελεγκτή RISC με firmware (λογισμικό) για να εκτελεί τις τυπικές δικτυακές λειτουργίες, RS232/RS485 σειριακή διασύνδεση για απομακρυσμένο έλεγχο, ETHERNET διασύνδεση για σύνδεση σε βιομηχανικά Ethernet ή Intranet συστήματα, και ένα bus επέκτασης για τον τοπικό μικροϋπολογιστή. Επίσης, περιέχει ρολόι πραγματικού χρόνου για εφαρμογές που απαιτούν χρόνο.

Το δίκτυο Linet χρησιμοποιεί το πρωτόκολλο επικοινωνίας time division. Όλα τα nodes έχουν το δικό τους κομμάτι χρόνου στην κυματομορφή του σήματος. Κάθε πακέτο δεδομένων (frame) αρχίζει με τον χαρακτήρα συγχρονισμού και ακολουθούν τα bit εξυπηρέτησης συστήματος. Στη συνέχεια ακολουθούν 200 bits δεδομένων, ένα για κάθε node που υπάρχει στο δίκτυο. Το συνολικό μήκος των δεδομένων που στέλνονται είναι 253 bits. Κατά την διάρκεια ενός frame κάθε node μπορεί να στέλνει και να λαμβάνει ένα bit πληροφορίας, επίσης μπορεί να στέλνει και να λαμβάνει 80 bits/second συνεχώς. Η συχνότητα του φέροντος (carrier) είναι 20 kHz έτσι ο full duplex ρυθμός μετάδοσης των δεδομένων είναι 200x80 bits/sec.

## 2.5 ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ CONTROLLER LIC03

<b>Παροχή ισχύος προς το δίκτυο</b>	
Παροχή τάσης	24 ( $\pm 5\%$ ) V <sub>DC</sub>
Μέγιστη παροχή ρεύματος με 200 nodes	0,5 A <sub>DC</sub>
<b>Linnet bus</b>	
Παροχή τάσης	40 V <sub>PP</sub>
Μέγιστη παροχή ρεύματος (200 nodes)	0,7 A (peak)
Μέγιστη κατανάλωση ισχύος (200 nodes)	12 W
Συχνότητα bus, ημιτονοειδές κυματομορφή	20 kHz
Μέγιστος αριθμός nodes	200
Μέγιστο μήκος καλωδίωσης (CAT5E)	1000 m
<b>Σειριακή διασύνδεση</b>	
Τυπική διασύνδεση	RS232
Τυπική ταχύτητα διασύνδεσης	19200 bits/sec
Τυπικές ρυθμίσεις διασύνδεσης (data,stop,parity)	8, 1, N
Προαιρετική διασύνδεση	RS485
Απομόνωση	100 V <sub>DC</sub>
<b>Ethernet διασύνδεση</b>	
Network interface	10BASE-T
Network connector	RJ-45



Εικόνα 10: Διαστάσεις σε χιλιοστά και connectors του controller.

## 2.6 CONNECTIONS

### 2.6.1 Είσοδος ισχύος (J12)

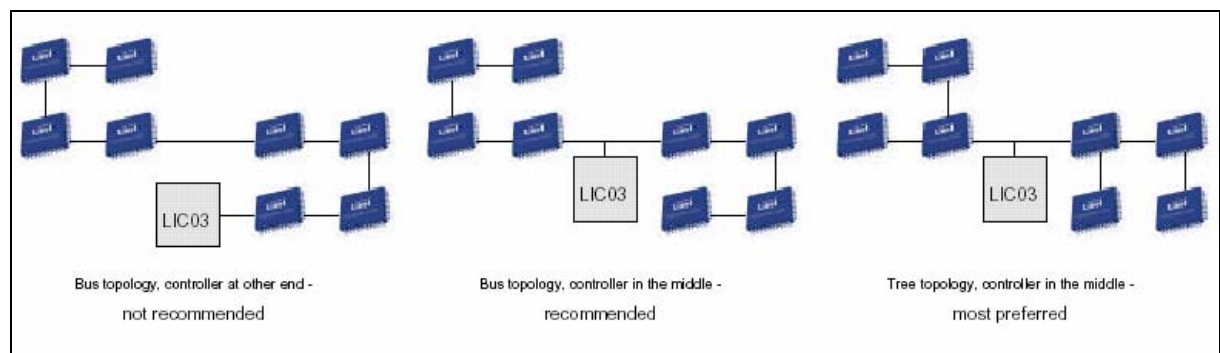
Ο connector J12 είναι η είσοδος ισχύος στον controller. Τα καλώδια που χρησιμοποιούνται δεν χρειάζεται να είναι μεγαλύτερα από  $1.0 \text{ mm}^2$ , πρέπει όμως να δώσουμε προσοχή ώστε να τα συνδέσουμε με σωστή πολικότητα στον controller. Η είσοδος προστατεύεται από μια δίοδο zener, έτσι σε περίπτωση που συνδέσουμε τα καλώδια με ανάστροφη πολικότητα για μικρό χρονικό διάστημα η δίοδος zener προστατεύει τον controller. Αν η λάθος σύνδεση παραμείνει για μεγάλο χρονικό διάστημα ίσως έχουμε καταστροφή του controller και της πηγής ισχύος.

## 2.6.2 Linet network connector (J3)

Ο connector J3 είναι ο σύνδεσμος του δικτύου Linet, δηλαδή συνδέει τον controller με τα nodes που δικτύου δίνοντας ισχύ και δεδομένα. Κάθε διπολικό καλώδιο  $2 \times 0,5 \text{mm}^2$  τηλεφωνικού τύπου unshielded, μπορεί να χρησιμοποιηθεί σαν καλώδιο του δικτύου (προτείνεται τύπου CAT5). Η σύνδεση δεν εξαρτάται από την πολικότητα και την τοπολογία του δικτύου, επίσης δεν χρειάζονται τερματικές αντιστάσεις. Το μέγιστο μήκος του καλωδίου που μπορούμε να χρησιμοποιήσουμε, για να έχουμε σωστή λειτουργία είναι 1000 μέτρα.

Όπως είπαμε και προηγουμένως το δίκτυο δεν εξαρτάται από την τοπολογία, έτσι ο controller μπορεί να τοποθετηθεί σε οποιαδήποτε θέση. Παρόλα αυτά η καλύτερη θέση για τον controller είναι στη μέση του δικτύου ώστε η απόσταση μεταξύ αυτού και των nodes του να μοιράζεται.

Στο καλώδιο του δικτύου Linet δεν πρέπει να συνδέεται τίποτα άλλο πέρα από τον controller και τα nodes. Πρέπει επίσης να είναι απομονωμένο από κάθε άλλο κύκλωμα.



Εικόνα 11: Παραδείγματα σύνδεσης

## 2.6.3 Σειριακή διασύνδεση RS232/485 (J4)

Ο connector J4 είναι η RS232/485 σειριακή διασύνδεση. Η ασύγχρονη σειριακή πόρτα (port) μπορεί να διαμορφωθεί ώστε να λειτουργεί είτε με RS232 είτε με RS485 διασύνδεση. Ένας τυπικός controller χρησιμοποιεί την RS232 διασύνδεση, έτσι τα Pins (7,8) που χρησιμοποιούνται στην RS485 είναι ασύνδετα.

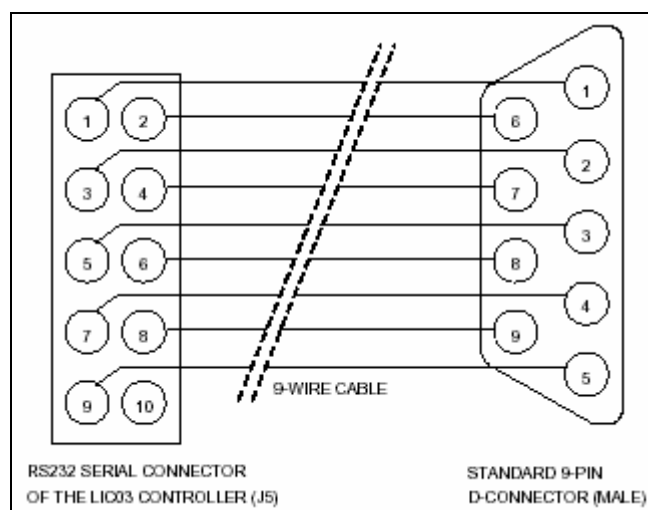
Το λειτουργικό πρόγραμμα του controller χρησιμοποιεί την σειριακή πόρτα σαν βασικό μέσο για τη διαμόρφωση του συστήματός μας. Ο χρήστης με ένα PC που τρέχει κάποιο πρόγραμμα terminal, μπορεί να συνδεθεί στον controller μέσω της σειριακής πόρτας και να διαμορφώσει το σύστημα. Μετά την διαμόρφωση το PC μπορεί να αποσυνδεθεί από τον controller.

Εκτός από την διαμόρφωση του συστήματος, με την σειριακή πόρτα μπορούμε να δούμε και να μεταβάλουμε τις τιμές των nodes. Μπορούμε π.χ. να δούμε αν κάποιο node είναι on ή off, να δούμε την τιμή του A/D converter σε κάποιου node ή να δώσουμε εμείς τιμή on σε κάποιο άλλο node.

Η σειριακή διασύνδεση μπορεί να είναι είτε απομονωμένη είτε μη απομονωμένη. Η απομονωμένη διασύνδεση προτείνεται όταν ο controller είναι διαρκώς συνδεδεμένος σε κάποιο κεντρικό υπολογιστή (host), ενώ η μη απομονωμένη διασύνδεση όταν ο controller συνδέεται στον κεντρικό υπολογιστή μόνο κατά την διαδικασία διαμόρφωσης. Να σημειώσουμε ότι η απομονωμένη διασύνδεση χρησιμοποιείτε μόνο για την μείωση του θορύβου και δεν θεωρείται σκόπιμο να ακολουθηθεί τυποποιημένους κανόνες ασφαλείας.

**Πίνακας 6:** RS232/485 connectors

ΣΥΜΒΟΛΟ	ΑΡΙΘΜΟΣ	ΕΙΔΟΣ	ΟΝΟΜΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ
-	1	-	-
-	2	-	-
RxD	3	OUTPUT	EIA-232
RTS	4	OUTPUT	EIA-232
TxD	5	INPUT	EIA-232
CTS	6	INPUT	EIA-232
D-	7	IN/OUT	EIA-485 (option)
D+	8	IN/OUT	EIA-485 (option)
sGND	9	-	EIA-232/EIA-485 (isolated/un-isolated) Γείωση
Isol. +5V	10	-	Απομονωμένη έξοδος 5V για EIA-485



**Εικόνα 12:** Το σειριακό καλώδιο

Όταν χρησιμοποιούμε την σειριακή διασύνδεση για να συνδέσουμε τον controller με ένα PC, το καλώδιο που θα χρειαστούμε είναι αυτό που φαίνεται στην παραπάνω εικόνα και ονομάζεται null modem cable. Τα Pins 1 και 2 του connector J4 δεν χρησιμοποιούνται καθώς επίσης και τα Pins 7 και 8 που είναι για σύνδεση RS485. Οπότε μπορούν να παραμείνουν ασύνδετα.



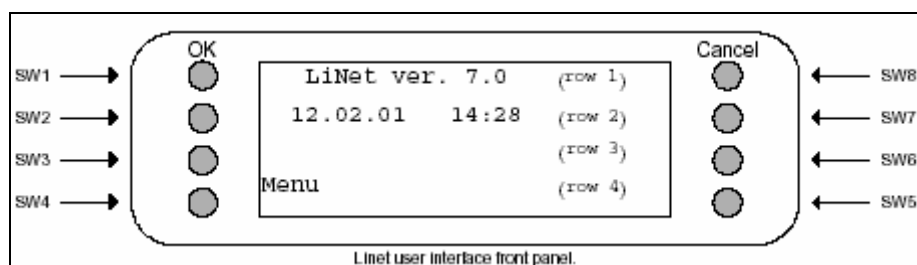
## 2.6.4 Διάυλος επέκτασης (J1)

Ο connector J1 είναι ο διάυλος επέκτασης του μικροελεγκτή (που βρίσκεται στον controller). Σε αυτόν μπορεί να συνδεθεί προαιρετικά ένα user interface που αποτελείται από μια LCD οθόνη και 8 μπουτόν. Η οθόνη είναι 20 χαρακτήρων και τεσσάρων γραμμών και μπορεί να είναι μόνιμα συνδεδεμένη στο controller ή να συνδέεται κατά την διαδικασία διαμόρφωσης του συστήματος.

Ένας τυπικός controller χρησιμοποιεί τον διάυλο επέκτασης σαν δευτερεύον μέσο για την διαμόρφωση του συστήματός μας.

**Πίνακας 7:** Διάυλος επέκτασης μικροελεγκτή

ΣΥΜΒΟΛΟ	ΑΡΙΘΜΟΣ	ΕΙΔΟΣ	ΟΝΟΜΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ
GND	1	-	-
GND	2	-	-
+5V <sub>DC</sub>	3	OUTPUT	-
+5V <sub>DC</sub>	4	OUTPUT	-
WR/	5	OUTPUT	Write signal
RD/	6	OUTPUT	Read signal
CS1/	7	OUTPUT	Chip select signal
CS2/	8	OUTPUT	Chip select signal
RESET/	9	OUTPUT	Reset signal
ALE	10	OUTPUT	Address latch enable signal
AD0	11	I/O	Address/data
AD1	12	I/O	Address/data
AD2	13	I/O	Address/data
AD3	14	I/O	Address/data
AD4	15	I/O	Address/data
AD5	16	I/O	Address/data
AD6	17	I/O	Address/data
AD7	18	I/O	Address/data
-	19	-	Reserved
-	20	-	Reserved



**Εικόνα 13:** Οθόνη LCD με μπουτόν που μπορούμε να χρησιμοποιήσουμε



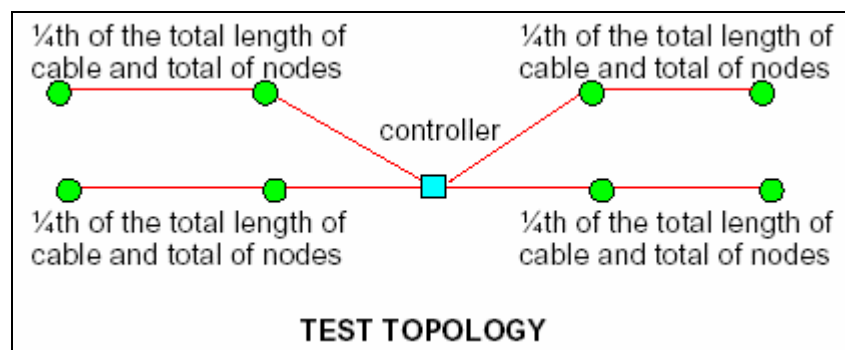
## 2.6.5 Ethernet connector (J7)

Ο controller μπορεί να συνδεθεί σε 10BASE-T δίκτυα Ethernet μέσω του RJ45 connector που υπάρχει. Το λογισμικό του, διαθέτει το πρωτόκολλο telnet για υποστήριξη του συστήματος σε σύνδεση Ethernet. Επίσης, διαθέτει το πρωτόκολλο UDP/IP για μεταφορά δεδομένων του δικτύου Linet μέσα και έξω από αυτό.

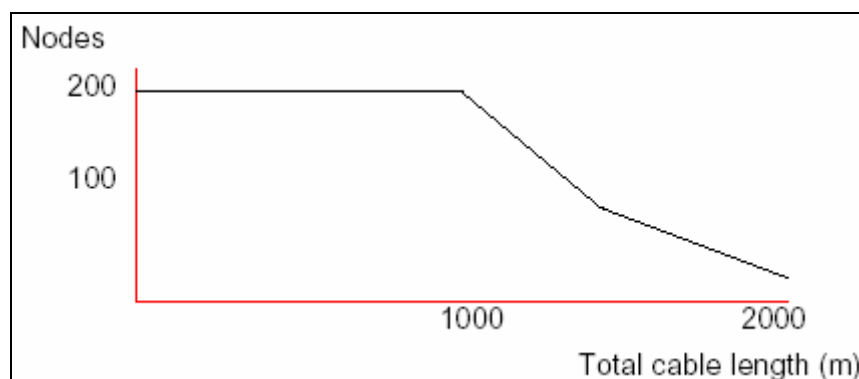
## 2.7 ΜΗΚΟΣ ΔΙΚΤΥΟΥ

Το Linet χρησιμοποιεί ημιτονοειδές φέρον (carrier) χαμηλής συχνότητας για να μεταφέρει δεδομένα και ισχύ στα nodes. Το μέγιστο μήκος του δικτύου έχει μετρηθεί σε bus τοπολογία. Ο συνολικός αριθμός των nodes χωρίστηκε σε τέσσερις ομάδες οι οποίες συνδέθηκαν σε bus τοπολογία και τοποθετήθηκαν σε όλο το μήκος του δικτύου. Τα αποτελέσματα φαίνονται στις παρακάτω εικόνες (15 και 16).

Το καλώδιο που χρησιμοποιήθηκε είναι CAT5E, το οποίο είναι unshielded διπολικό καλώδιο.



Εικόνα 15: Τοπολογία για την ανεύρεση μέγιστου μήκους δικτύου

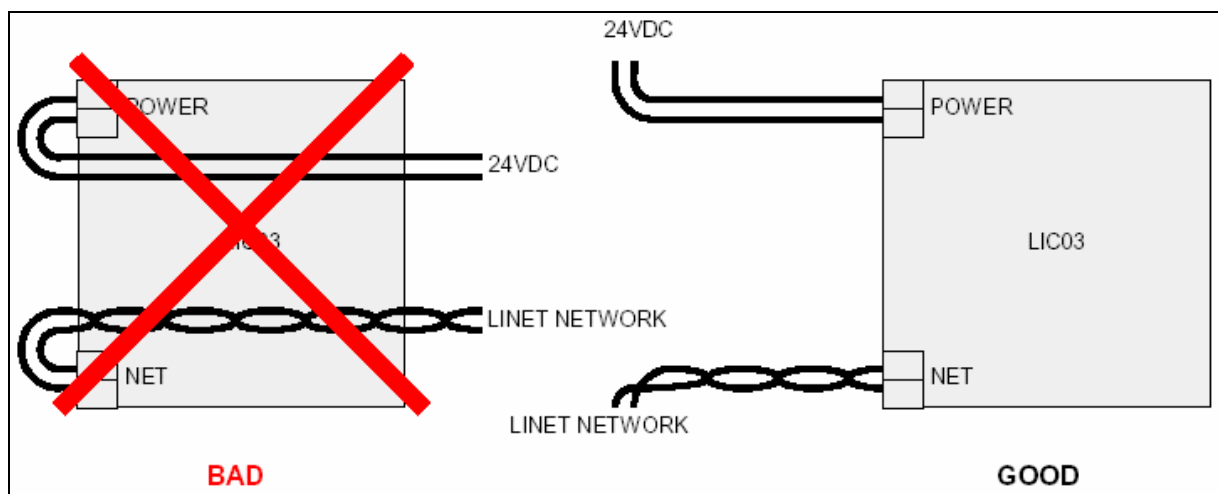


Εικόνα 16: Μέγιστο μήκος δικτύου σε συνάρτηση με τον αριθμό των nodes

## 2.8 ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΗ ΣΥΜΒΑΤΟΤΗΤΑ (EMC)

Στο δίκτυο Linet, η μεγαλύτερη ποσότητα του θορύβου προέρχεται από τον controller. Η πηγή αυτού του θορύβου είναι ο κρύσταλλος του ρολογιού, ο οποίος βρίσκεται στο κέντρο του controller. Για να αποφύγουμε λοιπόν τυχόν παρεμβολές στο σύστημά μας, θα πρέπει η καλωδίωση του δικτύου να είναι όσο πιο μακριά γίνεται από το ρολόι του controller.

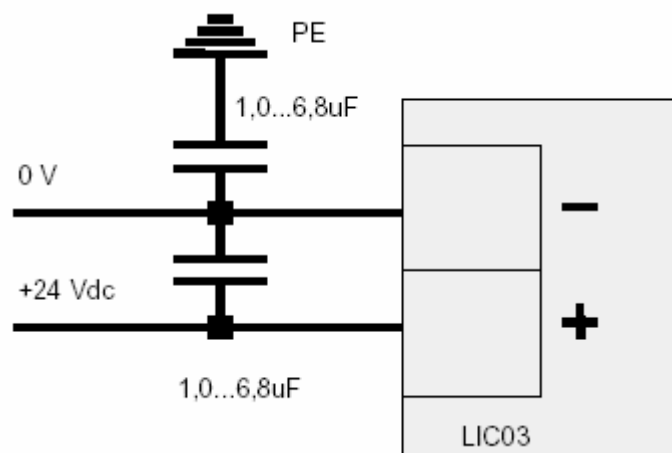
Ο controller δεν είναι απαραίτητο να τοποθετηθεί σε κάποιο μεταλλικό κουτί, αν όμως κάτι τέτοιο χρειαστεί να γίνει, θα πρέπει το κουτί να είναι όσο πιο μικρό γίνεται και να γειωθεί. Επίσης, η καλωδίωση στο εσωτερικό του κουτιού θα πρέπει να είναι όσο το δυνατόν πιο μικρή. Αν θέλουμε να έχουμε ακόμα μεγαλύτερη προστασία από Ηλεκτρομαγνητικές Παρεμβολές (EMI), μπορούμε να χρησιμοποιήσουμε προστατευμένα καλώδια (shielded) στο εσωτερικό του κουτιού και δακτυλίδια σιδηρίτη στο σημείο εισαγωγής του καλωδίου.



Εικόνα 17: Παράδειγμα καλωδίωσης

Πυκνωτές στην είσοδο ισχύος του controller μπορούν να μειώσουν την Ηλεκτρομαγνητική Συμβατότητα προς τα καλώδια παροχής. Οι πυκνωτές που χρειάζονται εξαρτώνται από την εγκατάσταση. Στις περισσότερες περιπτώσεις, τιμές πυκνωτών από 1  $\mu\text{F}$  μέχρι 6,8  $\mu\text{F}$  είναι επαρκής.

Ο controller εκπληρώνει τα όρια EMC που έχουν οριστεί σύμφωνα με τα standards EN5008-1 (για οικιακή χρήση) και EN5008-2 (για βιομηχανική χρήση) χωρίς την χρήση εξωτερικού περιβλήματος (κουτί).



Εικόνα 18: Φίλτρο εισόδου ισχύος

## **ΚΕΦΑΛΑΙΟ 3**

### **3.1 ΠΡΟΕΤΟΙΜΑΣΙΑ ΠΡΙΝ ΤΗ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΔΙΚΤΥΟΥ**

#### **3.1.1 Έλεγχος των nodes με dc τάση**

Όταν τοποθετούμε τα nodes ή συσκευές που περιέχουν nodes στο δίκτυο, είναι καλό να ελέγχουμε και την σωστή λειτουργία τους. Για να γίνει αυτό αποσυνδέουμε τον controller από το δίκτυο και στην θέση του βάζουμε μια dc πηγή τάσης 18 V (μπορούμε να συνδέσουμε και δυο μπαταρίες των 9 V σε σειρά).

Όταν ένα node είναι συνδεδεμένο σε dc τάση και δεν μπορεί να ανιχνεύσει κάποιο ac σήμα (carrier), μπαίνει σε κατάσταση διάγνωσης. Έτσι πατώντας το μπουτόν που διαθέτει, η έξοδος γίνεται on και off σε δεύτερο πάτημα του μπουτόν. Με τον τρόπο αυτό μπορούν να ελεγχθούν γρήγορα όλα τα nodes του συστήματος καθώς και το δίκτυο γενικότερα.

#### **3.1.2 Διαδικασία Lamp test**

Η διαδικασία lamp test είναι ένα χρήσιμο εργαλείο για να ελέγξουμε τη σωστή λειτουργία του δικτύου. Μέσω ενός PC συνδεόμαστε στο λειτουργικό του controller και επιλέγουμε Configure net - Network debug - Set debug command - Set service command - Lamp test. Με την διαδικασία αυτή η έξοδος όλων των node που υπάρχουν στο δίκτυο γίνεται on. Αν κάποιο node είναι συνδεδεμένο στο δίκτυο αλλά δεν έχει διαμορφωθεί, η έξοδος του θα γίνει και πάλι on.

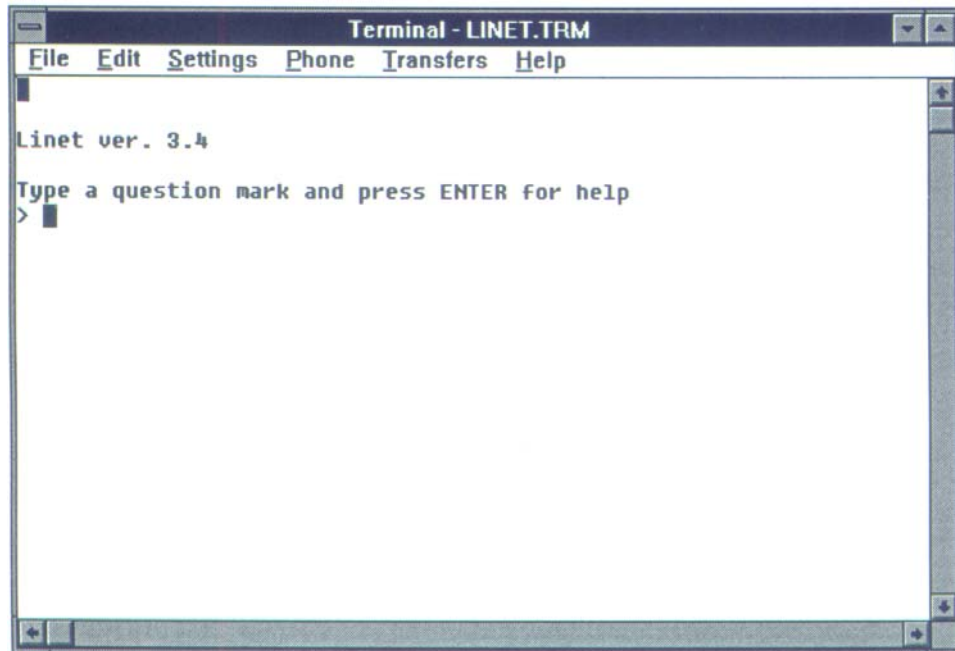
Όταν λοιπόν ελεγχθεί το δίκτυο και θέλουμε να βγούμε από την διαδικασία lamp test, επιλέγουμε Configure net - Network debug - Set debug command - Set service command - None.

### **3.2 ΑΡΧΙΖΟΝΤΑΣ ΤΗΝ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΔΙΚΤΥΟΥ (CONFIGURATION MODE)**

#### **3.2.1 Εγκατάσταση**

Το πρώτο πράμα που πρέπει να κάνουμε είναι να εγκαταστήσουμε και να τροφοδοτήσουμε το δίκτυό μας. Στη συνέχεια συνδέουμε τον controller με κάποιο PC μέσω ενός καλωδίου null modem. Ανοίγουμε το PC και τρέχουμε το πρόγραμμα terminal που έχουμε εγκαταστήσει σε αυτό.

Οι ρυθμίσεις που θα πρέπει να κάνουμε στο terminal είναι: 19200 baud, 8 data bits, 1 stop bit, no parity check, no flow control. Όταν γίνουν όλα σωστά, στο terminal εμφανίζεται η οθόνη που φαίνεται στην εικόνα 1.



Εικόνα 1

### 3.2.2 Ξεκινώντας την διαμόρφωση

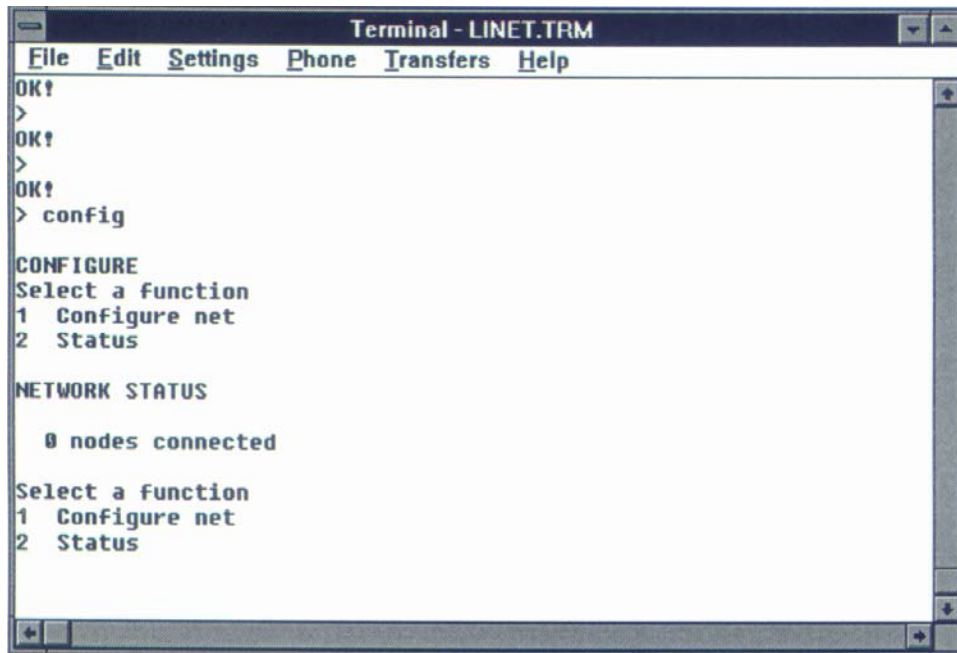
Όπως είπαμε και προηγουμένως ο controller μπορεί να χρησιμοποιηθεί για διαμόρφωση του δικτύου (configuration mode) ή για έλεγχο (command mode). Όταν λοιπόν ανοίγουμε το terminal, ερχόμαστε σε command mode. Το terminal περιμένει από εμάς μια εντολή για το τι θέλουμε να κάνει. Για να περάσουμε σε configuration mode πληκτρολογούμε την εντολή 'GONFIG' και πατάμε ENTER. Μας εμφανίζεται λοιπόν το Configure – menu.

Η Διαμόρφωση είναι βασισμένη σε ένα σύστημα οδηγούμενο από μενού. Ξεκινάς από το αρχικό μενού και μπαίνεις στα επιμέρους πατώντας τον ανάλογο αριθμό. Για να γυρίσεις στο προηγούμενο μενού πατάς ENTER.

Κάθε καινούριο node είναι απρογραμματίστο όμως καλό θα ήταν να ελέγχουμε την κατάσταση του δικτύου μας πριν αρχίσουμε την διαμόρφωση. Έτσι πατάμε '2' (Status) και εμφανίζεται η οθόνη της εικόνας 2. Εάν μας εμφανίζει αριθμό συνδεδεμένων node μεγαλύτερο από 0, τότε υπάρχουν κάποια προγραμματισμένα nodes στο δίκτυο. Για να 'καθαρίσουμε' το δίκτυο επιλέγουμε Configure net - clear all nodes.

#### Σημείωση:

Όταν βρισκόμαστε σε command mode το δίκτυο μας είναι ενεργό, εκτελεί κάθε εργασία του και περιμένει εντολές από τον χρήστη. Κατά το configuration mode σταματάει κάθε λειτουργία του δικτύου μέχρι να διαμορφωθεί και να επανέλθει σε command mode.



Εικόνα 2

### 3.2.3 Ομάδες, διευθύνσεις και λειτουργίες

Κάθε node έχει τρεις βασικές λειτουργίες οι οποίες είναι:

- Διακόπτης on/off (Toggle mode)
- Pwm (διαμόρφωση εύρους παλμών) για DC φορτία (Dimmer mode)
- Σειριακή μετάδοση δεδομένων (Data mode)

Κάθε node μπορεί να κάνει οποιαδήποτε από αυτές τις λειτουργίες αλλά μόνο μια λειτουργία μπορεί να ενεργοποιηθεί σε κάθε node ή ομάδα nodes κάθε φορά. Δεν μπορούμε δηλαδή να έχουμε ένα node σε Toggle mode και Data mode ταυτόχρονα. Μπορούμε όμως το ίδιο node να το έχουμε σε Toggle mode ενώ σε μια επόμενη εφαρμογή μας σε Data mode.

Σε μια εφαρμογή μας μπορούμε να έχουμε απομονωμένα nodes που κάνουν κάποιες λειτουργίες ή να τα εντάξουμε σε ομάδες αν οι λειτουργίες σχετίζονται μεταξύ τους. Για παράδειγμα ένας διακόπτης λάμπας και οι λάμπες που ελέγχονται από τον διακόπτη μπορούν να οριστούν σαν Toggle group.

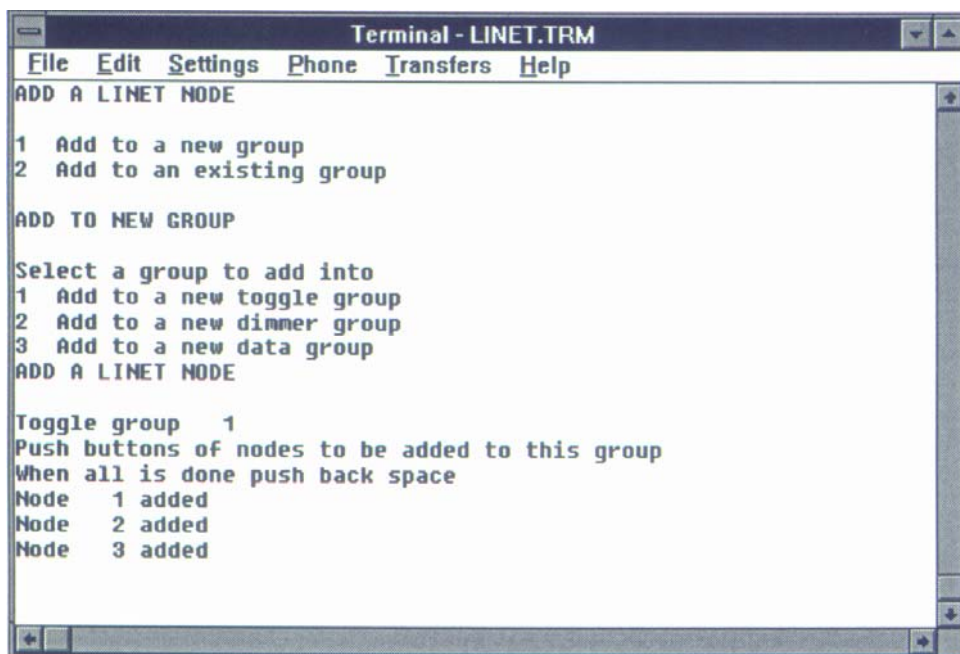
Όταν διαμορφώνουμε μια νέα ομάδα, για να ορίσουμε τα nodes από τα οποία αποτελείται πατάμε τα μπουτόν τους. Αν κάποιο node συνδέεται σε κάποιο διακόπτη (ή εξωτερικό μπουτόν, κτλ), μπορούμε τον να χρησιμοποιήσουμε αντί του μπουτόν.

Όσα nodes βρίσκονται στην ίδιο ομάδα, συμπεριφέρονται ομοίως. Παράδειγμα αν πατήσουμε οποιοδήποτε από τα μπουτόν (εσωτερικά στα nodes ή εξωτερικά) που βρίσκονται σε ένα Toggle group, οι έξοδοι όλων των node από off θα γίνουν on και αντίστροφα. Κάθε ομάδα έχει ένα αριθμό διεύθυνσης (group address, GID) από 1 έως 200.

Επίσης, κάθε node έχει μια μοναδική δικτυακή διεύθυνση (network address, NID) από 1 έως 200. Κατά την διαμόρφωση, αυτή η διεύθυνση αποθηκεύεται σε μια μνήμη EEPROM, έτσι σε περίπτωση που έχουμε διακοπή ρεύματος δεν χάνεται.

### 3.2.4 Δημιουργώντας ένα Toggle group

Για να δημιουργήσουμε ένα Toggle group επιλέγουμε Configure net - Add a Linet node - Add to a new group - Add to a toggle group και το terminal μας εμφανίζει το μήνυμα 'push buttons of nodes to be added to this group'. Πατάμε το μπουτόν ενός από τα node που θέλουμε να συμπεριλάβουμε στο Toggle group, το κρατάμε πατημένο για λίγο. Αν η καταχώρισή μας γίνει σωστά θα δούμε το led του node να ανάβει για λίγο και θα εμφανιστεί μήνυμα στην οθόνη του terminal ότι το node προστέθηκε στο Toggle group. Κάνουμε το ίδιο και με τα υπόλοιπα nodes που θέλουμε να συμπεριλάβουμε, για να επιστρέψουμε σε command mode πατάμε ENTER τρεις φορές.



Εικόνα 3

### 3.2.5 Δημιουργώντας ένα Dimmer group

Για να δημιουργήσουμε ένα Dimmer group επιλέγουμε Configure net - Add a Linet node - Add to a new group - Add to a new dimmer group. Πατάμε τα μπουτόν των nodes που θέλουμε να συμπεριλάβουμε στο Dimmer group, για έξοδο από το μενού πατάμε τρεις φορές ENTER.

Σε ένα Dimmer group όταν πατάμε στιγμιαία κάποιο από τα μπουτόν των nodes, η έξοδος γίνεται on με 50% duty cycle ενώ σε δεύτερο πάτημα γίνεται off. Για να ρυθμίσουμε το duty cycle, κρατάμε το μπουτόν πατημένο για περισσότερο χρόνο.



### 3.2.6 Αλλαγές στις ομάδες που έχουν δημιουργηθεί

Πολλές φορές χρειάζεται να αφαιρέσουμε ή να προσθέσουμε κάποιο node σε μια ομάδα που ήδη υπάρχει. Ας πούμε ότι έχουμε ένα node σε Toggle group, ένα δεύτερο node σε Dimmer group και θέλουμε να καταργήσουμε το node από το Toggle group και να το συμπεριλάβουμε στο Dimmer group. Πρώτα επιλέγουμε Configure net - Delete a Linet node και πατάμε το μπουτόν του node στο Toggle group, το node καταργείται από το Toggle group και δεν ανήκει σε καμία ομάδα. Για να το συμπεριλάβουμε στο Dimmer group επιλέγουμε Add a Linet node - Add to an existed group. Πρώτα πατάμε το μπουτόν του node στο Dimmer group για να προσδιορίσουμε σε ποια ομάδα θέλουμε να προσθέσουμε το νέο node, στη συνέχεια πατάμε το μπουτόν του node που θέλουμε να προσθέσουμε στην ομάδα. Όταν τελειώσουμε πατάμε το ENTER τρεις φορές και επιστρέφουμε στο κύριο μενού.

### 3.2.7 Δημιουργώντας Data groups

Τα βασικά data groups που μπορούν να ενταχθούν τα nodes είναι:

- I/O node group
- 8 bit data group
- 12 bit data group
- 16 bit data group
- Data exchange group

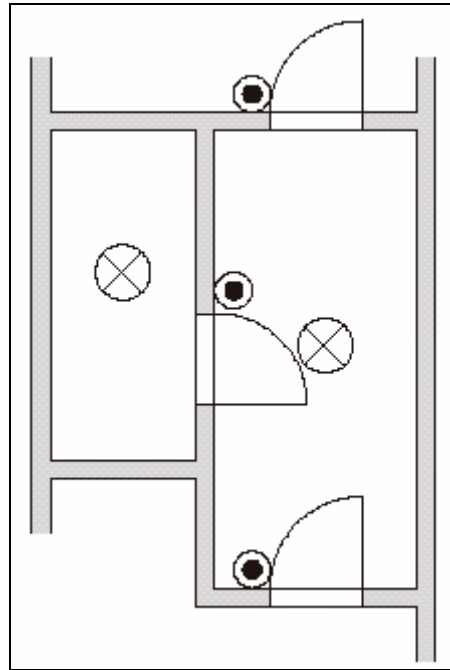
Το I/O node group αποτελείται από μια είσοδο και μια έξοδο, διαφέρει από το Toggle group στο ότι το μπουτόν δεν επηρεάζει την κατάσταση της εξόδου. Τα Data modes χρησιμοποιούνται στο να στέλνουν και να δέχονται 8, 12, ή 16 bit δεδομένα μεταξύ των node και του controller. Το Data exchange mode χρησιμοποιείται για ανταλλαγή σειριακών δεδομένων μεταξύ δύο node.

Ο αριθμός των node που μπορούμε να έχουμε σε ένα Toggle group, Dimmer group και I/O node group είναι απεριόριστος. Σε ένα Data exchange group είναι δύο και σε Data mode είναι ένα.

### 3.2.8 Παράδειγμα διαμόρφωσης σε ένα σπίτι

Στην παρακάτω εικόνα (εικόνα 4) έχουμε ένα τμήμα από τον όροφο ενός σπιτιού (τον διάδρομο και το λουτρό). Υπάρχουν μπουτόν κοντά στις δυο εισόδους του διαδρόμου και στην είσοδο του λουτρού, επίσης ένα φωτιστικό στον διάδρομο και άλλο ένα στο λουτρό.

Σε κάθε ένα από τα μπουτόν του χώρου υπάρχει ένα node. Τα φωτιστικά συνδέονται στα κουτιά διακλάδωσης τα οποία περιέχουν από ένα node με ρελέ για τον έλεγχο του φωτιστικού. Η καλωδίωση είναι σχετικά απλή: φέρνουμε καλώδιο παροχής στα κουτιά διακλάδωσης (για τα φωτιστικά) και διπολικό καλώδιο (twisted pair) σε όλα τα nodes για να τα ενώσουμε στο δίκτυο.



**Εικόνα 4:** Τμήμα ορόφου σπιτιού

Αφού τελειώσουμε με την εγκατάσταση του συστήματος, συνεχίζουμε με την διαμόρφωση. Θέλουμε τα δυο μπουτόν που βρίσκονται στο διάδρομο, να ελέγχουν το φωτιστικό του διαδρόμου, οπότε θα τα εντάξουμε στην ίδια ομάδα (Toggle group).

Πληκτρολογούμε 'CONFIG' στο terminal του υπολογιστή (που είναι συνδεδεμένος στον controller) και πατάμε ENTER.

```

TERMINAL

Linet ver. 6.2
Type a question mark and press ENTER for help
> OK!
> CONFIG

CONFIGURE

1  Configure net
2  Status
3  Configure input/output
```

Για να δημιουργήσουμε ένα Toggle group, επιλέγουμε Configure net...

```

                                TERMINAL
OK!
>
> OK!
> CONFIG

CONFIGURE

1  Configure net
2  Status
3  Configure input/output

CONFIGURE NET

1  Add a Linet node
2  Delete a Linet node
3  Identify node
4  Set group master controls
5  Set frame size
6  Set clock
7  Save net configuration
8  Restore net configuration
9  Set delay group time
A  Network debug

```

...στη συνέχεια επιλέγουμε Add a Linet node...

```

                                TERMINAL
CONFIGURE

1  Configure net
2  Status
3  Configure input/output

CONFIGURE NET

1  Add a Linet node
2  Delete a Linet node
3  Identify node
4  Set group master controls
5  Set frame size
6  Set clock
7  Save net configuration
8  Restore net configuration
9  Set delay group time
A  Network debug

ADD A LINET NODE

1  Add to a new group
2  Add to an existing group

```

...Το σύστημά μας δεν έχει άλλες ομάδες, οπότε επιλέγουμε Add to a new group...

```

                                TERMINAL
1  Add a Linet node
2  Delete a Linet node
3  Identify node
4  Set group master controls
5  Set frame size
6  Set clock
7  Save net configuration
8  Restore net configuration
9  Set delay group time
A  Network debug

ADD A LINET NODE

1  Add to a new group
2  Add to an existing group

ADD TO NEW GROUP

Select a group to add into
1  Add to a new toggle group
2  Add to a new dimmer group
3  Add to a new data group
4  Add to a new extended group

```

...Μας εμφανίζει τις διαθέσιμες ομάδες που θα μπορούσαμε να δημιουργήσουμε. Εμείς θέλουμε να δημιουργήσουμε μια ομάδα toggle, οπότε επιλέγουμε Add to a toggle group...

```

                                TERMINAL
6  Set clock
7  Save net configuration
8  Restore net configuration
9  Set delay group time
A  Network debug

ADD A LINET NODE

1  Add to a new group
2  Add to an existing group

ADD TO NEW GROUP

Select a group to add into
1  Add to a new toggle group
2  Add to a new dimmer group
3  Add to a new data group
4  Add to a new extended group
ADD A LINET NODE

Toggle group  1
Push buttons of nodes to be added to this group
When all is done push back space

```

...Στη συνέχεια το terminal μας ζητά να δηλώσουμε ποια από τα nodes θέλουμε να συμπεριλάβουμε στο Toggle group 1 (GID 1), πατώντας τα μπουτόν τους...

```

TERMINAL

9 Set delay group time
A Network debug

ADD A LINET NODE

1 Add to a new group
2 Add to an existing group

ADD TO NEW GROUP

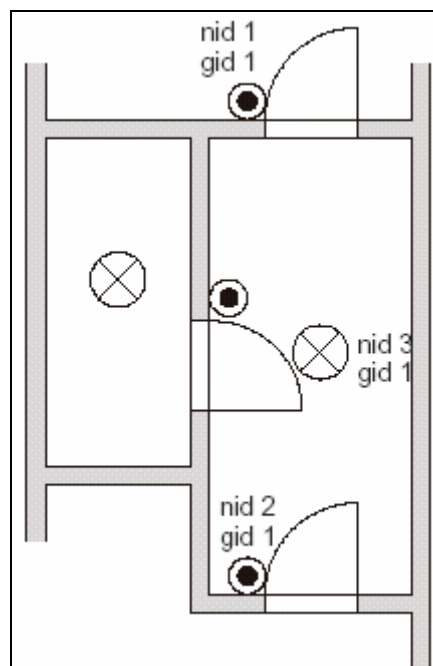
Select a group to add into
1 Add to a new toggle group
2 Add to a new dimmer group
3 Add to a new data group
4 Add to a new extended group
ADD A LINET NODE

Toggle group 1
Push buttons of nodes to be added to this group
When all is done push back space
Node 1 added
Node 2 added
Node 3 added

```

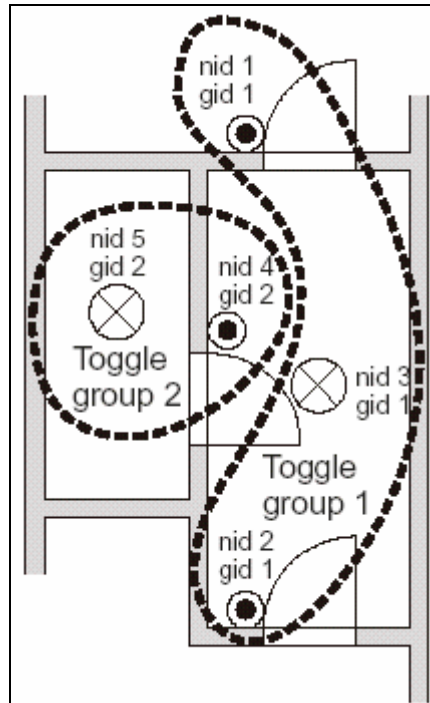
...Πατάμε τα μπουτόν των nodes που θέλουμε να εντάξουμε στην ομάδα (μπουτόν και φωτιστικό διαδρόμου). Για κάθε νέο node που προστίθεται, το terminal μας εμφανίζει την δικτυακή του διεύθυνση (NID).

Το Toggle group 1 έχει δημιουργηθεί, τώρα πατώντας κάποιο από τα μπουτόν του διαδρόμου μπορούμε αν ανάβουμε και να σβήνουμε το φωτιστικό.



Εικόνα 5: Toggle group 1

Κάνουμε την ίδια διαδικασία για το φωτιστικό και το μπουτόν του λουτρού και δημιουργούμε ένα δεύτερο Toggle group (GID 2). Τώρα το σύστημά μας που αποτελείται από δυο Toggle groups έχει οριστεί πλήρως και είμαστε έτοιμοι να το χρησιμοποιήσουμε.



**Εικόνα 6:** Το σύστημα μας ολοκληρωμένο

### 3.3 ΒΑΣΙΚΕΣ ΡΥΘΜΙΣΕΙΣ

#### 3.3.1 Αποθήκευση και φόρτωση ρυθμίσεων διαμόρφωσης

Κάθε node κρατάει 12 bits δεδομένων στην μνήμη EEPROM που διαθέτει. Τα 8 bits είναι η δικτυακή διεύθυνση (NID) του node και τα 4 bits είναι τα προγραμματιζόμενα bits (configuration bits). Η δικτυακή διεύθυνση αποθηκεύεται στην μνήμη κατά την διαμόρφωση του node, ενώ τα προγραμματιζόμενα bits όταν χρησιμοποιούμε την λειτουργία 'Set node configuration'.

Οι πληροφορίες που κρατάει ο controller είναι οι ομάδες που υπάρχουν στο δίκτυο καθώς και τα nodes που αντιστοιχούν σε κάθε ομάδα. Αυτά είναι αποθηκευμένα στην RAM του controller. Για να τα αποθηκεύσουμε στην μνήμη EEPROM επιλέγουμε `Configure net - Save net configuration`.

Όταν δίνουμε ισχύ στον controller αυτός παίρνει τις πληροφορίες που είναι αποθηκευμένες EEPROM, τις αντιγράφει στην RAM και αρχίζει την λειτουργία του. Αν θέλουμε να κάνουμε αυτή τη διαδικασία χειροκίνητα (π.χ. έχουμε κάνει κάποιο λάθος και θέλουμε να επαναφέρουμε το σύστημά μας στην προηγούμενη κατάσταση) επιλέγουμε `Configure net - Restore net configuration - Restore from active save` και ανακτούμε τις τελευταίες αποθηκευμένες πληροφορίες. Για να ανακτήσουμε τις προτελευταίες αποθηκευμένες πληροφορίες επιλέγουμε `Restore from backup`.

#### 3.3.2 Σβήσιμο των ρυθμίσεων

Για να σβήσουμε τις ρυθμίσεις διαμόρφωσης, που υπάρχουν στον controller, τις διευθύνσεις των node καθώς και τα προγραμματιζόμενα bits επιλέγουμε `Configure net - Network debug - Clear node addresses and delete connections`. Μετά από αυτή τη διαδικασία στον controller και τα nodes δεν υπάρχει καμία πληροφορία, είναι κενά.

#### 3.3.3 Προγραμματιζόμενα bits

Σε κάθε node υπάρχουν τέσσερα προγραμματιζόμενα bits τα οποία όταν ρυθμιστούν αποθηκεύονται στην EEPROM που έχουν τα nodes για αυτό το σκοπό. Όταν ένα node ενταχθεί ή σβηστεί από μια ομάδα, δεν γίνονται αλλαγές στα προγραμματιζόμενα bits του. Όμως αν σβηστεί ολόκληρο το δίκτυο, τότε τα προγραμματιζόμενα bits παίρνουν τιμή 0.

Σε πολλές εφαρμογές θα χρειαστεί να προγραμματίσουμε αυτά τα bits. Όπως ξέρουμε το bit PO3 χρησιμοποιείται για να ενεργοποιήσει τον A/D-converter που έχει κάθε node και το bit PO1 για εξωτερική χρήση.

Για να ορίσουμε τα προγραμματιζόμενα bits ενός node, επιλέγουμε `Configure net - Network debug - Set debug data - Set configuration data`,

πληκτρολογούμε την κατάλληλη λέξη (π.χ. για να ενεργοποιήσουμε τον A/D-converter πληκτρολογούμε '4') και πατάμε δυο φορές ENTER. Στη συνέχεια επιλέγουμε Set node configuration και πατάμε το μπουτόν του node που θέλουμε να ορίσουμε.

### 3.3.4 Αρχείο Backup

Όταν χρησιμοποιούμε κάποιο πρόγραμμα terminal για την διαμόρφωση του δικτύου, έχουμε την δυνατότητα να αποθηκεύσουμε τις ρυθμίσεις διαμόρφωσης του δικτύου μας σε ένα αρχείο κειμένου (αρχείο backup). Το αρχείο αυτό μπορεί να χρησιμοποιηθεί σαν αντίγραφο ασφαλείας και να φορτωθεί στον controller σε περίπτωση που χρειαστεί.

Για να δημιουργήσουμε το αρχείο κειμένου, πληκτρολογούμε 'get' και πατάμε ENTER, το terminal εμφανίζει στην οθόνη του τις ρυθμίσεις διαμόρφωσης του δικτύου. Για να ολοκληρωθεί η διαδικασία, χρησιμοποιούμε το text capture tool του terminal και τα περιεχόμενα της οθόνης μετατρέπονται σε κείμενο και αποθηκεύονται στο αρχείο που έχουμε ορίσει.

Αν τώρα θέλουμε να φορτώσουμε τις ρυθμίσεις μας, από το αρχείο κειμένου, στον controller χρησιμοποιούμε το text send tool του terminal.

#### Σημείωση :

Το κείμενο που θα μεταφέρουμε στο backup αρχείο μας πρέπει να αρχίζει με τους χαρακτήρες 'put' και να τελειώνει με τον χαρακτήρα 'e'.

```

                                TERMINAL
CONFIGURE
 1 Configure net
 2 Status
 3 Configure input/output

OK!
> GET

put
; Linet Oy, LINET controller, version 6.2, serial no. 0
; At: 02.10.01 09:57:29
v 2
m 200
g 1 1
n 1 1
n 2 1
n 3 1
e

OK!
>
```



## ΚΕΦΑΛΑΙΟ 4

### 4.1 ΕΛΕΓΧΟΣ ΤΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΟ TERMINAL (COMMAND MODE)

#### 4.1.1 Γενική περιγραφή του Command mode

Όταν βρισκόμαστε σε command mode μπορούμε να διαβάσουμε την κατάσταση των nodes ή να την αλλάξουμε χρησιμοποιώντας κάποιες εντολές. Οι εντολές αυτές είναι μικρές ASCII συμβολοσειρές οι οποίες στέλνονται από και προς την σειριακή πόρτα του controller. Μπορούμε να στείλουμε αυτές τις εντολές στον controller μέσω του terminal ή κάποιου προγράμματος που έχει σχεδιαστεί για αυτό το σκοπό (πρόγραμμα host).

Αφού στείλουμε στον controller μια εντολή, αυτός κάνει τις απαραίτητες ενέργειες και μας στέλνει απάντηση 'OK' αν η ενέργεια έγινε σωστά ή μας στέλνει 'ERROR' αν έγινε κάποιο λάθος. Πρέπει να περιμένουμε απάντηση από τον controller πριν του στείλουμε επόμενη εντολή.

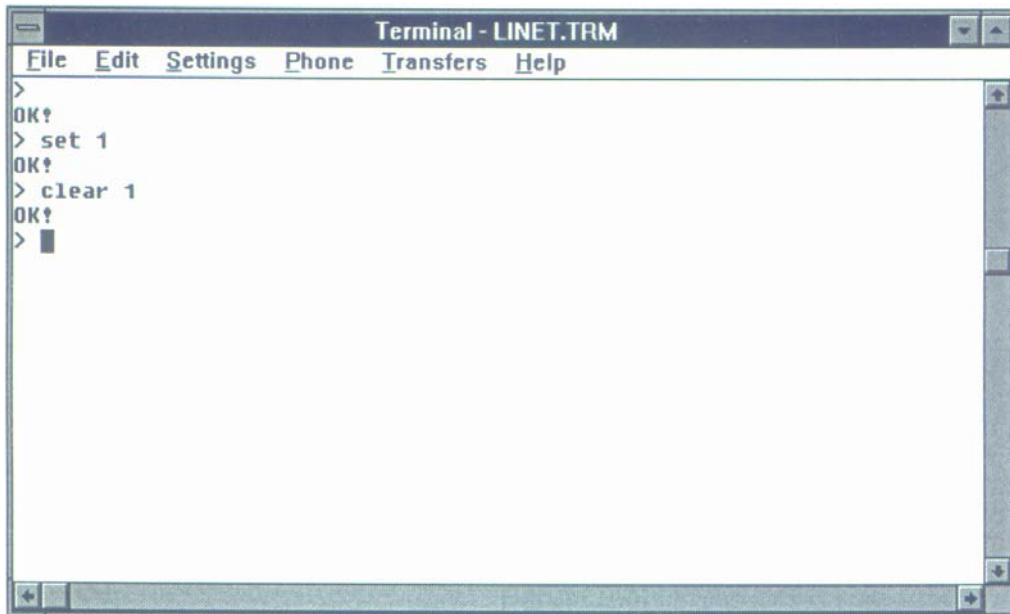
Οι εντολές που έχουν σχέση με έλεγχο των nodes χρησιμοποιούν τον αριθμό ομάδας (GID) για να προσδιορίσουν σε ποια ομάδα nodes αναφερόμαστε. Αν για παράδειγμα έχουμε πέντε nodes σε ένα Toggle group με αριθμό ομάδας 1 και θέλουμε να κάνουμε την έξοδο από off σε on, πληκτρολογούμε την εντολή 'Set 1' όπου '1' είναι ο αριθμός της ομάδας (GID).

Για να μας εμφανιστεί η λίστα με τις διαθέσιμες εντολές, πληκτρολογούμε '?' ή 'help' και πατάμε ENTER. Να σημειώσουμε ότι οι εντολές μπορούν να χρησιμοποιηθούν σε κάθε ομάδα node εκτός από το Data exchange group που δεν έχει καμία εντολή.

Τέλος, μπορούμε να χρησιμοποιήσουμε σαν αριθμητικό σύστημα το δεκαδικό, το δεκαεξαδικό ή το οκταδικό.

#### 4.1.2 Εντολές Set και Clear

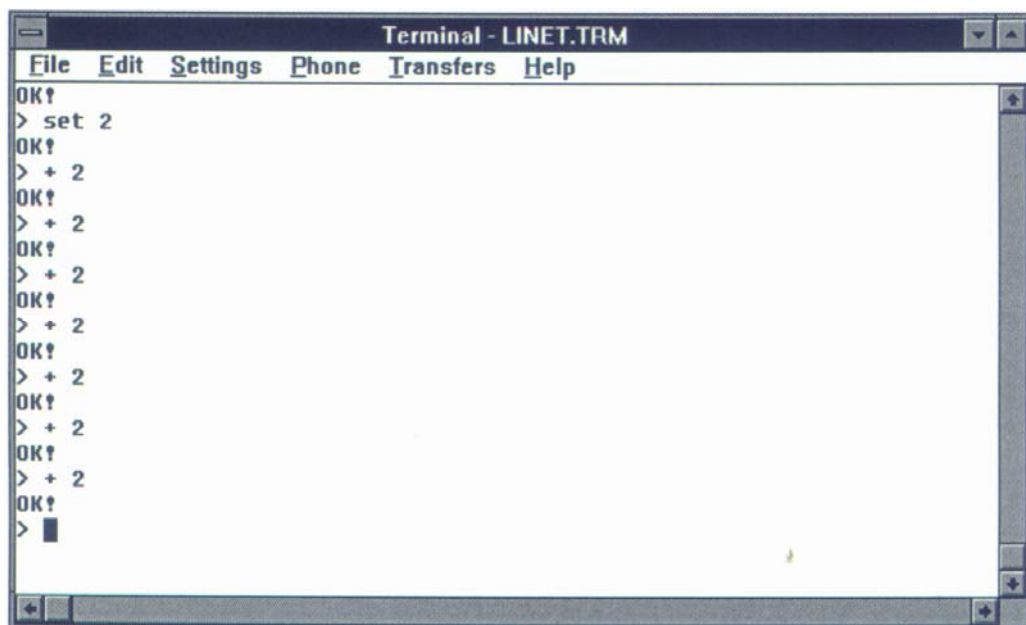
Οι εντολές Set και Clear είναι διαθέσιμες στα Toggle groups, Dimmer groups και I/O-nodes. Για να ενεργοποιήσουμε την έξοδο κάποιας ομάδας, πληκτρολογούμε 'SET g' ή 'S g' όπου 'g' είναι ο αριθμός της ομάδας (GID) και πατάμε ENTER. Για να απενεργοποιήσουμε την έξοδο μιας ομάδας, πληκτρολογούμε 'Clear g' ή 'C g' και πατάμε ENTER.



```
Terminal - LINET.TRM
File Edit Settings Phone Transfers Help
>
OK?
> set 1
OK?
> clear 1
OK?
> █
```

### 4.1.3 Εντολές Step up και Step down

Οι εντολές Step up και Step down είναι διαθέσιμες μόνο για το Dimmer group. Όταν δίνουμε εντολή Set σε ένα Dimmer group, το duty cycle της εξόδου γίνεται 50%. Η τιμή του duty cycle μπορεί να αυξηθεί με την εντολή **'Step up g'** ή **'+ g'** και να μειωθεί με την εντολή **'Step down g'** ή **'- g'**. Το όριο ρύθμισης από 0% μέχρι 100% χωρίζεται σε 32 βήματα. Αν για παράδειγμα από duty cycle 50% (16/32) θέλουμε να πάμε στο 75% (24/32), θέλουμε 8 βήματα.



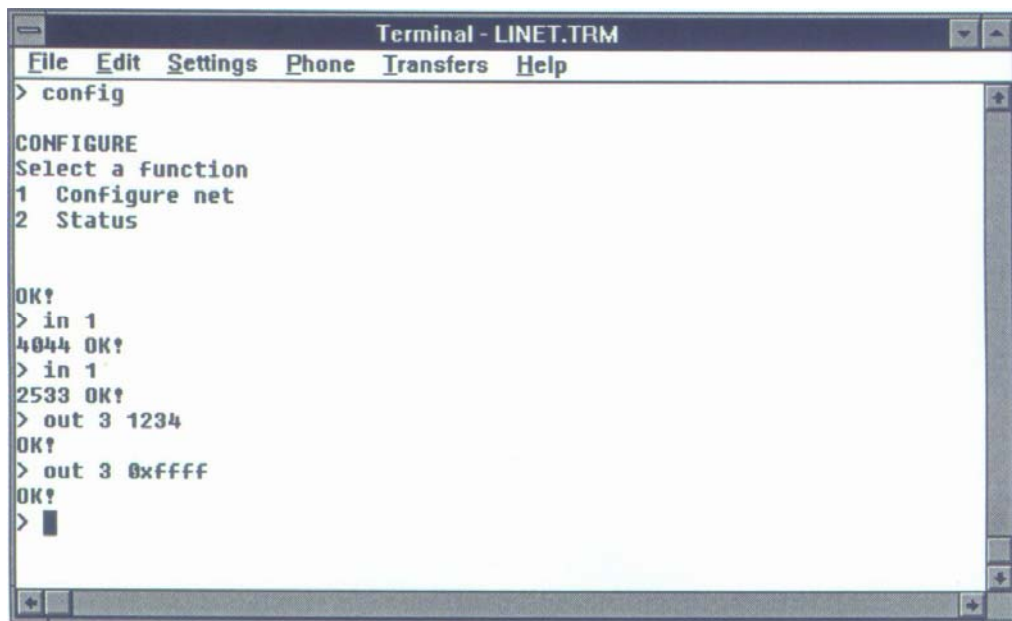
```
Terminal - LINET.TRM
File Edit Settings Phone Transfers Help
OK?
> set 2
OK?
> + 2
OK?
> + 2
OK?
> + 2
OK?
> + 2
OK?
> + 2
OK?
> + 2
OK?
> + 2
OK?
> + 2
OK?
> █
```

#### 4.1.4 Εντολές In και Out

Οι εντολές In και Out είναι διαθέσιμες για όλες τις ομάδες εκτός από το Data exchange group. Για να διαβάσουμε την κατάσταση ενός Toggle group, Dimmer group ή I/O- node, πληκτρολογούμε **'In g'** ή **'I g'**. Να σημειώσουμε ότι ένα Dimmer group μας επιστρέφει '1' αν η έξοδος είναι on ή '0' αν είναι off, δεν μας επιστρέφει το duty cycle. Επίσης δεν μπορούμε να χρησιμοποιήσουμε την εντολή 'Out' για να μεταβάλουμε το duty cycle, αυτό μπορεί να γίνει μόνο με τις εντολές Step up και Step down.

Στα Data groups, η εντολή 'In' χρησιμοποιείται για να διαβάσουμε τα δεδομένα μιας ομάδας. Για παράδειγμα, αν έχουμε ορίσει ένα node σαν data 12-bit group 1, έχουμε συνδέσει ένα ποτενσιόμετρο στην αναλογική του είσοδο και θέλουμε να διαβάσουμε την τιμή του ποτενσιόμετρου, πληκτρολογούμε 'In 1'.

Αντιστοίχως, με την εντολή 'Out' μπορούμε να στείλουμε δεδομένα σε μια ομάδα. Η σύνταξή της είναι **'Out g d'** ή **'O g d'**, όπου 'g' το GID ομάδα και 'd' τα δεδομένα που θέλουμε να στείλουμε. Για παράδειγμα, έχουμε ορίσει ένα node σαν data 12-bit group 2 και έχουμε συνδέσει μια οθόνη. Για να στείλουμε τον αριθμό '1234' στην οθόνη, πληκτρολογούμε 'Out 2 1234'



```
Terminal - LINET.TRM
File Edit Settings Phone Transfers Help
> config
CONFIGURE
Select a function
1 Configure net
2 Status

OK?
> in 1
4044 OK?
> in 1
2533 OK?
> out 3 1234
OK?
> out 3 0xffff
OK?
>
```

#### 4.1.5 Άλλες εντολές

- ECHO ON και ECHO OFF – terminal echo

Η προεπιλογή στο terminal είναι echo off, έτσι δεν μπορούμε να δούμε τους χαρακτήρες που πληκτρολογούμε σε αυτό. Αυτό γίνεται για να μειωθεί ο χρόνος που χρειάζεται ώστε να φορτωθεί η σειριακή διασύνδεση (serial interface), όταν ο controller συνδέεται σε υπολογιστή που τρέχει κάποιο πρόγραμμα host.

Με την εντολή 'Echo on' ενεργοποιούμε το terminal echo και έτσι βλέπουμε τους χαρακτήρες που πληκτρολογούμε.

- INIT – αρχικοποίηση συστήματος

Η εντολή 'Init' πραγματοποιεί επανεκκίνηση στο σύστημά μας, είναι δηλαδή ανάλογο με το να διακόψουμε την παροχή ισχύος στον controller και να την ξαναδώσουμε. Μετά την 'Init' έχουμε αρχικοποίηση του συστήματος, σβήνεται ότι δεν έχει αποθηκευτεί στην EEPROM και ερχόμαστε σε αρχική κατάσταση.

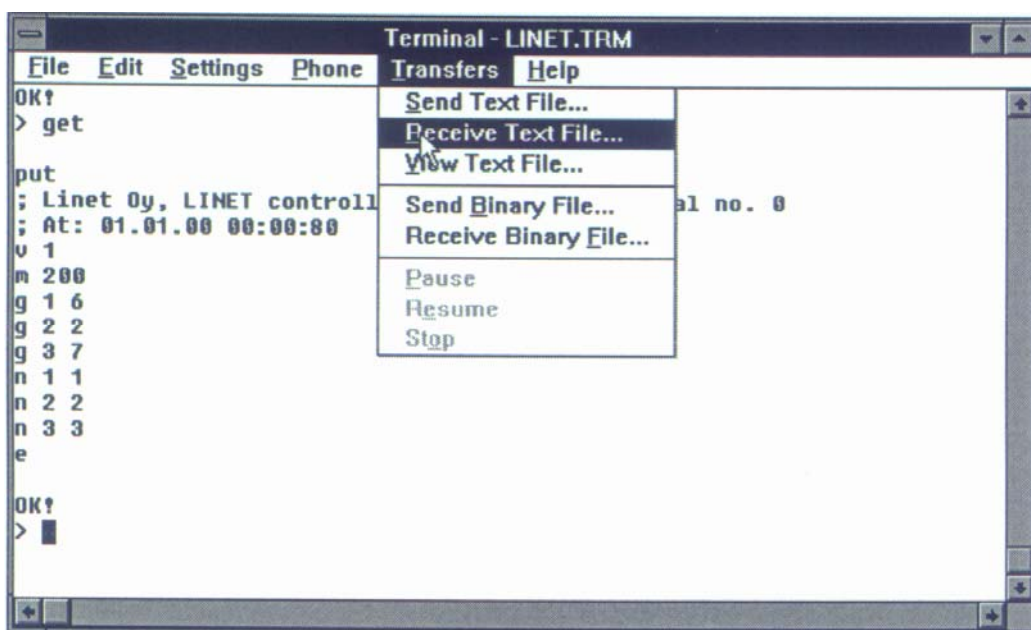
Εάν θέλουμε να συμπεριλάβουμε στο δίκτυό μας καινούρια nodes που δεν έχουν διαμορφωθεί, καλό θα ήταν να χρησιμοποιήσουμε την εντολή 'Init' για να γίνει επανεκκίνηση πριν την διαμόρφωση αυτών των nodes.

- GET και PUT – Αποθήκευση και φόρτωση ρυθμίσεων διαμόρφωσης

Οι εντολές 'Get' και 'Put' χρησιμοποιούνται για να αποθηκεύσουν, σε δισκέτα ή αρχείο (backup), τις ρυθμίσεις που έχουμε κάνει κατά την διαμόρφωση του συστήματος και να τις επανακτήσουν όποτε αυτό χρειαστεί. Αυτό είναι χρήσιμο σε περίπτωση κάποιου σφάλματος ή όταν ο controller πρέπει να αντικατασταθεί.

Με την 'Get' μπορούμε επίσης να έχουμε μια γενική εικόνα του συστήματος μας, να δούμε τις ομάδες και τα nodes που έχουμε εγκαταστήσει. Δίνοντας την εντολή 'Get' μας εμφανίζεται:

```
v version
m frame size
g group_id group_type
...
n node_id group_id
...
e
```



Σημείωση :

Οι τιμές που μπορούν να πάρουν οι μεταβλητές 'g' (αριθμός ομάδας) και 'd' (δεδομένα) είναι:

$1 \leq g \leq 200$    ή    $0x01 \leq g \leq 0xC8$

$0 \leq d \leq 255$    ή    $0x00 \leq d \leq 0xFF$    **(data 8-bit mode)**

$0 \leq d \leq 4095$    ή    $0x000 \leq d \leq 0xFFF$    **(data 12-bit mode)**

$0 \leq d \leq 65535$    ή    $0x0000 \leq d \leq 0xFFFF$    **(data 16-bit mode)**

## ΚΕΦΑΛΑΙΟ 5

### 5.1 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΙΑΘΕΣΙΜΩΝ ΟΜΑΔΩΝ

#### 5.1.1 Βασικές ομάδες – Toggle, Analog input, Dimmer, Data

##### TOGGLE GROUP

Menu – Configure – Network – Add – Add to new Group – Toggle

###### I/O Pins

FOUT – έξοδος

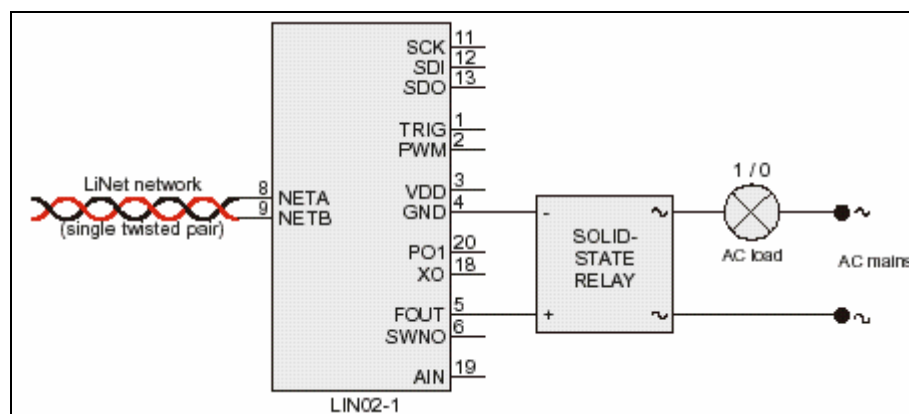
SWNO – είσοδος

###### Περιγραφή λειτουργίας

Κάθε node που ανήκει σε ένα Toggle group μπορεί να βρεθεί σε δυο καταστάσεις, με έξοδο on ή off. Αυτό ισχύει για όλα τα nodes που είναι στο ίδιο Toggle group, θα είναι on ή off την ίδια στιγμή. Η κατάσταση την ομάδα αλλάζει (από on σε off ή αντίστροφα) όταν ανιχνευτεί rising edge στην είσοδο SWNO κάποιου από τα nodes της ομάδας.

###### Παράδειγμα εφαρμογών

Έλεγχος φωτιστικών



##### ANALOG INPUT GROUP

Menu – Configure – Network – Add – Add to new Group – Data – Data 12

###### I/O Pins

AIN – αναλογική είσοδος

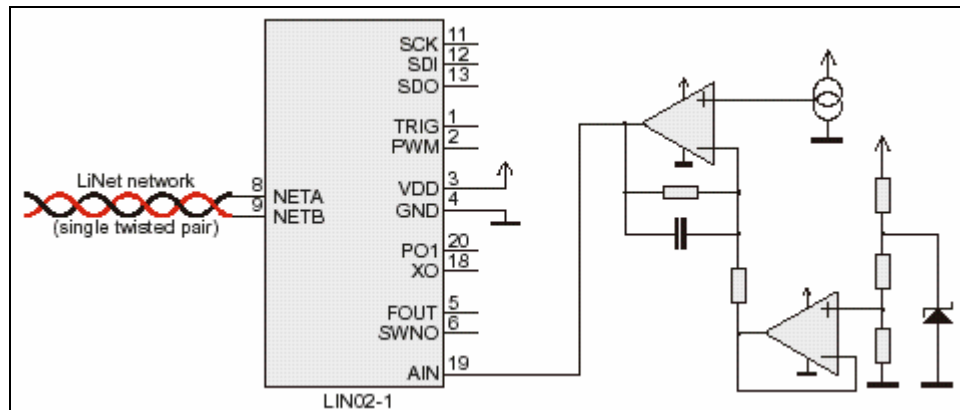
ADE – ενεργοποίηση του A/D converter

### Περιγραφή λειτουργίας

Κάθε node περιέχει στο εσωτερικό του ένα A/D converter που μετατρέπει τάση από 0 μέχρι 1,25 V, στην είσοδο AIN, σε 12-bit αριθμό. Ο αριθμός αυτός στέλνεται στο δίκτυο και μπορεί να αξιοποιηθεί ανάλογα με την εφαρμογή μας. Ο A/D converter ενεργοποιείται δίνοντας '1' στο προγραμματιζόμενο bit Rb3.

### Παράδειγμα εφαρμογών

Αισθητήρας θερμοκρασίας



## DIMMER GROUP

Menu - Configure - Network - Add - Add to new Group - Dimmer

### I/O Pins

PWM - έξοδος

TRIG - είσοδος σκανδαλισμού

SWNO - είσοδος

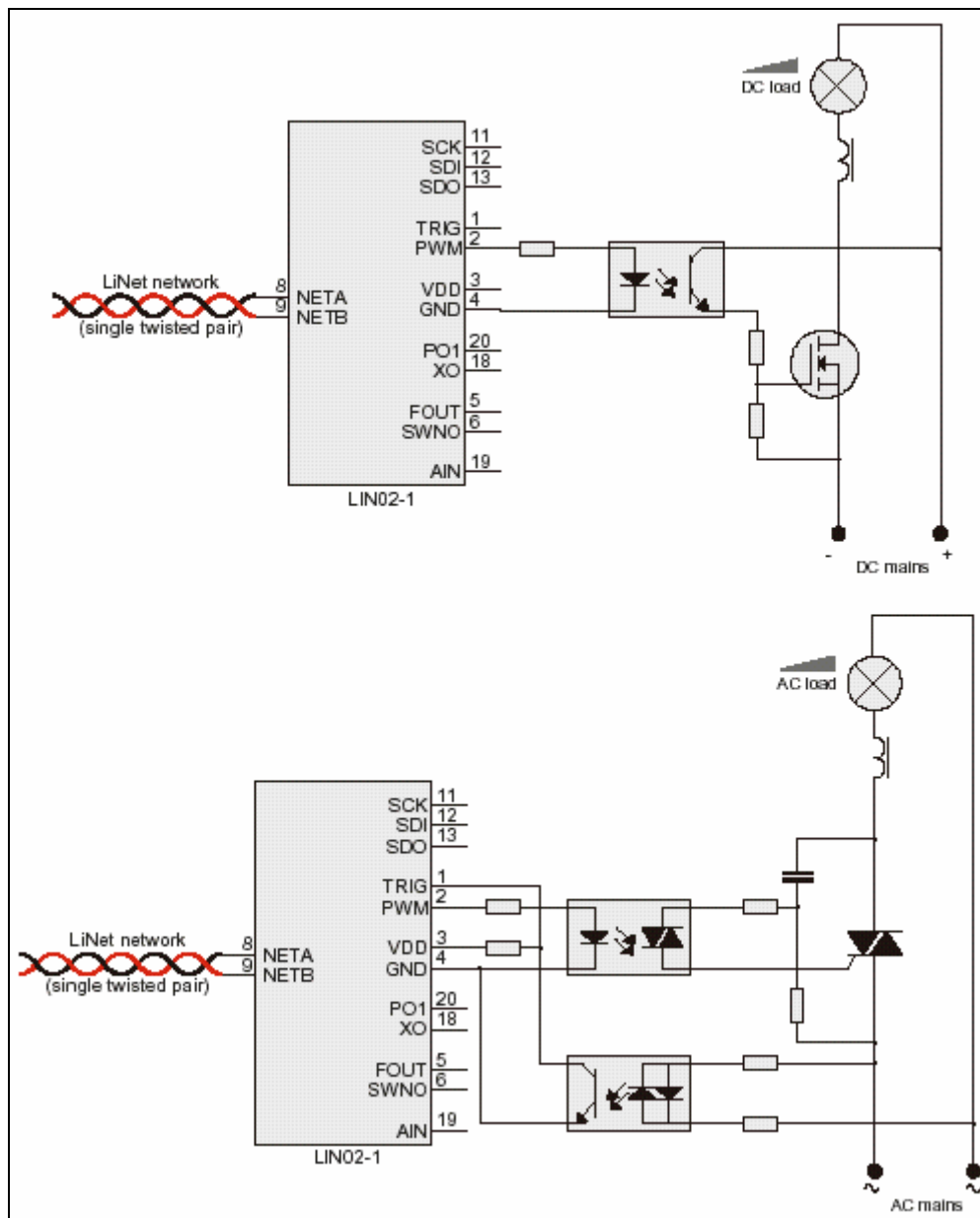
### Περιγραφή λειτουργίας

Όλα τα nodes που είναι στο ίδιο Dimmer group έχουν την ίδια κατάσταση εξόδου κάθε στιγμή. Η έξοδος εμφανίζεται διαμορφωμένη κατά εύρος παλμών στα 100 Hz και χωρίζεται σε 32 βήματα. Όταν πατήσουμε το μπουτόν κάποιου node που βρίσκεται σε Dimmer group, όλη η ομάδα αλλάζει κατάσταση. Αν είναι off γίνεται on και λειτουργεί με 50% duty cycle, ενώ αν είναι on γίνεται off. Σε περίπτωση που θέλουμε να μεταβάλουμε το duty cycle πατάμε το μπουτόν για περισσότερη ώρα, μέχρι την τιμή που θέλουμε.

### Παραδείγματα εφαρμογών

Φωτιστικά που θέλουμε να ελέγχουμε την ένταση του φωτισμού

Έλεγχος κινητήρων



## DATA GROUPS

Menu - Configure - Network - Add - Add to new Group - Data - Data8  
 Menu - Configure - Network - Add - Add to new Group - Data - Data12  
 Menu - Configure - Network - Add - Add to new Group - Data - Data16  
 Menu - Configure - Network - Add - Add to new Group - Data - Data exchange

## I/O Pins

SCK - σειριακό ρολόι  
 SDI - είσοδος σειριακών δεδομένων  
 SDO - έξοδος σειριακών δεδομένων  
 XO - έξοδος ρολογιού



## Περιγραφή λειτουργίας

Κάθε node μπορεί να λαμβάνει και να στέλνει δυαδικά δεδομένα με ένα σταθερό ρυθμό. Τα nodes που έχουν ορισθεί σαν Data8, Data12 και Data16 ανταλλάσσουν με τον controller δεδομένα 8, 12, και 16 bits αντίστοιχα. Στο Data exchange group έχουμε δυο nodes που ανταλλάσσουν δεδομένα μεταξύ τους.

Αν θέλουμε να συνδέσουμε κάποια συσκευή (π.χ. μικροελεγκτή) σε ένα node, αυτή πρέπει να δέχεται και να στέλνει δεδομένα της μορφής:

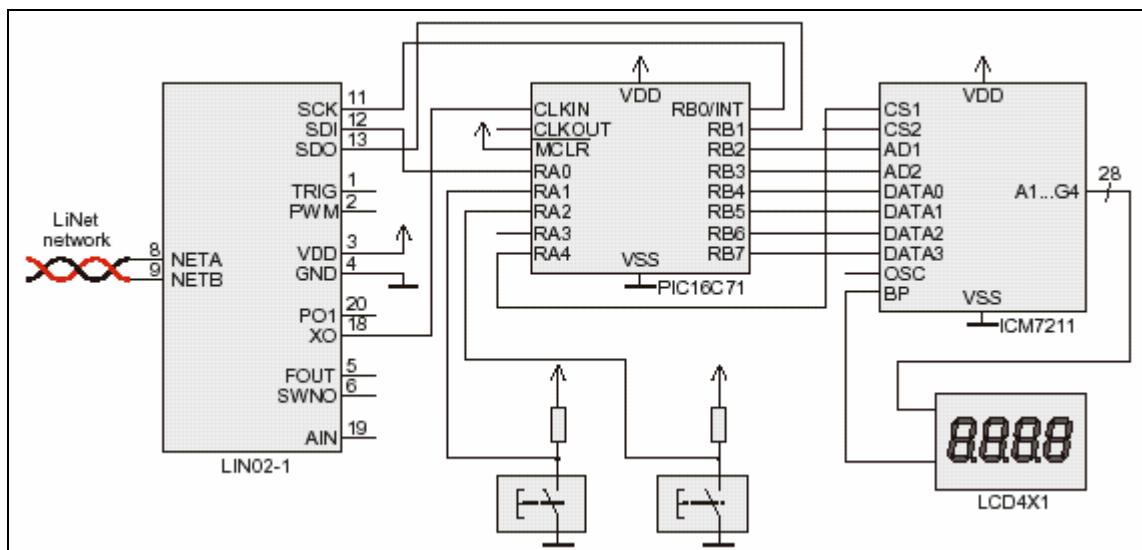
0 1 1 1 1 1 0 X7 X6 X5 X4 0 X3 X2 X1 X0 0 (8 bits)

0 1 1 1 1 1 0 X11 X10 X9 X8 0 X7 X6 X5 X4 0 X3 X2 X1 X0 0 (12 bits)

0 1 1 1 1 1 0 X15 X14 X13 X12 0 X11 X10 X9 X8 0 X7 X6 X5 X4 0 X3 X2 X1 X0 0 (16 bits)

## Παραδείγματα εφαρμογών

Μονάδες επικοινωνίας



### 5.1.2 Πρόσθετες ομάδες – I/O node, lamp group, call group, AD/state, control

#### I/O NODE GROUP

Menu - Configure - Network - Add - Add to new Group - Data - I/O-node

#### I/O Pins

FOUT – έξοδος

SWNO – είσοδος

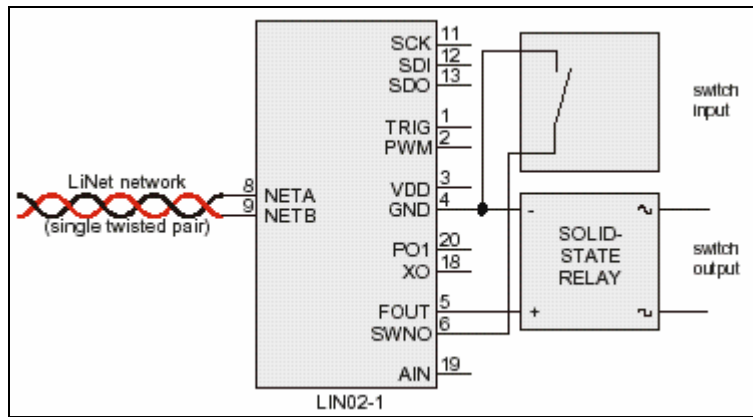
#### Περιγραφή λειτουργίας

Το I/O node group αποτελείται από μια είσοδο και μια έξοδο, διαφέρει από το Toggle group στο ότι το μπουτόν δεν επηρεάζει την κατάσταση της εξόδου.

#### Παραδείγματα εφαρμογών

Κεντρικός διακόπτης φωτιστικών (master)

Έλεγχος σωστής λειτουργίας συσκευών του δικτύου



## LAMP NODE

Menu - Configure - Network - Add - Add to new Group - Extended - Lamp

### I/O Pins

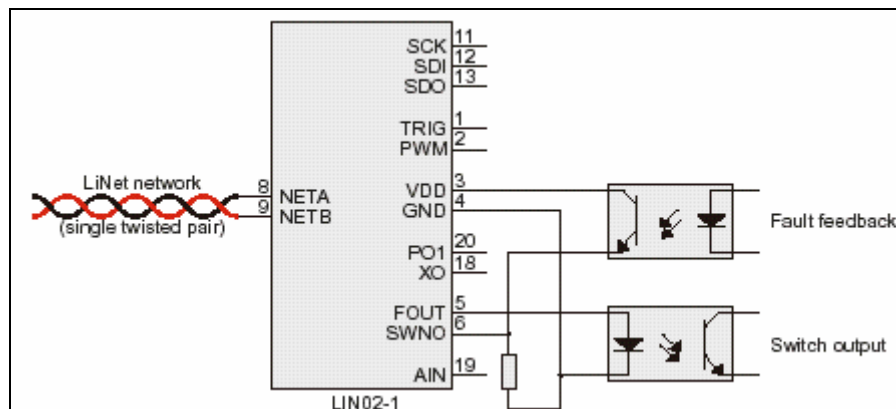
- FOUT – έξοδος
- PWM – έξοδος
- SWNO – είσοδος σφάλματος

### Περιγραφή λειτουργίας

Ένα Lamp node χρησιμοποιεί την έξοδο FOUT ή PWM για να ελέγξει ένα φορτίο. Όταν ανιχνευτεί rising edge από την είσοδο SWNO, αυτό υπολογίζεται σαν σήμα σφάλματος και εμφανίζεται στο terminal το μήνυμα 'lamp fault at *nid*'. Ένα Lamp node μπορεί να ορισθεί σαν slave σε ένα Toggle ή Dimmer group.

### Παραδείγματα εφαρμογών

Συστήματα φωτισμού



## CALL NODE

Menu - Configure - Network - Add - Add to new Group - Extended - Call

### I/O Pins

FOUT - έξοδος

SWNO - είσοδος

### Περιγραφή λειτουργίας

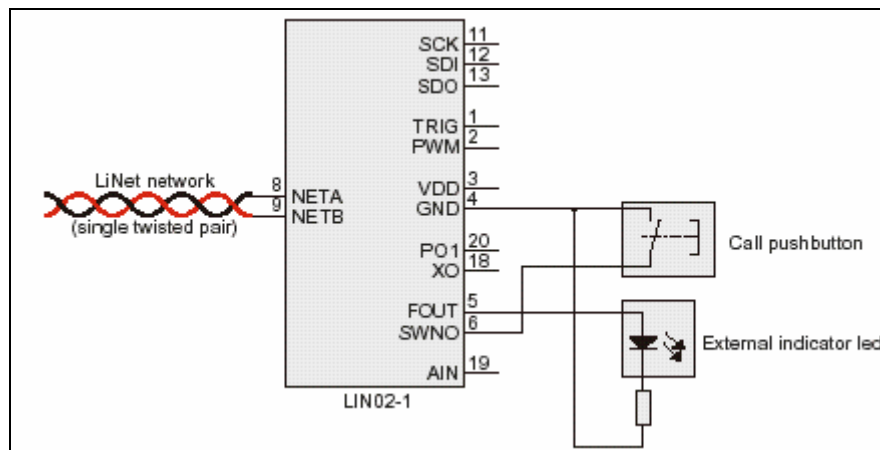
Σε ένα Call node, όταν ανιχνευτεί raising edge την είσοδο SWNO, η έξοδος γίνεται on και εμφανίζεται στο terminal το μήνυμα 'call at gid'. Όταν το μήνυμα αναγνωριστεί από τον χρήστη, όταν δηλαδή έχει δει την κλήση που έγινε, τότε η έξοδος γίνεται off.

Για να καταλάβει ο controller ότι έχουμε δει το μήνυμα κλήσης, πατάμε ENTER.

### Παραδείγματα εφαρμογών

Συστήματα συναγερμού

Συστήματα ειδοποίησης (ξενοδοχεία, νοσοκομεία)



## AD/STATE

Menu - Configure - Network - Add - Add to new Group - Data - AD/state

### I/O Pins

FOUT - έξοδος

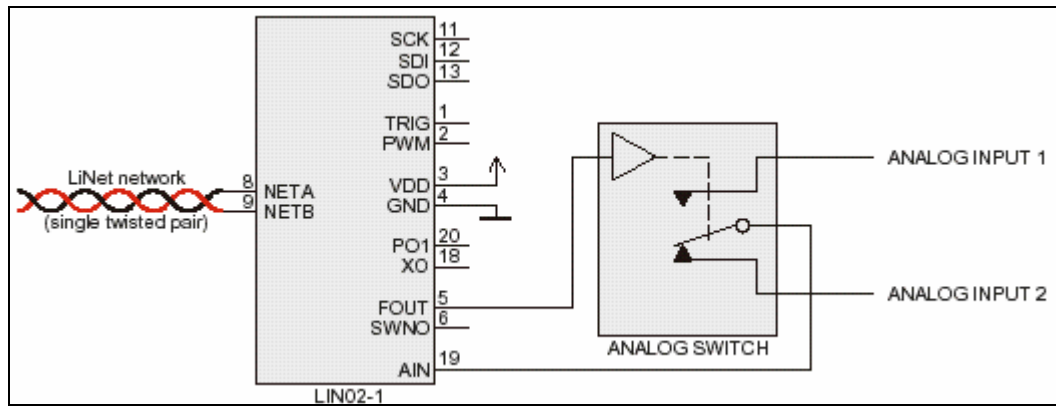
AIN - αναλογική είσοδος

### Περιγραφή λειτουργίας

Το AD/state node, είναι ένα αναλογικής εισόδου-διακοπτόμενης εξόδου node. Όταν ορίσουμε ένα node να είναι AD/state, ο A/D converter του node ενεργοποιείται αυτόματα.

### Παραδείγματα εφαρμογών

Αισθητήρες που μετρούν θερμοκρασία και υγρασία μαζί



## CONTROL

Menu - Configure - Network - Add - Add to new Group - Data - Control

### I/O Pins

FOUT - έξοδος

AIN - αναλογική είσοδος

### Περιγραφή λειτουργίας

Το Control group χρησιμοποιείται για να αντικαταστήσει τους θερμοστάτες, όταν θέλουμε να ελέγξουμε την θερμοκρασία ενός χώρου. Η αποστολή του Control group είναι ο έλεγχος σε πραγματικό χρόνο. Ένα σύστημα host, συνδέεται στον controller μέσω της σειριακής πόρτας ή πόρτας Ethernet και μπορούμε με την βοήθειά του να μεταβάλουμε την 'set value' της ομάδας (είναι δηλαδή σαν να βάζουμε την επιθυμητή τιμή σε ένα θερμοστάτη). Μπορούμε επίσης να μετατρέψουμε τα δεδομένα που έχει το Control group σε βαθμούς Κελσίου και να τα εμφανίσουμε σε μια οθόνη.

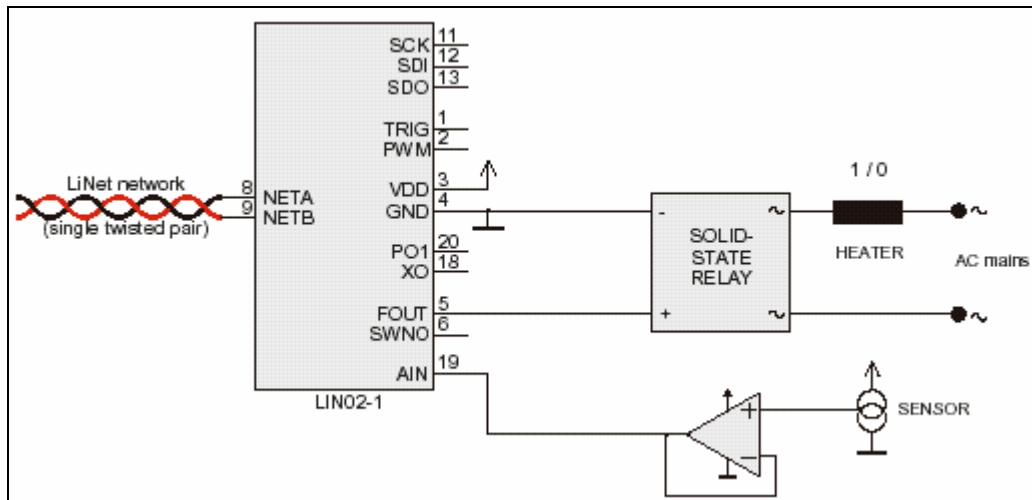
Το Control group μπορεί να δεχτεί και να στείλει δεδομένα ομοίως με ένα Data12 group. Όταν μια τιμή στέλνεται στο Control group, την παίρνει σαν μια νέα τιμή set value. Όταν διαβάζουμε από το Control group, η τιμή που διαβάζουμε είναι η τιμή του A/D converter του node.

Όταν η τιμή του set value είναι μεγαλύτερη από την τιμή του A/D converter, η έξοδος FOUT γίνεται on. Όταν η τιμή του set value είναι μικρότερη από την τιμή του A/D converter, η έξοδος FOUT γίνεται off.

Σε ένα Control group μπορούμε να έχουμε ένα ή περισσότερα nodes. Στην περίπτωση που έχουμε περισσότερα από ένα nodes, το πρώτο δίνει την μετρούμενη από τον A/D converter τιμή.

### Παραδείγματα εφαρμογών

Συστήματα ελέγχου θερμοκρασίας



### 5.1.3 Δημιουργία ομάδων Master/Slave

Ένα Lamp node μπορεί να ορισθεί σαν slave σε ένα Toggle ή Dimmer group. Για να γίνει αυτό επιλέγουμε `Configure net - Set group master controls`, πατάμε το μπουτόν του Lamp node (slave), στη συνέχεια πατάμε το μπουτόν ενός node από το Toggle ή Dimmer group (master). Όταν τελειώσει η διαδικασία, η έξοδος του Lamp node είναι στην ίδια κατάσταση με την έξοδο του master group, η είσοδος του Lamp node υπολογίζεται σαν σήμα σφάλματος.

Ένα I/O node μπορεί να ορισθεί σαν διακόπτης master-on ή master-off για ένα συνδυασμό από Toggle και/ή Dimmer groups. Μέχρι δύο τέτοιου είδους master διακόπτες μπορούν να ορισθούν στο σύστημα. Το ίδιο node μπορεί να είναι master-on σε μια ομάδα και master-off σε μια άλλη.

Για να ορίσουμε ένα I/O node σαν master μιας ομάδας, επιλέγουμε `Configure net - Set group master controls`, πατάμε το μπουτόν ενός από τα nodes του slave group, στη συνέχεια επιλέγουμε λειτουργία master-on ή master-off και τέλος πατάμε το μπουτόν του master node.

## ΚΕΦΑΛΑΙΟ 6

### **6.1 ΕΠΙΠΡΟΣΘΕΤΕΣ ΔΥΝΑΤΟΤΗΤΕΣ ΤΟΥ LINET**

Όπως έχουμε προαναφέρει, το Linet είναι ένα σύστημα που χρησιμοποιείται σε απλές εφαρμογές, έχει προγραμματιστεί να κάνει κάποιες συγκεκριμένες λειτουργίες τις οποίες και επιλέγουμε κατά την διαδικασία διαμόρφωσης. Τι γίνεται όμως σε μια περισσότερο απαιτητική εφαρμογή που δεν μπορεί να υλοποιηθεί με τις υπάρχουσες λειτουργίες;

Το Linet είναι ένα ανοιχτό σύστημα, μπορεί δηλαδή να δεχτεί και να στείλει δεδομένα σε οποιαδήποτε συσκευή αρκεί τα δεδομένα αυτά να είναι σε μορφή που τα αναγνωρίζει. Για τον σκοπό αυτό ο controller διαθέτει μια θήρα σειριακής επικοινωνίας και μια θήρα Ethernet. Η συσκευή που θα συνδέσουμε μπορεί να είναι ένας μικροελεγκτής, ένα PLC ή ακόμα και ένα PC που τρέχει κάποιο πρόγραμμα host. Μπορούμε λοιπόν σε μια απαιτητική εφαρμογή να συνδέσουμε έναν μικροελεγκτή στον controller, μέσω της σειριακής θήρας και να αναλάβει αυτός την “δύσκολή” δουλειά.

Στην συνέχεια παραθέτω κάποια ενδεικτικά προγράμματα για να παρουσιάσω αυτή τη δυνατότητα του Linet. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Visual Basic 6 και το λειτουργικό σύστημα τα Windows 98.

### **6.2 LINET MONITOR**

Το Linet Monitor είναι ένα πρόγραμμα με το οποίο μπορούμε να δούμε ή να αλλάξουμε την κατάσταση του δικτύου μας, μπορεί δηλαδή να αντικαταστήσει το Command mode. Σκοπός του Linet Monitor είναι να μας δείξει πως μπορούμε να συνδέσουμε το Linet με ένα PC μέσω της σειριακής θήρας και να ανταλλάξουμε δεδομένα. Ποια είναι όμως η λογική που χρησιμοποιήθηκε για την ανάπτυξη του Linet Monitor;

Για να κάνουμε έλεγχο του δικτύου μας, συνδέουμε ένα PC στον controller μέσω της σειριακής θήρας και με ένα πρόγραμμα terminal κάνουμε έλεγχο του δικτύου χρησιμοποιώντας τις εντολές που υποστηρίζει ο controller. Με τον ίδιο τρόπο λειτουργεί και το Linet Monitor, συνδέεται στον controller μέσω της σειριακής, στέλνει εντολές προς αυτόν και εμφανίζει τα αποτελέσματα.

#### **6.2.1 Κώδικας του Linet Monitor**

Για να έχουμε πρόσβαση στην σειριακή θήρα του PC με την Visual Basic, χρησιμοποιήσαμε το PORT.DLL το οποίο είναι μια βιβλιοθήκη από υπορουτίνες προγραμματισμού, γραμμένες από τον H.-J. Berndt. Όλες οι διαδικασίες (procedures) και λειτουργίες (functions) που χρησιμοποιούνται στη DLL, γνωστοποιούνται σε Visual Basic με την εντολή Declare. Στην περίπτωση μας αυτό γίνεται με την ενιαία λογισμική μονάδα (module) PORTS.BAS την οποία εντάσσουμε στο πρόγραμμά μας. Παρακάτω έχουμε τον κώδικά του PORTS.BAS.

```

Declare Function OPENCOM Lib "Port" (ByVal A$) As Integer
Declare Sub CLOSECOM Lib "Port" ()
Declare Sub SENDBYTE Lib "Port" (ByVal b%)
Declare Function READBYTE Lib "Port" () As Integer
Declare Sub DTR Lib "Port" (ByVal b%)
Declare Sub RTS Lib "Port" (ByVal b%)
Declare Sub TXD Lib "Port" (ByVal b%)
Declare Function CTS Lib "Port" () As Integer
Declare Function DSR Lib "Port" () As Integer
Declare Function RI Lib "Port" () As Integer
Declare Function DCD Lib "Port" () As Integer
Declare Sub DELAY Lib "Port" (ByVal b%)
Declare Sub TIMEINIT Lib "Port" ()
Declare Sub TIMEINITUS Lib "Port" ()
Declare Function TIMEREAD Lib "Port" () As Long
Declare Function TIMEREADUS Lib "Port" () As Long
Declare Sub DELAYUS Lib "Port" (ByVal l As Long)
Declare Sub REALTIME Lib "Port" (ByVal i As Boolean)

```

Η OPENCOM ενεργοποιεί την σειριακή θήρα του PC, ενώ η CLOSECOM την απενεργοποιεί. Οι SENDBYTE και READBYTE μεταφέρουν δεδομένα. Οι διαδικασίες DTR, RTS και TxD ελέγχουν τα σχετικά σήματα εξόδου και οι διαδικασίες CTS, DSR, RI και DCD διαβάσουν τα σχετικά σήματα εξόδου. Οι υπόλοιπες διαδικασίες χρησιμοποιούνται για μέτρηση χρόνου.

Χρησιμοποιήσουμε και ένα δεύτερο module στο πρόγραμμά μας όπου υπάρχουν κάποιες συναρτήσεις που χρησιμοποιούνται συχνά. Το module αυτό το ονομάζεται Synarthseis.BAS και περιλαμβάνει τρεις συναρτήσεις, παρακάτω έχουμε τον κώδικά του.

‘ Η συνάρτηση SendData δέχεται σαν όρισμα μια συμβολοσειρά την οποία και την στέλνει στον controller. Η συμβολοσειρά αυτή είναι κάποια εντολή’

```

Function SendData (data As String)
Dim i As Integer

For i = Len (data) To 1 Step -1
SEND BYTE Asc (Right$(data, i))
Next i

End Function

```

‘ Η συνάρτηση GroupType δέχεται σαν όρισμα ένα αριθμό, ο οποίος συμβολίζει μια ομάδα και επιστρέφει την ομάδα αυτή’

```

Function GroupType (code As Double) As String

Select Case code
Case 0
GroupType = "None"

```

```

Case 1
  GroupType = "Toggle"
Case 2
  GroupType = "Dimmer"
Case 3
  GroupType = "I/O Group"
Case 4
  GroupType = "Lamp group with fault detect"
Case 5
  GroupType = "Lamp fault monitor"
Case 6
  GroupType = "Delay timer"
Case 7
  GroupType = "Waiter call"
Case 8
  GroupType = "Data exchange"
Case 9
  GroupType = "8 bit data"
Case 10
  GroupType = "12 bit data"
Case 11
  GroupType = "16 bit data"
Case 12
  GroupType = "Analog in/Toggle out"
Case 13
  GroupType = "Control group"
End Select

```

**End Function**

‘Η συνάρτηση GroupValue δέχεται σαν όρισμα έναν αριθμό, που συμβολίζει μια ομάδα και επιστρέφει την τιμή της ομάδας’

```

Function GroupValue (Group As Double) As String
Dim dat As Integer

```

```

SendData ("i " & Group & " ")
SENDBYTE 13

```

```

Do
  dat = READBYTE
  If (dat >= 48) And (dat <= 57) Then
    GroupValue = GroupValue + Chr$(dat)
  End If
Loop Until dat = -1
  GroupValue = Val (GroupValue)

```

**End Function**



Στην συνέχεια έχουμε τον κώδικα του κυρίως προγράμματος:

**Private Sub Form\_Load ()**

**Timer2.Enabled = False**

**Dim i As Integer**

**Dim x As Double, y As Double, q As String**

**Dim dat As Integer, s As String**

**Dim textstart As Long, seekstring As String**

**Dim enter As Long, cnt As Long, str As Long, stp As Long, num As Long**

‘Ενεργοποίηση της σειριακής θήρας του PC με τις κατάλληλες παραμέτρους για να μπορεί να ανταλλάσσει δεδομένα με τον controller’

**i = OPENCOM("COM1,19200,N,8,1")**

**If i = 0 Then**

**i = OPENCOM("COM2,19200,N,8,1")**

**Option2.Value = True**

**End If**

‘Μόλις το πρόγραμμα φορτωθεί στην μνήμη του PC, θέλουμε να ελέγχει ποια nodes είναι συνδεδεμένα στο δίκτυο, σε ποιες ομάδες βρίσκονται και να μας εμφανίζει τα αποτελέσματα. Αυτό το πετυχαίνουμε με την εντολή GET. Στέλνοντας την εντολή αυτή στον controller, μας στέλνει κάποια κωδικοποιημένα δεδομένα (βλ. σελίδα 44) τα οποία και επεξεργαζόμαστε στην συνέχεια για να δούμε και να εμφανίσουμε την κατάσταση του δικτύου.’

‘Στέλνουμε την εντολή GET στον controller’

**REALTIME True**

**Call SendData ("get")**

**SENDBYTE 13**

‘Λαμβάνουμε τα κωδικοποιημένα δεδομένα από τον controller και τα αποθηκεύουμε σε μια μεταβλητή’

**Do**

**dat = READBYTE**

**If (dat > -1) And (dat <> 13) And (dat <> 19) Then**

**s = s + Chr\$(dat)**

**End If**

**If dat = 13 Then**

**s = s + Chr\$(13)**

**End If**

**Loop Until dat = -1**

‘Σε αυτό το κομμάτι του κώδικα, χρησιμοποιούμε συναρτήσεις επεξεργασίας συμβολοσειρών ώστε να πάρουμε από τα δεδομένα που λάβαμε τα τμήματα που μας ενδιαφέρουν και να τα αποκωδικοποιήσουμε.’

‘Αντλούμε πληροφορίες για τις ομάδες που υπάρχουν στο δίκτυο μας και τις εμφανίζουμε’

```
Do  
  seekstring = "g"  
  textstart = InStr(textstart + 1, s, seekstring)  
  enter = InStr(textstart + 1, s, Chr(13))  
  cnt = enter - textstart  
  If textstart > 0 Then  
    q = Mid(s, textstart + 1, cnt)  
    str = InStr(2, q, Chr(32))  
    stp = InStr(2, q, Chr(13))  
    num = stp - str  
    x = Val(Right(q, num))  
  
    str = InStr(1, q, Chr(32))  
    stp = InStr(2, q, Chr(32))  
    num = stp - str  
  
    y = Val(Left(q, num))  
  
    Set litem = ListView1.ListItems.Add(, , GroupType(x))  
    litem.SubItems(1) = y  
    litem.SubItems(2) = GroupValue(y)  
  
  End If  
  Loop Until textstart <= 0  
  textstart = 100
```

‘Αντλούμε πληροφορίες για τα nodes που υπάρχουν στο δίκτυο καθώς και σε ποια ομάδα αντιστοιχούν, εμφανίζουμε τα αποτελέσματα’

```
Do  
  seekstring = "n"  
  textstart = InStr(textstart + 1, s, seekstring)  
  enter = InStr(textstart + 1, s, Chr(13))  
  cnt = enter - textstart  
  If textstart > 0 Then  
    q = Mid(s, textstart + 1, cnt)  
    str = InStr(2, q, Chr(32))  
    stp = InStr(2, q, Chr(13))  
    num = stp - str  
    x = Val(Right(q, num))  
  
    str = InStr(1, q, Chr(32))  
    stp = InStr(2, q, Chr(32))  
    num = stp - str
```

```
y = Val(Left(q, num))  
  
Set litem = ListView2.ListItems.Add(, y)  
litem.SubItems(1) = x
```

```
End If  
Loop Until textstart <= 0  
textstart = 100
```

‘Αντλούμε πληροφορίες για τα Delay groups και Control groups που υπάρχουν στο δίκτυο μας και εμφανίζουμε τα αποτελέσματα’

```
Do  
seekstring = "d"  
textstart = InStr(textstart + 1, s, seekstring)  
enter = InStr(textstart + 1, s, Chr(13))  
cnt = enter - textstart  
If textstart > 0 Then  
q = Mid(s, textstart + 1, cnt)  
str = InStr(2, q, Chr(32))  
stp = InStr(2, q, Chr(13))  
num = stp - str  
x = Val(Right(q, num))  
  
str = InStr(1, q, Chr(32))  
stp = InStr(2, q, Chr(32))  
num = stp - str  
  
y = Val(Left(q, num))  
  
Set litem = ListView3.ListItems.Add(, y)  
litem.SubItems(1) = x  
  
End If  
Loop Until textstart <= 0
```

‘Αντλούμε πληροφορίες για τα Master off groups που υπάρχουν στο δίκτυο μας και εμφανίζουμε τα αποτελέσματα’

```
Do  
seekstring = "m0"  
textstart = InStr(textstart + 1, s, seekstring)  
enter = InStr(textstart + 1, s, Chr(13))  
cnt = (enter - textstart) - 1  
If textstart > 0 Then  
q = Mid(s, textstart + 2, cnt)  
str = InStr(2, q, Chr(32))  
stp = InStr(2, q, Chr(13))  
num = stp - str  
x = Val(Right(q, num))
```

```
str = InStr(1, q, Chr(32))
stp = InStr(2, q, Chr(32))
num = stp - str
```

```
y = Val(Left(q, num))
```

```
Set litem = ListView4.ListItems.Add(, , x)
litem.SubItems(1) = y
```

**End If**

**Loop Until textstart <= 0**

‘Αντλούμε πληροφορίες για τα Master on groups που υπάρχουν στο δίκτυο μας και εμφανίζουμε τα αποτελέσματα’

**Do**

```
seekstring = "m1"
textstart = InStr(textstart + 1, s, seekstring)
enter = InStr(textstart + 1, s, Chr(13))
cnt = (enter - textstart) - 1
```

**If textstart > 0 Then**

```
q = Mid(s, textstart + 2, cnt)
str = InStr(2, q, Chr(32))
stp = InStr(2, q, Chr(13))
num = stp - str
x = Val(Right(q, num))
```

```
str = InStr(1, q, Chr(32))
stp = InStr(2, q, Chr(32))
num = stp - str
```

```
y = Val(Left(q, num))
```

```
Set litem = ListView5.ListItems.Add(, , x)
litem.SubItems(1) = y
```

**End If**

**Loop Until textstart <= 0**

**REALTIME False**

**Timer2.Enabled = True**

**End Sub**

‘Όταν επιλέξουμε ένα στοιχείο από τη λίστα και πατήσουμε το δεξί μπουτόν του ποντικιού εμφανίζεται ένα μενού με επιλογές’

**Private Sub ListView1\_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)**

```
Timer2.Enabled = False
If Button = vbRightButton Then
    PopupMenu mnuPopup
End If
```

```
End Sub
```

‘Επιλογή Set Value του μενού’

```
Private Sub mnuPopupSet_Value_Click()
```

```
    SendData ("o " & ListView1.SelectedItem.SubItems(1) & " " & InputBox("Enter
    value"))
    SENDBYTE 13
    Timer2.Enabled = True
```

```
End Sub
```

‘Επιλογή Toggle του μενού’

```
Private Sub mnuPopupToggle_Click()
```

```
    If ListView1.SelectedItem.SubItems(2) = 0 Then
        SendData ("o " & ListView1.SelectedItem.SubItems(1) & " 1")
        SENDBYTE 13
        Timer2.Enabled = True
    Else
        SendData ("o " & ListView1.SelectedItem.SubItems(1) & " 0")
        SENDBYTE 13
        Timer2.Enabled = True
    End If
```

```
End Sub
```

‘Χρήση της θήρας COM1 για σειριακή επικοινωνία’

```
Private Sub Option1_Click()
```

```
    i = OPENCOM("COM1,19200,N,8,1")
    If i = 0 Then MsgBox ("COM1 not available")
```

```
End Sub
```

‘Χρήση της θήρας COM2 για σειριακή επικοινωνία’

```
Private Sub Option2_Click()
```

```
    i = OPENCOM("COM2,19200,N,8,1")
    If i = 0 Then MsgBox ("COM2 not available")
End Sub
```

‘Χρησιμοποιούμε έναν Timer ο οποίος κάθε 500 msec μας εμφανίζει τις τιμές των ομάδων’

```
Private Sub Timer2_Timer()  
    Timer2.Enabled = False  
  
    Dim i As Integer  
    Dim litem As ListItem  
  
    For i = 1 To ListView1.ListItems.Count  
        Set litem = ListView1.ListItems(i)  
        litem.SubItems(2) = GroupValue(litem.SubItems(1))  
    Next  
  
    Timer2.Enabled = True  
End Sub
```

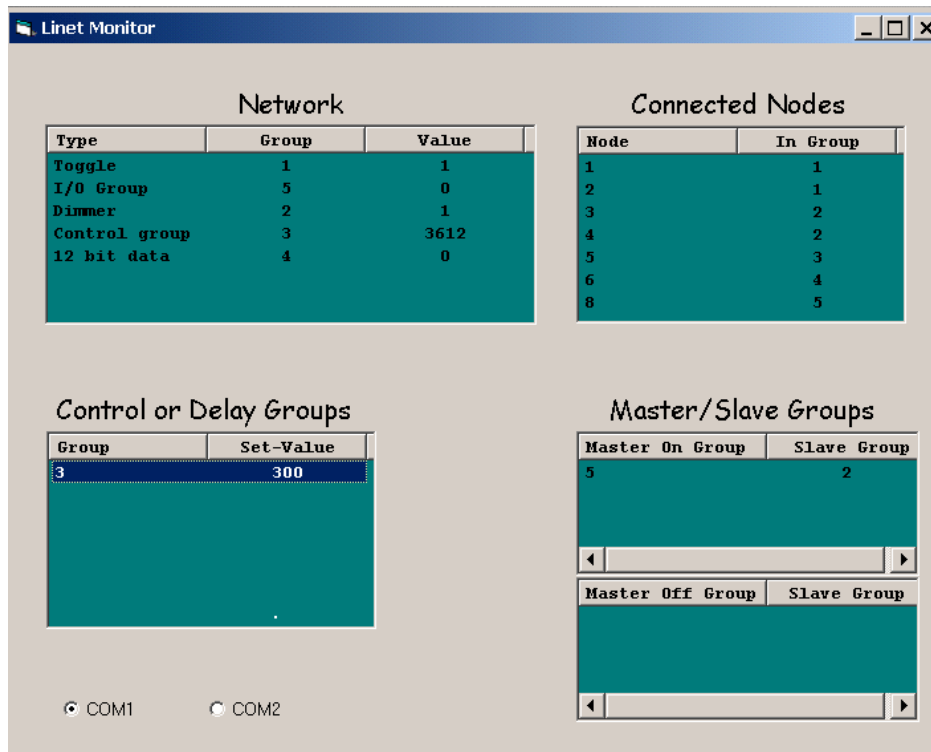
‘Όταν κλείνουμε το πρόγραμμά μας, κλείνουμε και την θήρα επικοινωνίας ώστε να μπορεί να χρησιμοποιηθεί από κάποια άλλη εφαρμογή’

```
Private Sub Form_Unload(Cancel As Integer)  
  
    REALTIME False  
    CLOSECOM  
  
End Sub
```

## 6.2.2 Χρήση του Linet Monitor

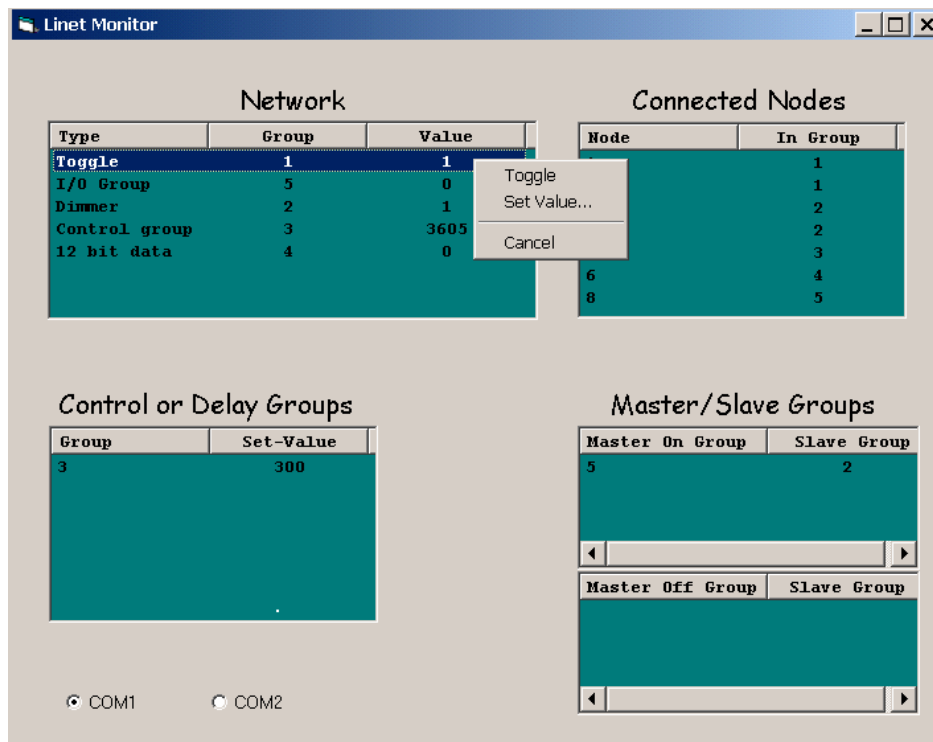
Στο Linet Monitor υπάρχουν τέσσερα παράθυρα στα οποία εμφανίζεται η κατάσταση του δικτύου μας και δύο κουμπιά για επιλογή σειριακής θήρας COM1/COM2.

Το παράθυρο Network εμφανίζει σε λίστα όλες τις ομάδες που υπάρχουν στο δίκτυο καθώς και τις τιμές που έχουν. Οι τιμές των ομάδων ανανεώνονται κάθε 500 msec περίπου. Το παράθυρο Connected Nodes εμφανίζει σε λίστα όλα τα nodes που υπάρχουν στο δίκτυο καθώς και σε ποια ομάδα ανήκουν. Αν στο δίκτυό μας υπάρχουν ομάδες Control ή Delay, αυτές εμφανίζονται στο παράθυρο Control or Delay Groups μαζί με τις τιμές set value που έχουν οριστεί. Τέλος, αν στο δίκτυό μας έχουμε ορίσει ομάδες master/slave, αυτές εμφανίζονται στο παράθυρο Master/Slave groups (εικόνα 1).



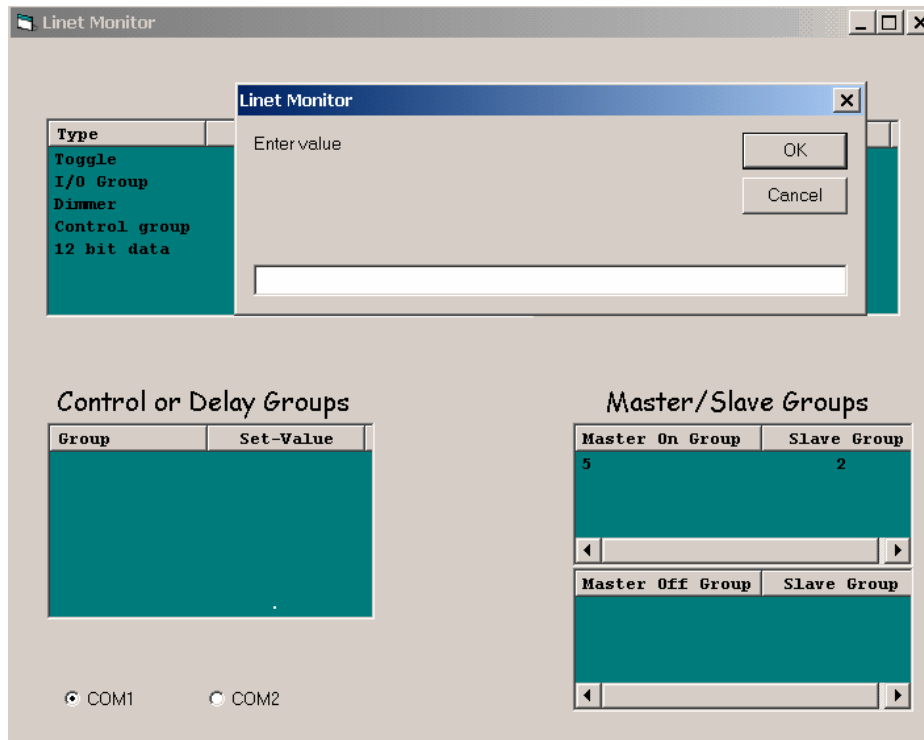
Εικόνα 1: Το Linet Monitor σε λειτουργία

Εκτός όμως από παρακολούθηση του δικτύου, το Linet Monitor μας δίνει και την δυνατότητα ελέγχου. Από το παράθυρο Network επιλέγουμε την ομάδα που θέλουμε και κάνοντας δεξιά κλικ με το ποντίκι, εμφανίζεται ένα μενού με τρεις επιλογές (εικόνα 2).



Εικόνα 2: Μενού για έλεγχο των ομάδων

Η επιλογή Toggle αντιστρέφει την έξοδο, δηλαδή από off γίνεται on και αντίστροφα. Έχει εφαρμογή σε ομάδες Toggle, Dimmer και I/O Node. Η επιλογή Set Value εμφανίζει ένα παράθυρο στο οποίο εισάγουμε μια νέα τιμή για την ομάδα που θέλουμε και έχει εφαρμογή σε όλες τις ομάδες (εικόνα 3). Τέλος, Με την επιλογή Cancel βγαίνουμε από το μενού.



Εικόνα 3: Επιλογή Set Value

### 6.3 LINET CLIENT

Το Linet Client είναι ένα πρόγραμμα που κάνει την ίδια δουλειά με το Linet Monitor (παρακολούθηση και έλεγχο του δικτύου), με την διαφορά ότι χρησιμοποιεί την θήρα Ethernet για ανταλλαγή δεδομένων μεταξύ controller και PC. Αυτό μας δίνει την δυνατότητα να μπορούμε να ελέγχουμε το δίκτυό μας από μεγάλη απόσταση, γνωρίζοντας την IP address του controller. Ας δούμε όμως πως γίνεται η επικοινωνία με την θήρα αυτή.

Ο controller χρησιμοποιεί το πρωτόκολλο επικοινωνίας IP/UDP, έχει λοιπόν μια UDP πόρτα για αυτό το σκοπό. Ο αριθμός πόρτας και η IP address του controller μπορούν να αλλάξουν κατά την διαμόρφωση του δικτύου, εκτός και αν πρόκειται για δοκιμαστική έκδοση controller (όπως ο controller που επεξεργάστηκα), όπου η IP address είναι 192.168.42.13 και ο αριθμός πόρτας είναι 1313.

Ο controller στέλνει δεδομένα μόνο όταν του ζητηθεί από το δίκτυο (ή το host PC) που είναι συνδεδεμένος. Για κάθε πακέτο αίτησης (request packet) στέλνεται ένα πακέτο απάντησης (response packet). Το response packet στέλνεται στην IP address και στον αριθμό πόρτας που έστειλε το request packet.



- UDP packet header

Σε κάθε πακέτο υπάρχει μια κοινή επικεφαλίδα (header) 4 bytes που περιλαμβάνει:

BYTE OFFSET	BYTES	FIELD
0	1	Έκδοση πρωτοκόλλου , για τώρα 0
1	1	Τύπος πακέτου
2	2	Σημαίες, δεσμευμένα bytes

Οι τύποι πακέτων που έχουν οριστεί είναι:

CODE	TYPE
0	Status request
1	Status response
2	Structure request
3	Structure response

- Network status packets

Ένα Network status packet διαβάζει ή αλλάζει τις τιμές των ομάδων που υπάρχουν στο δίκτυό μας. Αποτελείται από το packet header ακολουθούμενο από ένα πίνακα 200 θέσεων, σε κάθε μία από αυτές τις θέσεις αντιστοιχούν 4 bytes. Κάθε θέση περιγράφει και μια ομάδα, η πρώτη καταχώρηση είναι η ομάδα 1. Τα 4 bytes περιλαμβάνουν:

BYTE OFFSET	BYTES	FIELD
0	1	Τύπος ομάδας
1	1	Σημαία
2	2	Τιμή ομάδας

Όταν το bit 0 της σημαίας είναι on, έχουμε δηλαδή αίτημα για αλλαγή της τιμής της ομάδας, η τιμή αλλάζει και γίνεται ίση με την τιμή ομάδας που βρίσκεται στο πακέτο. Όταν είναι off, η τιμή ομάδας παραμένει ίδια.

Τύποι ομάδων που έχουν οριστεί:

TYPE CODE	GROUP TYPE	VALUE TYPE
0	NONE	None
1	TOGGLE	On/off
2	DIMMER	0=off, 1=on, 2=step up, 3=step down
3	IOGROUP	1 bit I/O
4	XLAMP	1 bit I/O
5	LMON	1 bit I/O
6	XDELAY	16 bit, delay value /10 ms
7	WCALL	Waiter call
8	DATA EXCH.	Not usable
9	DATA8	8 bit, data value
10	DATA12	12 bit, data value
11	DATA16	16 bit, data value
12	AD/STATE	AD value (12 bit)
13	CONTROL	AD value (12 bit)

### 6.3.1 Κώδικας του Linet Client

Χρησιμοποιούμε ένα module (το Linet.BAS) όπου δηλώνουμε τις μεταβλητές, τις σταθερές, τις δομές καθώς και τις συναρτήσεις που χρησιμοποιούνται συχνά στο πρόγραμμά μας. Παρακάτω έχουμε τον κώδικα του module.

```
Declare Sub MoveMemory Lib "kernel32" Alias "RtlMoveMemory" (pDest As Any, pSource As Any, ByVal dwLength As Long)
```

```
Public Const LINET_DEFAULT_ADDRESS = "192.168.1.13"  
Public Const LINET_DEFAULT_PORT = 1313
```

```
Public Const LINET_CURRENT_PROTOCOL_VERSION = 0  
Public Const LINET_STATUS_REQUEST = 0  
Public Const LINET_STATUS_RESPONSE = 1  
Public Const LINET_STRUCTURE_REQUEST = 2  
Public Const LINET_STRUCTURE_RESPONSE = 3
```

‘Επικεφαλίδα (header)’

```
Public Type tLinetHeader  
    ProtocolVersion As Byte  
    PacketType As Byte  
    Flags(0 To 1) As Byte  
End Type
```

‘Δεδομένα ομαδας’

```
Public Type tLinetStatusPacket  
    GroupType As Byte  
    Flags As Byte  
    GroupValue(0 To 1) As Byte  
End Type
```

‘Ολοκληρωμένο πακέτο δεδομένων’

```
Public Type tLinetPacket  
    Header As tLinetHeader  
    StatusPackets(0 To 199) As tLinetStatusPacket  
End Type
```

‘Μετατροπή του Linet packet σε χαρακτήρες για αποστολή προς το δίκτυο’

```
Public Function LinetPacketToString(lp As tLinetPacket) As String  
    Dim sOut As String  
    Dim I As Integer
```

```
sOut = Chr(lp.Header.ProtocolVersion) & _  
      Chr(lp.Header.PacketType) & _  
      Chr(lp.Header.Flags(0)) & _  
      Chr(lp.Header.Flags(1))
```

```
For I = 0 To 199  
  With lp.StatusPackets(I)  
    sOut = sOut & Chr(.GroupType) & _  
            Chr(.Flags) & _  
            Chr(.GroupValue(0)) & _  
            Chr(.GroupValue(1))  
  End With  
Next I
```

```
LinetPacketToString = sOut  
End Function
```

‘Μετατροπή χαρακτήρων σε δομή tLinetPacket’

```
Public Function StringToLinetPacket(ByVal strPacket As String) As tLinetPacket  
  Dim lp As tLinetPacket  
  Dim I As Integer
```

```
  With lp.Header  
    .ProtocolVersion = Asc(Mid(strPacket, 1, 1))  
    .PacketType = Asc(Mid(strPacket, 2, 1))  
    .Flags(0) = Asc(Mid(strPacket, 3, 1))  
    .Flags(1) = Asc(Mid(strPacket, 4, 1))  
  End With
```

```
  For I = 0 To 199  
    With lp.StatusPackets(I)  
      .GroupType = Asc(Mid(strPacket, I * 4 + 5, 1))  
      .Flags = Asc(Mid(strPacket, I * 4 + 6, 1))  
      .GroupValue(0) = Asc(Mid(strPacket, I * 4 + 7, 1))  
      .GroupValue(1) = Asc(Mid(strPacket, I * 4 + 8, 1))  
    End With  
  Next I
```

```
  StringToLinetPacket = lp  
End Function
```

Στην συνέχεια έχουμε τον κώδικα του κυρίως προγράμματος:

### **Option Explicit**

‘ Linet UDP packet’

### **Private Packet As tLinetPacket**

‘Όταν πατηθεί το μπουτόν Connect..’

```
Private Sub cmdConnect_Click()  
If cmdConnect.Caption = "&Connect" Then  
cmdConnect.Caption = "Dis&connect"  
cmdConnect.Enabled = False
```

‘Bind στην UDP port του controller’

```
WinsockClient.Bind 0  
WinsockClient.RemotePort = txtPort  
WinsockClient.RemoteHost = txtAddress  
  
tmrPoll_Timer  
cmdConnect.Enabled = True  
Else  
WinsockClient.Close  
tmrPoll.Enabled = False  
cmdConnect.Caption = "&Connect"  
End If  
End Sub
```

‘Όταν αλλάζουμε το μέγεθος της φόρμα του προγράμματος, προσαρμόζονται αυτόματα και τα ελεγκτήρια’

```
Private Sub Form_Resize()  
If Not Me.WindowState = vbMinimized Then  
lsvwNodes.Width = Me.ScaleWidth - lsvwNodes.Left * 2  
lsvwNodes.Height = Me.ScaleHeight - lsvwNodes.Top - lsvwNodes.Left  
End If  
End Sub
```

‘Όταν επιλέξουμε ένα στοιχείο από τη λίστα και πατήσουμε το δεξί μπουτόν του ποντικιού εμφανίζεται ένα μενού με επιλογές’

```
Private Sub lsvwNodes_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)  
If Button And vbRightButton Then  
If Not lsvwNodes.SelectedItem Is Nothing Then  
PopupMenu mnuPopup  
End If
```

```
End If
End Sub
```

‘Επιλογή Toggle του μενού’

```
Private Sub mnuPopupToggle_Click()
    With Packet.StatusPackets(IsvwNodes.SelectedItem.Text - 1)
        .GroupValue(0) = 0

        If .GroupValue(1) Then
            .GroupValue(1) = 0
        Else
            .GroupValue(1) = 1
        End If

        .Flags = 1
    End With
End Sub
```

‘Επιλογή Set Value του μενού’

```
Private Sub mnuPopupSetValue_Click()
    Dim iIn As Integer

    iIn = Val(InputBox("Enter node value", "Set value"))

    With Packet.StatusPackets(IsvwNodes.SelectedItem.Text - 1)
        .GroupValue(0) = iIn \ 255
        .GroupValue(1) = iIn And 255
        .Flags = 1
    End With
End Sub
```

‘Έλεγχος της κατάστασης του Linet κάθε 1000 msec’

```
Private Sub tmrPoll_Timer()
    tmrPoll.Enabled = False

    With Packet.Header
        .ProtocolVersion = LINET_CURRENT_PROTOCOL_VERSION
        .PacketType = LINET_STATUS_REQUEST
    End With

    WinsockClient.SendData LinetPacketToString(Packet)
    tmrPoll.Enabled = True
End Sub
```

‘Έλεγχος για σωστή δήλωση του UDP port’

```
Private Sub txtPort_LostFocus()  
    If Not IsNumeric(txtPort) Then  
        MsgBox "Port must be numeric. Default is " & LINET_DEFAULT_PORT  
        txtPort.SetFocus  
    End If  
End Sub
```

‘Λήψη και επεξεργασία δεδομένων’

```
Private Sub WinsockClient_DataArrival(ByVal bytesTotal As Long)  
    Dim sDataIn As String  
    Dim I As Integer  
    Dim Item As MSCComctlLib.ListItem  
  
    WinsockClient.GetData sDataIn  
    Packet = StringToLinetPacket(sDataIn)  
  
    'IsvwNodes.ListItems.Clear  
  
    For I = 0 To 199  
        Set Item = FindListItem(I + 1)  
  
        Select Case Packet.StatusPackets(I).GroupType  
            Case 4, 7, 8  
                ' do nothing  
  
            Case Else  
                If Packet.StatusPackets(I).GroupType = 0 Then  
                    If Not Item Is Nothing Then  
                        IsvwNodes.ListItems.Remove Item.Index  
                    End If  
                Else  
                    If Item Is Nothing Then  
                        Set Item = IsvwNodes.ListItems.Add(, I + 1)  
                    End If  
  
                    With Item  
                        .SubItems(1) = LoadResString(Packet.StatusPackets(I).GroupType)  
  
                        Select Case Packet.StatusPackets(I).GroupType  
                            Case 1, 2, 4, 5  
                                If Packet.StatusPackets(I).GroupValue(1) Then  
                                    .SubItems(2) = "on"  
                                Else  
                                    .SubItems(2) = "off"  
                                End If  
  
                            Case 8
```

```

        .SubItems(2) = "not usable"

    Case Else
        .SubItems(2) = Packet.StatusPackets(I).GroupValue(0) * 256 + _
            Packet.StatusPackets(I).GroupValue(1)

    End Select
End With
End If
End Select
Next I
End Sub

```

‘Αναζήτηση ενός στοιχείου του ελεγκτηρίου ListView, με βάση το κείμενό του (text property).’

```

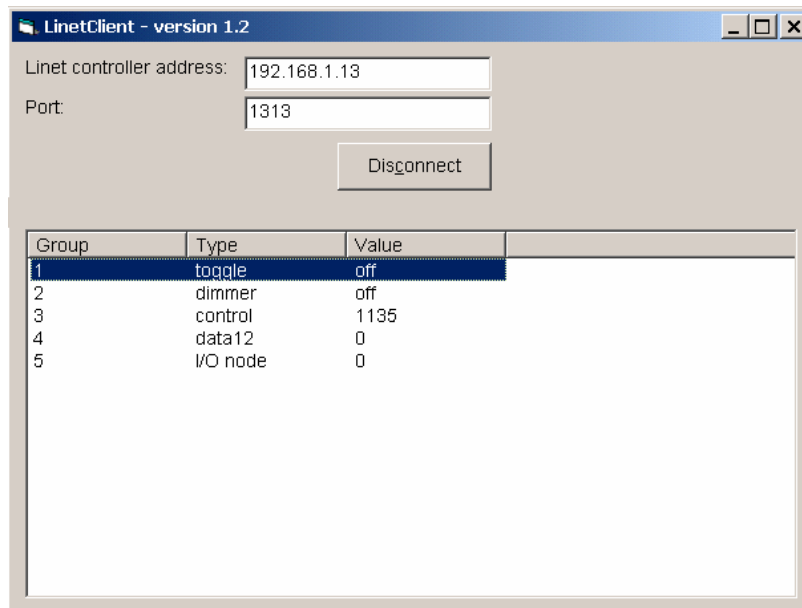
Private Function FindListItem(ByVal Text As String) As MSComctlLib.ListItem
    Dim TempListItem As MSComctlLib.ListItem

    For Each TempListItem In lsvwNodes.ListItems
        If TempListItem.Text = Text Then
            Set FindListItem = TempListItem
            Exit For
        End If
    Next TempListItem
End Function

```

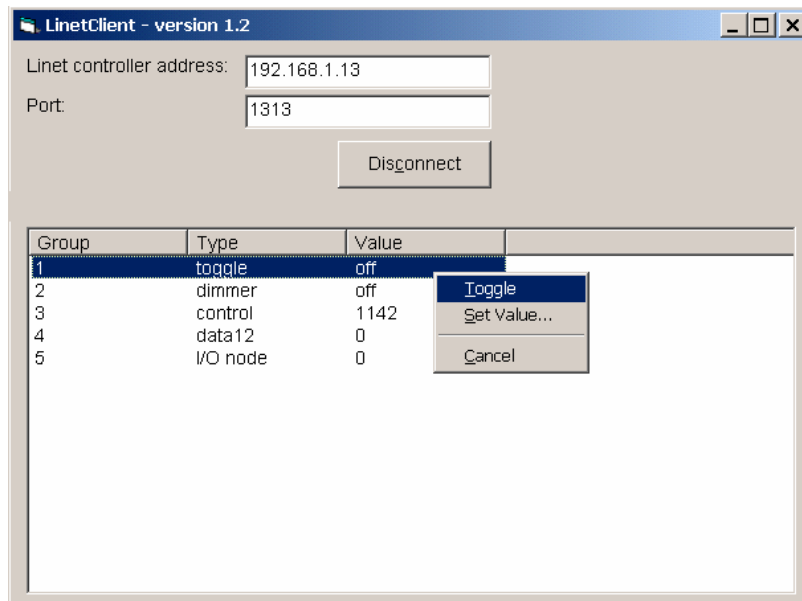
### 6.3.2 Χρήση του Linet Client

Στο Linet Client υπάρχουν δύο κελιά για εισαγωγή του IP address και UDP port του controller μας, ένα μπουτόν για σύνδεση και αποσύνδεση από το δίκτυο (ή το host PC) και τέλος ένα παράθυρο όπου βλέπουμε ή μεταβάλλουμε την κατάσταση του δικτύου μας (εικόνα 4).



**Εικόνα 4:** Το Linet Client σε λειτουργία

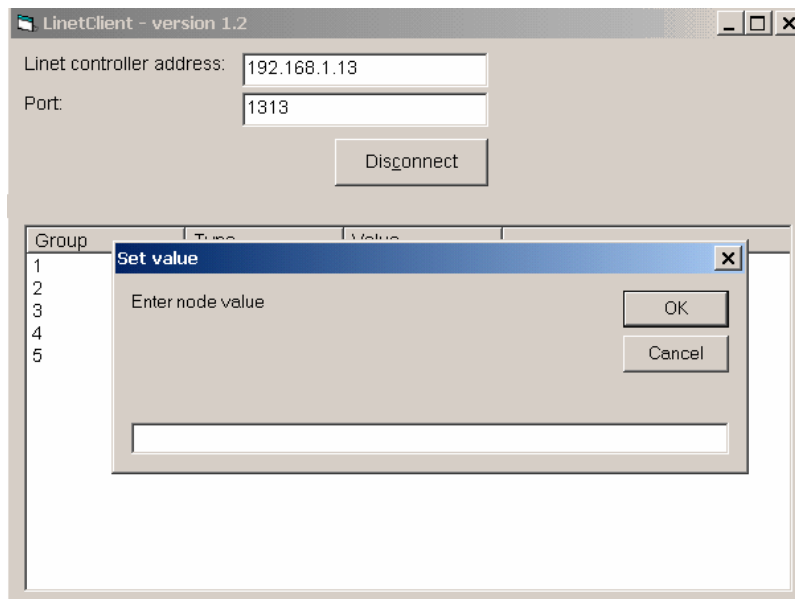
Για να μεταβάλουμε την τιμή μιας ομάδας, επιλέγουμε την ομάδα που θέλουμε και κάνοντας δεξί κλικ με το ποντίκι, εμφανίζεται ένα μενού με τρεις επιλογές (εικόνα 5).



**Εικόνα 5:** Μενού για έλεγχο των ομάδων

Η επιλογή Toggle αντιστρέφει την έξοδο, δηλαδή από off γίνεται on και αντίστροφα. Έχει εφαρμογή σε ομάδες Toggle, Dimmer και I/O Node. Η επιλογή Set Value εμφανίζει ένα παράθυρο στο οποίο εισάγουμε μια νέα τιμή για την ομάδα που θέλουμε και έχει εφαρμογή σε όλες τις ομάδες (εικόνα 6). Τέλος, Με την επιλογή Cancel βγαίνουμε από το μενού.





Εικόνα 6: Επιλογή Set Value

## 6.4 TEMP\_SENSOR

Το Temp\_sensor είναι ένα πρόγραμμα το οποίο μετράει θερμοκρασία με το αισθητήριο LM35 και την εμφανίζει σε ένα LCD display καθώς και στην οθόνη του υπολογιστή. Θα χρειαστεί να δεσμεύσουμε δυο nodes για τον σκοπό αυτό. Στο ένα θα συνδέσουμε το LM35 στην αναλογική του είσοδο και στο άλλο ένα LCD display. Και τα δυο nodes θα ορισθούν σαν 12 bit data group. Θα χρησιμοποιήσουμε την σειριακή θήρα του controller για επικοινωνία με το PC. Ας δούμε όμως ποια είναι η λογική του προγράμματός.

Το LM35 είναι ένα αισθητήριο θερμοκρασίας το οποίο αφού τροφοδοτηθεί με 4-20 V στην είσοδό του, μας δίνει 10mV ανά βαθμό Κελσίου στην έξοδο. Από την άλλη, όταν στην αναλογική είσοδο του node εφαρμοστεί μια τάση 0-1,25 V , ενώ έχει ενεργοποιηθεί ο A/D converter του node, η τάση αυτή μετατρέπεται σε 12 bit αριθμό. Έτσι όταν συνδέσουμε την έξοδο του LM35 με την αναλογική είσοδο του node, έχουμε ένα 12 bit αριθμό για κάθε θερμοκρασία που μετρά το LM35. Αυτός ο αριθμός στέλνεται από τον controller στο PC μέσω της σειριακής, το πρόγραμμα αναλαμβάνει να τον μετατρέψει σε βαθμούς Κελσίου και να τον στείλει πίσω στον controller για να τον εμφανίσει στο LCD display σαν βαθμό Κελσίου.

### 6.4.1 Κώδικας του Temp\_sensor

Το Temp\_sensor χρησιμοποιεί δύο modules, το PORTS.BAS και το Synarthseis.BAS, τα οποία και εξετάσαμε προηγουμένως. Ας δούμε λοιπόν τον κώδικα του κυρίως προγράμματος.

```
Option Explicit  
Dim i As Integer  
Dim y As Variant  
Dim z As Variant  
Dim c As Integer
```

```
Private Sub Form_Load()  
CLOSECOM
```

‘Ζητάμε από τον χρήστη την ομάδα του αισθητήρα, την ομάδα του LCD display καθώς και τον αριθμό της θήρας που θα χρησιμοποιήσουμε’

```
z = InputBox("Group of sensor?")  
y = InputBox("Group of display?")  
c = InputBox("COM?")
```

‘Ανοίγουμε την θήρα σειριακής επικοινωνίας που επέλεξε ο χρήστης’

```
If c = 1 Then  
OPENCOM ("COM1,19200,N,8,1")  
ElseIf c = 2 Then  
OPENCOM ("COM2,19200,N,8,1")  
ElseIf c = 3 Then  
OPENCOM ("COM3,19200,N,8,1")  
Else: MsgBox ("Error!")  
End If
```

```
DTR 0  
RTS 0
```

```
End Sub
```

‘Κάθε 2000 msec παίρνουμε τις μετρήσεις από το αισθητήριο, τις μετατρέπουμε σε βαθμούς Κελσίου και τις εμφανίζουμε στο LCD display’

```
Private Sub Timer2_Timer()  
Dim x As Variant, dat As Integer
```

```
REALTIME True
```

‘Δίνουμε εντολή στον controller να μας στείλει την μέτρηση του αισθητήρα’

```
Call SendData("i " + z + " ")  
SENDBYTE 13
```

‘Παίρνουμε την μέτρηση από τον αισθητήρα, σε 12 bit αριθμό και την αποθηκεύουμε σε μια μεταβλητή’

```
Do  
dat = READBYTE
```

```

If (dat >= 48) And (dat <= 57) Then
  x = x + Chr$(dat)
End If
Loop Until dat = -1

```

‘Μετατρέπουμε τον 12 bit αριθμό σε βαθμούς Κελσίου’

```

If IsEmpty(x) = True Or (x = "0") Or (x = "1") Then
  Text1.Text = "Error!"
  CLOSECOM
Else
  x = ((Val(x) + 83.4) / 32) + 4
  Text1.Text = Format(x, "####.000") + " °C"
End If

x = Str(CInt(x))

```

‘Στέλνουμε την θερμοκρασία στο LCD display (σε βαθμούς Κελσίου)’

```

Call SendData("o " + y + " ")
Call SendData(x + " ")
SENDBYTE 13

```

```

REALTIME False
End Sub

```

‘Όταν κλείνουμε το πρόγραμμά μας, κλείνουμε και την θήρα επικοινωνίας ώστε να μπορεί να χρησιμοποιηθεί από κάποια άλλη εφαρμογή’

```

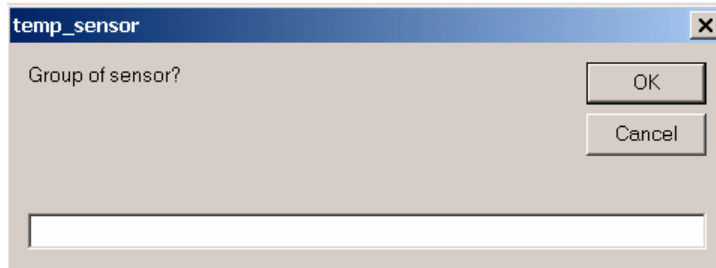
Private Sub Form_Unload(Cancel As Integer)
  CLOSECOM
  REALTIME False
End Sub

```

#### 6.4.2 Χρήση του Temp\_sensor

Στο Temp\_sensor υπάρχει ένα παράθυρο όπου εμφανίζεται η μετρούμενη θερμοκρασία από το LM35, δεν υπάρχει κάποια άλλη επιλογή ή δυνατότητα.

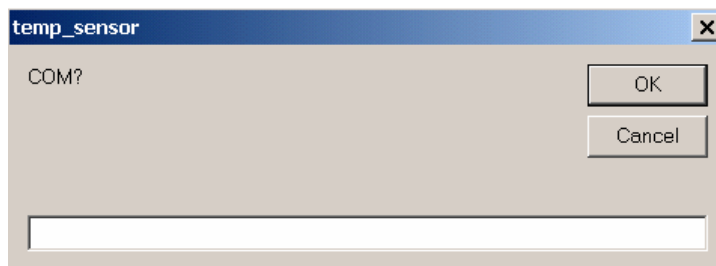
Με την εκκίνηση του προγράμματος, μας ζητείται να εισάγουμε τον αριθμό ομάδας που βρίσκεται ο αισθητήρας (εικόνα 7), τον αριθμό ομάδας που βρίσκεται το LCD display (εικόνα 8), καθώς και τον αριθμό της θήρας που θα χρησιμοποιήσουμε για σειριακή επικοινωνία (εικόνα 9). Όταν όλα τα παραπάνω γίνουν σωστά, το πρόγραμμά εμφανίζει την μετρούμενη τιμή στο παράθυρο που διαθέτει για αυτό το σκοπό (εικόνα 10), ενώ παράλληλα την εμφανίζει και στο LCD display.



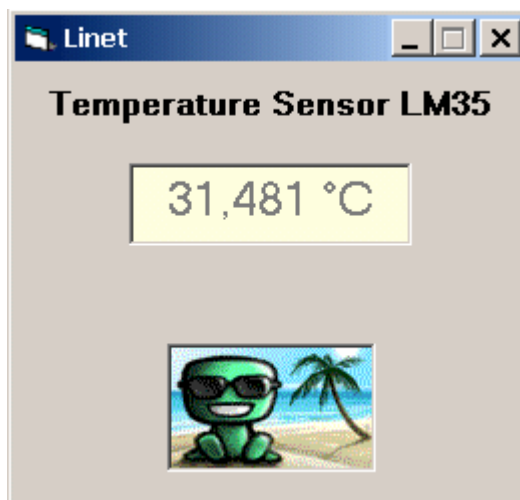
**Εικόνα 7:** Το πρόγραμμα ζητάει να εισάγουμε την ομάδα του αισθητήρα



**Εικόνα 8:** Το πρόγραμμα ζητάει να εισάγουμε την ομάδα του LCD display



**Εικόνα 9:** Το πρόγραμμα ζητάει να εισάγουμε τον αριθμό της θήρας σειριακής επικοινωνίας



**Εικόνα 10:** Το Temp\_sensor σε λειτουργία

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Διαβάζοντας κάποιος την εργασία αυτή, θα έχει σίγουρα βγάλει τα συμπεράσματα του για το Linet, πως μπορεί να το χρησιμοποιήσει, σε ποιες εφαρμογές και αν τον συμφέρει οικονομικά μια τέτοια κίνηση. Θα ήθελα όμως να αναφέρω επιγραμματικά και τα προσωπικά μου συμπεράσματα

Το Linet είναι ένα σύστημα που μπορεί κάποιος εύκολα να κατανοήσει τον τρόπο λειτουργίας του και να το χειριστεί, δεν χρειάζονται γνώσεις προγραμματισμού ή δικτύων. Το κόστος υλοποίησης μιας εφαρμογής με Linet είναι χαμηλό, σε σχέση με άλλα παρόμοια συστήματα, αφού δεν χρειάζονται εξειδικευμένα προγράμματα ή άλλες συσκευές. Ακόμα και η διαμόρφωση του δικτύου γίνεται με οποιοδήποτε πρόγραμμα terminal, που τρέχει σε κάποιο ηλεκτρονικό υπολογιστή. Σε μια εφαρμογή που έχει υλοποιηθεί με το Linet μπορούμε εύκολα να κάνουμε αλλαγές, διορθώσεις ή ακόμα και να την επεκτείνουμε χωρίς μεγάλο οικονομικό κόστος. Τέλος, το Linet έχει αποθηκευμένες κάποιες συγκεκριμένες λειτουργίες τις οποίες και επιλέγουμε κατά την φάση της διαμόρφωσης. Αυτές οι λειτουργίες είναι αρκετές για να την κατασκευή πολλών εφαρμογών, όταν όμως θέλουμε κάτι παραπάνω μπορούμε να συνδέσουμε στον controller κάποια άλλη συσκευή (π.χ. μικροελεγκτή) και να αναλάβει την “δύσκολη” δουλειά.

Ας δούμε όμως και τα αρνητικά του σημεία. Το Linet είναι αρκετά αργό σύστημα και αυτό οφείλεται κυρίως στο ότι τα δεδομένα που διακινούνται μεταξύ του controller και των nodes μεταφέρουν πληροφορίες για 200 nodes ακόμα και αν έχουμε συνδέσει λιγότερα στο σύστημά μας. Ένα δεύτερο αρνητικό του Linet είναι ο AD/converter που έχουν ενσωματωμένα τα nodes, είναι χαμηλής ακρίβειας που τον κάνει ουσιαστικά άχρηστο σε μια εφαρμογή με απαιτήσεις. Το ότι το Linet έχει κάποιες συγκεκριμένες λειτουργίες μπορεί να είναι θετικό για την κατανόηση και χρήση του, είναι όμως αρνητικό για την λειτουργικότητα και την χρησιμοποίησή του σε πιο περίπλοκες εφαρμογές. Για να καλύψει αυτή του την αδυναμία, μπορεί να δεχτεί βοήθεια από κάποια άλλη συσκευή αλλά τότε ανεβαίνει το κόστος την εφαρμογής ενώ παράλληλα χρειάζονται γνώσεις προγραμματισμού και ηλεκτρονικών.

Από όλα τα παραπάνω προκύπτει ότι η επιλογή του Linet για μια εφαρμογή συμφέρει (οικονομικά και σε χρόνο κατασκευής), υπό κάποιες προϋποθέσεις. Να είναι απλή εφαρμογή, ώστε να μπορεί να καλυφθεί από τις λειτουργίες του Linet και να μην χρειάζεται κάποια βοηθητική συσκευή. Τέλος, να μην χρειάζεται ακρίβεια στις μετρήσεις, ούτε μεγάλη ταχύτητα στην μεταφορά δεδομένων.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- E. Petroustos, Πλήρες εγχειρίδιο της Visual Basic 6
- S. Warner, Teach yourself Visual Basic 6
- B. Kainka, Ηλεκτρονικά με τη Visual Basic – πειράματα με το PC
- Γ. Μηναδάκης, Εργαστήριο Μικροϋπολογιστών
- Θ. Τσιλιγκιρίδης / Γ. Αλεξίου, Μετάδοση δεδομένων και δίκτυα υπολογιστών I & II
  
- <http://www.linet-network.com>
- <http://www.Planet-Source-Code.com>
- <http://www.elektor.gr>