A decorative graphic on the right side of the page. It features three concentric blue circles of different sizes, each with a lighter blue outer ring. Two thin blue lines originate from the top left and extend diagonally towards the circles. A large, partially visible blue circle is at the bottom right.

Σχεδίαση και κατασκευή μιας πλατφόρμας υλοποίησης εφαρμογών BACnet και πειραματική χρήση της

Design and implementation of a BACnet
application platform and experimental usage

Μιχαήλ Νίκος
Πολυτεχνείο Κρήτης
Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών
Υπολογιστών
12/12/2014

ΕΠΙΒΛΕΠΩΝ:

ΚΑΛΑΪΤΖΑΚΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ – ΚΑΘΗΓΗΤΗΣ

Εξεταστική επιτροπή

Καθηγητής Κωνσταντίνος Καλαϊτζάκης
Επίκουρη Καθηγήτρια Διονυσία Κολοκοτσά
Διδάσκων Π.Δ 407/80 Αντώνης Τσικαλάκης

Περιεχόμενα

Περίληψη.....	6
Κεφάλαιο 1	
Εισαγωγή	8
1.1 Γενικά για την εργασία	8
1.2 Προηγούμενες Εργασίες πάνω στο αντικείμενο	9
Κεφάλαιο 2	
Το πρωτόκολλο BACnet.....	11
2.1 Εισαγωγή στο πρωτόκολλο BACnet.....	11
2.2 Αρχιτεκτονική του πρωτοκόλλου	12
2.3 BACnet Application Layer	13
2.3.1 <i>BACnet Object Model</i>	13
2.3.2 <i>BACnet Object Properties</i>	15
2.3.3 <i>BACnet Object Types</i>	15
2.3.4 <i>BACnet Services</i>	16
2.3.5 <i>BACnet Service descriptions</i>	18
2.3.6 <i>BACnet Service Types</i>	20
2.4 BACnet Network Layer.....	21
2.4.1 <i>NL Protocol Data Unit</i>	21
2.4.2 <i>Περιγραφή των μηνυμάτων του Network Layer</i>	23
2.4.3 <i>Διανομή μηνυμάτων σε πολλαπλούς αποδέκτες</i>	25
2.4.3.1 <i>BACnet Multicast</i>	26
2.4.3.2 <i>BACnet Broadcast</i>	26
2.5 BACnet Data Link Layer.....	27
2.5.1 <i>Master Slave / Token Passing (MS/TP)</i>	27
2.5.2 <i>Point-To-Point (PTP)</i>	30
2.6 BACnet Physical Layer.....	34
2.6.1 <i>Εισαγωγή στα σειριακά πρωτόκολλα</i>	34
2.6.2 <i>Το πρωτόκολλο RS-232</i>	34
2.6.3 <i>Το πρωτόκολλο RS-485</i>	36
2.7 Ειδικεύοντας το BACnet.....	38

2.7.1 Interoperability Areas (IAs)	38
2.7.2 Device Profiles	39
2.7.3 BACnet Interoperability Building Block (BIBB)	40
2.7.4 Protocol Implementation Conformance Statement (PICS).....	41
Κεφάλαιο 3	42
Σχεδίαση και Υλοποίηση Πλατφόρμας ανάπτυξης εφαρμογών	42
3.1 Εισαγωγή	42
3.2 Ανάλυση σχεδίασης.....	43
3.2.1 Ολοκληρωμένα που χρησιμοποιήθηκαν	43
3.2.2 Σχεδίαση του κυκλώματος.....	45
3.3 Υλοποίηση κυκλώματος.....	50
3.4 Αναλυτική κοστολόγηση.....	55
Κεφάλαιο 4
Υλοποίηση επιπέδου εφαρμογής	58
4.1 Εισαγωγή	58
4.2 BACnet Stack.....	58
4.3 BACnet MS/TP Atmega 644	59
4.3.1 Stack Documentation	59
4.3.2 Ανάλυση του κώδικα του Stack	60
4.4 Ανάλυση απαιτήσεων επιπέδου εφαρμογής –Εξοπλισμός	66
4.4.1 Αισθητήρες που χρησιμοποιήθηκαν	66
4.4.1 Το πρωτόκολλο επικοινωνίας I2C.....	69
4.5 Επίπεδο Εφαρμογής /Υλοποίηση	70
Κεφάλαιο 5
Πειραματικά αποτελέσματα	75
5.1 Εισαγωγή	75
5.2 Επικοινωνία με τον Υπολογιστή	75
5.2.1 Windows	76
5.2.1 Linux.....	79
5.3 Πειραματικά αποτελέσματα αισθητήρων.....	82
Κεφάλαιο 6
Συμπεράσματα - Προτάσεις για μελλοντική έρευνα	85
Βιβλιογραφία, Ηλεκτρονικοί Σύνδεσμοι.....	86

Περίληψη

Τα συστήματα αυτοματισμών που βρίσκουν πρακτική εφαρμογή στα συστήματα διαχείρισης κτιρίων (Building management Systems, BMS) αποτελούν ένα πολύ ενδιαφέρον και χρήσιμο πεδίο ακαδημαϊκής έρευνας και ενασχόλησης για τους ηλεκτρονικούς μηχανικούς. Στην εποχή που το κόστος της ενέργειας (οικονομικό, περιβαντολογικό) θα αρχίσει να αυξάνεται γεωμετρικά, λόγω εξάντλησης των ορυκτών καυσίμων, τα συστήματα αυτοματισμών ανοίγουν τον δρόμο για μια πιο ορθολογική διαχείριση της ενέργειας που παράγεται.

Τα συστήματα αυτοματισμών είναι όμως και μια τεράστια αγορά με εκατοντάδες διαφορετικούς κατασκευαστές. Για να γίνει πιο αποδοτική η σχεδίαση και η κατασκευή συστημάτων με την συμμετοχή στο ίδιο σύστημα κομματιών από πολλούς κατασκευαστές η αγορά έχει αρχίσει να κατευθύνεται προς την υιοθέτηση ανοιχτών πρωτοκόλλων επικοινωνίας που επιτρέπουν την δυνατότητα αυτή.

Ένα από τα πρώτα και το πιο γνωστά τέτοιου είδους πρωτόκολλα είναι το BACnet (Building Automation and Control networking protocol). Στην παρούσα εργασία δημιουργήθηκε ένα σύστημα που μπορεί να χρησιμοποιηθεί σαν εργαλείο ανάπτυξης εφαρμογών με το συγκεκριμένο πρωτόκολλο σε ένα τοπικό MS/TP (Master Slave Token Passing) δίκτυο. Το σύστημα αυτό κατασκευάστηκε από την αρχή, είναι πλήρως λειτουργικό και δημιουργήθηκε μια πρότυπη εφαρμογή καταγραφής δεδομένων από αισθητήρες.

Με την διπλωματική αυτή ο υπογράφων θεωρεί ότι αφήνει μια καλή παρακαταθήκη για όποιον θέλει να ασχοληθεί με το συγκεκριμένο πρωτόκολλο και να δημιουργήσει πρακτικές εφαρμογές σε ένα BMS σύστημα.

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά για την εργασία

Το παρόν κείμενο είναι μια πλήρης τεχνική αναφορά της διπλωματικής εργασίας που εκπόνησε ο υπογράφων, Μιχαήλ Νικόλαος, στη Σχολή Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών του Πολυτεχνείου Κρήτης. Η εργασία εκπονήθηκε με την επίβλεψη και αξιολόγηση του καθηγητή του τμήματος Κωνσταντίνου Καλαϊτζάκη αλλά και σε συνεργασία με τον υποψήφιο διδάκτορα του τμήματος Μηχανικών Περιβάλλοντος του Πολυτεχνείου Κρήτης, Κωνσταντίνου Γομπάκη. Η εργασία εκπονήθηκε από τον Φεβρουάριο του 2014 μέχρι τον Δεκέμβριο του ίδιου χρόνου

Αντικείμενο της εργασίας είναι η δημιουργία μιας ολοκληρωμένης πλατφόρμας ανάπτυξης εφαρμογών του πρωτοκόλλου επικοινωνίας BACnet και η πειραματική χρήση της πλατφόρμας σε ένα πραγματικό MS/TP (Master Slave Token Passing) δίκτυο. Η ανάπτυξη της πλατφόρμας περιλαμβάνει την σχεδίαση εκτύπωση και κατασκευή πλακέτας, με μικροεπεξεργαστή αρχιτεκτονικής AVR μαζί με τα απαραίτητα περιφερειακά ολοκληρωμένα για μια επιτυχημένη επικοινωνία σε ένα δίκτυο, την δημιουργία μιας ολοκληρωμένης εφαρμογής στην πλατφόρμα που να διαβάζει μια πληθώρα εξωτερικών αισθητήρων διαφόρων τύπων (Αναλογικοί, Ψηφιακοί) και οι καταγραφή των πειραματικών δεδομένων από το σύστημα για ένα ικανοποιητικό χρονικό διάστημα.

Στα επόμενα 5 κεφάλαια του παρόντος κειμένου θα ακολουθήσει μια πλήρης περιγραφή όλων των σταδίων της εργασίας. Πιο αναλυτικά:

Κεφάλαιο 2-Το πρωτόκολλο BACnet: Περιγραφή του πρωτοκόλλου BACnet. Ανάλυση της σχεδίασης του και της διαστρωμάτωσης του σε επίπεδα. Περιγραφή της λειτουργικότητας κάθε επιπέδου και των ξεχωριστών πρωτοκόλλων που αντιστοιχούν σε αυτά. Αναφορά και ανάλυση της τεκμηρίωσης συσκευών που υλοποιούν το πρωτόκολλο BACnet.

Κεφάλαιο 3-Σχεδίαση και Υλοποίηση πλατφόρμας εφαρμογών: Περιγραφή της σχεδίασης των ηλεκτρονικών κυκλωμάτων του συστήματος αναφορά σε όλα τα κυκλώματα και σε όλα τα περιφερειακά ολοκληρωμένα που χρησιμοποιήθηκαν. Περιγραφή των εργαλείων που χρησιμοποιήθηκαν και των σχεδιαστικών

παραμέτρων. Αναφορά στο κατασκευαστικό κομμάτι. Κοστολόγηση του συστήματος.

Κεφάλαιο 4 – Υλοποίηση επιπέδου εφαρμογής: Αναφορά στην BACnet βιβλιοθήκη που χρησιμοποιήθηκε για την τελική ανάπτυξη της εφαρμογής αναλυτική περιγραφή της αρχικής λειτουργικότητας αλλά και των αλλαγών που απαιτήθηκαν στο κώδικα για να λειτουργήσει σωστά με βάση τις απαιτήσεις που προδιαγράφηκαν για το σύστημα. Περιγραφή των αισθητήρων που χρησιμοποιήθηκαν σαν είσοδοι στο σύστημα και του ψηφιακού σειριακού πρωτοκόλλου I2C που χρησιμοποιούσαν κάποιοι από τους αισθητήρες και η υλοποίηση του σε επίπεδο εφαρμογής.

Κεφάλαιο 5- Πειραματικά αποτελέσματα: Αποτελέσματα της επικοινωνίας και της χρήσης του ενσωματωμένου σε μια ολοκληρωμένη επικοινωνία σε τοπικό δίκτυο με δεύτερο κόμβο έναν Ηλεκτρονικό Υπολογιστή που τρέχει έναν client για BACnet MS/TP. Περιγραφή των δυσκολιών που αντιμετωπίστηκαν και περιγραφή των αποτελεσμάτων στα λειτουργικά που χρησιμοποιήθηκαν (Windows, Linux). Αναφορά στις μετρήσεις των αισθητήρων που καταγράφηκαν για δεκαπέντε συναπτές μέρες.

Κεφάλαιο 6-Συμπεράσματα Προτάσεις για μελλοντική έρευνα: Αναφορά στα προβλήματα που αντιμετωπίστηκαν και λύθηκαν κατά την διάρκεια της εργασίας. Προτάσεις για συνέχιση της εργασίας σε άλλες κατευθύνσεις.

1.2 Προηγούμενες Εργασίες πάνω στο αντικείμενο

Η περιοχή των αυτοματισμών κτιρίων είναι μια πολύ ενδιαφέρουσα και ζωντανή περιοχή ακαδημαϊκών μελετών είτε πρόκειται για άρθρα σε επιστημονικά περιοδικά είτε για paper καθώς παρουσιάζει πρακτικές εφαρμογές που μπορούν και βρίσκουν άμεσα έξοδο στην αγορά. Η ανάγκη για εξοικονόμηση ενέργειας μέσω συσκευών αυτοματισμού βρίσκει πλέον εφαρμογές όχι μόνο σε μεγάλα εταιρικά κτίρια αλλά και στον οικιακό τομέα (smart houses) όπου σταδιακά υπάρχουν ολοένα και περισσότερες προτάσεις για αυτοματισμούς που έχουν θέση στην οικιακή καθημερινότητα.

Το BACnet μπορεί να χρησιμοποιηθεί με μεγάλη ευελιξία στον τομέα αυτό. Για παράδειγμα στο *Experimental Case Study of a BACnet-Based Lighting Control System T.J. Park, Seung* [1.1] παρουσιάζεται χρήση του πρωτοκόλλου για έλεγχο συστήματος φωτισμού και κεντρικής διαχείρισης του όπου και επιτυγχάνεται μείωση του κατανάλωσης ενέργειας για το σύστημα φωτισμού έως και 40%.

Σε μια άλλη δημοσίευση *BACnet–EnOcean Smart Grid Gateway and its application to demand response in buildings (Chang Lin,Jo Hong)* [1.2] βλέπουμε μια πειραματική εγκατάσταση που χρησιμοποιεί BACnet αλλά και το πρωτόκολλο EnOcean για την δημιουργία ενός Demand Response Δικτύου που αποτελεί μέρος ενός Smart Grid όπως και στο *Experimental evaluation of BZ-GW (BACnet-ZigBee smart grid gateway) for demand response in building -S.H. Hong, S.H Kim, G.M Kim, H.L Kim)* [1.3] που βλέπουμε την πειραματική αξιολόγηση ενός τέτοιου συστήματος.

Στο *A reference model of fire detection and monitoring system using BACnet – Seon Song, Ho Hong* [1.4] βλέπουμε ενός BACnet MSTP/ARCnet συστήματος για τον έλεγχο πυρασφάλειας κτιρίων.

Φυσικά το ακαδημαϊκό ενδιαφέρον δεν εξαντλείται στο πεδίο διαχείρισης συστημάτων αλλά και της έρευνας σε πρωτόκολλα επικοινωνίας και της αξιολόγησης της απόδοσης τους όπως στο *Performance evaluation of BACnet MS/TP protocol using experimental model - Industrial Technology (T.J. Park, Young ,Chan Kwon Seung, Ho Hong)* [1.5] όπου αναλύουν την απόδοση του BACnet MSTP με βάση πειραματικά μοντέλα και αξιολογούν την απόδοση του.

Κεφάλαιο 2

Το πρωτόκολλο BACnet

Σημείωση 1: Στο κεφάλαιο αυτό περιγράφεται την λειτουργία του πρωτοκόλλου BACnet και την λειτουργικότητα του σε κάθε επίπεδο σχεδίασης. Το παρακάτω κείμενο δανείζεται μεγάλα αποσπάσματα κυρίως από τρεις πηγές:

1) BACnet The Global Standard for Building Automation and Control Networks

Michael H. Newman - Momentum Press 2013 [2.1]

2) Building Automation Communication systems with EIBKNX, LON and BACnet

Συλλογικό έργο – Springer 2009 [2.2]

3) ANSI/ASHRAE Standard 135-2012, BACnet-A Data Communication Protocol for Building Automation and Control Networks –ANSI/ASHRAE 2012 [2.3]

Ο γράφων αντί να παραπέμπει σε αυτά τα κείμενα σε κάθε κατάλληλο σημείο θεωρεί καταλληλότερο την αναφορά τους εδώ και παραπομπές βιβλιογραφίας να δίνονται μόνο για πηγές ξεχωριστές από αυτά τα κείμενα

Σημείωση 2: Στο κείμενο χρησιμοποιήθηκε η αγγλική ορολογία όπου θεωρήθηκε αδόκιμη και νεολογισμός η αντίστοιχη ελληνική μετάφραση, καθώς και στα σημεία που η αγγλική ορολογία είναι ευρέως αποδεκτή και επισκιάζει την αντίστοιχη ελληνική.

2.1 Εισαγωγή στο πρωτόκολλο BACnet

Το BACnet είναι ένα πρωτόκολλο επικοινωνίας και αυτοματισμού σε δίκτυα. Το όνομα του είναι ακρωνύμιο για το Building Automation and Control networking protocol. Ένα πρωτόκολλο επικοινωνίας σε δίκτυα είναι ένα σύνολο κανόνων που ρυθμίζουν την ανταλλαγή πληροφοριών μεταξύ υπολογιστών σε ένα δίκτυο. Το BACnet αναπτύχθηκε τον οργανισμό American Society of Heating and Air-Conditioning Engineers (ASHRAE) με την πρώτη έκδοση του πρωτοκόλλου να χρονολογείται το 1987. Το BACnet έχει εγκριθεί από το American National Standards Institute (ANSI) και η τελευταία έκδοση του πρωτοκόλλου είναι επισήμως γνωστή σαν “ANSI/ASHRAE Standard 135-2012, BACnet-A Data Communication Protocol for Building Automation and Control Networks”

Το BACnet είναι ένα ανοιχτό πρωτόκολλο (είναι ελεύθερο για υλοποίηση από τον καθένα χωρίς άδεια από κάποιον οργανισμό) που σχεδιάστηκε για να διευκολύνει την επικοινωνία και των έλεγχου συσκευών που ανήκουν σε

διαφορετικούς κατασκευαστές και συμμετέχουν σε ένα σύστημα αυτοματισμών ενός κτηρίου. Η ανάγκη που καλύπτει είναι η εξασφάλιση της ανεξαρτησίας από έναν και μόνο κατασκευαστή και η εξασφάλιση λειτουργικότητας σε συστήματα που συμμετέχουν συσκευές περισσότερων του ενός κατασκευαστών.

Πριν την εισαγωγή ενός ανοιχτού πρωτοκόλλου σαν το BACnet οι αυτοματισμοί κτηρίων κυριαρχούνταν από ιδιωτικές λύσεις που είχε ο κάθε κατασκευαστής. Το αποτέλεσμα ήταν ότι για παράδειγμα ο ελεγκτής θερμοκρασίας του ενός κατασκευαστή δεν μπορούσε να επικοινωνήσει με ένα κέντρο ελέγχου (control center) από έναν διαφορετικό κατασκευαστή. Συνεπώς σε μια εγκατάσταση αυτοματισμών τα συστήματα εξαερισμού, φωτισμού η συναγερμού έπρεπε να σχεδιαστούν ξεχωριστά και ανεξάρτητα μεταξύ τους χωρίς την δυνατότητα καθολικής επίβλεψης τους από ένα κοινό interface.

Προφανώς οι σχεδιαστές συστημάτων είχαν ένα σημαντικό μειονέκτημα είτε θα έπρεπε να εγκαταλείψουν την προσπάθεια κεντρικού συντονισμού των υποσυστημάτων ή ήταν υποχρεωμένοι να βασίσουν όλη την σχεδίαση σε έναν μόνο κατασκευαστή, ενώ όταν θα ήθελαν να επεκτείνουν μια εγκατάσταση θα ήταν δεσμευμένοι να αγοράσουν εξοπλισμό μόνο από τον αρχικό κατασκευαστή.

Σε αυτά τα προβλήματα δίνει λύση το BACnet το οποίο είναι ένα ανοιχτό πρωτόκολλο που υποστηρίζεται από πολλούς κατασκευαστές και εξασφαλίζει την συνεργασία συσκευών από διαφορετικούς πωλητές.

Ακριβώς η ίδια λογική έχει επικρατήσει και στα δίκτυα υπολογιστών με την διάδοση ανοιχτών πρωτοκόλλων όπως το TCP/IP το οποίο υποστηρίζουν το σύνολο του hardware και του software που χρησιμοποιούνται στα δίκτυα από όλους τους κατασκευαστές.

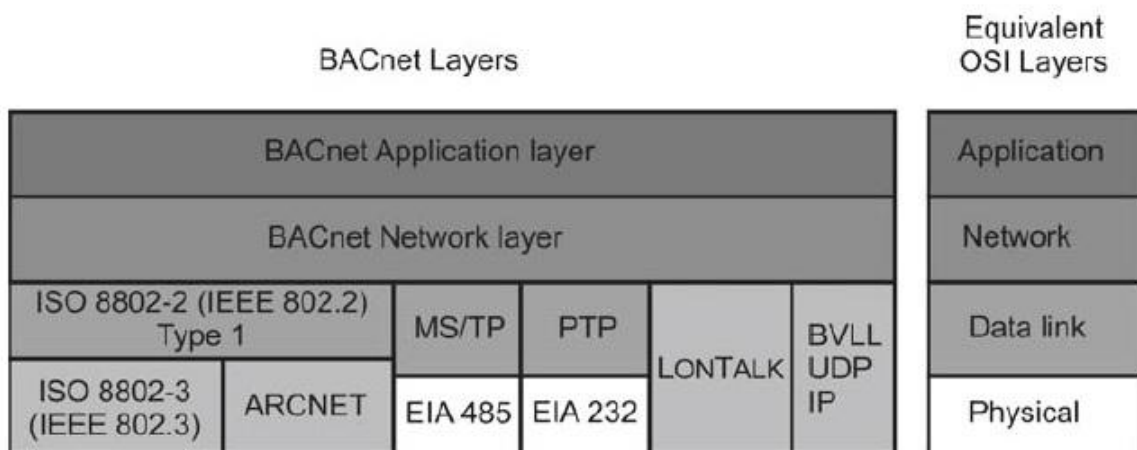
2.2 Αρχιτεκτονική του πρωτοκόλλου

Η αρχιτεκτονική των επιπέδων στο πρωτόκολλο BACnet είναι επηρεασμένη από το ISO Open Systems Interconnection Basic Reference Model (OSI/BRM) ([1.4] για το OSI/BRM). Αυτό που κάνει το BRM είναι να διαιρεί τις διάφορες ανάγκες που υπάρχουν στην ανταλλαγή δεδομένων μεταξύ οντοτήτων δικτύου σε 7 επίπεδα.

Τα επίπεδα αυτά είναι τα εξής: Physical (Αφορά την μετάδοση και την λήψη των bitπάνω στο μέσο επικοινωνίας), Data Link (ελέγχει την σειρά πρόσβασης στο μέσο), Network (Εγκαθιστά λογικά κυκλώματα και την δρομολόγηση μεταξύ των οντοτήτων), Transport (Ελέγχει για λάθη και τμηματοποιεί τα δεδομένα), Session (Συγχρονίζει τα δεδομένα με σημεία ελέγχου), Presentation (αποκρυπτογραφεί /κρυπτογραφεί τα δεδομένα και τα αναδιοργανώνει) και τέλος Application (που είναι και η πραγματική διεπαφή με το πρόγραμμα του χρήστη). Το κάθε επίπεδο έχει την

δική του επικεφαλίδα και προσθέτει στο συνολικό μήκος του κάθε πακέτου. Το BRM είναι μια προσπάθεια να η τυποποίηση των πρωτοκόλλων δικτύου να “σπάσει” σε μια σειρά από μικρότερα προβλήματα που είναι πιο διαχειρίσιμα.

Το BACnet έχει μια αρχιτεκτονική απευθείας εμπνευσμένη από το BRM παραλείποντας τα επίπεδα που είναι περιττά από την φύση του πρωτοκόλλου, για παράδειγμα λόγω του ότι τα μηνύματα που ανταλλάσσονται μεταξύ συσκευών BACnet είναι μικρά και χωράνε σε ένα πακέτο δεδομένων στο φυσικό μέσο τα επίπεδα Session, Transport είναι περιττά. Η ενθυλάκωση των πρωτοκόλλων σε κάθε επίπεδο και γενικά η αρχιτεκτονική του BACnet παρουσιάζεται στην Εικόνα 2.1.



Εικόνα 2.1 Διαστρωμάτωση και ενθυλάκωση πρωτοκόλλων στο BACnet

Όπως είναι φανερό το επίπεδο Application και Network είναι κοινό για όλα τις ζεύξεις δεδομένων (Data link) καθώς και ότι υποστηρίζονται 6 διαφορετικά πρωτόκολλα ζεύξεως δεδομένων.

2.3 BACnet Application Layer

Υπάρχουν δύο κομμάτια στο επίπεδο εφαρμογών του BACnet τα BACnet Objects (αντικείμενα) και τα BACnet Services (υπηρεσίες)

2.3.1 BACnet Object Model

Κεντρικό κομμάτι του πρωτοκόλλου είναι τα αντικείμενα (Objects). Τα αντικείμενα γενικά στον αντικειμενοστραφή προγραμματισμό είναι ένας αφηρημένος τύπος δεδομένων (Abstract Data Type, ADT) όπου η πληροφορία αποθηκεύεται σαν ιδιότητες του αντικειμένου. Σχεδιαστικά τα αντικείμενα λύνουν

το πρόβλημα της αναπαράστασης των λειτουργιών μιας BACnet συσκευής με ένα τυποποιημένο τρόπο ώστε να μπορούν να χρησιμοποιηθούν από άλλες συσκευές στο δίκτυο.

Πιο συγκεκριμένα ένα αντικείμενο στο BACnet είναι μια συλλογή πληροφοριών που σχετίζεται με μια υποστηριζόμενη λειτουργία μιας συσκευής BACnet. Οι ιδιότητες αυτές αποκαλούνται properties. Το πρωτόκολλο ορίζει τύπους αντικειμένων (object types) οι οποίοι όταν υλοποιούνται σε μια πραγματική συσκευή αναφέρονται πλέον σαν objects (αντικείμενα). Η διάκριση αυτή (object type-object) είναι σημαντική καθώς ένα object σε μια πραγματική συσκευή δεν είναι απαραίτητο να έχει όλα τα properties που αντιστοιχούν στο object type αλλά όσα κριθούν απαραίτητα και αναγκαία κατά την υλοποίηση. Το πρωτόκολλο είναι ανοιχτό σε υλοποιήσεις.

Μια συσκευή BACnet (BACnet device) είναι μια συλλογή από Objects που πρέπει να περιέχει υποχρεωτικά ένα Device Object που περιέχει πληροφορίες οι οποίες περιγράφουν την συσκευή αλλά δεν μπορούν να προσκολληθούν σε άλλου είδους property. Τέτοια property περιλαμβάνουν Vendor_Name (όνομα κατασκευαστή) , Firmware_Revision(έκδοση firmware), Location, Description, Protocol_Services_Supported (Υποστηριζόμενες υπηρεσίες) κ.α. Μια πραγματική BACnet συσκευή περιέχει πολλά άλλα object αναλόγως του σκοπού της όπως Analog Input, Binary Input, Schedule, Calendar κ.α. Ακολουθεί στην εικόνα 2.2 μια αναπαράσταση ενός BACnet Object με την μορφή πίνακα.

Object name	Room temperature
Object type	Analog input
Current value	25.3
Status	Normal
Upper limit	35.0
Lower limit	0.0

Εικόνα 2.2 Αναπαράσταση BACnet Object με την μορφή πίνακα

Κάθε αντικείμενο έχει ένα object identifier (μοναδικό αναγνωριστικό) το οποίο το χαρακτηρίζει μοναδικά μέσα στην συσκευή. Ο συνδυασμός μοναδικού σε όλο το δίκτυο αναγνωριστικού για τα device και μοναδικού αναγνωριστικού για κάθε διαφορετικό αντικείμενο εντός της ίδιας συσκευής είναι που επιτρέπει την αδιαμφισβήτητη προσπέλαση κάθε αντικειμένου μέσα στο δίκτυο.

2.3.2 BACnet Object Properties

Κάθε property χαρακτηρίζεται από τρία attribute: Property Identifier (αναγνωριστικό αντικειμένου), Property Data type (Τύπο δεδομένων) καθώς και ένα Conformance Code.

Το Property Identifier είναι ένα string (αλφαριθμητική πληροφορία) που αποτελείται από μια ή περισσότερες λέξεις που ξεκινάνε με κεφαλαίο χαρακτήρα και διαχωρίζονται με κάτω underscore (κάτω παύλα π.χ. «Present_Value» κ.ο.κ.). Κάθε τέτοιο string αντιστοιχεί σε μια τιμή στην απαρίθμηση (enumeration) που καθορίζεται από την επίσημη περιγραφή του πρωτοκόλλου (Clause 21, Formal Description of Application Data Units [2.1]) για παράδειγμα Property Identifier, Present Value αντιστοιχεί ο κωδικός 85. Οι κωδικοί 1-511 είναι δεσμευμένοι από τους κατασκευαστές του πρωτοκόλλου ενώ οι υπόλοιποι μέχρι τον 4194303 είναι διαθέσιμοι για κάθε υλοποίηση.

Το Property Data Type μπορεί να είναι κάθε απλός τύπος δεδομένων όπως NULL, Unsigned, REAL κ.τ.λ. είτε κάποιος σύνθετος τύπος που ορίζεται στο πρωτόκολλο όπως BACnetAddress που αποτελείται από δύο αντικείμενα που αντιστοιχούν σε απλούς τύπους δεδομένων μια διεύθυνση δικτύου (Unsigned16) και μια MAC-address (Medium Access Address) τύπου OCTET STRING

Το τρίτο attribute είναι το Conformance Code (CC) που παίρνει της εξής τιμές: R, W ή O. Το Conformance Code δηλώνει την προσβασιμότητα ενός property από τα services.

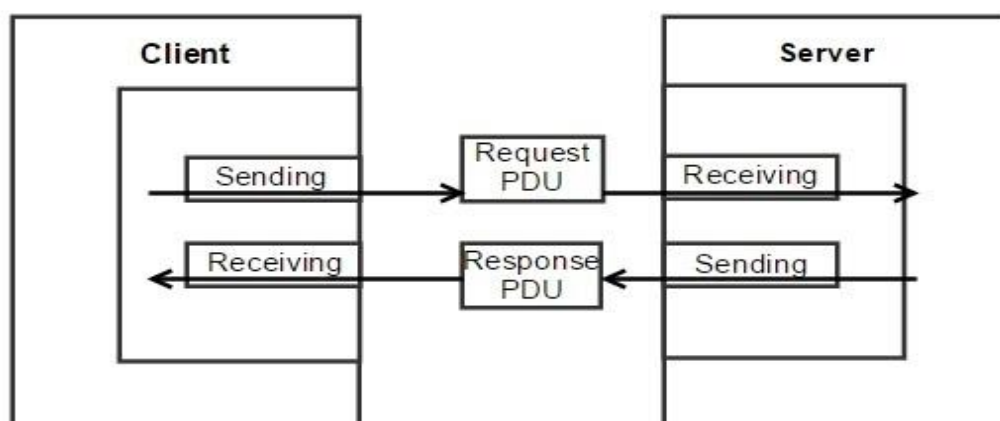
- Το R δηλώνει ότι ένα property απαιτείται να υπάρχει σε μια υλοποίηση και μπορεί να είναι προσβάσιμο για ανάγνωση.
- Το W σημαίνει ότι το property πρέπει να υπάρχει και να είναι προσβάσιμο για ανάγνωση και για εγγραφή.
- Το O δηλώνει ότι το property είναι προαιρετικό.

2.3.3 BACnet Object Types

Στο BACnet 2012 ορίζονται 54 object types ενώ υπάρχει κατεύθυνση για την αύξηση του αριθμού των αντικειμένων. Τα αντικείμενα αυτά καλύπτουν τις πιθανές ανάγκες από την χρήση του πρωτοκόλλου ενώ για κάθε αντικείμενο είναι υποχρεωτικό να υπάρχει τουλάχιστον ένα Device Object που έχει πληροφορίες για την ίδια την συσκευή. Άλλα Object Types που είναι συνηθισμένα και υπάρχουν συνήθως είναι Analog Input, Analog Output, Analog Value, File, Averaging, Accumulator, Calendar, Schedule κ.α.

2.3.4 BACnet Services

Τα services είναι ο πυρήνας του πρωτοκόλλου. Έχουν σχεδιαστεί με βάση το μοντέλο client-server όπου μια συσκευή client(πελάτης) στέλνει ένα service request σε μια άλλη συσκευή που δρα σαν server και απαντά με ένα service response όπως στην εικόνα 2.3. Οι ανταλλαγές πληροφορίας που ακολουθούν το μοντέλο αυτό αποκαλούνται confirmed services(επιβεβαιωμένες υπηρεσίες) επειδή απαιτείται κάποιου είδους acknowledgement(ή ACK, πακέτο πληροφορίας που επιβεβαιώνει ότι μια συσκευή έλαβε το πακέτο που της στάλθηκε).

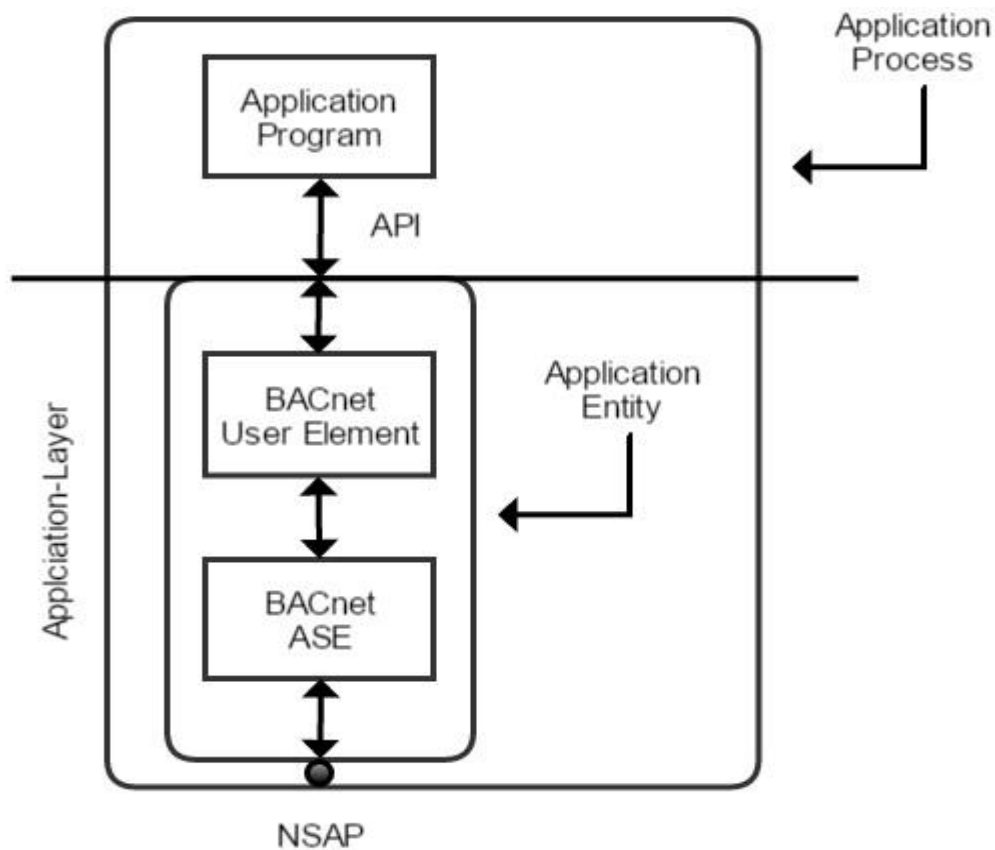


Εικόνα 2.3 Το μοντέλο Client-Server

Σε πιο αφαιρετικό επίπεδο το επίπεδο εφαρμογής στο BACnet δομείται στην εξής λογική: Το Application Process (μια εκτέλεση μια εφαρμογής BACnet) κάνει την επεξεργασία που χρειάζεται η εφαρμογή για να χρησιμοποιεί το BACnet για την επικοινωνία. Το μέρος του Application Process που βρίσκεται μέσα στο Application Layer του BACnet και το υλοποιεί άμεσα καλείται Application Entity. Η τελική εφαρμογή επικοινωνεί με την Application Entity μέσα από το Application Program Interface (API). Η διεπαφή αυτή δεν καθορίζεται από το πρωτόκολλο αλλά εξαρτάται από την υλοποίηση. Για παράδειγμα σε μια πραγματική υλοποίηση του BACnet θα ήταν πιθανότατα μια συνάρτηση, μια διαδικασία ή μια κλήση υπορουτίνας. Ακολουθεί σχηματική αναπαράσταση των παραπάνω στην εικόνα 2.4.

Όπως φαίνεται στην πιο κάτω εικόνα το Application Entity αποτελείται από δυο κομμάτια: το BACnet UserElement (UE) και τα BACnet Application Service Element (ASE). Το UE περιέχει την λογική για να υποστηριχθεί το API καθώς υλοποιεί τα service procedures για κάθε υποστηριζόμενη BACnet service ενώ πρέπει να κρατά και πληροφορία ID's σχετικά με τα process που ξεκινούν ένα service αλλά και να

χειρίζεται counters που χρειάζονται σε περιπτώσεις retransmission (επαναποστολής πακέτου). Το ASE αντιπροσωπεύει το σύνολο των services που είναι υποστηρίζονται στην συγκεκριμένη υλοποίηση . Στην εικόνα επίσης φαίνεται το NSAP (Network Service Access Point) που είναι στην ουσία το interface του Application Layer με το Network Layer.



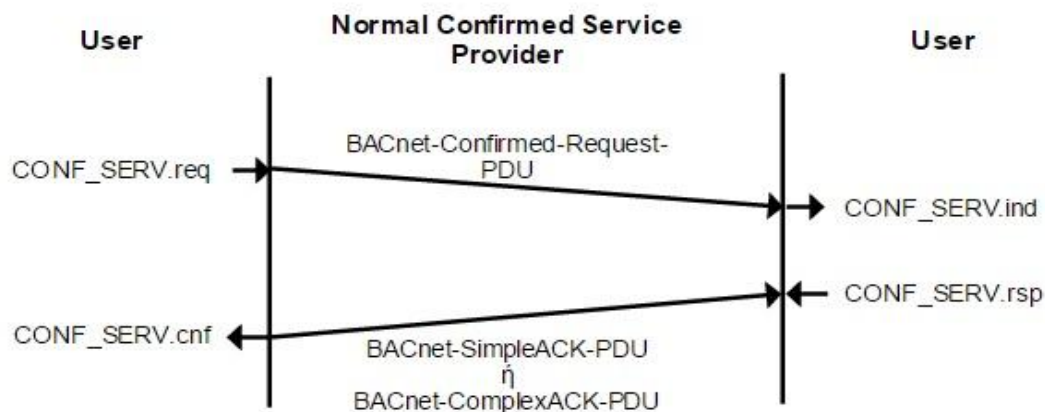
Εικόνα 2.4 Εννοιολογικό μοντέλο μιας διαδικασίας BACnet application

Η κεντρική ιδέα είναι τα Application Processes από διαφορετικά BACnet devices απευθείας ως peers (ισότιμες συσκευές). Η πληροφορία που ανταλλάσσεται μεταξύ δυο peer Application Processes αναπαρίσταται στο BACnet σαν ανταλλαγή αφηρημένων service primitives (απλού τύπου πληροφορία). Αυτά τα primitives χρησιμοποιούνται για να μεταφέρουν παραμέτρους όπως αυτές ορίζονται στην περιγραφή κάθε ξεχωριστού service. Υπάρχουν τεσσάρων ειδών service primitive: request (req) , indication(ind), response (rsp) και confirm (cnf). Η πληροφορία που περιέχουν τα primitives μεταφέρετε μέσω μιας πληθώρας από διαφορετικά Protocol Data Unit (PDU, πακέτα δεδομένων) που περιγράφονται από το πρωτόκολλο. Αυτά είναι τα εξής που φαίνονται στον παρακάτω πίνακα.

Τύπος PDU	Συντομογραφία
BACnet-Confirmed-Request -PDU	CONF_SERV
BACnet-Unconfirmed-Request-PDU	UNCONF_SERV
BACnet-Simple-ACK-PDU	SIMPLE_ACK
BACnet-ComplexACK-PDU	COMPLEX_ACK
BACnet-SegmentACK-PDU	SEGMENT_ACK
BACnet-Error-PDU	ERROR
BACnet-Reject-PDU	REJECT
BACnet-Abort-PDU	ABORT

Πίνακας 2.1 Τα PDU (Protocol Data Unit) που χρησιμοποιεί το BACnet

Στην εικόνα 2.5 φαίνεται μια επιτυχημένη συναλλαγή μεταξύ δύο συσκευών BACnet. Οι παράμετροι του CONF_SERV.ind στέλνονται μέσω του αντίστοιχου πακέτου BACnet-Confirmed-Request –PDU σε μια άλλη BACnet συσκευή. Αυτή λαμβάνει το πακέτο και αφού το αποκωδικοποιήσει και παράγει ένα ACK το οποίο ανάλογα με την φύση της υπηρεσίας μπορεί να είναι είτε Simple είτε Complex οπότε και περιέχει δεδομένα. Συνεπώς κωδικοποιεί τα δεδομένα και τα στέλνει μέσω ενός BACnet-Simple-ACK-PDU είτε ένα BACnet-Complex-ACK-PDU που παραδίνεται στον αρχικό αποστολέα όπου και παίρνει τα CONF_SERV.cnf primitive και καταλαβαίνει ότι έγιναν όλα σωστά.



Εικόνα 2.5 Μία επιτυχημένη Confirmed Service συναλλαγή

2.3.5 BACnet Service descriptions

Τα service primitives που αναφέρθηκαν δεν είναι υλοποιημένα στην πραγματικότητα. Η γνώση όμως της ορολογίας του πρωτοκόλλου και το πώς λειτουργεί σε αφαιρετικό επίπεδο είναι χρήσιμη καθώς τα services descriptions, που

συνοδεύουν τα services και που περιγράφουν μια υπηρεσία, αναφέρονται στην ορολογία αυτή.

Κάθε service description έχει 4 μέρη:

- 1)Μια εξήγηση για τον σκοπό της υπηρεσίας
- 2)Έναν πίνακα που δείχνει την δομή των service primitives
- 3)Μια γραπτή περιγραφή κάθε παραμέτρου
- 4)Την περιγραφή της συνάρτησης που αντιστοιχεί στην υπηρεσία

Παραδείγματος χάριν τα service description που ανεφέρονται από πάνω για το read property service είναι τα εξής:

- 1)Στην παράγραφο 15 του πρωτοκόλλου (βλέπε [2.1] παράγραφος 15) αναφέρεται στο πρωτότυπο για το ReadProperty Service:

The ReadProperty service is used by a client BACnet-user to request the value of one property of one BACnet Object. This service allows read access to any property of any object, whether a BACnet-defined object or not

(Απόδοση: «Το ReadProperty service χρησιμοποιείται από ένα πελάτη, χρήστη BACnet για να ζητήσει την τιμή μιας ιδιότητας ενός BACnet αντικειμένου. Το service επιτρέπει την πρόσβαση ανάγνωσης για μία ιδιότητα ενός αντικειμένου είτε είναι ορισμένο από το BACnet είτε όχι»)

2)Στην εικόνα 2.6 φαίνονται τα service primitives Read Property. Στην πρώτη στήλη φαίνονται οι παράμετροι για το request και για τις πιθανές απαντήσεις. Για τις επιβεβαιωμένες απαντήσεις (confirmed responses) αντιστοιχούν οι παράμετροι στο Result(+) εάν έχουμε επιτυχή επικοινωνία διαφορετικά έχουμε τις παραμέτρους στο Result(-) εάν έχουμε κάποιου είδους λάθος στην επικοινωνία. Για ανεπιβεβαίωτες υπηρεσίες (unconfirmed services) δεν απαιτούνται καθόλου οι παράμετροι Result(+)/Result(-) καθώς δεν έχουμε κάποιου είδους απάντηση. Στις υπόλοιπες τέσσερις στήλες φαίνονται σε ποια κατηγορία service primitive από αυτές που προαναφέρθηκαν αλλά και το είδος της παραμέτρου σε σχέση με την σημασία της. Το M δηλώνει υποχρεωτική υλοποίηση (Mandatory), το S επιλεκτική (Selective), το U επιλογή του χρήστη (User option)

Parameter-Name	Request	Indication	Response	Confirmation
Argument	M	M (=)		
Subscriber Process Identifier	M	M (=)		
Monitored Object	M	M (=)		
Issue ConfirmedNotifications	U	U (=)		
Lifetime	U	U (=)		
Result (+)			S	S (=)
Result (-)			S	S (=)
Error Type			M	M (=)

Εικόνα 2.6 Service primitive του Read Property service

3)Κάθε παράμετρος περιγράφεται και σε πληροφορία κειμένου. Για παράδειγμα για την παράμετρο Object Identifier αναφέρεται (2.1] παράγραφος 15 σε απόδοση): «Η παράμετρος αυτή, του τύπου BACnetIdentifier, θα παρέχει τα μέσα αναγνώρισης του αντικειμένου η οποία ιδιότητα είναι αναγνώσιμη και επιστρέφεται σε έναν χρήστη BACnet».

4)Τέλος η παράγραφος που περιγράφει την διαδικασία για κάθε service περιγράφει την θα πρέπει να κάνει ο πάροχος του service όταν έρθει service request και πρέπει να το εξυπηρετήσει. Για το ReadProperty αναφέρεται ([1.1] παράγραφος 15 σε απόδοση): «Αφού επιβεβαιωθεί την εγκυρότητα του request ο BACnet-user που πρέπει να απαντήσει θα προσπαθήσει να προσπελάσει την συγκεκριμένη ιδιότητα για το συγκεκριμένο αντικείμενο. Εάν έχουμε επιτυχία ένα Result(+) primitive θα δημιουργηθεί και θα επιστραφεί μαζί με την τιμή που ανακτήθηκε. Εάν η πρόσβαση αποτύχει ένα Result(-) primitive θα δημιουργηθεί, που δείχνει το λόγο της επιτυχίας».

2.3.6 BACnet Service Types

Τα service ταξινομούνται στις εξής κλάσεις αναλόγως με την χρήση τους
Alarm and Event: Services που χρησιμοποιούνται για την διαχείριση της επικοινωνίας που σχετίζεται με Alarm (συναγερμούς) και Event(γεγονότα)

File Access: Services που χρησιμοποιούνται για την πρόσβαση και την διαχείριση αρχείων

Object Access: Services που χρησιμοποιούνται για την πρόσβαση και διαχείριση των ιδιοτήτων των αντικειμένων BACnet

Remote Device Management: Services που χρησιμοποιούνται για μια πληθώρα από ετερόκλητες, αλλά και σημαντικές, διαδικασίες όπως ο συγχρονισμός, έναρξης και

παύσης της επικοινωνίας, επαναρχικοποίησης των διαδικασιών, μεταφοράς μηνυμάτων προτεραιότητας και δυναμικής σύνδεσης (dynamic binding).

Virtual Terminal: Services που χρησιμοποιούνται πλέον σπάνια (αλλά περιλαμβάνονται στο πρωτόκολλο ακόμα) και αποφεύγονται στις υλοποιήσεις πλέον (καθώς υπερκαλύπτονται από την χρήση του Internet BACnet/IP) προσανατολισμένες στην δημιουργία Virtual Terminal, εικονικών τερματικών δηλαδή για την μεταφορά δεδομένων χαρακτήρων.

2.4 BACnet Network Layer

Σύμφωνα με το Open Systems Interconnection Basic Reference Model (BRM βλέπε την παράγραφο 2.2 καθώς και το [2.1]) ανάμεσα στο επίπεδο Εφαρμογής (Application) και Δικτύου (Network Layer ή NL) τα εξής: Παρουσίασης (Presentation), Συνόδου (Session) και Μεταφοράς (Transport). Στο BACnet παραλείπονται τα ενδιάμεσα αυτά επίπεδα παραλείπονται και οι απαραίτητες λειτουργίες ενσωματώνονται στο επίπεδο Δικτύου.

Το επίπεδο Δικτύου (NL) έχει τις εξής αρμοδιότητες:

1. Διασύνδεση των δικτύων BACnet ανεξαρτήτως του δικτύου Ζεύξης
2. Παρέχει την κατανομή των μηνυμάτων ανάμεσα σε πολλαπλούς χρήστες
3. Διαχειρίζεται την χρήση και την ρύθμιση των router του δικτύου
4. Παρέχει το μηχανισμό μεταφοράς αλλά και ελέγχου για την αρχιτεκτονική ασφάλειας του BACnet

2.4.1 NL Protocol Data Unit

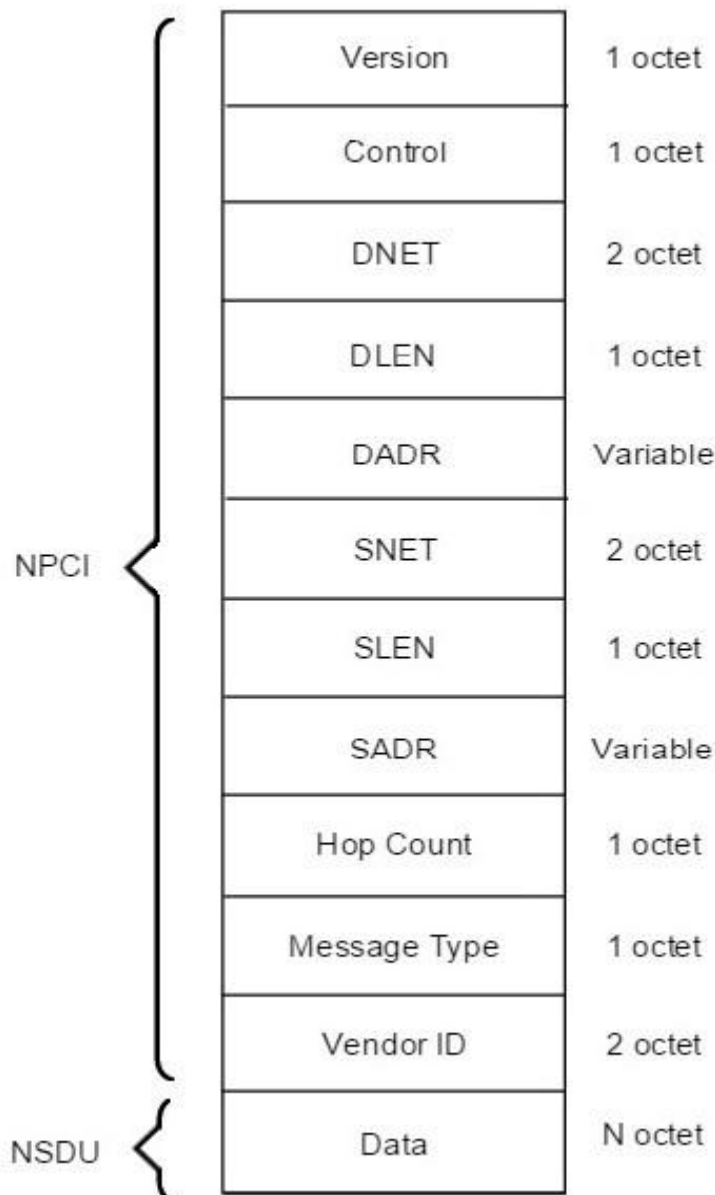
Τα πακέτα πληροφορίας στο επίπεδο αυτό καλούνται NL Protocol Data Unit ή NPDU και στην παράγραφο αυτήν θα εξεταστεί η δομή τους. Τα NPDU αποτελείται από δυο μέρη: Το NL Protocol Control Information (NPCI) που περιέχει δεδομένα επικεφαλίδας του πακέτου και το NL Service Data Unit (NLSDU) που περιέχει τα χρήσιμα δεδομένα της υπηρεσίας (βλέπε εικόνα 2.7).

Όπως φαίνεται στην πιο πάνω εικόνα το πακέτο αποτελείται από ομάδες 8-bit πληροφορίας που ονομάζονται octets. Η σημασία των octets είναι:

Version: Η έκδοση του πρωτοκόλλου ακέραιος μέχρι στιγμής είναι 1

Control: Πληροφορίες ελέγχου π.χ. το bit7 υποδηλώνει εάν έχουμε μήνυμα για το επίπεδο αυτό (NL) είτε είναι μήνυμα που πρέπει να προωθηθεί στο επίπεδο

εφαρμογής APDU. Ομοίως για τα bit 1,0 μας δίνουν την προτεραιότητα του μηνύματος (00 η μικρότερη, 11 η μεγαλύτερη)



Εικόνα 2.7 Η δομή ενός BACnet πακέτου δικτύου (NPDU)

DNET: Πληροφορία μήκους 2-octet που δείχνει τον τελικό δίκτυο προορισμού του πακέτου. Για Global Broadcast (μετάδοση σε όλα τα δίκτυα) το DNET έχει την τιμή Χ'FFFF' (σε δεκαεξαδικό)

DLEN: Πληροφορία μήκους 1-octet που δείχνει το μέγεθος του MAC address. (Το 0 μηδέν υποδηλώνει broadcast σε όλους τους κόμβους του δικτύου.

DADR: Πληροφορία μεταβλητού μήκους σε octet που αφορά την MAC-address προορισμού. Το μήκος σε octet καθορίζεται από το DLEN.

SNET: Ομοίως με τον DNET αλλά αφορά το δίκτυο πηγής του μηνύματος. Μήκους 2-octet.

SLEN: Μήκος του αρχικού MAC-address της πηγής

SADR: MAC-address της αρχικής πηγής. Το μήκος σε octet καθορίζεται από το SLEN.

Hop Count: Μετρητής μήκους 1-octet (δηλαδή 256 στοιχείων από 250 μέχρι 0) για την αποφυγή κυκλικών διαδρομών πακέτων δηλαδή που περιφέρονται στο δίκτυο χωρίς να μπορούν να φτάσουν στο τελικό προορισμό απασχολώντας πόρους άσκοπα. Αρχικοποιείται στο 255 και μειώνεται με κάθε "άλμα" στον επόμενο router. Εάν φτάσει στο 0 το πακέτο γίνεται discard απορρίπτεται δηλαδή.

Message Type: Εάν έχουμε NL message (αυτό καθορίζονται από το bit 7 του control όπως αναφέρθηκε πριν) τότε η πληροφορία εδώ, δείχνει τον τύπο του μηνύματος (περισσότερα για τους τύπους μηνυμάτων στην επόμενη παράγραφο). Παραδείγματος χάριν τιμή X'00' αντιπροσωπεύει το μήνυμα Who-is-Router-To-Router.

Vendor ID: 2-octet αναγνωριστικό του ενός εγνωσμένου BACnet πωλητή που δεσμεύει τον κωδικό αυτό με παραχώρηση από την ASHRAE. Το πεδίο αυτό χρησιμοποιείται μόνο όταν έχουμε το bit 7 του Control 1 που υποδηλώνει NSDU και Message Type ανάμεσα X'80' και X'FF' που υποδηλώνει ιδιόκτητα μηνύματα.

Data: Εάν στο Control πεδίο έχουμε το bit 7 ίσο με 0 τότε δεν έχουμε NSDU και έχουμε πακέτο δεδομένων APDU. Η πληροφορία για το application level ακολουθεί σε αυτό το πεδίο που έχει μεταβλητό μέγεθος ανάλογα με τα δεδομένα.

2.4.2 Περιγραφή των μηνυμάτων του Network Layer

Ακολουθεί μια σύντομη περιγραφή των μηνυμάτων επιπέδου δικτύου

1) Who-Is-Router-To-Network (WIRTN): Το WIRTN χρησιμοποιείται από όλες τις BACnet συσκευές σε ένα δίκτυο με σκοπό να βεβαιώσουν την διεύθυνση τον επόμενο router σε δίκτυο προορισμού αλλά και από τους router για να ενημερώνουν τους πίνακες δρομολόγησης τους (routing table).

2) I-am-Router-To-Network (IARTN): Το IARTN χρησιμοποιείται για να υποδηλώσει τον αριθμό και την διεύθυνση των δικτύων (μέσω μια λίστας με τα DNET) που είναι προσβάσιμα μέσω του router που γεννάει το μήνυμα αυτό.

3)I-Could-Be-Router-To-Network (ICBRTN): Το ICBRTN χρησιμοποιείται σαν απάντηση στο WIRTN που αναφέρθηκε πριν δηλώνοντας ότι ο κόμβος που γεννάει την απάντηση αυτή ότι είναι σημείο πρόσβαση για ένα συγκεκριμένο πρόσβασης.

4)Reject-Message-To-Network (RMTN): Το RMTN κατευθύνεται από έναν router στον κόμβο που του έστειλε ένα πακέτο το οποίο απορρίφθηκε για διάφορους λόγους. Μαζί επιστρέφεται και ένα πεδίο που υποδηλώνει τον λόγο απόρριψης.

5)Router-Busy-To-Network (RBTN): Το RBTN δηλώνει ότι ο router που το γεννά δεν μπορεί να προωθήσει την κίνηση που προωθούσε κανονικά. Το RBTN περιέχει μια λίστα από τα DNET των δικτύων που δεν μπορεί να εξυπηρετήσει πλέον. Εάν δεν υπάρχει λίστα DNET τότε σημαίνει ότι έχουμε flow control (έλεγχου του ρυθμό κίνησης) σε όλα τα δίκτυα που εξυπηρετούσε κανονικά ο router.

6)Router-Available-To-Network (RATN): Το RATN μεταδίδεται τοπικά (local broadcast) όπως και το RBTN και υποδηλώνει το τέλος του flow control που είχε επιβληθεί πριν για ένα συγκεκριμένο router.

7)Initialize-Routing-Table (IRT): Χρησιμοποιείται για δυο σκοπούς την αρχικοποίηση του πίνακα δρομολόγησης αλλά και για την ερώτηση των περιεχομένων του.

8)Initialiaze-Routing-Table-Ack (IRTA): Χρησιμοποιείται για την επιβεβαίωση μιας αλλαγής στο πίνακα δρομολόγησης.

9)Establish-Connection-To-Network (ECTN): Το ECTN χρησιμοποιείται για να φτιάξει μια PTP σύνδεση με έναν half-router που πλέον μπορεί να χρησιμοποιηθεί σαν router για ένα συγκεκριμένο δίκτυο (Half router είναι μια συσκευή που μπορεί να χρησιμοποιηθεί σαν το ένα μέρος σε μια Point-to-Point σύνδεση. Δύο half routers σε PTP σύνδεση αποτελούν έναν router).

10)Disconnect-Connection-To-Network (DCNT): Χρησιμοποιείται για τον τερματισμό της σύνδεσης PTP με τον half router που δημιουργήθηκε με τον ECTN

11)Challenge-Request (CR): Το CR είναι μήνυμα που χρησιμοποιείται από τον προορισμό ενός μηνύματος για λόγους ασφαλείας ώστε να επιβεβαιώσει ότι μια συγκεκριμένη πηγή έστειλε όντως το μήνυμα που έλαβε. Το μήνυμα αυτό μπορεί να είναι ακόμα και κρυπτογραφημένο.

12)Security Payload(SP): Το SP περιέχει ένα κρυπτογραφημένο μήνυμα δεδομένων το οποίο μπορεί να είναι είτε NPDU είτε APDU.

13)Security Response(SR): Το SR στέλνεται σαν απάντηση τύπου επιβεβαίωσης (ACK) σε μήνυμα ασφαλείας το οποίο στάλθηκε με την SP

14)Request-Key-Update (RKU), Update-Key-Set (UKS), Update-Distribution-Key (UDK), Request-Master-Key (RMK) Set-Master-Key (SMK): Όλα αυτά του τύπου

τα μηνύματα χρησιμοποιούνται για την δημιουργία και την διαχείριση των κλειδιών της κρυπτογράφησης. Το BACnet χρησιμοποιεί το Advanced Encryption Standard (AES) το οποίο είναι μια τεχνική κρυπτογραφίας συμμετρικού κλειδιού, γεγονός που σημαίνει ότι το ίδιο κλειδί που χρησιμοποιείται για την κρυπτογράφηση του μηνύματος χρησιμοποιείται και για την αποκρυπτογράφηση του. (Περισσότερα για το AES [2.6])

15)What-Is-Network-Number (WINN): Αρκετές φορές εγκαταστάσεις με συσκευές BACnet λειτουργούν σε ένα δίκτυο και δεν υπάρχει η ανάγκη για router. Σε αυτές τις περιπτώσεις η αρχιτεκτονική των μηνυμάτων δικτύου απλοποιείται, καθώς τα DNET,DLEN,DADR και τα SNET,SLEN,SADR παραλείπονται. Εφόσον δεν υπάρχει ανάγκη για επικοινωνία με άλλο δίκτυο μια BACnet συσκευή δεν χρειάζεται να ξέρει τον αριθμό του δικτύου της. Για την υποστήριξη της ασφαλούς επικοινωνίας μεταξύ συσκευών όμως είναι απαραίτητη η γνώση του αριθμού του δικτύου οπότε μια συσκευή που δεν γνωρίζει το αριθμό δικτύου της χρησιμοποιεί το WINN για να τον μάθει. Μια συσκευή που δεν χρησιμοποιεί ασφαλή επικοινωνία και δεν συμμετέχει σε επικοινωνία με άλλα δίκτυα δεν χρειάζεται να ξέρει τον αριθμό δικτύου. Το WINN μεταδίδεται πάντα τοπικά (local broadcast) και αγνοείται από router.

16)Network-Number-Is (NNI): Μια συσκευή που δέχεται ένα WINN μήνυμα και γνωρίζει το αριθμό του δικτύου απαντάει με NNI σε local broadcast. Ομοίως το μήνυμα αυτό αγνοείται από τους router.

17)Υπόλοιπα Message Type: Οι κωδικοί του από X''00'' μέχρι X''13'' χρησιμοποιούνται για τα Message Type που προαναφέρθηκαν. Οι κωδικοί από το X''14'' μέχρι X''7F'' είναι δεσμευμένοι από την ASHRAE ενώ οι υπόλοιποι κωδικοί από X''80'' μέχρι X''FF'' είναι διαθέσιμοι για ιδιωτικά μηνύματα του εκάστοτε δημιουργού μιας BACnet συσκευής.

2.4.3 Διανομή μηνυμάτων σε πολλαπλούς αποδέκτες

Όπως έχει αναφερθεί ήδη υπάρχουν αρκετά μηνύματα που ενδέχεται να έχουν πάνω από ένα παραλήπτη και χρειάζονται πολλαπλή διανομή. Τέτοια μηνύματα είναι τα unconfirmed Application Layer messages (όπως το Who-is κ.α.) και κάποια NL messages (όπως τα WIRTN,IARTN και το RBTN). Το BACnet υποστηρίζει και Multicast αλλά και Broadcast. (Το multicast αφορά μηνύματα που παραδίδονται σε μια συγκεκριμένη ομάδα παραληπτών που αποτελούν την multicast ομάδα, ενώ το broadcast αφορά μηνύματα που στέλνονται σε όλες τις συσκευές που ανήκουν στο ίδιο δίκτυο είτε πρόκειται για το τοπικό δίκτυο (local

broadcast) είτε σε άλλο δίκτυο (remote broadcast) (για broadcast/multicast[1.6]). Ακολουθεί μια σύντομη περιγραφή κάθε τύπου πολλαπλής επικοινωνία (one-to-many) που υποστηρίζεται στο BACnet.

2.4.3.1 BACnet Multicast

Από τα data-link type που υποστηρίζονται στο BACnet multicast υποστηρίζουν τα εξής: Ethernet, LonTalk, Zigbee και το BACnet/IP. Παρόλα αυτά στην πράξη δεν χρησιμοποιείται συχνά το multicast γιατί είναι πολύ ακριβό υπολογιστικά. Κάθε συσκευή που θα πρέπει να μπορεί να ρυθμίζεται έτσι ώστε να ξέρει εάν είναι μέρος μιας multicast ομάδας όπως και οι router θα πρέπει να ξέρουν εάν σε ένα από τα δίκτυα που παρέχουν πρόσβαση υπάρχει μέρος μια multicast ομάδας.

2.4.3.2 BACnet Broadcast

Υπάρχουν τριών ειδών broadcast: το local broadcast που αφορά μετάδοση σε όλους τους σταθμούς στο ίδιο δίκτυο, το remote broadcast που αφορά μετάδοση σε όλους τους σταθμούς σε ένα DNET (Destination Network) και το global broadcast που αφορά την μετάδοση σε όλους τους σταθμούς σε όλα τα δίκτυα. Σε όλους τους τύπους broadcast έχουμε συγκεκριμένα broadcast address π.χ. για Local broadcast σε MSTP χρησιμοποιείται το address X''FF'' ενώ για remote address χρησιμοποιείται το DNET του δικτύου προορισμού με DLEN ίσο με 0. Τέλος για Global broadcast χρησιμοποιείται το DNET X''FF''.

2.5 BACnet Data Link Layer

Όπως αναφέρθηκε στην παράγραφο 2.2 το BACnet υποστηρίζει 6 διαφορετικού τύπου πρωτόκολλα ζεύξης δεδομένων (data link). Στο παρόν κείμενο θα αναφερθούν τα δυο Data Link πρωτόκολλα που χρησιμοποιήθηκαν στην συγκεκριμένη εργασία: το Master Slave/Token Passing (MS/TP) και το Point-to-Point (PTP).

2.5.1 Master Slave / Token Passing (MS/TP)

Το πρωτόκολλο MS/TP απαιτεί στο φυσικό επίπεδο (physical layer) να υπάρχει hardware που να υποστηρίζει Universal Asynchronous Receiver/Transmitter (UART), έναν πομποδέκτη (transceiver) EIA-485 ή πιο απλά RS-485 και ένα timer με δυνατότητα χρονικής ανάλυσης μικρότερης ίσης των 5 millisecond.

Όπως γίνεται εμφανές από την ονομασία του πρωτοκόλλου υπάρχουν δυο διακριτά είδη συσκευών, οι κυρίαρχοι (master) και οι σκλάβοι (slave). Κάθε συσκευή μπορεί να είναι μόνο ένα από τα δυο.

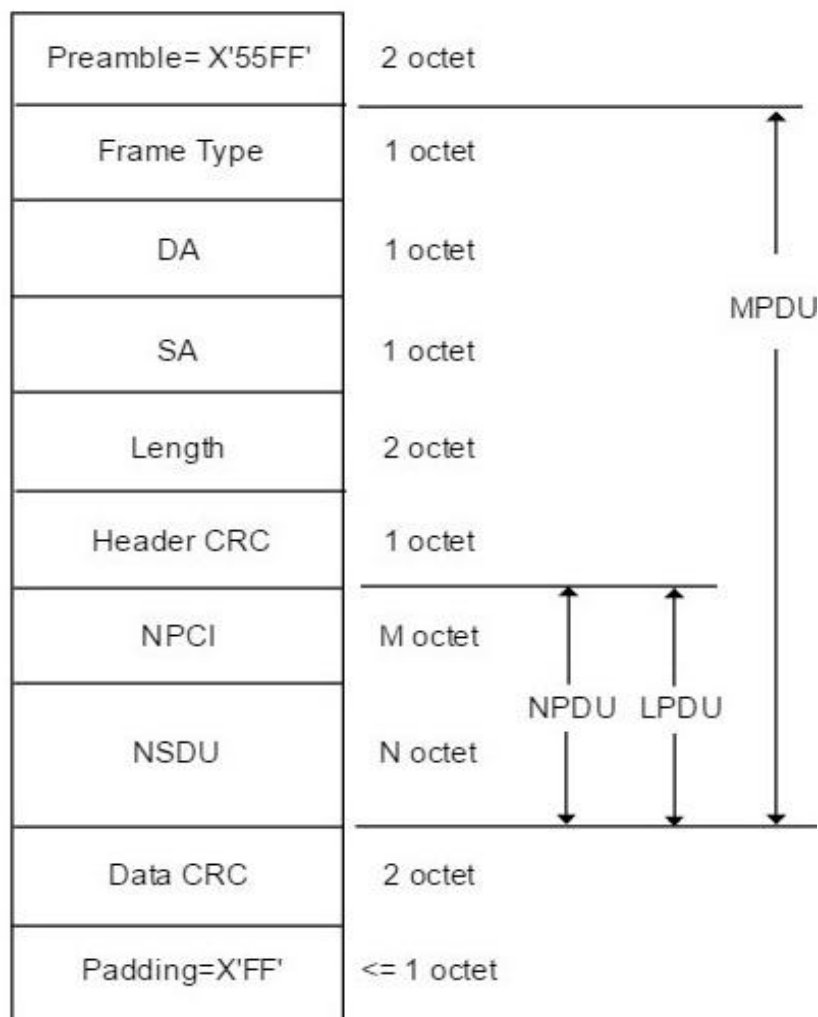
Οι master συμμετέχουν σε ένα Peer-To-Peer (PTP) δίκτυο όπου ανταλλάσσουν κυκλικά ένα κουπόνι (token) ενώ μπορούν να ξεκινάνε αλλά και να απαντάνε σε Confirmed Service Request. Το token είναι ένα ειδικού τύπου πακέτο το οποίο χρησιμοποιούν οι master για να μιλούν με αποκλειστικότητα στον δίαυλο (bus). Ένας master περιμένει μέχρι να του αποσταλεί το πακέτο token. Όταν λάβει ένα token εάν δεν έχει κάτι να αποστείλει (να ξεκινήσει κάποιο request για παράδειγμα) στέλνει το πακέτο token στον επόμενο κόμβο που συμμετέχει στον δακτύλιο (ring). Εάν θέλει να αποστείλει κάτι το στέλνει είτε μέχρι να στείλει ένα μέγιστο αριθμό frame που ορίζεται στην αρχή για όλους τους κόμβους ή μέχρι να επέλθει ένα token timeout (μέγιστος χρόνος κατοχής του token από έναν κόμβο) οπότε και στέλνει το token στον επόμενο κόμβο. Εάν στείλει ένα Confirmed Request μπορεί να περιμένει μέχρι να του αποσταλούν τα δεδομένα σαν απάντηση.

Οι slave μπορούν μόνο να απαντάνε σε Confirmed Service Request που τους απευθύνει κάποιος master και δεν μπορούν να ξεκινήσουν κάποιο request. Μεταδίδουν στο κοινό μέσο όταν ο master τους στείλει ένα confirmed request και δεσμεύσει το κοινό μέσο μέχρι να του αποσταλούν τα δεδομένα που ζήτησε. Ο λόγος για την ύπαρξη συσκευών slave είναι ότι είναι σημαντικό φθηνότερο από άποψη κόστους να αναπτυχθεί μια συσκευή slave (μια συσκευή slave δεν χρειάζεται να λαμβάνει υπόψη της περιπτώσεις απώλειας από του token για παράδειγμα).

Πλέον όμως τείνει να επικρατήσει η άποψη ότι οι μειωμένες δυνατότητες των slave δεν αντισταθμίζονται από το οριακό κέρδος στο κόστος, ακόμη περισσότερο ειδικά όταν κάποιος έχει αναπτύξει ήδη το λογισμικό για μια συσκευή master δεν υπάρχει λόγος να μην το χρησιμοποιήσει ανεξαρτήτως του σκοπού της συσκευής. Έτσι πλέον οι περισσότερες συσκευές είναι master.

Πρέπει επίσης να υποστηρίζονται τουλάχιστον Baud Rate για την επικοινωνία στο φυσικό επίπεδο 9.600 bps και 384000 bps. Μπορούν να υποστηρίζονται μεγαλύτεροι ρυθμοί μέχρι 115.200 bps σε δίκτυα μήκους μέχρι 1000 μέτρα για αξιόπιστη επικοινωνία (για μικρότερους ρυθμούς μετάδοσης το όριο ανεβαίνει στα 1200 μέτρα).

Το format ενός πακέτου MS/TP φαίνεται στην εικόνα 2.8. Η ακολουθία αρχής X'55FF' είναι μια ακολουθία bit επιλεγμένη έτσι να μην είναι πιθανό να προκύψει από τυχαίο θόρυβο και χρησιμοποιείται σαν ασφάλεια (μαζί με την ακολουθία ηρεμίας idle) σε λάθος πακέτο δεδομένων.



Εικόνα 2.8 Το Frame του Format MS/TP

Το επόμενο πεδίο είναι το frame type. Αυτή την στιγμή υπάρχουν 8 τύποι frame. Ακολουθεί το MAC address που έχει μέγεθος ενός octet (8-bit). Οι διευθύνσεις από 0-127 ανατίθενται σε masters αποκλειστικά ενώ από 128 μέχρι 254 μπορούν να ανατεθούν είτε σε master είτε σε slave ενώ η διεύθυνση 255 είναι δεσμευμένη για broadcast.

Ένα MS/TP frame έχει πάντα ένα τουλάχιστον CRC (Cyclic Redundancy Check). Το CRC είναι ένας τύπος έλεγχου δεδομένων για την απόφαση εάν ένα πακέτο έχει ληφθεί σωστά. Ένα n-bit CRC είναι κατάλληλο για την ανίχνευση εκρηκτικού θορύβου (burst noise) ,που είναι και ο συνηθέστερος σε ένα κανάλι επικοινωνίας λόγω ηλεκτρομαγνητικού θορύβου ,όχι μεγαλύτερου των n-bits (για το CRC [1.7]). Υπάρχει ένα CRC 8-bit για τα δεδομένα επικεφαλίδας του MSTP και ένα CRC 16-bit για τον έλεγχο του NPDU.

Στην εικόνα 2.8 φαίνεται ότι μέρος του MS/TP frame είναι το field Length που αναφέρετε στο μέγεθος NPDU σε octets και μπορεί να κυμαίνεται το μέχρι 501 octetto ανώτερο. Τέλος το προαιρετικό padding με X'FF' έχει σκοπό να ευκολύνει την χρήση του UART. Το MS/TP απαιτεί το κλείσιμο του RS-485 transmitter για κάποια bit times, συνήθως 15 bit times (bit time καλείται η ποσότητα ($\frac{1}{\text{Baud Rate}}$) που έχει διαστάσεις χρόνου). Εάν το padding γίνεται με άσσους και είναι γνωστό ότι η κατάσταση idle που είναι ανενεργή η USART έχει πάλι άσσους στο δίαυλο τότε υπάρχει μια δυνατότητα ευελιξίας στο πότε μπορεί να αφήσει τον δίαυλο η συσκευή καθώς το τέλος ενός frame και η κατάσταση idle του διαύλου είναι αδιαχώριστες.

Ακολουθούν τα frame types Που χρησιμοποιεί το MSTP.

Token (Frame Type 0): Το Token είναι το πακέτο που χρησιμοποιείται για να παρέχει στους masters δικαίωμα μετάδοσης στον δίαυλο και αποστέλλεται κυκλικά στον δίαυλο σε όλους τους master.

Poll for Master (Frame Type 1): Το Poll For Master είναι το frame που μεταδίδεται από όλους τους Master που θέλουν να συμμετέχουν στον δακτύλιο κατά την αρχικοποίηση αλλά και περιοδικά κατά την κανονική λειτουργία του δικτύου. Χρησιμοποιείται για την ανακάλυψη των υπολοίπων Master στο δίκτυο και να αποφασιστεί ο διάδοχος κόμβος που θα αποσταλεί το Token στην συνέχεια. Το Frame. Μεταδίδεται με Local Broadcast σε όλους τους κόμβους του δικτύου. Οι Slave κόμβοι απλά αγνοούν το συγκεκριμένο πακέτο.

Reply To Poll For Master (Frame Type 2): Το συγκεκριμένο frame χρησιμοποιείται σαν απάντηση στο Poll For Master και δηλώνει ότι ο κόμβος που απαντάει με αυτό το frame ότι θέλει να συμμετέχει στον δακτύλιο.

Test Request (Frame Type 3): Το frame αυτό χρησιμοποιείται για να ξεκινήσει ένα Loopback test για το έλεγχο του μονοπατιού μετάδοσης (Loopback test καλείται η διαδικασία που μια συσκευή στέλνει ένα πακέτο και περιμένει να της επιστρέψει αφού έχει κάνει τον γύρω του δικτύου).

Test Response (Frame Type 4): Είναι το frame απάντησης στο Test Request frame. Το Data Field περιέχει τα ίδια δεδομένα με αυτά που στάλθηκαν στο Test Request τα οποία αποφασίζει η συσκευή που εκκινεί το Loopback Test.

BACnet Data Expecting Reply (Frame Type 5): Αυτό είναι το frame type που χρησιμοποιείται για την μεταφορά των confirmed service request.

BACnet Data Not Expecting Reply (Frame Type 6): Το frame type που χρησιμοποιείται για την μεταφορά των unconfirmed service request.

Reply Postponed (Frame Type 7): Το frame αυτό χρησιμοποιείται από εάν master για σηματοδοτήσει ότι δεν θα απαντήσει σε ένα Data Expecting Reply frame που έχει λάβει προηγουμένως.

2.5.2 Point-To-Point (PTP)

Το Point-To-Point είναι ένα πρωτόκολλο που χρησιμοποιείται για την δημιουργία επικοινωνίας μεταξύ δύο half-router χρησιμοποιώντας κάποιον μηχανισμό σημείου με σημείου επικοινωνίας στο φυσικό επίπεδο. Συνήθως το πρωτόκολλο στο φυσικό επίπεδο που χρησιμοποιείται είναι το EIA-232 ή RS-232 μέσω μιας UART. Το PTP είναι σημαντικό διαφορετικό από ένα τυπικό δίκτυο LAN ή και από το ring δίκτυο που αναφέρθηκα και πριν.

Καταρχήν ο αριθμός των κόμβων που συμμετέχουν στην επικοινωνία είναι πάντα δυο και η επικοινωνία μπορεί να είναι είτε half είτε full duplex. Η κύρια διαφορά είναι όμως ότι η σύνδεση δεν είναι μόνιμη αλλά παροδική γεγονός που σημαίνει ότι παρέχεται από το πρωτόκολλο όχι μόνο ο μηχανισμός ανταλλαγής δεδομένων αλλά και διαχείρισης της σύνδεσης (άνοιγμα σύνδεσης, κλείσιμο σύνδεσης). Επίσης η σύνδεση αυτή είναι αργή οπότε πρέπει να υπάρχει πρόβλεψη όταν σχεδιάζεται μια εφαρμογή που θα έχει πρόσβαση στο δίκτυο ώστε να αναμένονται αυτές οι καθυστερήσεις και να μην έχουμε dropped messages.

Στην αρχή της δημιουργίας της σύνδεσης η συσκευή που την εκκινεί παίρνει υπόψη της ότι η άλλη συσκευή μπορεί να μην είναι σε θέση να δεχθεί αυτού του είδους την δυαδική πληροφορία που αποστέλλει. Οπότε για την αρχή της σύνδεσης στέλνονται χαρακτήρες εκτυπώσιμοι και αποφεύγονται οι χαρακτήρες control (όπως οι χαρακτήρες XON (X"11") και XOFF (X"13")) οι οποίοι θα μπορούσαν να προκαλέσουν να κάνουν ένα modem να κλείσει η να απορρίψει τα δεδομένα μόνο

ακριβώς γιατί δεν βλέπει τα δεδομένα που έρχονται σαν δεδομένα αλλά σαν control πληροφορία.

Εφόσον έχουμε σύνδεση σε φυσικό επίπεδο ανταλλάσσονται frame μέχρι να επιτευχθεί μια BACnet σύνδεση που επιτρέπει στις δυο συσκευές πλέον να ανταλλάσουν BACnet PDU. Καθεμία από τις δυο συσκευές που συμμετέχουν στην σύνδεση μπορεί να τερματίσει την σύνδεση. Η σύνδεση παραμένει σε ισχύ μέχρι είτε και οι δυο συσκευές να επιβεβαιώσουν την αίτηση τερματισμού είτε μέχρι να αποφασίσει η μια συσκευή ότι έχει χαθεί η επαφή στο φυσικό μέσο. Υπάρχει ακόμα η περίπτωση η σύνδεση να τερματιστεί μονομερώς εάν δεν υπάρχει επικοινωνία για μεγάλο χρονικό διάστημα υποδηλώνοντας ότι το άλλο μέρος της σύνδεση είναι ανενεργό.

Σε αντίθεση με τα υπόλοιπα πρωτόκολλα Data Link που χρησιμοποιεί το BACnet το PTP frames είναι acknowledged γεγονός που σημαίνει ότι κάθε frame που αποστέλλεται πρέπει να απαντηθεί με το κατάλληλο acknowledge frame από τον δέκτη του. Αυτό γίνεται με μια προσέγγιση που προσομοιάζει το Alternating Bit Protocol (ABP). Το ABP είναι ένα πρωτόκολλο δικτύου που χρησιμοποιεί δυο ξεχωριστού τύπου frame τα τύπου 0 και τα τύπου 1. Ο πομπός μπορεί να έχει μέχρι ένα ανεπιβεβαίωτο frame από κάθε τύπο στο queue του. Αυτό σημαίνει ότι εάν για παράδειγμα ξεκινήσει στέλνοντας το πρώτο frame 0, συνεχίζει στέλνοντας το δεύτερο frame τύπου 1 και σταματάει μέχρι να του έρθει Acknowledgement τύπου 0 για το πρώτο frame. Μόνο τότε μπορεί να συνεχίσει την αποστολή στέλνοντας το τρίτο frame σαν τύπου 0 ή εάν επέλθει timeout χωρίς να λάβει acknowledgement για το frame 0 προχωρά σε retransmission του πρώτου frame(για το ABP [2.8]). Το PTP που χρησιμοποιεί το BACnet έχει ακριβώς αυτήν την λογική.

Όπως αναφέρθηκε πριν υπάρχουν τρία διακριτά μέρη στο PTP, η δημιουργία σύνδεσης, η ανταλλαγή δεδομένων και ο τερματισμός της σύνδεσης. Ακολουθεί μια σύντομη περιγραφή κάθε μέρους.

- *Δημιουργία σύνδεσης (Data Links Establishment)*

Εφόσον έχουμε φυσική σύνδεση μεταξύ των δυο συσκευών η συσκευή που εκκινεί την σύνδεση στέλνει την ακολουθία έναρξης που είναι τα γράμματα "BACNET<CR>" σύμφωνα με την κωδικοποίηση ANSI X3.4 ([για το ANSI X3.4 [2.9]) και ο χαρακτήρας CR Carriage Return. Εάν η συσκευή που δέχεται την ακολουθία έναρξης καταλάβει την σημασία της τότε απαντάει με ένα Connect Request Frame (περισσότερα για τα frame types αμέσως μετά) που μπορεί προαιρετικά να περιέχει έναν κωδικό για την επικοινωνία. Η συσκευή που ξεκίνησε την κλήση τότε μπορεί

εάν θέλει να αλλάξει τον κωδικό αλλά σε κάθε περίπτωση απαντά με ένα Connect Response Frame. Εάν και οι δύο συσκευές επιθυμούν την συνέχεια της σύνδεσης ανταλλάσσουν "Heartbeat (XON)" frame που σηματοδοτούν ότι είναι ενεργές συσκευές και ότι μπορούν να δεχτούν δεδομένα

- *Ανταλλαγή δεδομένων (Data Exchange)*

Σε αυτό το σημείο οι συσκευές μπορούν να ξεκινήσουν να ανταλλάσσουν BACnet PDU (τα οποία έχουν binary format σε αντίθεση με την προηγούμενη φάση που είχαμε ASCII format). Τα BACnet PDU ενσωματώνονται σε Data Request 0/1 όπως αναφέρθηκε προηγουμένως τα οποία γίνονται Acknowledged με Data Response 0/1 αντιστοίχως. Επιπλέον για να εξασφαλιστεί ότι μια συσκευή δεν θα κλείσει την σύνδεση επειδή πιστεύει ότι το άλλο μέρος της σύνδεσης είναι ανενεργό στέλνονται Heartbeat Frames (Frames που δείχνουν ότι η σύνδεση είναι "ζωντανή" εξού και Heartbeat) με περιοδικότητα που κυμαίνεται από το 1/3 μέχρι το 1/4 του χρόνου διακοπής της σύνδεσης. Τα πακέτα αυτά περιέχουν και πληροφορία flow control (ροής δεδομένων) που δείχνει ένα η συσκευή μπορεί να δεχτεί άλλα πακέτα.

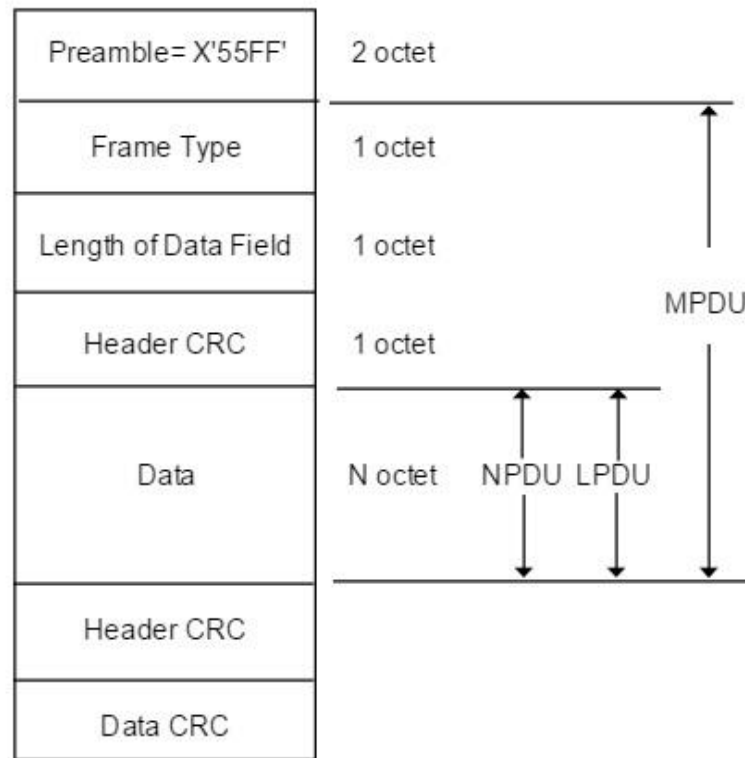
- *Τερματισμός σύνδεσης (Data Link Termination)*

Οποιαδήποτε από της δυο συσκευές μπορεί να τερματίσει την σύνδεση χρησιμοποιώντας τα Disconnect Request Frames και Disconnect Response. Η σύνδεση παραμένει ενεργή μέχρι να αποδεχθούν και οι δύο συσκευές το κλείσιμο της σύνδεσης, να χαθεί η σύνδεση σε φυσικό επίπεδο ή να έχουμε timeout που υποδηλώνει ότι η άλλη συσκευή δεν είναι ενεργή.

Τέλος να σημειωθεί ότι στα frames του PTP υπάρχουν δυο πεδία που ελέγχουν για την ακεραιότητα των δεδομένων μέσω CRC (Cyclic Redundancy Control, όπως στο MS/TP, βλέπε 2.4.1 σελ.26) ένα πεδίο για το header και ένα για τα data. Στην εικόνα φαίνεται το format ενός PTP frame.

Ακολουθούν τα Frame Types του PTP μαζί με μια σύντομη περιγραφή:

Heartbeat (XOFF/XON) Frame Type (0/1): Frame Type που αποστέλλεται περιοδικά όταν δεν υπάρχουν δεδομένα προς μετάδοση για να μην κλείσει η σύνδεση λόγω timeout. Τα XON,XOFF είναι η flow control πληροφορία που αναφέρθηκε πριν και υποδηλώνει ότι η συσκευή μπορεί να δεχθεί πακέτα (XON) ή ότι δεν μπορεί να δεχθεί (XOFF) αντίστοιχα.



Εικόνα 2.9 To frame format του PTP

DATA 0/1 (Frame Type 2/3): Frame μετάδοσης NPDU με αριθμό ακολουθία 0 ή 1 αντίστοιχα.

DATA ACK0/1 XOFF/XON (Frame Type 4-7): Θετικό Acknowledgement σε Data 0/1 αντίστοιχα μαζί με την αντίστοιχη flow control πληροφορία (XON,XOFF).

DATA NACK0/1 XOFF/XON (Frame Type 9-11): Αρνητικό Acknowledgement σε Data 0/1 αντίστοιχα μαζί με την αντίστοιχη flow control πληροφορία (XON,XOFF). Αρνητικό Acknowledgement σηματοδοτεί απόρριψη του πακέτου που έφτασε λόγω ενδεχομένως κάποιου λάθους σε κάποιο field του πακέτου η λόγω discard εξαιτίας γεμάτου buffer.

Connect Request (Frame Type 12): Το frame type που χρησιμοποιείται σαν απάντηση στην ακολουθία έναρξης σύνδεσης.

Connect Response (Frame Type 13): Frame type που είναι η απάντηση στο Connect Request.

Disconnect Request (Frame Type 14): Frame type που σηματοδοτεί ότι μια συσκευή θέλει να κλείσει την σύνδεση. Στο data field του πακέτου κωδικοποιείται και ο λόγος για την διακοπή σύνδεσης (π.χ. Δεν χρειάζεται να αποσταλούν άλλα δεδομένα, Λάθος κωδικός αποστολή κ.α.).

Disconnect Response (Frame Type 15): Απάντηση σε Disconnect Request που επιβεβαιώνει το κλείσιμο της σύνδεσης.

Test Request/Response (Frame Type 20/21): Frame που χρησιμοποιούνται για έναρξη loopback test και σαν απάντηση αντίστοιχα. Το Data Field είναι το ίδιο

και στα δύο πακέτα και το αποφασίζει η συσκευή που εκκινεί το test. (Loopback test καλείται η διαδικασία που μια συσκευή στέλνει ένα πακέτο και περιμένει να της επιστρέψει αφού έχει κάνει τον γύρω του δικτύου).

2.6 BACnet Physical Layer

2.6.1 Εισαγωγή στα σειριακά πρωτόκολλα

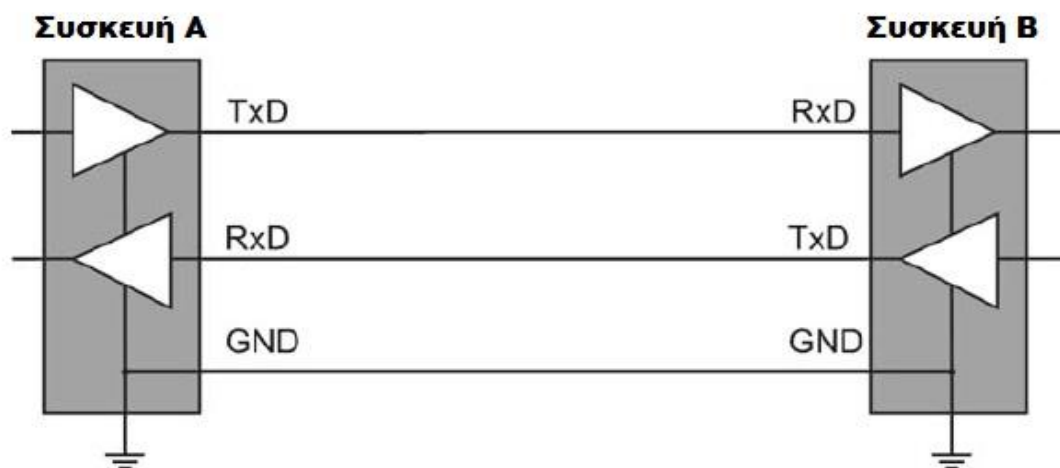
Όπως αναφέρθηκε στη παράγραφο 2.2 το BACnet υποστηρίζει 7 διαφορετικού τύπου πρωτόκολλα φυσικού επιπέδου (Physical Layer). Στο παρόν κείμενο θα αναφερθούν τα δυο Physical Layer πρωτόκολλα που χρησιμοποιήθηκαν στην συγκεκριμένη εργασία: Το RS-232 στην συμπλήρωμα στο φυσικό επίπεδο του PTP και το RS-485 αντίστοιχα για το MS/TP. Τα δύο αυτά πρωτόκολλα είναι σειριακά, δηλαδή μεταφέρουν bit πληροφορίας με σειριακό τρόπο το ένα μετά το άλλο. Γενικά ένα σειριακό πρωτόκολλο προσπαθεί να δημιουργήσει ένα μοναδικό μονοπάτι για τα δεδομένα ανεξαρτήτως του μέσου (είτε ενσύρματα είτε ασύρματα).

Τα σειριακά πρωτόκολλα βρίσκουν ακόμα εφαρμογές γιατί πρακτικά είναι δεν μπορεί να επιτευχθεί αξιόπιστη επικοινωνία σε υψηλούς ρυθμούς μετάδοσης με παράλληλο τρόπο. Παρόλο που τα RS 232/485 είναι από τα πιο παλιά πρωτόκολλα φυσικού επιπέδου και σε μεγάλο βαθμό αντικαθιστούνται από το USB πρωτόκολλο, που κυριαρχεί πλέον στην ενσύρματες επικοινωνίες μικρών αποστάσεων, εξακολουθούν και βρίσκουν εφαρμογές ακόμα και σήμερα.

2.6.2 Το πρωτόκολλο RS-232

Το RS-232 είναι ένα από το παλιότερα σειριακά πρωτόκολλα μεταφοράς δεδομένων. Η επίσημη ονομασία του είναι EIA/TIA-232. Το πρωτόκολλο μπορεί να μεταφέρει δεδομένα αποκλειστικά μεταξύ δυο συσκευών χρησιμοποιώντας τρεις συνδέσεις απαραίτητα οι οποίες φαίνονται στην εικόνα 2.10:

- Rx ή Receive Data που είναι το κανάλι όπου λαμβάνονται δεδομένα
- Tx ή Transmit Data που είναι το κανάλι αποστολής δεδομένων και
- GND ή Ground που είναι το κοινό σημείο αναφοράς για την τάση (γείωση)



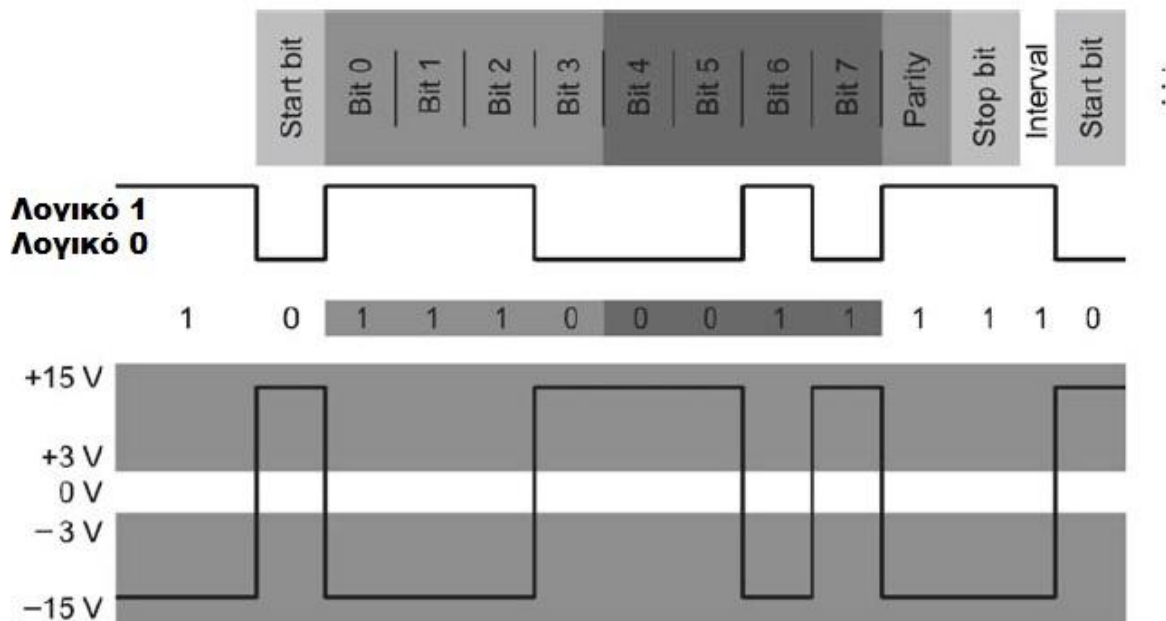
Εικόνα 2.10: RS-232 με 3 συνδέσεις: Tx, Rx και GND

Λόγω των ξεχωριστών καναλιών για λήψη/αποστολή δεδομένων έχουμε επικοινωνία full duplex (full duplex ονομάζεται η επικοινωνία όπου ταυτόχρονα μπορούν αν στέλνονται και να λαμβάνονται δεδομένα).

Όπως αναφέρθηκε πριν το πρωτόκολλο στέλνει δεδομένα σειριακά αντιστοιχώντας της ακολουθίες από bit σε επίπεδα τάσης δηλαδή σε διαφορά δυναμικού από το σημείο αναφοράς που είναι το GND. Μια τάση ανάμεσα στα -3 με -15 Volt ορίζεται σαν λογικό "1" (αντίθετα από την διαίσθηση που θα περίμενε κάποιος το 0) ενώ μια τάση ανάμεσα στα +3 με +15 Volt αναγνωρίζεται σαν λογικό "0". Η αντιστοιχία λογικών bit 0,1 σε επίπεδα τάσεων φαίνεται καθαρά στην εικόνα 2.9. Οποιαδήποτε τάση ανάμεσα στα +3 με τα -3 Volt απορρίπτεται σαν θόρυβος για αυτό το λόγο συνηθισμένα επίπεδα τάσεων είναι +/-5 Volt και τα +/-12Volt έτσι ώστε να υπάρχει περιθώρια ασφαλείας σε αυξομειώσεις της τάσης.

Μέσω του πρωτοκόλλου μεταδίδονται ομάδες 8-bit χρήσιμης πληροφορίας κάθε φορά μαζί με ένα Start Bit που είναι 0 (το bus στην κατάσταση idle είναι σε λογικό 1 συνήθως εξού και 0 το start bit), ένα bit ισοτιμίας (parity bit: δηλώνει το αριθμό των άσων στα αρχικά 8 data bit και δρα σαν ανιχνευτής περιττού αριθμού λαθών) και ένα stop bit που είναι 1 όσο και η idle κατάσταση του bus. Συνεπώς στέλνονται 11 bit για κάθε 8 bit πληροφορίας (βλέπε εικόνα 2.11).

Το φυσικό μέσο ανάμεσα στις δύο συσκευές μπορεί να είναι απλά παράλληλα καλώδια ή ένα συνεστραμμένο ζεύγος καλωδίων. Το μήκος του καλωδίου εξαρτάται από το data rate καθώς όσο μεγαλύτερο το data rate τόσο μικρότερη πρέπει να είναι η απόσταση για να εξασφαλιστεί αξιόπιστη επικοινωνία. Στην πράξη η απόσταση δεν πρέπει να υπερβαίνει τα 15 μέτρα ενώ το data rate κυμαίνεται μεταξύ 9600 Bd μέχρι 115 kBd.



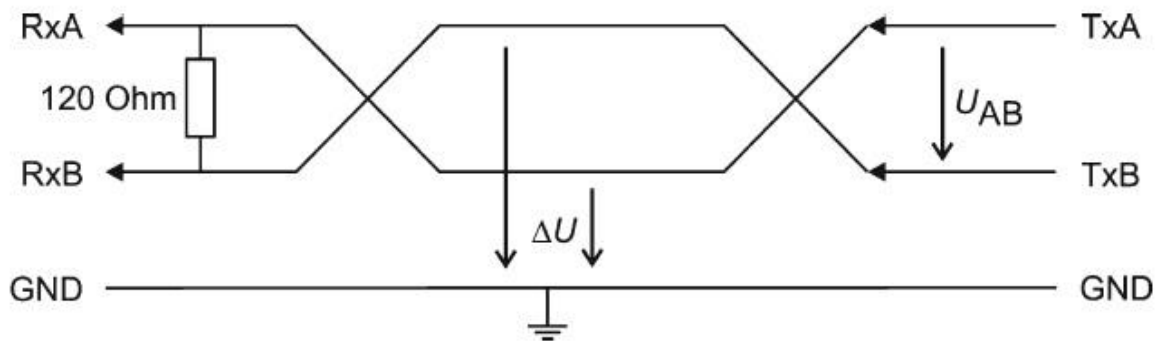
Εικόνα 2.11: Μεταφορά δεδομένων μέσω RS-232. Φαίνεται η αρχική λογική σειρά από 0,1 μαζί με την control πληροφορία και η αντιστοίχιση τους σε επίπεδα τάσης μεταξύ +/- 3-15 Volt. Οι γκρι ζώνες είναι τα επιτρεπτά επίπεδα τάσεων και οι λευκές αγνοούνται σαν θόρυβος

2.6.3 Το πρωτόκολλο RS-485

Το δεύτερο σειριακό πρωτόκολλο που χρησιμοποιήθηκε είναι το RS-485 σαν συμπλήρωμα του MS-TP στο φυσικό επίπεδο. Το RS-485 μοιάζει αρκετά στο RS-232 που είδαμε πριν αλλά έχει κάποιες σημαντικές διαφορές.

Καταρχάς το RS-485 είναι διαφορετικό πρωτόκολλο που σημαίνει ότι η κωδικοποίηση δεν γίνεται με βάση την διαφορά δυναμικού ενός κοινού σημείου διαφοράς αλλά με βάση την διαφορά δυναμικού των δύο καλωδίων που μεταφέρουν την πληροφορία μεταξύ τους. Πιο συγκεκριμένα υπάρχουν δυο καλώδια ανάμεσα σε δυο RS-485 συσκευές έστω A και B. Εάν η διαφορά δυναμικού U_{AB} είναι μεγαλύτερη από +200mV έχουμε λογικό 0 ενώ όταν U_{AB} είναι μικρότερα των -200mV έχουμε λογικό 1 (βλέπε εικόνα 2.12).

Αυτή η λειτουργία του έχει δύο άμεσα αποτελέσματα, το πρωτόκολλο γίνεται αναγκαστικά half duplex δηλαδή δεν επιτρέπεται ταυτόχρονη μετάδοση και λήψη πληροφορία (για μετάδοση με δυο καλώδια καθώς με τέσσερα καλώδια θα μπορούσαμε να έχουμε full duplex επικοινωνία) καθώς δεν υπάρχουν άλλα διαθέσιμα κανάλια ενώ η διαφορετική κωδικοποίηση ακυρώνει αποτελεσματικά τον θόρυβο καθώς εάν ένα σημείο του καναλιού χτυπηθεί με θόρυβο ας πούμε το κανάλι A είναι πολύ πιθανό ότι εξίσου θα χτυπηθεί και το κανάλι B και συνεπώς η διαφορά δυναμικού μεταξύ τους θα μείνει η ίδια.

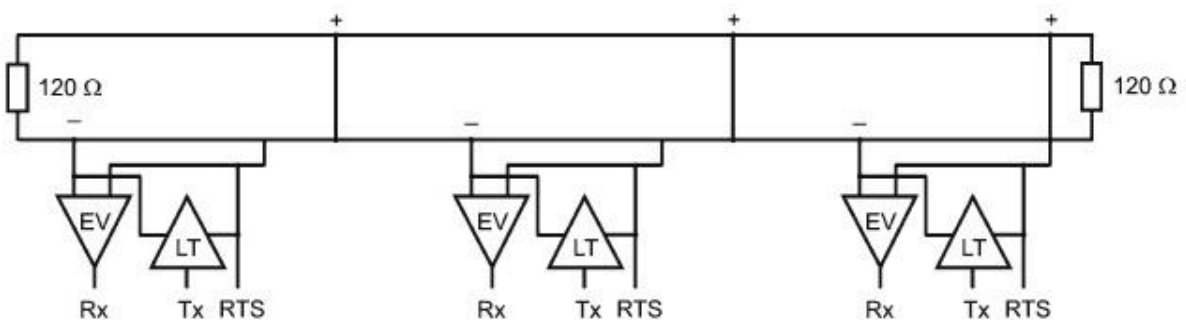


Εικόνα 2.12 RS-485 με την χρήση τριών καλωδίων (half duplex)

Αυτό το εγγενές πλεονέκτημα είναι που επιτρέπει στο πρωτόκολλο μεγαλύτερα baud rates σε μεγαλύτερες αποστάσεις και το κάνει πρακτικά πιο εφαρμόσιμο. Το τελικό baud rate εξαρτάται από το ίδιο το καλώδιο και την ποιότητα κατασκευής του αλλά μπορούν να επιτευχθούν ρυθμοί μετάδοσης μέχρι 10 Mbit/s, Η μέγιστη απόσταση που μπορεί να επιτευχθεί είναι 1200 μέτρα περίπου σε έναν data rate 120 Kbit/s.

Το RS-485 δεν υποστηρίζει μόνο επικοινωνία μεταξύ δύο σημείων αλλά μπορούν να υπάρχουν παραπάνω από δύο κόμβοι που μοιράζονται το ίδιο bus. Το πρωτόκολλο αναφέρει ότι μπορεί να υποστηριχθούν μέχρι 32 κόμβοι σε κάθε τμήμα (segment, βλέπε εικόνα 2.13). Κάθε τμήμα όμως θα πρέπει να έχει μια αντίσταση τερματισμού στο τέλος του.

Ένα ηλεκτρικό σήμα που ταξιδεύει σε ένα καλώδιο με μέση ταχύτητα διάδοση στα δύο τρίτα της ταχύτητας του φωτός. Σύμφωνα με τον νόμο διατήρησης της ενέργειας, που αναφέρει ότι η ενέργεια δεν μπορεί να καταστραφεί η να δημιουργηθεί αλλά μπορεί να αλλάζει μορφή, εάν ένα σήμα φθάσει στην άκρη ενός καλωδίου τότε, μιας και έχει ενέργεια, θα ανακλαστεί (σαν ένα ακουστικό σήμα) και υπερτεθεί στο εαυτό του στο αρχικό δηλαδή σήμα μετάδοσης. Η υπέρθεση αυτή προκαλεί θόρυβο στον δέκτη. Για να αποτραπεί αυτό χρησιμοποιούνται η αντιστάσεις τερματισμού που αναφέρθηκαν πριν. Η αντιστάσεις αυτές μετατρέπουν την ενέργεια του σήματος σε θερμότητα σταματώντας την ανάκλαση του. Το μέγεθος της αντίστασης καθορίζεται από την εμπέδηση κύματος (wave impedance) που είναι χαρακτηριστικό του καλωδίου.



Εικόνα 2.13 Δίκτυο RS-485

Η εμπέδηση κύματος είναι ο λόγος τάσεως, ρεύματος που χρησιμοποιείται από το ηλεκτρομαγνητικό κύμα για την μεταφορά δεδομένων. Το γινόμενο τάσεως ρεύματος εκφράζει την ενέργεια του σήματος. Εάν η εμπέδηση της αντίστασης τερματισμού ισούται με την εμπέδηση κύματος τότε όλη η ενέργεια του σήματος μετατρέπεται σε θερμότητα και έχουμε μηδενική ανάκλαση. Στην εικόνα 2.11 φαίνεται ένα segment RS-485 με αντιστάσεις τερματισμού μεγέθους 120 Ωm.

2.7 Ειδικεύοντας το BACnet

2.7.1 Interoperability Areas (IAs)

Μέχρι στιγμής έχει γίνει φανερό ότι το BACnet είναι ανοιχτό σε υλοποιήσεις και ότι μια BACnet συσκευή δεν χρειάζεται να υποστηρίζει όλες τις δυνατότητες του πρωτοκόλλου παρά μόνο όσες απαιτούνται για την συγκεκριμένη εφαρμογή. Εν τούτοις παραμένει επιτακτική η ανάγκη να είναι δυνατό η διαφορετικές δυνατότητες μιας BACnet συσκευής να μπορούν να περιγραφούν με έναν μοναδικό και δομημένο τρόπο έτσι ώστε να είναι στον καθένα προφανές τι δυνατότητες έχει μια συγκεκριμένη συσκευή. Από την ανάγκη αυτή της μοναδικής περιγραφής γεννήθηκε η κατηγοριοποίηση σε Interoperability Areas (περιοχές διαλειτουργικότητας IAs) της λειτουργίας κάθε συσκευής.

Η BACnet διαλειτουργικότητα όπως αυτή εκφράζεται μέσω του συνδυασμού συγκεκριμένων Object και Services που αλληλεπιδρούν με τα Object αυτά και συγκεκριμένα property τους μπορεί αν χωριστεί σε πέντε διακριτές κατηγορίες βάση συνάφειας και που κάθε κατηγορία έχει μια αυτονομία από τι υπόλοιπες. Οι πέντε Interoperability Areas είναι η εξής:

- **Data Sharing** (Μοίρασμα δεδομένων)
- **Alarm and Event Management** (Διαχείριση προειδοποιήσεων και γεγονότων)
- **Scheduling** (Δρομολόγηση)
- **Trending** (Καταγραφή των τάσεων του συστήματος)
- **Device Network Management** (Διαχείριση Δικτύου)

Κάθε IA σηματοδοτεί ένα σύνολο δυνατοτήτων, Κάθε δυνατότητα αντίστοιχα απαιτεί συγκεκριμένα στοιχεία του BACnet να είναι υλοποιημένα (Object, Services) για να μπορεί να συλλειτουργήσει με μια άλλη συσκευή που υλοποιεί το ίδιο interoperability με έναν εκ των προτέρων γνωστό και προβλέψιμο τρόπο. Το ποιες

λειτουργίες πρέπει να υποστηρίζονται σε καθένα από τα πέντε ΙΑ περιγράφεται μέσω των device profiles που αναφέρονται στην συνέχεια.

2.7.2 Device Profiles

Στο κεφάλαιο L του πρωτοκόλλου περιγράφονται οχτώ τυποποιημένοι τύποι BACnet συσκευών μαζί με την κωδικοποιημένη συντομογραφία τους.

- **BACnet Operator Workstation (B-OWS)**
- **BACnet Advanced Operator Workstation (B-AWS)**
- **BACnet Operator Display (B-OD)**
- **BACnet Building Controller (B-BC)**
- **BACnet Advanced Application Controller (B-AAC)**
- **BACnet Application Specific Controller (B-ASC)**
- **BACnet Smart Actuator (B-SA)**
- **BACnet Smart Sensor (B-SS)**

Καθεμία από τις πιο πάνω κατηγοριοποιήσεις σε Device Profile αντιστοιχίζεται σε συγκεκριμένη υποστήριξη λειτουργικότητας σε καθένα από τα πέντε Interoperability Area. Τα profile αυτά αναλύουν ποια services υποστηρίζει κάθε συσκευή και με ποια λειτουργικότητα (π.χ. ποιος εκκινεί το service). Στον παρακάτω πίνακα (2.2) φαίνονται για παράδειγμα τα οχτώ device profile μαζί με το ποια services απαιτείται να υποστηρίζουν για να κατηγοριοποιηθούν αναλόγως.

B-AWS	B-OWS	B-OD	B-BC	B-AAC	B-ASC	B-SA	B-SS
DS-RPM-A,B	DS-RP-A,B	DS-RP-A,B	DS-RP-A,B	DS-RP-B	DS-RP-B	DS-RP-B	DS-RP-B
DS-RPM-A	DS-RPM-A		DS-RPM-A,B	DS-RPM-B			
DS-WP-A	DS-WP-A	DS-WP-A	DS-WP-A,B	DS-WP-B	DS-WP-B	DS-WP-B	
DS-WPM-A	DS-WPM-A		DS-WPM-B	DS-WPM-B			
DS-AV-A	DS-V-A	DS-V-A					
DS-AM-A	DS-M-A	DS-M-A					

Πίνακας 2.2 Τα οχτώ Device Profile και τα απαιτούμενα BIBB για το Data Sharing (DS)

Το σύνολο των service που υποστηρίζεται σε ένα Device Profile καλείται και εκφράζουν την ελάχιστη λειτουργικότητα που πρέπει να υποστηρίζεται καλείται BACnet Interoperability Building Block (BIBB) και αναλύονται στην επόμενη παράγραφο.

2.7.3 BACnet Interoperability Building Block (BIBB)

Τα BIBB είναι ένας δομημένος τρόπος περιγραφής της λειτουργικότητας μιας συσκευής που ανταποκρίνεται στην κατηγοριοποίηση σε Device Profile. Στο πρωτόκολλο περιγράφονται 111 ξεχωριστά BIBB αυτή την στιγμή.

Ένα BIBB περιγράφεται από ένα κωδικό όπως στην εικόνα 1.12. Για παράδειγμα το BIBB, DS-RP-B σημαίνει τα εξής:

DS: Αναφέρεται σε service που αφορά το Interoperability Area Data Sharing

RP: Είναι η ονομασία του service, Read Property

B : Σημαίνει ότι η συσκευή λειτουργεί σαν server και όχι σαν client στο συγκεκριμένο BIBB.

Γενικότερα ο πρώτο κωδικός περιγράφει το Interoperability Area με το DS να σηματοδοτεί Data Sharing, το AE Alarm and Event, το SCHED Scheduling, το T Trending και τέλος το DM ή NM Device and Network Management.

Ο δεύτερος κωδικός περιγράφει το service (π.χ. WP Write Property). Ο κωδικός αυτός επειδή θα μπορούσε να μην είναι προφανές σε ποιο service αναφέρεται στην περιγραφή του BIBB αποσαφηνίζεται με μια αναφορά που δείχνει ξεκάθαρα σε ποιο service αντιστοιχεί. Μια τέτοια αναφορά θα ήταν μια εκχώρηση σε ένα πίνακα, για παράδειγμα στο λήμμα DS-RP-A στο κεφάλαιο L βλέπουμε το εξής:

<u>BACnet Service</u>	<u>Initiate</u>	<u>Execute</u>
ReadProperty	x	

Τέλος το τελευταίο κομμάτι του κωδικού του BIBB δείχνει εάν το service πρέπει να υποστηρίζεται σαν client οπότε έχουμε A, σαν server οπότε και B ή και τα δύο οπότε έχουμε κωδικό A,B.

Επίσης κάποια BIBB έχουν συγκεκριμένες απαιτήσεις σε “παρουσίαση” δηλαδή να παρουσιάζουν με συγκεκριμένο τρόπο πληροφορίες για τα για τα Object Properties των αντικειμένων που χρησιμοποιούν (το range των τιμών για παράδειγμα). Ένα μέρος ενός τέτοιου πίνακα απαιτήσεων φαίνεται στην που ακολουθεί από το BIBB του Data Sharing-Advanced View-A (DS-AV-A)

Enumerated: Present the complete range of standard values defines for all standard enumeration types for the Protocol_Revision claimed by the A device. The actual presentation of the values is unregistered (text, numeric, iconic, etc) as long as the individual values are distinguishable.

Real, Double: Present the complete value range, including special values as +-INF and NaN, unless specifically restricted by the standard for the property being displayed

Επιπλέον στην περιγραφή του BIBB περιγράφεται και το με ποια άλλα BIBB είναι interoperable (συνεργάσιμο). Για παράδειγμα για το DS-AV-A αναφέρεται ότι είναι interoperable με το BIBB DS-RP-B. Συνεπώς κατά την σχεδίαση μιας BACnet εφαρμογής ανάλογα με τα απαιτούμενα χαρακτηριστικά είναι δυνατόν μέσω τον BIBB να βρεθεί ο κατάλληλος συνδυασμός δυνατοτήτων για να ανταποκρίνονται στις ζητούμενες απαιτήσεις.

2.7.4 Protocol Implementation Conformance Statement (PICS)

Όλα τα παραπάνω που έχουν αναφερθεί και το σύνολο της λειτουργικότητας μιας συσκευής πρέπει να είναι προσβάσιμα σε οποιονδήποτε ενδιαφέρεται για μια BACnet συσκευή. Η τεκμηρίωση αυτή γίνεται μέσω τον Protocol Implementation Conformance Code (PICS) που συνοδεύουν μια πιστοποιημένη συσκευή BACnet. Στο κεφάλαιο 22 του πρωτοκόλλου περιγράφεται επακριβώς τι πρέπει να έχει ένα τέτοιο έγγραφο. Έτσι κάθε ενδιαφερόμενος μπορεί να ανατρέξει στο αντίστοιχο PICS της συσκευής που τον ενδιαφέρει και να εξακριβώσει ποιες λειτουργίες υποστηρίζει αλλά και εάν είναι Interoperable με μια άλλη συσκευή τουλάχιστον στο IA που τον ενδιαφέρουν.

Κεφάλαιο 3

Σχεδίαση και Υλοποίηση Πλατφόρμας ανάπτυξης εφαρμογών

3.1 Εισαγωγή

Στα πλαίσια αυτής της διπλωματικής εργασίας μου ανατέθηκε η σχεδίαση ενός ενσωματωμένου συστήματος κατάλληλου για ανάπτυξη εφαρμογών με το πρωτόκολλο BACnet που αναλύθηκε στο προηγούμενο κεφάλαιο.

Ζητήθηκαν οι εξής λειτουργικές απαιτήσεις:

- Μικρό κόστος ανάπτυξης εφαρμογών
- Ευελιξία ως προς τα πρωτόκολλα επιπέδου 2 (Data Link Layer) που υποστηρίζει η συσκευή. Το BACnet υποστηρίζει 7 πρωτόκολλα Data Link (βλέπε παράγραφο 2.2) και ζητήθηκε η υποστήριξη τουλάχιστον δύο πρωτοκόλλων: Master Slave/Token Passing (MSTP), που απαιτεί το πρωτόκολλο RS-485 στο επίπεδο 1, φυσικό επίπεδο) και το Point to Point (PTP) που απαιτεί αντιστοίχως στο RS-232. Τα δυο αυτά πρωτόκολλα αναλύθηκαν στο προηγούμενο κεφάλαιο (βλέπε παράγραφο 2.5).
- Μικρό τελικό κατασκευαστικό κόστος. Στην τελευταία παράγραφο του κεφαλαίου αυτού θα ακολουθήσει αναλυτική κοστολόγηση της κατασκευής.

Για τους παραπάνω λόγους επιλέχτηκε σαν υπόδειγμα σχεδίασης το BACnet MS/TP development kit [3.1]. Η συγκεκριμένη πλατφόρμα ικανοποιεί τις λειτουργικές απαιτήσεις, που αναφέρθηκαν πιο πριν. Το μικρό κόστος ανάπτυξης προκύπτει από το την ύπαρξη βιβλιοθήκης για την συγκεκριμένη πλακέτα που υποστηρίζει αρκετές από τις απαραίτητες λειτουργίες του πρωτοκόλλου. Η βιβλιοθήκη φιλοξενείται στο SourceForge[3.2] και υποστηρίζει Windows αλλά και Linux καθώς και μια πληθώρα ενσωματωμένων συστημάτων (περισσότερα για την βιβλιοθήκη στο κεφ.4). Επίσης έχουμε υποστήριξη εγγενώς από τα δύο απαιτούμενα πρωτόκολλα (MSTP, PTP)

3.2 Ανάλυση σχεδίασης

3.2.1 Ολοκληρωμένα που χρησιμοποιήθηκαν

1. Atmega 644P

Ο επεξεργαστής που είναι και η βάση της σχεδίασης(πλήρες datasheet [3.3]). Το package που επιλέχθηκε είναι το 44-lead TQFP. Τα χαρακτηριστικά του συγκεκριμένου επεξεργαστή που μας ενδιαφέρουν ως προς την υλοποίηση του πρωτοκόλλου είναι τα εξής.

- Εντολές πλάτους 8-bit
- Λειτουργική ταχύτητα μέχρι 20Mhz με τροφοδοσία από εξωτερικό κρύσταλλο. (Επιλέχθηκε κρύσταλλος 18.432 Mhz)
- Εσωτερική σειριακή μνήμη EEPROM (Erasable Programmable ROM) συνολικής χωρητικότητας 2KB
- JTAG interface για hardware debugging
- 32 γραμμές εισόδου/εξόδου όπου συνδεθήκαν οι binary αισθητήρες.
- Υποστήριξη 8 καναλιών ADC (Analog to Digital) ανάλυσης μέχρι 10-bit. Τα κανάλια αυτά είναι απαραίτητα γιατί σε αυτά συνδέονται οι αναλογικοί αισθητήρες που χρησιμοποιήθηκαν. (Περισσότερα για τους αισθητήρες στην παράγραφο 4.4.1)
- Δύο USART από τις οποίες χρησιμοποιήθηκε η μία (USART0) ως interface για το RS-485 (με baudrate 36800) όπου τρέχει το πρωτόκολλο και η δεύτερη (USART1) ως monitor για debugging και έλεγχο
- Two-wire serial Interface (TWI) που χρησιμοποιήθηκε για τους αισθητήρες που χρησιμοποιούν το πρωτόκολλο αυτό.

Να σημειωθεί ότι το TWI είναι το ίδιο πρωτόκολλο με το I²C απλά έχουν διαφορετική ονομασία για λόγους πνευματικών δικαιωμάτων (το I²C είναι νομικά κατοχυρωμένο από την Phillips Electronics). Στην πράξη το TWI είναι απολύτως συμβατό με το I²C αλλά και με μια οικογένεια πρωτοκόλλων όπως τα SMBus, PMBus κ.α.

- Σημαντικά χαμηλή κατανάλωση ισχύος. Σε ταχύτητα κρυστάλλου 1Mhz με τροφοδοσία 1.8 V και στους 25° C ο επεξεργαστής θέλει τροφοδοσία στην ενεργή του κατάσταση (που είναι η πιο ενεργοβόρα) 0.4mA

2. 24C16N Serial EPPROM

Σειριακή μνήμη που χρησιμοποιείται πρωτόκολλο I²C συνολικής χωρητικότητας 16KB (datasheet [3.4]). Η απαιτούμενη τροφοδοσία είναι 5V και υποστηρίζει ταχύτητες μέχρι 100KHz. Το package που επιλέχθηκε είναι το 8-pin DIP.

Να σημειωθεί ότι πέρα από τις τυπικές εισόδους (Vs για την τροφοδοσία, SDA, SCL για το I²C) τα τρία τελευταία ψηφία (LSB) της διεύθυνσης για το I²C είναι hardwired και αντιστοιχούν στις εισόδους A0, A1, A2 οι οποίες συνδέθηκαν στην γείωση καταστρώνοντας την διεύθυνση πάνω στον δίαυλο της μνήμης 0xA0 (αντίστοιχα στην γενική περίπτωση και σε δυαδικό 0b 1010 [A2][A1][A0][R/W] περισσότερα για το I²C στην παράγραφο 4.4.1)

3. MAX 481 RS-485 Transceiver

Χαμηλής κατανάλωσης transceiver (πομποδέκτης) της Maxim που χρησιμοποιείται για την μετατροπή από TTL (Transistor to Transistor Logic που είναι η έξοδος του επεξεργαστή) σε RS-485 που θέλουμε να είναι η τελική μας έξοδος (datasheet [3.5]). Ο συγκεκριμένος transceiver εγγυάται Error-Free επικοινωνία σε ταχύτητες μέχρι 250kbrps ενώ χρειάζεται και αυτός τροφοδοσία 5V. Το package που επιλέχθηκε το 8 pin DIP (Dual in Line Package).

4. MAX 202 RS-232 Transceiver

Transceiver που χρησιμοποιείται για την μετατροπή από TTL σε RS-232 που μπορεί να χρησιμοποιηθεί για επικοινωνία με τον υπολογιστή (είτε απευθείας καθώς υπάρχουν υπολογιστές που έχουν RS-232 είτε με ένα απλό RS-232 σε USB καλώδιο) είτε για πρωτόκολλο είτε για debugging. Έχει λειτουργική ταχύτητα μέχρι 150 kbps και απαιτεί τροφοδοσία 5V. Χρειάζεται 4 εξωτερικούς πυκνωτές 0.1μF ο καθένας ενώ επιλέχθηκε το package PDIP-16. (datasheet [3.6])

5. MCP 2200 USB to UART Converter

Transceiver που χρησιμοποιείται για την μετατροπή της εξόδου από UART σε USB για απευθείας σύνδεση με τον υπολογιστή χωρίς κάποιο ενδιάμεσο με ένα απλό USB-USB καλώδιο (datasheet [3.7]). Υποστηρίζει ταχύτητα μέχρι 12Mb/s και είναι συμβατό με το πρωτόκολλο USB 2.0. Χρειάζεται τροφοδοσία 5V και τροφοδοσία από εξωτερικό κρύσταλλο συχνότητας 12 MHz. Το εξάρτημα

χρησιμοποιήθηκε στην σχεδίαση αλλά δεν χρησιμοποιήθηκε στην πλακέτα καθώς δεν έφτασε στην κατοχή μου.

6. LM7800 Voltage Regulator

Voltage Regulator (Ρυθμιστής τάσης) που χρησιμοποιείται για να έχουμε τα επιθυμητά επίπεδα τάσης από την τροφοδοσία του κυκλώματος. Για την τροφοδοσία χρησιμοποιείται ένας απλός μετασχηματιστής που συνδέεται στο ηλεκτρικό δίκτυο (~220 VoltRMS, 50 Hz) και μας μετασχηματίζει την τάση στα 12V σταθερό με 350mA ρεύμα στην έξοδο. Η είσοδος τότε συνδέεται στην είσοδο του LM7800 και έχουμε σταθερή έξοδο 5V. Πέρα από το επίπεδο τάσης που προσαρμόζεται στα κατάλληλα επίπεδα για την λειτουργία του κυκλώματος αλλά και για την προστασία του από της διακυμάνσεις του δικτύου. Απαιτεί πυκνωτή 330μF για φιλτράρισμα της εισόδου αλλά και στην έξοδο για την βελτίωση της σταθερότητας του σήματος. Χρησιμοποιήθηκε το 3-pin package TO-220 (datasheet [3.8]).

3.2.2 Σχεδίαση του κυκλώματος

Τα παραπάνω εξαρτήματα χρησιμοποιήθηκαν για την σχεδίαση του συστήματος. Η σχεδίαση έγινε χρησιμοποιώντας το εργαλείο Proteus της Labcenter Electronics [3.9]. Το εργαλείο αυτό επιλέχθηκε για τα εγγενή του πλεονεκτήματα σε σχέση με άλλες ανταγωνιστικά σχεδιαστικά εργαλεία καθώς επιτρέπει και την σχεδίαση αλλά και την προσομοίωση ενός κυκλώματος πράγμα σημαντικά χρήσιμο στις αρχικές φάσεις υλοποίησης πριν υλοποιηθεί το σύστημα.

Παρακάτω επεξηγείτε η σχεδίαση με βάση τα χαρακτηριστικά των ολοκληρωμένων όπως αναλύθηκαν στις προηγούμενες παραγράφους.

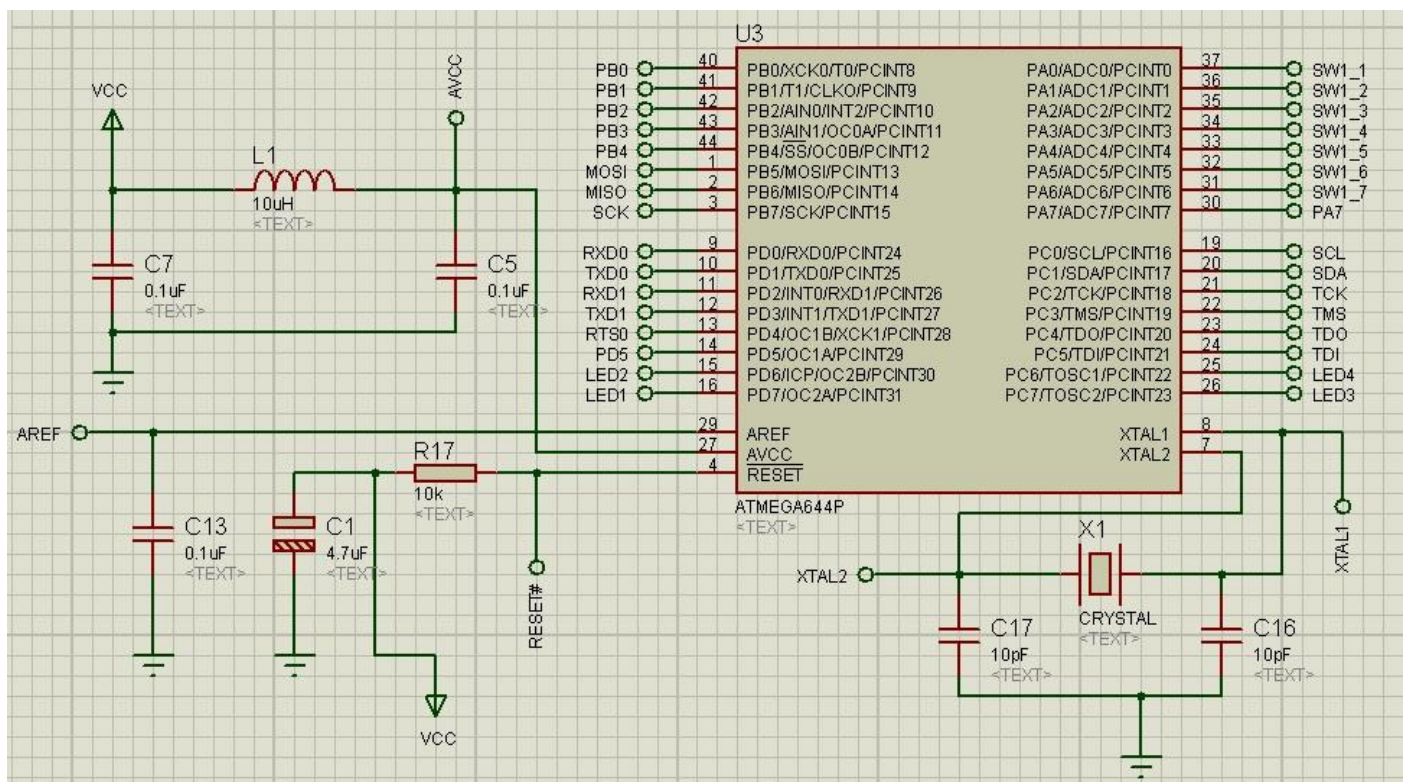
Στην εικόνα 3.1 φαίνεται η σχεδίαση του κυκλώματος για τον επεξεργαστή. Η είσοδος AVCC (είσοδος 29) είναι η τάση τροφοδοσίας για το PORT A του επεξεργαστή και γενικά για τον Analog-to-digital Converter. Όπως βλέπουμε από το datasheet του επεξεργαστή [2.1] στην σελίδα 7 (που περιγράφονται τα pin) ,απαιτείται για να λειτουργήσει σωστά το Analog-to-Digital Conversion (ADC) ένα χαμηλοπερατό φίλτρο.

Το χαμηλοπερατό φίλτρο υλοποιήθηκε με ένα LC φίλτρο σε διάταξη π (βλέπε [3.11] γενικά για χαμηλοπερατά φίλτρα π). Η τιμές των πυκνωτών και των πηνίων επιλέχθηκαν με βάση την σχεδίαση του BACnet development kit που αναφέρθηκε σαν υπόδειγμα της σχεδίασης [3.12].

Στην εικόνα Εικόνα 3.1 φαίνονται επίσης ένας πυκνωτής 0.1μF για τον έλεγχο της τάσης του σήματος AREF το οποίο είναι το σήμα αναφοράς για το ADC για την διαδικασία διαδοχικής εκτίμησης (βλέπε [3.13] για την λειτουργία του ADC).

Αυτός ο πυκνωτής λειτουργεί σαν πυκνωτής απόζευξης (bypass capacitor) απορροφώντας μικρές διακυμάνσεις του σήματος λειτουργώντας σαν «αποθήκη τάσης». Έτσι όταν έχουμε μια στιγμιαία πτώση τάσης ο πυκνωτής συμβάλει με το φορτίο του και έτσι γεμίζει τα κενά της τάσης ([3.14] γενικά για Bypass Capacitors). Μια άλλη ισοδύναμη ερμηνεία της συμβολής των πυκνωτών αυτού του τύπου, είναι ότι οι πυκνωτές αυτοί απορροφούν τα εναλλασσόμενα ρεύματα που δημιουργούνται από τις κυματώσεις των τάσεων. Τέτοιοι πυκνωτές απόζευξης έχουν χρησιμοποιηθεί σε όλες τις εισόδους των ολοκληρωμένων για την αποφυγή τυχόν κυμάτων στα σήματα εισόδου.

Επίσης ξανά για τον έλεγχο και ομαλοποίηση της τάσης χρησιμοποιούνται πυκνωτές και στον κρύσταλλο, που είναι συχνότητας 12.432 Mhz. Τέλος βλέπουμε

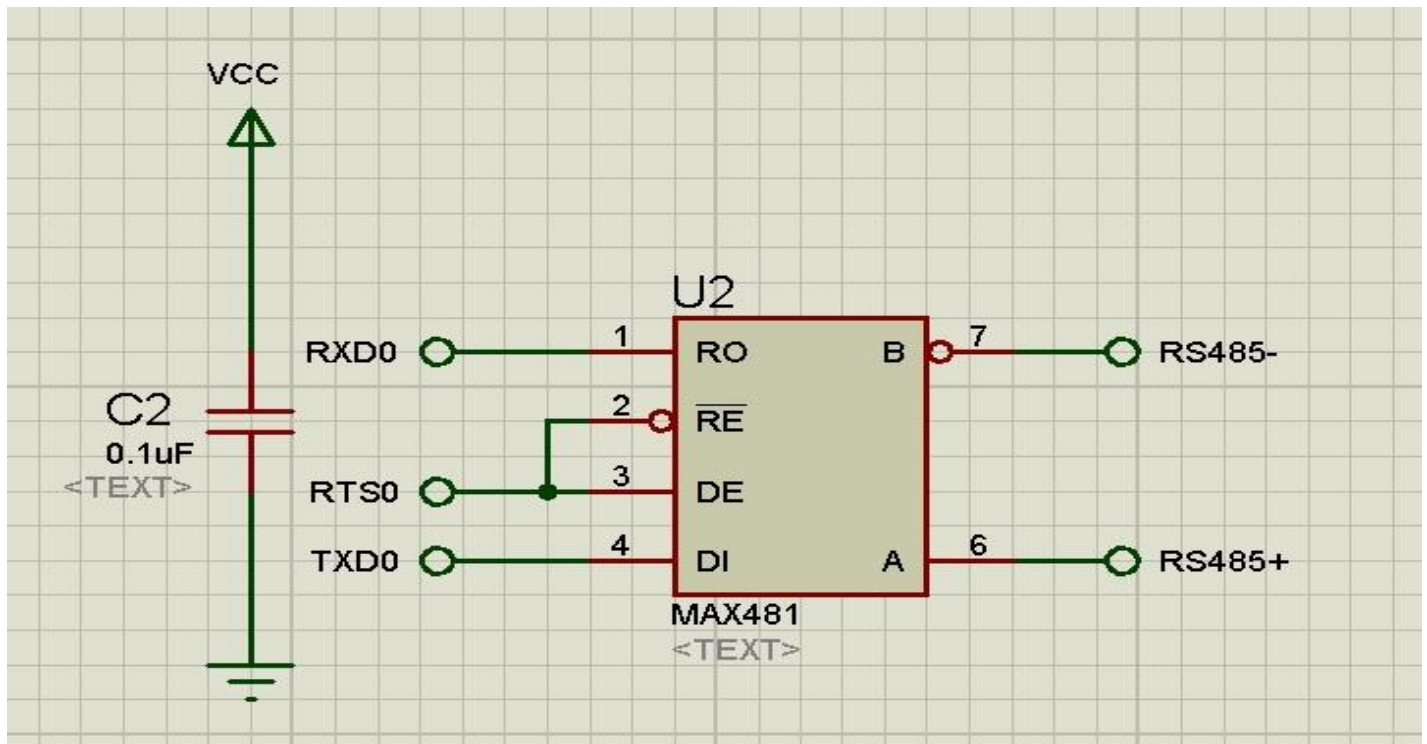


Εικόνα 3.1 Κύκλωμα Επεξεργαστή

έναν pull-up resistor στην είσοδο reset για την αποφυγή ενδιάμεσων επιπέδων τάσης στην κρίσιμη αυτή είσοδο καθώς ένα ενδιάμεσο επίπεδο τάσης στην είσοδο αυτή θα έκανε το σύστημα να λειτουργεί με ασταθή τρόπο.

Στην εικόνα 3.2 φαίνεται το ολοκληρωμένο MAX 481. Από το datasheet βλέπουμε ότι δεν χρειάζεται κάποιος πυκνωτής πέρα από τον C2 που

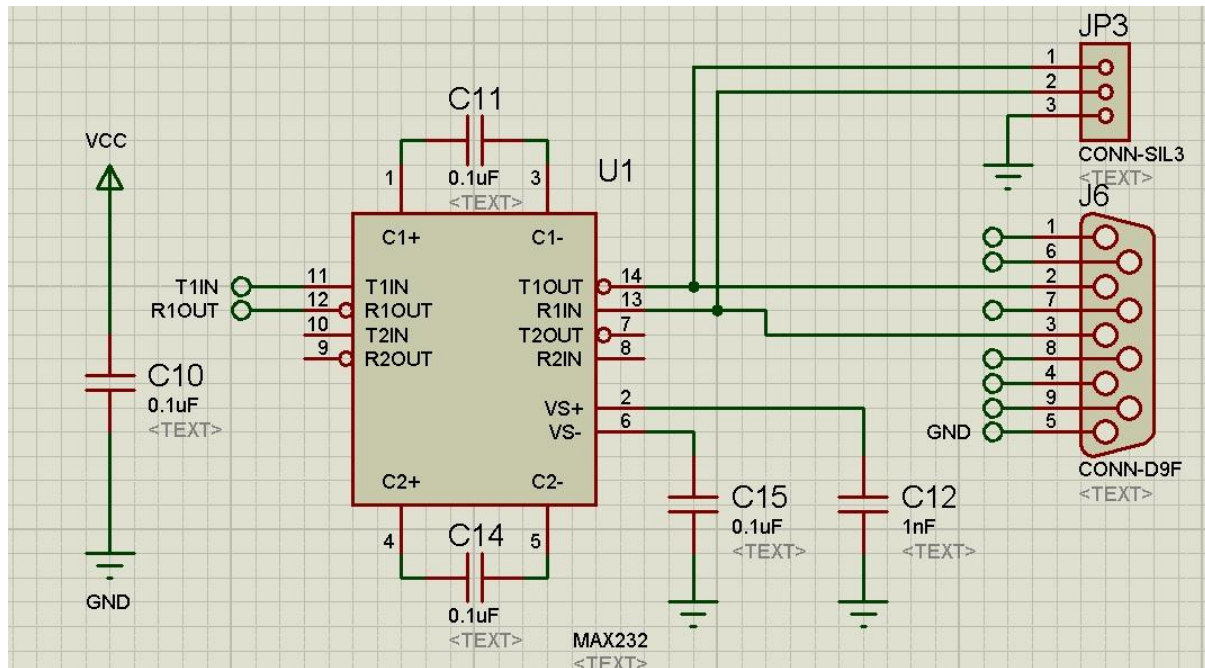
χρησιμοποιείται σαν bypass για τον έλεγχο του rippling της τάσης εισόδου όπως αναφέρθηκε και πιο πριν.



Εικόνα 3.2 Κύλωμα RS-485 Transceiver

Στην εικόνα 3.3 βλέπουμε την σχεδίαση του κυκλώματος για το ολοκληρωμένο MAX-232. Από το datasheet ([3.6] σελίδα 7) βλέπουμε ότι απαιτούνται πυκνωτές στα στις εισόδους C1+/-, C2+/- (C11, C14 στο σχηματικό) όπου όμως δεν είναι κρίσιμος ο τύπος του για την ομαλή λειτουργία του ολοκληρωμένου. Οποιοσδήποτε πυκνωτής μέχρι 10µF μπορεί να χρησιμοποιηθεί απρόσκοπτα παρόλαυτα προτείνεται η χρήση κεραμικού πυκνωτή 0.1µF όπου και επιλέχθηκε τελικώς.

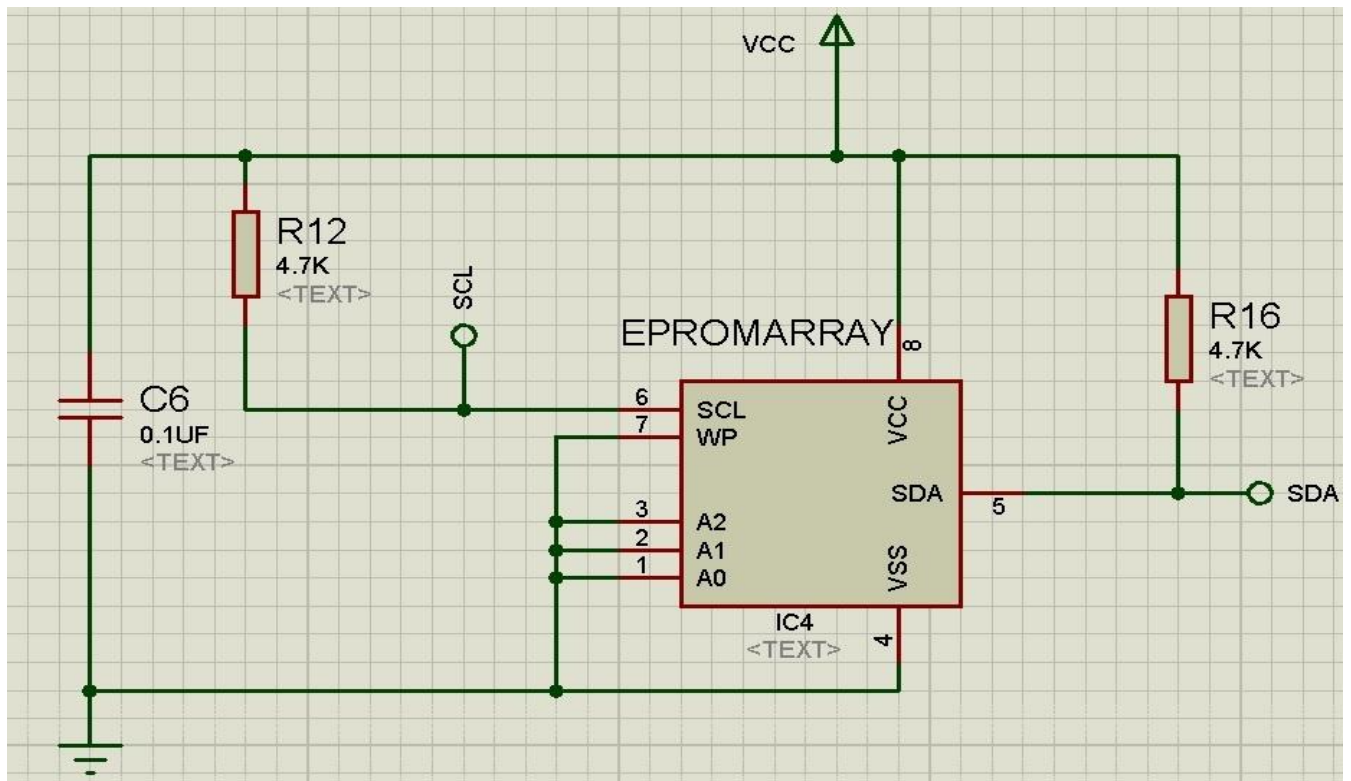
Ομοίως στα Vs+, Vs- προτείνεται η χρήση πυκνωτών 0.1 µF για τον περιορισμό της κυμάτωσης (C12, C15 στο σχηματικό). Επιπλέον στο χρησιμοποιείτε bypass πυκνωτής (απόζευξης) στην είσοδο Vcc. (Στην Εικόνα 3.3 δεν φαίνονται τα pin Vcc και GND καθώς υπάρχει δυνατότητα απόκρυψης το συγκεκριμένων pin από το σχεδιαστικό πρόγραμμα, έτσι ο bypass πυκνωτής ,που είναι ο C10, δεν φαίνεται να συνδέεται στο ολοκληρωμένο άμεσα ενώ είναι).



Εικόνα 3.3 Κύκλωμα RS-232 Transceiver

Επίσης βλέπουμε ότι οι έξοδοι κατευθύνονται σε δυο βάσεις την JP3 που είναι μια απλή βάση σύνδεσης με θηλυκούς κονέκτορες και την J6 που είναι μια έξοδος για την σύνδεση μέσω ενός απλού RS-232 καλωδίου σε υπολογιστή που έχει τέτοια είσοδο.

Στην Εικόνα 3.4 βλέπουμε το κύκλωμα για την μνήμη EEPROM. Η σειριακή μνήμη όπως αναφέρθηκε πριν επικοινωνεί μέσω του πρωτοκόλλου i2c (περισσότερα για το i2c στην παράγραφο 4.4.1).



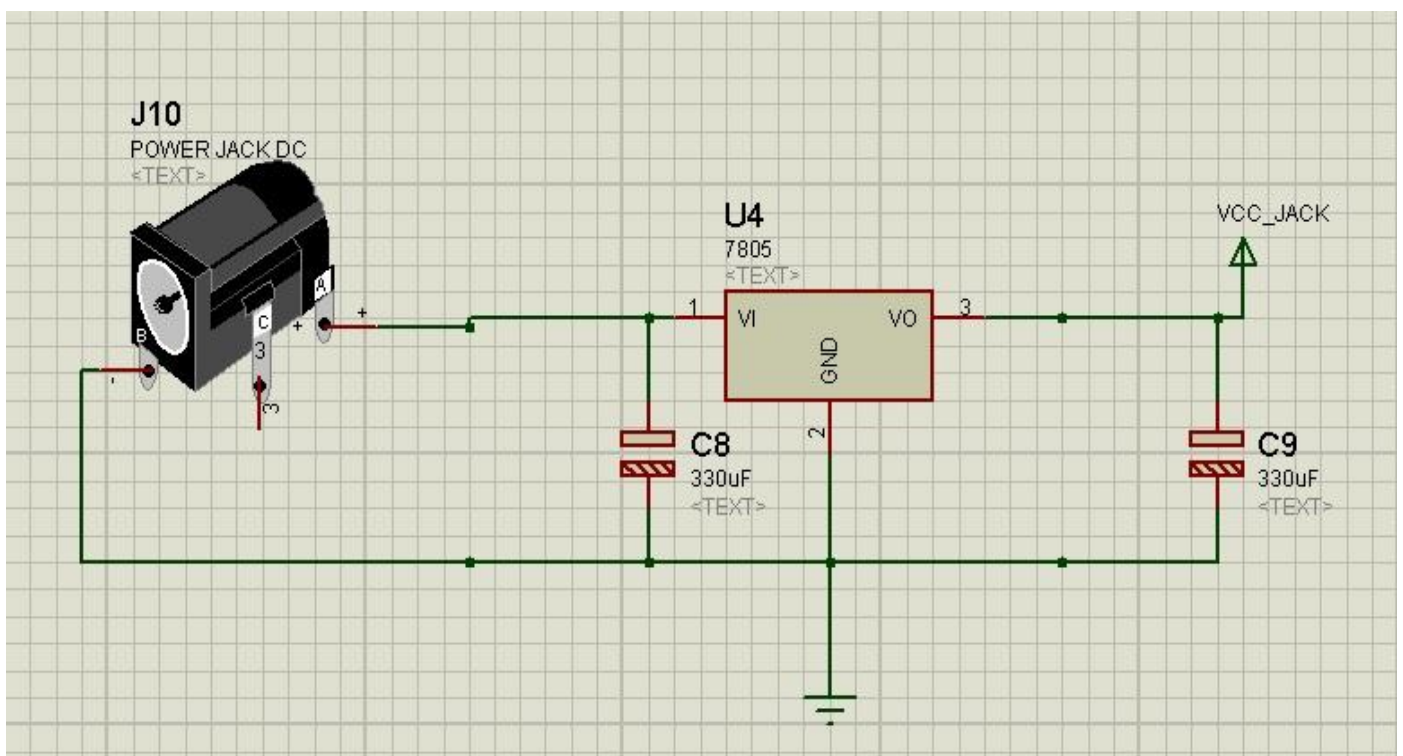
Εικόνα 3.4 Κύκλωμα σειριακής μνήμης EEPROM

Παρατηρούνται οι αντιστάσεις R12,R16 που είναι pull-up resistors για τις εισόδους SCL,SDA (Clock και Data για τα bus του i2c). Τα pull-up resistors χρησιμοποιούνται για να είναι πάντα η είσοδος του κυκλώματος σε ένα ορισμένο επίπεδο λογικής και να μην είναι σε ασταθείς καταστάσεις [3.10].

Η 8-bit διεύθυνση είναι όπως αναφέρθηκε πριν είναι 1010 [A2][A1][A0]συν το όγδοο bit που είναι το Read/Write. Όπως είναι εμφανές τα A2,A1,A0 είναι συνδεδεμένα στην γείωση οπότε για μια εγγραφή, η διεύθυνση είναι 0b10100000.

Στην Εικόνα 3.5 βλέπουμε το κύκλωμα του ρυθμιστή τάσης. Ο ρυθμιστής τάσης ότι δέχεται σαν είσοδο την έξοδο του βύσματος που έρχεται από τον μετασχηματιστή (12Vσταθερή τάση). Η είσοδος αυτή φιλτράρεται από έναν ηλεκτρολυτικό πυκνωτή bypass 330μF που εξομαλύνει την είσοδο . Ομοίως στην έξοδο έχουμε αντίστοιχα πυκνωτή 330μF για να σταθεροποιεί το σήμα εξόδου σαν αντίμετρο για μια στιγμιαία οριακή πτώση τάσης.

Οι ηλεκτρολυτικοί πυκνωτές επιλέχθηκαν λόγο της απαίτησης σε μέγεθος 330μF που είναι αρκετά μεγάλο για απλούς κεραμικούς πυκνωτές. Σημαντική παρατήρηση είναι ότι έχουν συγκεκριμένο προσανατολισμό.



Εικόνα 3.5 Κύκλωμα Ρυθμιστής Τάσης

3.3 Υλοποίηση κυκλώματος

Το επόμενο βήμα για την κατασκευή μας ήταν η δημιουργία του layout της κατασκευής δηλαδή της χωροταξικής διάταξης των ολοκληρωμένων και των ηλεκτρονικών στοιχείων που αναφερθήκαν προηγουμένως. Το σχεδιαστικό πρόγραμμα Proteus μας δίνει την δυνατότητα αυτή εκκινώντας μέσα από την αρχική σχεδίαση το πρόγραμμα ARRES που καθορίζει τρία πράγματα

1)Το PCB Layout (Printed Circuit Board Layout, δηλαδή της γραμμές που θα εκτυπωθούν ,τις φυσικές διαστάσεις κ.τ.λ.).

2)Την αντιστοίχιση μέσω του Netlist του Pinout των ολοκληρωμένων με τις πραγματικές γραμμές του PCB.

3)Το footprint των ολοκληρωμένων (περισσότερα για τα footprint στην συνέχεια)

3)Τις φυσικές διαστάσεις της πλακέτας.

Τα τελικά σχέδια που χρησιμοποιήθηκαν στην εκτύπωση υπάρχουν στα συνοδευτικά αρχεία της εργασίας στο path Files/Design.

Το footprint αναφέρεται στην φυσικά χαρακτηριστικά που αφορούν την σύνδεση ενός ολοκληρωμένου με την πλακέτα. Στην σχεδίαση αυτή χρησιμοποιήθηκαν δύο ειδών footprint:

- Through Hole που είναι η κατηγορία footprint που όπως υποδηλώνει το όνομα τους τα ολοκληρωμένα έχουν αρσενικές εξόδους που περνάνε μέσα από την αντίστοιχη θηλυκή τρύπα που έχει η πλακέτα [2.15]
- Surface Mount Device ή απλά SMD όπου το εξάρτημα ενώνεται απευθείας πάνω στην πλακέτα [3.16]

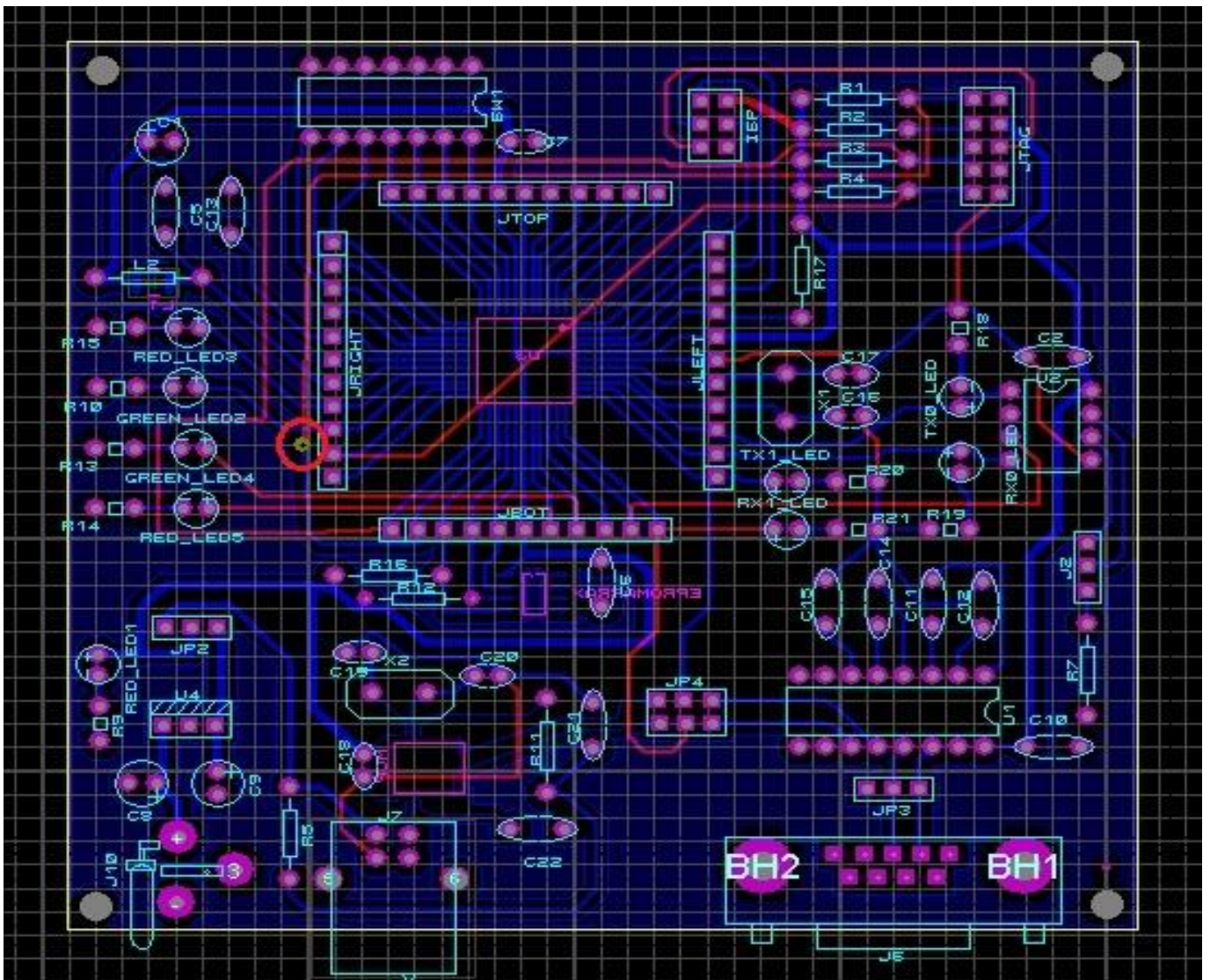
Οι πρώτες προσπάθειες σχεδίασης έγιναν με γνώμονα να επιτευχθεί Single Sided σχεδίαση δηλαδή τα component και οι γραμμές να υπάρχουν μόνο στην μία πλευρά της πλακέτας. Η Single Sided σχεδίαση προτιμάται περισσότερο γιατί είναι πιο εύκολο να εκτυπωθεί με χημικές μεθόδους καθώς δεν απαιτεί μεγάλη ακρίβεια σε αντίθεση με μια Dual Sided σχεδίαση που απαιτεί μεγάλη ακρίβεια στα σημεία που έχουμε drill holes που συνδέουν την πάνω με την κάτω πλευρά.

Λόγω της πολυπλοκότητας της σχεδίασης και της υψηλής της περιεκτικότητας σε components είτε αυτά είναι αντιστάσεις είτε ολοκληρωμένα έγινε αντιληπτό ότι η Single Sided δεν μπορούσε να επιτευχθεί με κανένα πολύπλοκο σχέδιο οπότε και με υπόδειξη του υπευθύνου της σχεδίασης κ.Γομπάκη ακολουθήθηκε η εξής πρακτική:

- Η σχεδίαση να είναι δύο επιπέδων (dual layer)

- Στο πάνω επίπεδο (top layer) να τοποθετούν τα ολοκληρωμένα που έχουν footprint through hole μαζί με τις αντιστάσεις τους πυκνωτές το πηνίο τα LED και της βάσεις σύνδεσης (που έχουν και αυτά through hole footprint)
- Στο κάτω επίπεδο (bottom layer) να κολληθούν τα ολοκληρωμένα που έχουν SMD footprint (η μνήμη, ο επεξεργαστής και το MCP)
- Το Common Ground ,η κοινή γείωση δηλαδή, να είναι στο bottom layer
- Να χρησιμοποιηθούν όσο το δυνατόν λιγότερα bypasses (παρακάμψεις) που συνδέουν το πάνω layer με το κάτω layer για να ελαχιστοποιηθεί ο κίνδυνος κάποιας αστοχίας κατά την εκτύπωση.

Με τις παραπάνω υποδείξεις έγινε η τοποθέτηση των στοιχείων στην πλακέτα η οποία στο τέλος είχε φυσικές διαστάσεις 102mm (χιλιοστόμετρα) οριζόντια και 97.5 mm κάθετα. Στην εικόνα 3.6 φαίνεται η τελική τοποθέτηση των component όπως φαίνεται μέσα από το σχεδιαστικό ARES.



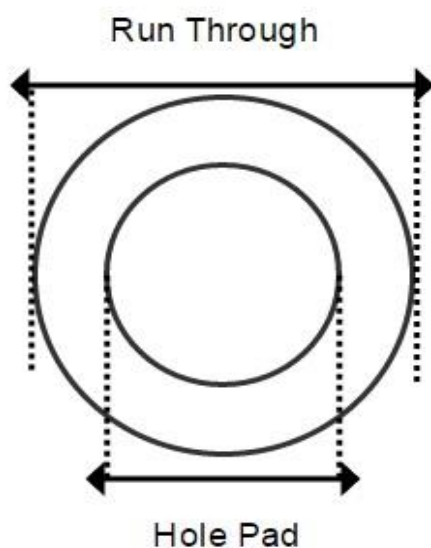
Εικόνα 3.6 Τελική σχεδίαση και PCB Layout. Βλέπουμε κυκλωμένο με κόκκινο το μοναδικό bypass που υπήρξε στην τελική σχεδίαση

Το τελικό layout που φαίνεται διαμορφώθηκε μετά από αρκετά revisions της σχεδίασης που προσέφεραν μικροβελτιώσεις. Παρατηρούμε τα εξής:

Οι μπλε γραμμές είναι το top layer. Στο ίδιο επίπεδο βρίσκεται και το Common Ground όπως έχει αναφερθεί από πριν. Το clearance η απόσταση του Common Ground από κάθε γειτονική διαδρομή του PCB είναι 30 th (0,762 mm στο SI), thousand of inch, εκατοστά της ίντσας που είναι μια συνηθισμένη μονάδα μέτρησης στα ηλεκτρονικά και ισούται με 0.0254 mm στο SI. Το clearance είναι μια πολύ σημαντική παράμετρος γιατί επηρεάζει άμεσα την απαιτούμενη τεχνική που χρειάζονται η κολλήσεις των εξαρτημάτων. Ένα πολύ μικρό clearance καθιστά εξαιρετικά δύσκολη την κόλληση καθώς υπάρχει πάντα ο κίνδυνος βραχυκυκλώματος μιας γραμμή με το Common Ground. Τα 30 th κρίθηκαν ικανοποιητική απόσταση για τις κολλήσεις.

Οι κόκκινες γραμμές αντιπροσωπεύουν το bottom layer. Παρατηρούμε ότι στο τελικό revision της σχεδίασης έχουμε μόνο ένα bypass που συνδέει απευθείας το μια γραμμή από το top στο bottom επίπεδο το οποίο είναι κυκλωμένο με κόκκινο χρώμα στην εικόνα 3.5. Οποιαδήποτε άλλη σύνδεση top/bottom επιπέδου έγινε μέσω των εξαρτημάτων που είχαν footprint through hole και ως εκ τούτου διατρέχουν και τα δυο επίπεδα της σχεδίασης.

Τέλος εξαιρετικά σημαντική είναι και η επιλογή της διαμέτρου των τρυπών για τα εξαρτήματα through hole. Δύο μεγέθη είναι σημαντικά εδώ το Run Through και το Hole Pad. Το τι αντιπροσωπεύει η καθεμία παράμετρος στο φυσικές διαστάσεις της τρύπας φαίνεται στο εικόνα 3.7.



Εικόνα 3.7 Οι παράμετροι Run Through και Hole Pad και η φυσική σημασία τους

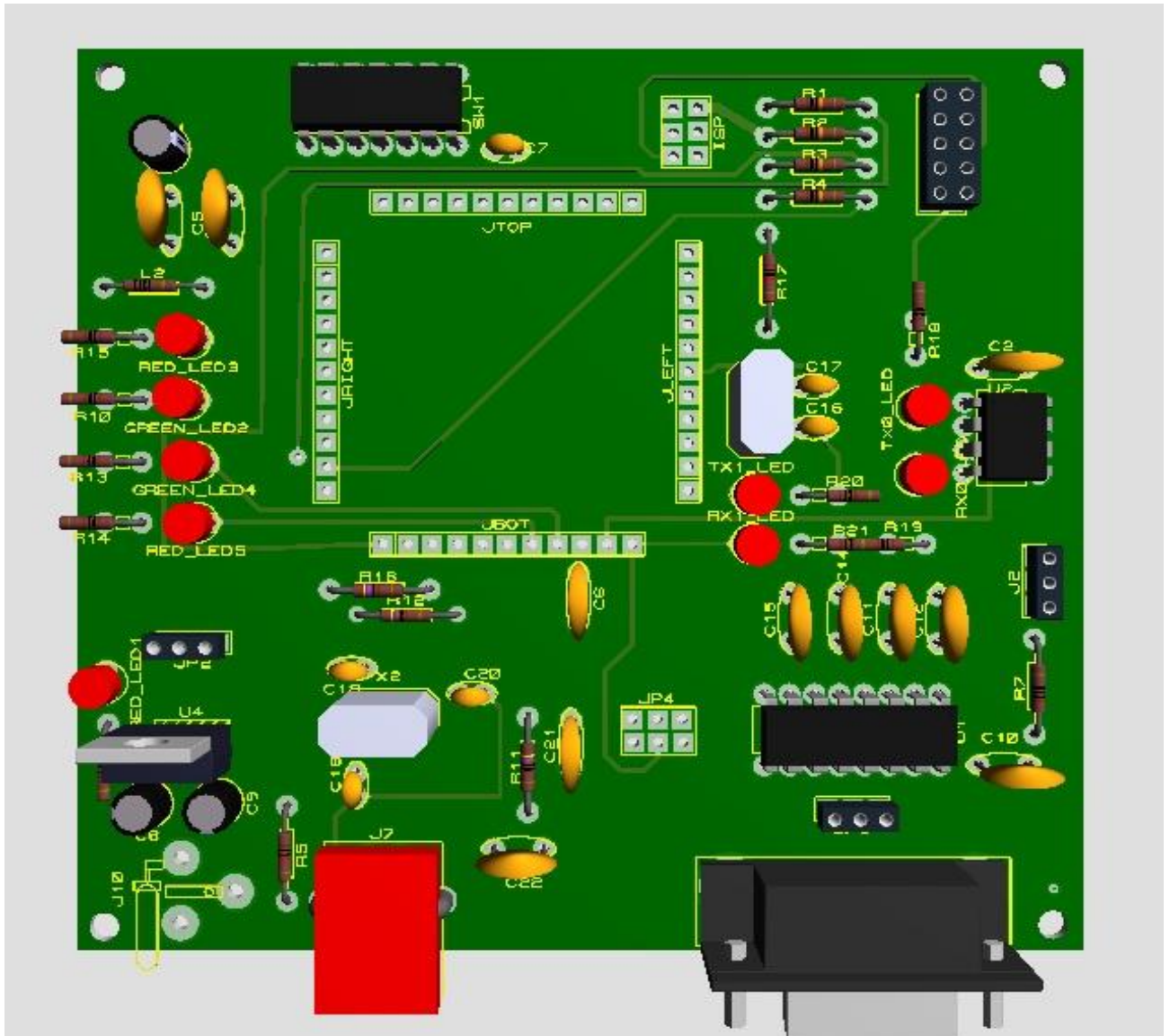
Όπως είναι εμφανές στην πιο δίπλα εικόνα το Run Through είναι συνολική διάμετρος της τρύπας μαζί με τον χαλκό που περιέχει γύρω της.

Το Hole Pad από την άλλη είναι η καθαρή διάμετρος της τρύπας μέγεθος που επηρεάζει άμεσα την μέγιστη διάμετρο των τρυπανιών που πρέπει να χρησιμοποιηθούν για να διανοιχθούν οι τρύπες αυτές. Όσο μικρότερο είναι το μέγεθος αυτό τόσο μεγαλύτερη ανάγκη σε ακρίβεια έχουμε.

Το Run Through από την άλλη επηρεάζει το πόσο χαλκό έχουμε διαθέσιμο για να πραγματοποιήσουμε την

κόλληση. Μια μεγάλη αναλογία Run Through/Pad (πάνω από 0.5) δημιουργεί μεγάλα προβλήματα στην σύνδεση. Συνεπώς μετά από δοκιμές και με βάση προηγούμενες εμπειρίες επιλέχθηκε το Run Through να είναι 90 th και το Pad είναι 40 th.

Στην εικόνα 3.8 φαίνεται η τελική προσομοίωση της σχεδίασης από το ARES.

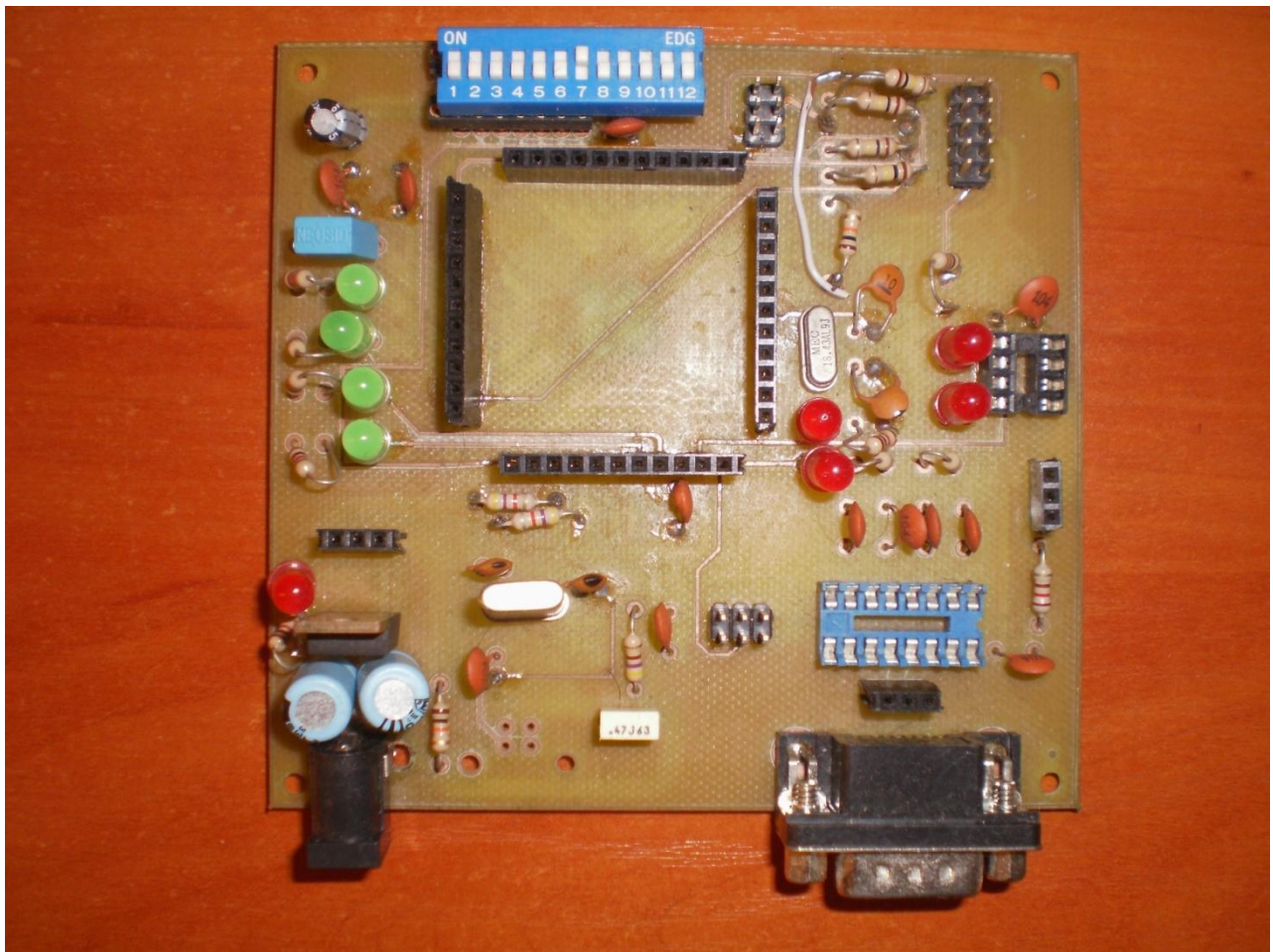


Εικόνα 3.8 Προσομοίωση τελικής σχεδίασης. Κάτοψη.

Το επόμενο βήμα ήταν η επιλογή της μεθόδου εκτύπωσης. Υπήρχαν δύο εναλλακτικές. Η επιλογή χημικής εκτύπωσης που γίνεται σε συνεργασία με εξωτερικό συνεργάτη του Πολυτεχνείου Κρήτης ή η επιλογή εκτύπωσης με μηχανήμα CNC (Computer Numerical Control) εντός του Πολυτεχνείου στο εργαστήριο τηλεπικοινωνιών του κ. Άγγελου Μπλέτσα.

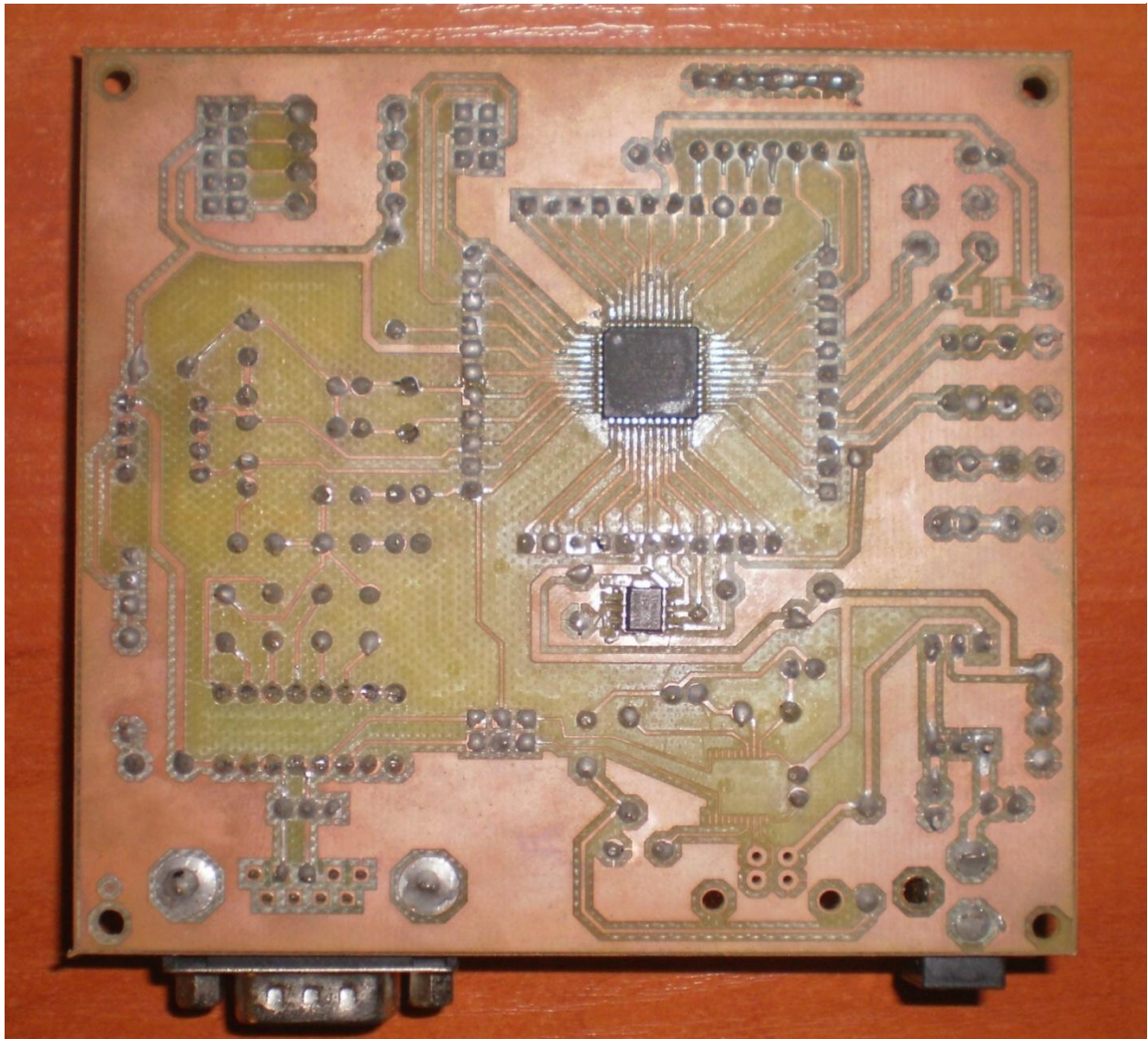
Προφανώς οι δεύτερη επιλογή έχει δύο πλεονεκτήματα, μεγαλύτερη ακρίβεια στην εκτύπωση και το γεγονός ότι την διάνοιξη των τρυπών την αναλαμβάνει το μηχάνημα. Εν τέλει εκτυπώθηκαν κάποιες δοκιμαστικές πλακέτες με την χημική μέθοδο αλλά η τελική πλακέτα εκτυπώθηκε με το μηχάνημα CNC στο εργαστήριο τηλεπικοινωνιών τον Ιούνιο του 2013.

Στην συνέχεια ο γράφων προμηθεύτηκε τα απαιτούμενα εξαρτήματα από προμηθευτή που συνεργάζεται με το Πολυτεχνείο και πραγματοποίησε τις κολλήσεις των Ιούνιο του 2013 στο Πολυτεχνείο Κρήτης. Το τελικό αποτέλεσμα φαίνεται στις πιο κάτω εικόνες. Στην εικόνα 3.9 φαίνεται η κάτοψη του συστήματος (η οπτική κοιτώντας προς τα κάτω) και στην 3.10 η άνοψη (η οπτική κοιτώντας προς τα πάνω)



Εικόνα 3.9 Κάτοψη τελικής σχεδίασης

Εικόνα 3.10 Άνοψη της τελικής σχεδίασης



3.4 Αναλυτική κοστολόγηση

Στην παράγραφο αυτή θα υπολογιστεί το κόστος των υλικών που χρησιμοποιήθηκαν. Για τις τρέχουσες τιμές των ηλεκτρονικών εξαρτημάτων χρησιμοποιήθηκαν οι τρέχουσες τιμές από δύο online καταστήματα: το <http://www.futurlec.com/> για τα ηλεκτρονικά εξαρτήματα (αντιστάσεις πυκνωτές, πηνίο, κρύσταλλοι και βύσματα) ενώ οι τιμές των ολοκληρωμένων πάρθηκαν από το www.findchips.com σε αντιστοιχία με τις τιμές των κατασκευαστών. Ακολουθεί ένας αναλυτικός πίνακας με όλα τα εξαρτήματα που χρησιμοποιήθηκαν και τις τρέχουσες τιμές τους

Όνομα Εξαρτήματος	Αριθμός	Κόστος	Σύνολο
Αντιστάσεις			
100 ΚΩ	4	\$0.10	\$0.40
10 ΚΩ	2	\$0.10	\$0.20
4.7 ΚΩ	2	\$0.10	\$0.20
1 ΚΩ	9	\$0.10	\$0.90
0.47 ΚΩ	1	\$0.10	\$0.10
0.12 ΚΩ	1	\$0.10	\$0.10
Πυκνωτές			
330 μF (Ηλεκτρολυτικός)	2	\$0.15	\$0.30
4.7 μF (Ηλεκτρολυτικός)	2	\$0.05	\$0.10
100 nf (Κεραμικός)	12	\$0.10	\$1.20
10 pf (Κεραμικός)	4	\$0.05	\$0.20
Πηνία			
10 μH	1	\$0.16	\$0.16
LED	9	\$0.08	\$0.72
Βύσματα			
Τροφοδοσίας	1	\$1.30	\$1.30
SIL-11-Female	4	\$0.25	\$0.25
SIL-3-Female	4	\$0.80	\$0.32
DIL3-Male	2	\$1.00	\$2.00
DIL5-Male	1	\$1.20	\$1.20
DIP switch 7	1	\$0.50	\$0.50
RS-232	1	\$1.25	\$1.25
USB	1	\$0.07	\$0.07
Κρύσταλλοι			
18.432 Mhz	1	\$0.75	\$0.75
12 Mhz	1	\$0.30	\$0.30
Ολοκληρωμένα			
Atmega 644P	1	\$4.94	\$4.94
MCP2200	1	\$1.59	\$1.59
MAX-481 (Maxim)	1	\$1.64	\$1.64
MAX-232 (T.I.)	1	\$0.51	\$0.51
EEPROM 24C08B	1	\$0.35	\$0.35
Σύνολο			\$21.55

Πίνακας 3.1 Αναλυτική κοστολόγηση κατασκευής

Η κοστολόγηση έγινε με μονάδα αναφοράς το USD καθώς τα περισσότερα εξαρτήματα ήταν κοστολογημένα σε αυτήν την μονάδα. Όσα εξαρτήματα βρέθηκαν σε άλλα νομίσματα μετατράπηκαν στο αντίστοιχο τους σε δολάρια με την τρέχουσα ισοτιμία (13/12/14). Η τελική τιμή που προέκυψε είναι 21.55 \$ ή 17.33 €. Η τιμή αυτή περιλαμβάνει το καθαρό υλικό κόστος του πρωτοτύπου ,όχι για μαζική

παραγωγή οπότε θα ήταν μικρότερο, χωρίς τις ανθρωποώρες για την σχεδίαση και την κόλληση του. Επίσης δεν λήφθηκε υπόψη κάποιο πρόσθετο κόστος για την εκτύπωση καθώς αυτή έγινε εντός του Πολυτεχνείου όπου υπάρχουν οι κατάλληλες υποδομές όπως αναφέρθηκε προγενέστερα. Η τιμή αυτή είναι ένα ενδεικτικό κάτω όριο για την τιμή μιας πλατφόρμας ανάπτυξης εφαρμογών για BACnet MS/TP και σαν τέτοιο μπορεί να συγκριθεί με την τιμή του BACnet Development Kit [3,1] που έχει αναφερθεί ήδη και η τιμή του είναι 179 \$, περίπου 9 φορές μεγαλύτερη.

Κεφάλαιο 4

Υλοποίηση επιπέδου εφαρμογής

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα περιγραφτούν τα εργαλεία λογισμικού που χρησιμοποιήθηκαν για να αναπτυχθεί η BACnet εφαρμογή στο ενσωματωμένο σύστημα που αναλύθηκε στα προηγούμενα κεφάλαια. Τα εργαλεία αυτά περιλαμβάνουν την open source βιβλιοθήκη BACnet Stack που φιλοξενείται στο source forge τα εργαλεία που χειρίστηκε ο γράφων για να την μετατρέψει σε εκτελέσιμο και να την χρησιμοποιήσει στο ενσωματωμένο αλλά και οι παραμετροποιήσεις του κώδικα ώστε να υποστηριχθούν οι αισθητήρες που χρησιμοποιήθηκαν για τις μετρήσεις.

4.2 BACnet Stack

Η BACnet Stack είναι μια open source βιβλιοθήκη που φιλοξενείται στο site sourceforge [2.2]. Περιλαμβάνει μια υλοποίηση των Application Layer, Network Layer και Media Access Control Layer βασικών services του BACnet που είναι συμβατά με αρκετά λειτουργικά συστήματα. Επίσης περιλαμβάνει υλοποιήσεις του πρωτοκόλλου για αρκετά ενσωματωμένα συστήματα, όπως για Atmega 168, arm7 και άλλα.

Όπως είδαμε στο κεφάλαιο 2 το BACnet υποστηρίζει μια γκάμα από πρωτόκολλα επιπέδου 2 ή ζεύξης δεδομένων (Data Link) αλλά ορίζει και έναν τυποποιημένο τρόπο για επικοινωνία UDP (User Datagram Protocol), TCP/IP (Transmission Control Protocol/Internet Protocol) και HTTP τα οποία είναι πρωτόκολλα για επικοινωνία μέσω διαδικτύου. Τα demo services που περιλαμβάνονται στο πρωτόκολλο είναι αρχικοποιημένες για το BACnet/IP χρησιμοποιώντας το TCP/IP για την μετάδοση των μηνυμάτων αλλά υπάρχουν και οι υλοποιήσεις για τα υπόλοιπα πρωτόκολλα Data Link.

Η βιβλιοθήκη αυτή είναι ελεύθερη και προστατεύεται με την αδεία για ελεύθερη διάδοση πνευματικού υλικού General Purpose License GPL [4.1] που σημαίνει πρακτικά ότι αλλαγές στον πυρήνα του πρωτοκόλλου πρέπει να είναι διαθέσιμες σε όλους αλλά το τελικό αποτέλεσμα δεν είναι υποχρεωτικό να

δεσμεύεται με την άδεια GPL, για παράδειγμα ένα τελικό προϊόν που αναπτύχθηκε με την BACnet stack δεν χρειάζεται να κάνει διαθέσιμο το κώδικα του εάν δεν το επιθυμεί παρά μόνο τα κομμάτια του πρωτοκόλλου που τροποποιηθήκαν. Να σημειωθεί ότι τα κομμάτια που αναμένεται να τροποποιηθούν προστατεύονται με την MIT άδεια λογισμικού [4.2] που επιτρέπει την χρήση σε του κώδικα σε ιδιωτικές υλοποιήσεις.

Η βιβλιοθήκη είναι γραμμένη σε C για να εξασφαλιστεί η φορητότητα ενώ οι εφαρμογές τρέχουν σε Linux, Windows και στους περισσότερους μικροελεγκτές.

4.3 BACnet MS/TP Atmega 644

4.3.1 Stack Documentation

Από το αρχικό directory της βιβλιοθήκης βλέπουμε δύο σημαντικούς φακέλους, τον φάκελο Ports που περιέχει υλοποιήσεις για ενσωματωμένα συστήματα και τον φάκελο bin που περιέχει τα εκτελέσιμα αρχεία για τα services. Στον φάκελο Ports υπάρχει ο φάκελος bdk-atxx4-mstp. Ο φάκελος περιέχει υλοποίηση του stack που είναι συμβατή με τον Atmega 644P που έχουμε χρησιμοποιήσει στο ενσωματωμένο σύστημα μας και μάλιστα χρησιμοποιεί το πρωτόκολλο MSTP σαν Data Link πρωτόκολλο.

Όπως είδαμε στο 2.6.3 μια BACnet υλοποίηση για να μπορέσει να δείξει με έναν τυποποιημένο τρόπο της δυνατότητες της συνοδεύεται από ένα documentation που καλείται Protocol Implementation Conformance Statement (PICS) που περιέχει μεταξύ άλλων το Device Profile της συσκευής που καθορίζει της δυνατότητες της αλλά και έναν πίνακα με όλα τα BIBB που υποστηρίζει (βλέπε 2.6.2). Η συγκεκριμένη υλοποίηση συνοδεύεται από ένα τέτοιο έγγραφο όπου διαβάζουμε τα εξής:

BACnet Standardized Device Profile (Annex L)

	BACnet Operator Workstation (B-OWS)
	BACnet Building Controller (B-BC)
	BACnet Advanced Application Controller (B-AAC)
x	BACnet Application Specific Controller (B-ASC)
	BACnet Smart Sensor (B-SS)
	BACnet Smart Actuator (B-SA)

Πίνακας 4.1 Κατηγοριοποίηση με βάση το Device Profile της υλοποίησης

List all BACnet Interoperability Building Blocks supported (see Annex K in BACnet Addendum 135d):

DS-RP-B Read Property
DS-WP-B Write Property
DM-DDB-B Dynamic Device Binding
DM-DOB-B Dynamic Object Binding
DM-DCC-B Device Communication Control
DS-RPM-B ReadPropertyMultiple
DM-RD-B Reinitialize Device

Πίνακας 4.2 Υποστηριζόμενα BIBB

Στους δύο πιο πάνω πίνακες που είναι από το PICS της υλοποίησης βλέπουμε ότι το BACnet Stack έχει Device Profile Application Specific Controller (B-ASC) και συγκεκριμένα τα BIBB που έχουν υλοποιηθεί για να ανήκει στην κατηγορία αυτή. Να σημειωθεί εδώ ότι τα device profile δίνουν ένα ελάχιστο όριο υλοποίησης δηλαδή δηλώνουν τι είναι απολύτως απαραίτητο για να χαρακτηριστεί μια συσκευή σε ένα profile, γεγονός που σημαίνει ότι μια συσκευή μπορεί να έχει παραπάνω λειτουργικότητα για αυτό είναι απαραίτητος ο δεύτερος πίνακας με την αναφορά έναν προς έναν τον υλοποιημένων BIBB καθώς μόνο το Device Profile δεν είναι αρκετό για να δείξει τι υποστηρίζει η συσκευή. Για παράδειγμα στο 1.6.2 στον πίνακα 1.2 είδαμε ότι είναι για να χαρακτηριστεί μια συσκευή B-ASC στο Interoperable Area Data Sharing πρέπει να υποστηρίζονται τα εξής BIBB, DS-RP-B και DS-WP-B ενώ εδώ βλέπουμε ότι πέρα από αυτά τα δυο έχουμε υλοποιημένο και το DS-RPM-B (Read Property Multiple).

4.3.2 Ανάλυση του κώδικα του Stack

Στην παράγραφο αυτή θα αναλυθούν τα αρχεία του stack και το τι υλοποιεί το καθένα. Όπως έχει προαναφερθεί ο κώδικας είναι γραμμένος σε C για να γίνει compile όμως απαιτείται ένας AVR-C compiler. AVR καλείται η αρχιτεκτονική που χρησιμοποιεί η οικογένεια των μικροεπεξεργαστών της Atmega και πρόκειται για μια 8-bit RISC (Reduced Instruction Set Computing) αρχιτεκτονική με έναν επεξεργαστή ([4.3] για AVR και [4.4] γενικά για RISC αρχιτεκτονικές). Η πλατφόρμα που επιλέχθηκε για την ανάπτυξη του κώδικα, το debugging και το compile είναι η πλατφόρμα της ίδιας της κατασκευάστριας των Atmega της Atmel και πρόκειται για το Atmel Studio όπου και χρησιμοποιήθηκε η έκδοση 6.1 ([4.5] το σελίδα download της τελευταίας έκδοσης του λογισμικού).

Στο path ports/bdk-atxx4-mstp/ του stack υπάρχουν τα απαραίτητα αρχεία για την υλοποίηση αυτή. Φυσικά υπάρχουν και οι βιβλιοθήκες στο path include/

που περιέχει τα header files που γίνονται include και src/ που περιέχει τον πηγαίο κώδικα για τα header files. Στο directory του project υπάρχει το αρχείο bacnet.atshn το οποίο είναι τύπου Atmel Studio Solution File και μπορεί να χρησιμοποιηθεί για να τρέξει το project μέσω Atmel Studio. Από εκεί και πέρα ανεξαρτήτως λογισμικού που θα χρησιμοποιηθεί για τον εκτέλεση του project έχουμε τα εξής αρχεία:

Παρατήρηση 1: Κάθε .c αρχείο που περιέχει τον πηγαίο κώδικα έχει και ένα αντίστοιχο .h header file που δίνει τον ορισμό των συναρτήσεων του αντίστοιχου αρχείου. Στην συνέχεια θα αναφερθούν η λειτουργία του κάθε αρχείου πηγαίου κώδικα που χρησιμοποιείται, αλφαβητικά.

Παρατήρηση 2: Στην συνέχεια θα αναφερθούν λεπτομερώς όπου κρίνεται απαραίτητο οι registers ελέγχου του 644 για κάθε συγκεκριμένη λειτουργία και οι τιμές απαιτείται να πάρουν. Όλες οι τιμές αυτές και οι λειτουργία κάθε καταχωρητή περιγράφονται αναλυτικότερα στο datasheet του επεξεργαστή [2.3]. Επιπλέον σε αρκετά σημεία χρησιμοποιείται η δυνατότητα για interrupt που έχει ο επεξεργαστής. Το πώς λειτουργεί το Interrupt γενικά θα αναφερθεί στην συνέχεια.

Παρατήρηση 3: Στον πηγαίο κώδικα συχνά βλέπουμε bitwise πράξεις που είναι ασυνήθιστες στην C αλλά χρησιμοποιούνται κατά κόρον στην AVR-C για να δίνονται τιμές σε συγκεκριμένα bit των register. Για μια αναλυτική περιγραφή των εντολών αυτών βλέπε [4.6]

adc.c: Σε αυτό το αρχείο γίνεται ο χειρισμός του Analog-to-Converter (ADC). Η μεταβλητή ADC_CHANNELS_MAX ορίζει τον μέγιστο αριθμό εισόδων-αναλογικών σημάτων που μπορούν να μετατραπούν παράλληλα ο οποίος είναι 8, όσες δηλαδή και οι ADC εισοδοί του 644. Στην συνάρτηση adc_init αρχικοποιούνται οι registers (καταχωρητές) που ελέγχουν το ADC. Πρόκειται για τους ADMUX, ADCSRA και ADCSRB. Ο ADMUX (ADC Multiplexer Selection Register) καθορίζει το voltage reference (την σήμα αναφοράς της τάσης) και παίρνει τιμή 0x01 που δηλώνει ότι έχουμε ξεχωριστή τροφοδοσία στην είσοδο AVCC και ότι το σήμα αναφοράς είναι το AREF το οποίο έχει εξωτερικό πυκνωτή. Οι ADCSRA,B (ADC Control and Status Register A,B) ελέγχουν την γενική λειτουργία του ADC. Ο Status Register A αρχικοποιείται έτσι ώστε να λειτουργήσει το ADC και να λειτουργεί με interrupt ενώ στον CSRB ελέγχει το mode του ADC και χρησιμοποιείται το free running mode που δηλώνει διαρκή λειτουργία του ADC χωρίς διακοπές (με το που τελειώνει μια μετατροπή ξεκινάει η επόμενη).

Οι συναρτήσεις ADC_result_8bit/10bit γυρνάνε το αποτέλεσμα με την αντίστοιχη ακρίβεια (8/10 bit) το αποτέλεσμα του ADC. Το αποτέλεσμα του ADC γράφεται στην πίνακα Sample_result[index] όπου index είναι ο κανάλι που γίνεται η μετατροπή κάθε φορά μέσω τις συνάρτησης ISR.

Σε αυτό το σημείο καλό θα ήταν να γίνει μια αναφορά στο το είναι και στο

πως ακριβώς λειτουργούν τα Interrupt. Τα interrupt (διακοπές) είναι σήματα που στέλνονται στον επεξεργαστή για να σηματοδοτήσουν ότι ένα γεγονός μόλις τελείωσε και απαιτεί την άμεση προσοχή του επεξεργαστή. Ο επεξεργαστής όταν δει ότι συνέβη ένα interrupt, διακόπτει ότι εκτελεί εκείνη την στιγμή σώζει την κατάσταση όλων των register στην μνήμη του και εκτελεί τον αντίστοιχο interrupt handler. Ένας interrupt handler είναι μια συνάρτηση, ένα κομμάτι κώδικα δηλαδή, που δηλώνει τι κάνει ο επεξεργαστής στην περίπτωση που γίνει ένα συγκεκριμένο interrupt. Το πιο interrupt συμβαίνει κάθε φορά το δηλώνει το interrupt vector ένα σήμα εισόδου που έχει διαφορετική τιμή αναλόγως πιο τμήμα του επεξεργαστή καλεί το interrupt. Για παράδειγμα σύμφωνα με τον πίνακα 9.1 του datasheet [2.3] βλέπουμε ότι για το ADC καλείτε ο Interrupt Handler με Vector Number 25 ([4.7] για το πώς λειτουργούν τα interrupt και οι interrupt handlers).

Χρησιμοποιώντας την συνάρτηση ISR (Interrupt Service Routine) από την βιβλιοθήκη avr/interrupt.h μπορούμε να γράψουμε το δικό μας interrupt handler ο οποίος καλείται όταν γεννιέται ένα interrupt σήμα από το ADC. Αυτό δηλώνεται δίνοντας το κατάλληλο όρισμα στην ISR που είναι το ADC_vect που αντιστοιχεί μέσω define στο 25 που αναφέρθηκε πριν.

ai.c: Στο αρχείο αυτό υλοποιείται το BACnet Object, Analog Input με τα properties που καθορίζει το πρωτόκολλο. Σημαντική μεταβλητή είναι η MAX_ANALOG_INPUTS που καθορίζει τον μέγιστο αριθμό αντικείμενων Analog Input που έχουμε στο σύστημα καθώς και τις δύο σημαντικές συναρτήσεις Analog_Input_Present_Value, Analog_Input_Present_Value_Set που η μεν πρώτη επιστρέφει το πιο σημαντικό property του

αντικείμενου το Present Value που περιέχει την τρέχουσα τιμή της αναλογικής εισόδου που είναι αντιστοιχισμένη με το αντικείμενο αυτό ενώ η δε δεύτερη ενημερώνει το property αυτό. Επιπλέον η συνάρτηση Analog_Input_Read_Property επιστρέφει το APDU (Application Data Unit) που απαντάει σε Read Property Request.

av.c: Στο αρχείο αυτό υλοποιείται το BACnet Object, Analog Value όπως το καθορίζει το πρωτόκολλο. Όπως και πριν το MAX_ANALOG_VALUES καθορίζει τον μέγιστο αριθμό analog value αντικείμενων που έχουμε, ενώ η συναρτήσεις Analog_Value_Present_Value/Set επιστρέφουν είτε την τιμή του αντικείμενου ή την ενημερώνουν. Τέλος έχουμε την συνάρτηση Analog_Value_Read_Property που επιστρέφει το APDU που απαντάει σε Read Property Request.

bacnet.c: Στο αρχείο αυτό γίνονται η πιο σημαντικές λειτουργίες. Στην συνάρτηση bacnet_init βλέπουμε της αρχικοποιήσεις που χρειάζονται για να τρέξει η υλοποίηση. Στην αρχή διαβάζεται το MAC address που διαβάζεται από τις εισόδους των Dip Switches και ελέγχεται για την εγκυρότητα της (μέγιστη τιμή για το

MAC address είναι το 127 γιατί όπως αναλύθηκε στο MS/TP μπορούν να υπάρχουν 128 κόμβοι ταυτόχρονα). Έπειτα αρχικοποιούνται όλα τα αντικείμενα μέσα από την Device_Init και θέτονται οι handler (οι συναρτήσεις που εκτελούνται όταν έχουμε μια το αντίστοιχο APDU) που χειρίζονται το confirmed services και τέλος στέλνεται ένα I-Am service που στέλνεται πάντα στην αρχικοποίηση για να δηλώσει την λειτουργία μιας νέας συσκευής στον δακτύλιο.

Επιπλέον υπάρχει η συνάρτηση bacnet_task η οποία ενημερώνει όλα τα αντικείμενα που έχουν φτιαχτεί με την αρχικοποίηση με νέες τιμές (οι οποίες μπορεί να έρχονται από το ADC για παράδειγμα ή τις binary εισόδους κ.ο.κ.) και στο τέλος ελέγχει εάν έχουμε πακέτο δεδομένων από το datalink επίπεδο για να καλέσει τον αντίστοιχο handler. Στην ουσία αυτή είναι η συνάρτηση που καλείτε διαρκώς και εκτελείται και καθορίζει τι κάνει στο ν Application Layer η συσκευή.

bi.c: Εδώ υλοποιείται το BACnet Object Binary Input. Όπως και για τα Analog Value και Analog Input έχουμε τις αντίστοιχες συναρτήσεις Binary_Input_Present_Value/Present_Value_Set και Binary_Input_Read_Property για το Read Property Service.

bo.c: Εδώ υλοποιείται το BACnet Object Binary Object. Ομοίως με τα υπόλοιπα BACnet αντικείμενα που έχουν αναφερθεί μέχρι στιγμής.

device.c: Εδώ υλοποιείται το BACnet Object Device το οποίο όπως έχουμε δει στο 1.3.1. είναι απαραίτητο να υπάρχει ένα σε κάθε BACnet συσκευή. Ομοίως με τα υπόλοιπα BACnet αντικείμενα που έχουν αναφερθεί μέχρι στιγμής. Να σημειωθεί εδώ ότι στην Device Init αρχικοποιούνται όλα τα αντικείμενα που υποστηρίζονται στην υλοποίηση.

dlimstp.c: Στο dlimstp υλοποιούνται οι λειτουργίες του datalink πρωτοκόλλου master slave token passing. Στην ουσία υλοποιείται η μηχανή πεπερασμένων καταστάσεων (Finite State Machine –FSM) του MSTP (βλέπε [4.8] για την υλοποίηση της FSM). Να σημειωθεί ότι στην αρχή ορίζονται μεταβλητές που παίζουν πολύ σημαντικό ρόλο στην λειτουργία του δακτυλίου και θα μας απασχολήσουν αργότερα.

eeeprom.c: Στο αρχείο αυτό υλοποιούνται δύο συναρτήσεις μια για εγγραφή και μια για να ανάγνωση από την μνήμη του επεξεργαστή

fuses.c: Τα fuses γενικά είναι μια οικογένεια control register που ελέγχουν αρκετές πλευρές του επεξεργαστή. Ο Atmega 644P έχει 4 byte fuses (32 bit). Στο αρχείο αυτό παίρνουν τιμές τα bit αυτά. (Για περισσότερα για τα fuses βλέπε το datasheet [3.3] στην παράγραφο 24.2 σελ 294).

hardware.h: Το αρχείο δεν υλοποιείται από κάποιο source file. Περιέχει διαφόρων τύπων defines όπως τα F_CPU (ταχύτητα επεξεργαστή, δηλαδή η ταχύτητα κρυστάλλου), η διεύθυνση της μνήμης για το πρωτόκολλο i2c κ.α.

iar2gcc.h: Το header αρχείο χρησιμοποιείται μόνο για να μπορεί να γίνει compile με τον iar workbench της IAR systems που δεν χρησιμοποιήθηκε καθόλου σε αυτήν την εργασία.

init.c: Στο αρχείο αυτό γίνεται κάποιες τελικές αρχικοποιήσεις. Ο clock prescaler αρχικοποιείται στο μηδέν οπότε και δεν έχουμε Clock Prescaling. Ο clock prescaler είναι ένας control register που χρησιμοποιείται για να ρυθμιστεί η συχνότητα που έρχεται από τον εξωτερικό κρύσταλλο στα επιθυμητά επίπεδα. Στην πραγματικότητα ο clock prescaler διαιρεί την συχνότητα του κρυστάλλου με έναν προβλέψιμο τρόπο. Από το datasheet [3.3] βλέπουμε (πίνακας 6.16 σελ. 41) ότι για παράδειγμα εάν ο CLKPR (Clock Prescaler) register πάρει την τιμή b0010 τότε έχουμε prescale factor 4 που σημαίνει ότι μόνο μια θετική ακμή ανά τέσσερις λαμβάνεται υπόψη από τον επεξεργαστή άρα η συχνότητα υποπολλαπλασιάζεται κατά τέσσερις φορές.

input.c: Στο αρχείο αυτό διαβάζονται οι είσοδοι του PORTA που έχουν Dip Switches και χρησιμοποιούνται για την δήλωση του MAC Address. Αφού αρχικοποιούνται τα αντίστοιχα PIN σαν είσοδοι. Με την συνάρτηση input_task ενημερώνεται η MAC address ενώ με την input_address επιστρέφεται η τιμή αυτή στα σημεία που χρειάζεται (Η συνάρτηση αυτή καλείται από την bacnet_task για ενημερώνει την τιμή του MAC Address στην επικεφαλίδα των APDU που στέλνει).

led.c: Περιλαμβάνει αρχικοποιήσεις και δηλώσεις σχετικά κάποιες εξόδους του επεξεργαστή που μπορούν να χρησιμοποιηθούν σαν output για LED.

main.c: Η κεντρική συνάρτηση που καλεί όλες τις υπόλοιπες. Στην αρχή γίνονται όλες οι αρχικοποιήσεις και ενεργοποιούνται τα Interrupt του επεξεργαστή. Κατόπιν υπάρχει ένας ατέρμον βρόγχος που καλεί διαρκώς όλα τα task που έχουν αναφερθεί μέχρι στιγμής.

nvddata.h: Ένα header αρχείο που έχει κάποια define για την εσωτερική μνήμη.

rs485.c/serial.c: Δύο αρχεία που κάνουν παρόμοια δουλειά και για αυτό αναλύονται μαζί. Όπως έχουμε δει ήδη στο 2.2.1 ο επεξεργαστής διαθέτει δύο εξόδους για σειριακή επικοινωνία. Και οι δύο εξοδοι είναι σε λογική TTL αλλά κατευθύνονται σε διαφορετικά ολοκληρωμένα, η μια στο MAX 481 και η άλλη στο MAX202 για επικοινωνία με RS-485 και RS-232 αντιστοίχως.

Στην συγκεκριμένη υλοποίηση μόνο το RS-485 χρησιμοποιείται για την αποστολή BACnet δεδομένων καθώς έχουμε υλοποίηση για BACnet MS/TP που χρησιμοποιεί δεσμευτικά το πρωτόκολλο αυτό στο φυσικό επίπεδο. Συνεπώς οι συναρτήσεις που υπάρχουν στο αρχείο serial.c και αφορούν την δεύτερη σειριακή έξοδο που κατευθύνεται για να μετατραπεί σε σήμα RS-232 είναι μόνο για στέλνονται απευθείας μηνύματα σε κάποιο text monitor και όχι για δεδομένα

BACnet. Επειδή τα δύο αρχεία είναι πανομοιότυπα στην δομή θα περιγραφτεί στην συνέχεια μόνο το ένα (rs485.c).

Η σειριακή επικοινωνία βασίζεται και αυτή σε interrupts και για να ενεργοποιηθεί τα bit του Universal Control Status Register B (UCSRnB όπου το n παίρνει τιμές 0 ή 1 αναλόγως ποια από τις δυο USART χρησιμοποιούμε) ενεργοποιούνται τα bit RXENn, TXENn, RXCIEn στην συνάρτηση rs_enable που ενεργοποιούν την λήψη την μετάδοση και τα Interrupt αντίστοιχα. Η πιο σημαντική συνάρτηση εδώ είναι η ISR που είναι ο Interrupt Handler για το περίπτωση λήψης δεδομένων και βλέπουμε ότι τα δεδομένα που έρχονται από το bus μπαίνουν σε μια δομή ουράς FIFO (First In First Out) για να παραδίδονται με την σειρά που ήρθαν. Τα δεδομένα αυτά αποθηκεύονται στον κατάλληλο πίνακα που δίνεται σαν όρισμα στην rs485_byte_available. Η συνάρτηση αυτή που ελέγχει εάν η στοίβα είναι μη κενή και εάν δεν είναι αποθηκεύει το δεδομένα στον πίνακα που τις έχει δοθεί σαν όρισμα.

Για την αποστολή δεδομένων χρησιμοποιείται η συνάρτηση rs485_bytes_send όπου αφού περιμένουμε να αδειάσει ο Transmit Buffer από τυχόν προηγούμενη μετάδοση και τα frame προς μετάδοση αποθηκεύεται στο UDRn (με το n=0 για το RS485 και n=1 για το RS232). Πριν την αποστολή frame δεδομένων καλείται η συνάρτηση rs485_turnaround_delay που εξασφαλίζει ότι έχουμε idle κατάσταση στο bus αφού στέλνονται τέσσερα dummy bytes δηλαδή byte (x"FF") μη πληροφορίας που χρησιμοποιούνται σαν padding ανάμεσα σε frame χρονικής διάρκειας 40 bit times (για το padding βλέπε 2.5.1 σελίδα 26 του παρόντος).

Το δεύτερο αρχείο που αφορά σειριακή επικοινωνία (serial.c) έχει την ίδια δομή χρησιμοποιώντας τις ίδιες συναρτήσεις, μόνο που στα αντίστοιχα σημεία έχουμε τους control register που ελέγχουν την δεύτερη UART με n=1 (π.χ. UCSR1A/B κ.τ.λ.). Συνάρτηση για silence στο bus (με αποστολή dummy byte) δεν υπάρχει καθώς δεν αποστέλλονται BACnet frame και δεν υπάρχει η ανάγκη για padding ανάμεσα σε frame. Τέλος λόγω των διαφορών πρωτοκόλλων όπως αναλύθηκαν στο 1.5 η δύο UART λειτουργούν με διαφορετικά baud rate. Το RS-485 δουλεύει στο 38,4 kbps ενώ το RS232 στα 9.6 kbps.

seeprom.c: Όπως έχει αναφερθεί και πιο πριν εκτός της εσωτερικής μνήμης του επεξεργαστή έχουμε και μια σειριακή μνήμη (serial eeprom) η οποία χρησιμοποιεί το i2c (ή TWI όπως είναι η εναλλακτική του ονομασία) πρωτόκολλο και έχει διεύθυνση μήκους 2 byte 0x"A0" (περισσότερα για το i2c στην συνέχεια). Στο αρχείο αυτό υλοποιείται η επικοινωνία με το περιφερειακό αυτό.

test.c: Στο αρχείο αυτό χρησιμοποιείται η δεύτερη σειριακή έξοδος που στέλνει μηνύματα μέσω RS-232 για debugging κυρίως. Εάν συνδεθεί η έξοδος αυτή σε ένα σειριακό monitor βλέπουμε πληροφορίες όπως το MAC address τα Baud Rate

των συνδέων και τέτοιου είδους πληροφορίες. Να σημειωθεί εδώ ότι ένα έχει οριστεί μέσω define σε κάποιο σημείο του κώδικα η μεταβλητή MSTP_MONITOR τότε τα BACnet frame στέλνονται και στις δύο UART οπότε τα δεδομένα μπορούν να καταγραφούν από το serial monitor και να αναλυθούν μέσω ενός αναλυτή πρωτοκόλλου. Προφανώς η επιλογή αυτή μπορεί να χρησιμοποιηθεί για debugging στα frame του πρωτοκόλλου.

timer.c: Στο αρχείο αυτό υλοποιείται ένας elapsed counter χρησιμοποιώντας το struct itimer που υπάρχει στην αντίστοιχη βιβλιοθήκη timer.h. Elapsed counter είναι ο μετρητής που δείχνει πόσο χρονικό διάστημα έχει παρέλθει από την ώρα εκκίνησης του. Οι πιο σημαντικές συναρτήσεις εδώ είναι οι timer_interval _start_seconds/minutes οι οποίες ξεκινάνε έναν counter μεγέθους που δίνεται σαν όρισμα, η timer_interval_expired που ελέγχει εάν το διάστημα αυτό που έχει δοθεί στον counter τελείωσε και η timer_interval_reset που μηδενίζει τον counter για να ξεκινήσει πάλι το μέτρημα.

timer2.c: Εναλλακτικός counter που υλοποιείται με τον Timer/Counter Control Register (TCCR) που έχει ο επεξεργαστής.

watchdog.c: Ο watchdog είναι ένας ειδικός register που χρησιμοποιείται για να μετράει ένα χρονικό διάστημα και στο τέλος να δημιουργεί ένα interrupt σηματοδοτώντας ότι τελείωσε το μέτρημα. Ο watchdog register είναι ο WDTCR και τα τέσσερα LSB του ρυθμίζουν το χρονικό διάστημα του μετρήματος (για παράδειγμα εάν είναι b'0101' τότε έχουμε μέτρημα ανά 0.5 sec)

4.4 Ανάλυση απαιτήσεων επιπέδου εφαρμογής – Εξοπλισμός

4.4.1 Αισθητήρες που χρησιμοποιήθηκαν

Η λειτουργικότητα το BACnet stack αναλύθηκε λεπτομερώς στην προηγούμενη παράγραφο. Στην παράγραφο αυτή θα αναφερθεί τι αλλαγές έπρεπε να γίνουν για στο επίπεδο εφαρμογής αλλά και γενικότερα για να έχουμε ένα σύστημα όπως περιγράφηκε στην παράγραφο 1.1. Όπως αναφέρθηκε στο κεφάλαιο 1 στόχος ήταν να δημιουργηθεί ένα BACnet σύστημα που θα δρα ως server σε service request που δημιουργούνται από έναν client ,τον ρόλο του οποίου μπορεί να παίξει ένας απλός Η/Υ που τρέχει τα service application που συνοδεύουν το stack. Τα service αυτά αφορούν μετρήσεις που έρχονται από διαφόρων τύπου αισθητήρες που συνδέονται στο ενσωματωμένο σύστημα.

Στην συνέχεια αναλύονται οι αισθητήρες που χρησιμοποιήθηκαν καθώς η κατανόηση των χαρακτηριστικών τους είναι αυτή που μας επιτρέπει να έχουμε μια εικόνα τι πρέπει να γίνει το επίπεδο εφαρμογής.

Αισθητήρας θερμοκρασίας TM75 της Texas instruments

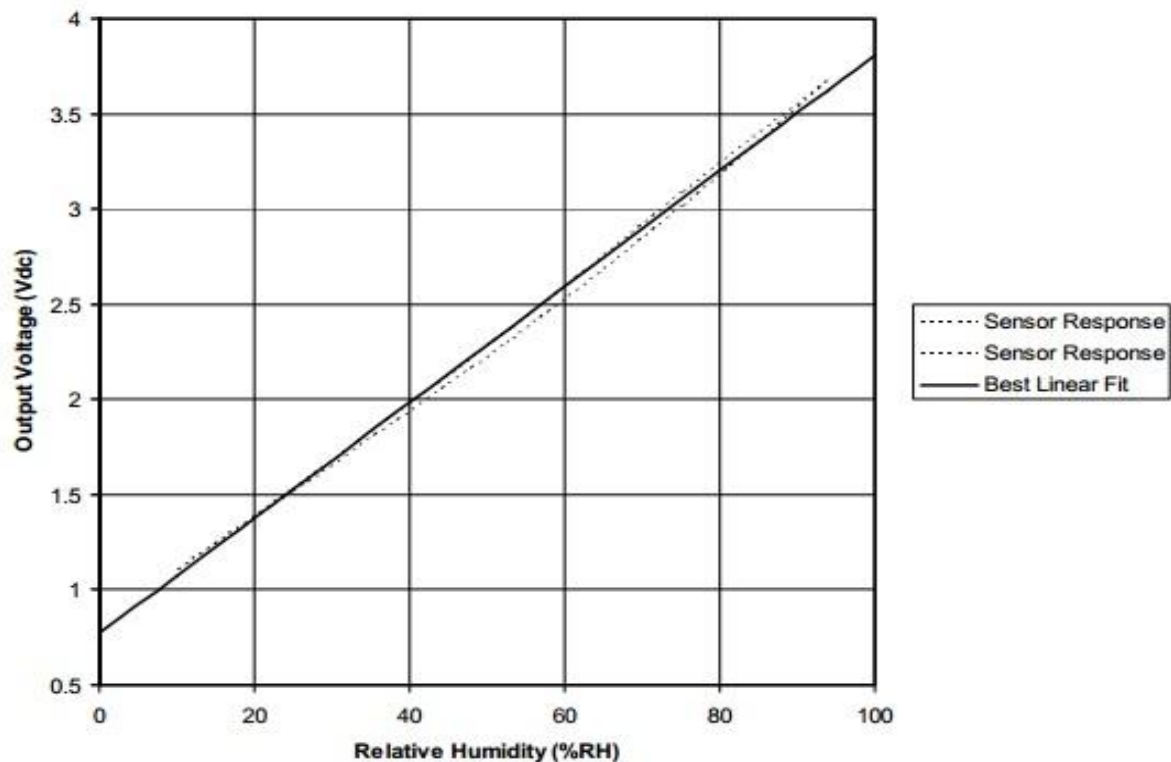
Το datasheet του αισθητήρα είναι στο [4.9]. Ο TM75 είναι ένας σειριακός αισθητήρας που χρησιμοποιεί το πρωτόκολλο I2C ή TWI (Two wire Interface). Το package που χρησιμοποιήθηκε είναι SO-8 με SMT footprint. Η μέγιστη ευαισθησία του αισθητήρα είναι 0.0625°C ή ισοδύναμα 12 bit ακρίβεια στο τελικό αποτέλεσμα. Η ευαισθησία αλλάζει γράφοντας μέσω του I2C bus στο configuration register που έχει εσωτερική διεύθυνση $0x'01''$.

Το pinout του αισθητήρα είναι το εξής: Δύο έξοδοι SDA,SCK που αφορούν το I2C (το οποίο θα αναλυθεί στην επομένη παράγραφο), τρεις έξοδοι A0,A1 και A2 που είναι το address για το I2C bus καθώς, δυο pin για την τροφοδοσία και μία έξοδος για το σήμα Alert που στην ουσία είναι ένα interrupt σήμα για την περίπτωση που θερμοκρασία ξεπεράσει κάποια συγκεκριμένα επίπεδα. Τα λειτουργικά επίπεδα θερμοκρασία είναι από -40°C έως $+125^{\circ}\text{C}$.

Αισθητήρας υγρασίας HIH-4030 της Honeywell

Το datasheet του αισθητήρα βρίσκεται στο [4.10]. Πρόκειται για έναν αναλογικό αισθητήρα δηλαδή έχει σαν έξοδο τάση. Έχει τρία pin, τροφοδοσία με GND και Vcc καθώς και το Output που είναι ένα σήμα ποσοστό του σήματος που δέχεται σαν είσοδο ανάλογα με την υγρασία. Στην εικόνα 4.1 (πηγή [4.1] σελ. 4) φαίνεται διαγραμματικά η σχέση του σήματος εισόδου Vcc με το σήμα εξόδου. Προφανώς επειδή οι αισθητήρες αυτοί παρασκευάζονται μαζικά δεν επιτρέπεται στον κατασκευαστή για λόγους κόστους να δώσει μια πιο εξειδικευμένη σχέση τάση εξόδου-θερμοκρασίας οπότε αναγκαστικά δίνει μια μέση απόκριση όπως έχει προκύψει από στατιστικές στους αισθητήρες μαζί με μια ανοχή. Παρατηρούμε στην εικόνα 4.1 ότι τα όρια αυτά της

ανοχής είναι με τις διακεκομμένες γραμμές. Επίσης το διάγραμμα αυτό παίρνει υπόψη του και την θερμοκρασία καθώς επηρεάζει την απόκριση του αισθητήρα.



Εικόνα 4.1 Σχέση Τάσης-Υγρασίας στον HIH-4030 σε θερμοκρασία 25 °C

Α

πό το διάγραμμα αυτό προσεγγίζεται η εξίσωση τάσης-υγρασίας σαν :

$$V_{out}(h) = 0.033 * h + 0.7$$

όπου h η υγρασία και $V_{out}(h)$ η τάση εξόδου στην υγρασία αυτή. Οπότε λύνοντας ως προς παίρνουμε:

$$Hum(V) = \frac{V-0.7}{0.033}$$

όπου V η τάση που παίρνουμε σαν έξοδο και $Hum(V)$ η υγρασία σαν ποσοστό επί τοις εκατό.

Αισθητήρας συγκέντρωσης Co2 K-30 της Co2 Meter

Το datasheet του αισθητήρα βρίσκεται στο [4.11]. Ο K-30 είναι ένας low-cost αισθητήρας συγκέντρωσης διοξειδίου του άνθρακα. Είναι ψηφιακός και η έξοδος του μπορεί να είναι είτε σειριακή (με RX.TX) είτε σε I2C. Η όρια των μετρήσεων είναι από 0 μέχρι 5000 ppm (parts per million, σωματίδια ανά εκατομμύρια σωματίδια ατμοσφαιρικού αέρα). Για την επικοινωνία με τον αισθητήρα αυτό ακολουθήθηκε το υπόδειγμα επικοινωνίας με Arduino που δίνεται σαν παράδειγμα από την ίδια την εταιρία [4.12].

Στο σημείο αυτό μιας και δυο αισθητήρες, που χρησιμοποιήθηκαν, χρησιμοποιούν το πρωτόκολλο I2C καλό θα ήταν να αναλυθεί το πρωτόκολλο αυτό

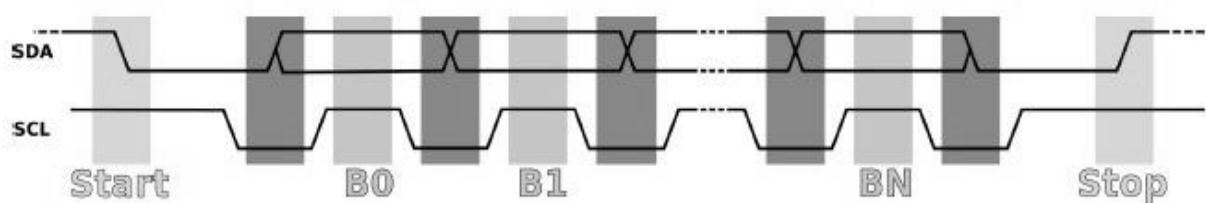
4.4.1 Το πρωτόκολλο επικοινωνίας I2C

Όπως έχουμε δει ήδη δύο αισθητήρες αλλά και η σειριακή μνήμη χρησιμοποιούν το πρωτόκολλο αυτό για να ανταλλάξου δεδομένα με τον επεξεργαστή συνεπώς είναι απαραίτητη η κατανόηση του πρωτοκόλλου προτού φτιαχτεί η αντίστοιχη βιβλιοθήκη που θα το υποστηρίζει. Στο κεφάλαιο αυτό θα αναλυθεί το πρωτόκολλο αυτό καθώς είναι απαραίτητο να υλοποιηθεί σε επίπεδο εφαρμογής για να είναι δυνατή η ανάγνωση των τιμών των αισθητήρων .

Το I2C (ή TWI για την ονομασία βλέπε παράγραφο 3.2.1 του παρόντος σελ.40) είναι ένα δημοφιλές σειριακό πρωτόκολλο επικοινωνίας. Είναι σύγχρονο πρωτόκολλο που σημαίνει ότι υπάρχει ρολόι στο δίαυλο (clock) και μια συσκευή που το ελέγχει clock master. Το ίδιο φυσικό μέσο μπορεί να υποστηρίξει παραπάνω από έναν master και παραπάνω από μια συσκευές που απευθύνονται οι master και καλούνται slave για αυτό και το πρωτόκολλο χαρακτηρίζεται σαν multi-master, multi-slave. :

Η επικοινωνία γίνεται μέσα από δυο σήματα το SCK, System Clock που συγχρονίζει τις συσκευές και SDA, System Data που μεταφέρει τα δεδομένα. Όταν ένας master θέλει να μεταδώσει δεδομένα ελέγχει το bus εάν υπάρχει μετάδοση δεδομένων και εάν καταλάβει ότι δεν μεταδίδει άλλος master στέλνει start.

Τα δεδομένα αλλάζουν τιμή (από high-1 σε low-0) μόνο κατά την φάση που το ρολόι έχει τιμή low-0 με μοναδικές εξαιρέσεις τα σήματα start, stop που καθορίζουν την αρχή μετάδοσης και το τέλος αντίστοιχα. Το start σηματοδοτείται από αρνητική ακμή στο SDA και high στο SCK ενώ αντίστοιχα το stop από ανοδική ακμή στο SCK και θετικό ρολόι ξανά. Οι χρονισμοί των αλλαγών αυτών όπως περιγράφηκαν φαίνονται στην εικόνα 4.2.



Εικόνα 4.2 Τα σήματα SDA,SCK κατά την διάρκεια μιας επικοινωνίας I2C

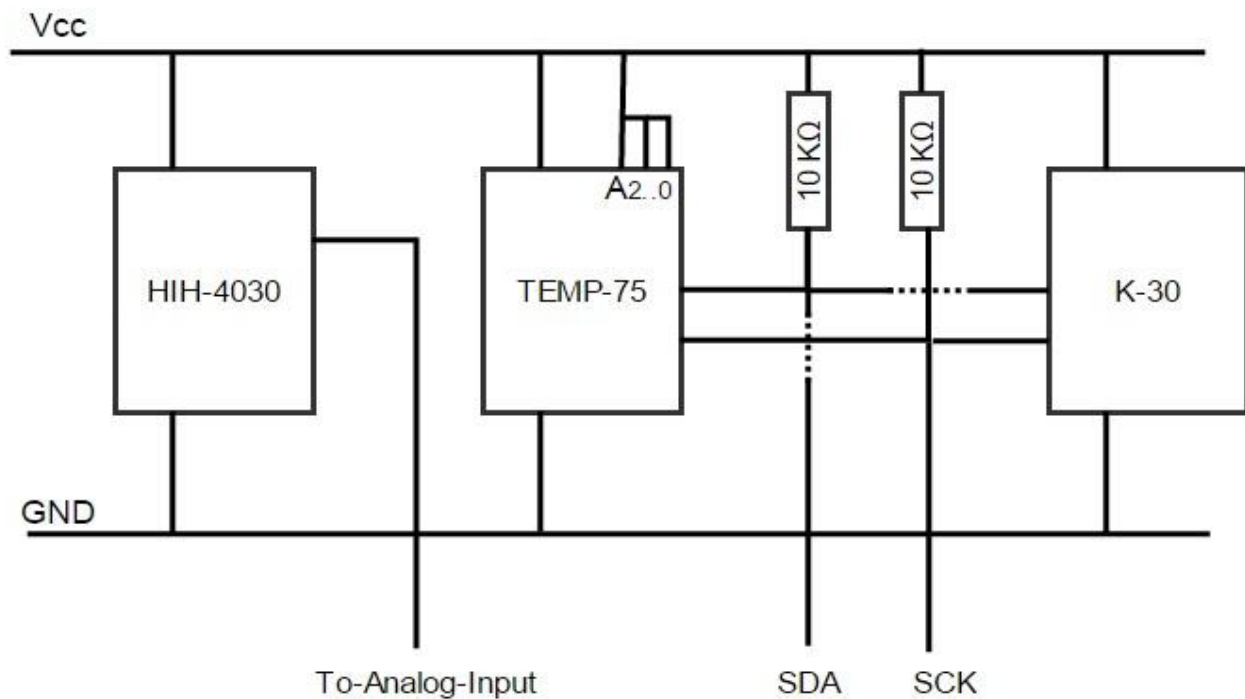
Προφανώς η παρουσία πολλαπλών συσκευών στον δίαυλο δημιουργεί την ανάγκη να υπάρχει ένας τρόπος διάκρισης τους. Έτσι κάθε συσκευή έχει μια μοναδική 7-bit διεύθυνση στον δίαυλο. Έτσι ένας master όταν θέλει να επικοινωνήσει με μια συσκευή στέλνει την 7-bit διεύθυνση της μαζί με άλλο ένα bit που δηλώνει το εάν θέλει να γράψει ή να διαβάσει δεδομένα από την συσκευή (R/W, Read/Write bit). Έτσι η slave συσκευή ξέρει εάν πρέπει να περιμένει δεδομένα ή να στείλει.

Τέλος σημαντικό κομμάτι του πρωτοκόλλου είναι ότι έχουμε επιβεβαιωμένη επικοινωνία δηλαδή οι slave στέλνουν είτε θετικές είτε αρνητικές επιβεβαιώσεις (ACK, NACK). Έτσι κάθε μετά τα 8-bit που στέλνει ο master ακολουθεί ένα 8 που στέλνει ο slave και είτε είναι high-1 που σημαίνει ότι τα δεδομένα λήφθηκαν σωστά (ACK) και αναμένονται και άλλα είτε low-0 που σημαίνει ότι τα δεδομένα είτε αλλοιώθηκαν είτε δεν είναι σωστά είτε η επικοινωνία έφτασε στο τέλος (NACK).

4.5 Επίπεδο Εφαρμογής /Υλοποίηση

Στις προηγούμενη παράγραφο αναλύθηκε ο εξοπλισμός που χρησιμοποιήθηκε για την εργασία αυτή όσον αφορά τους αισθητήρες, που καθορίζουν τις απαιτήσεις μας σε BACnet Object, αλλά και το I2C που χρησιμοποιούν οι αισθητήρες για την επικοινωνία. Στην παράγραφο αυτήν θα περιγραφούν οι αλλαγές που έγιναν στο επίπεδο εφαρμογής για να καλυφθούν οι απαιτήσεις αυτές.

Το πρώτο βήμα για την επικοινωνία αισθητήρων με το ενσωματωμένο σύστημα. Χρησιμοποιήθηκε ένα απλό breadboard μαζί με συνηθισμένους head-head connectors. Οι δύο αισθητήρες που χρησιμοποιούν το I2C συνδέθηκαν στα pin SCK, SDA (pin PC0 και PC1 αντίστοιχα) ενώ ο αναλογικός αισθητήρας θερμοκρασίας συνδέθηκε στο pin PA7 (όλο τα pin του PORTA υποστηρίζουν ADC). Η κυκλωματική διάταξη φαίνεται στην εικόνα 4.3.



Εικόνα 4.3 Διάταξη των αισθητήρων για επικοινωνία με το σύστημα

Να σημειωθεί εδώ ότι όπως είναι πρόδηλο χρησιμοποιήθηκαν αντιστάσεις 10 KΩ σαν pull up resistors για τα σήματα SDA,SCK. Τα pull resistors και η χρησιμότητα τους έχει αναφερθεί προγενέστερα (βλέπε και [3.10]) αλλά εν συντομία χρησιμοποιούνται για να υπάρχει ένα καθορισμένο επίπεδο τάσης κάθε στιγμή στο σήμα και όχι μετασταθείς καταστάσεις, δηλαδή το σήμα να είναι πάντα στις ζώνες τάσης που καθορίζονται σαν λογικό 1 ή 0. Η ύπαρξη τέτοιων αντιστάσεων είναι δεσμευτική για κάθε I2C συσκευή.

Επιπλέον φαίνεται η σύνδεση των bit διεύθυνσης με την τροφοδοσία του TEM-75. Τα τρία αυτά bit καθορίζουν τα LSB (Least Significant Bit) της 7-bit διεύθυνσης του αισθητήρα για το I2C οπότε με την σύνδεση τους στην τροφοδοσία η διεύθυνση παίρνει την τιμή 1001[1][1][1].

Εκτός της συνδεσμολογίας έπρεπε να γίνουν συγκεκριμένες αλλαγές στο επίπεδο του BACnet stack για να υποστηρίξει της συσκευές αυτές. Στην συνέχεια θα αναφερθούν οι αλλαγές αυτές με την ανάλυση των αρχείων που τροποποιήθηκαν όπως εξηγήθηκε προηγουμένως και για την λειτουργικότητα του BACnet stack. Ακολουθεί η περιγραφή των αρχείων που είτε δημιουργήθηκαν για να συμπληρώσουν την λειτουργικότητα του stack είτε τα αρχεία του stack που τροποποιήθηκαν. Τα αρχεία αυτά περιέχονται στην τελική μορφή που πήρε το BACnet Stack στα συνοδευτικά αρχεία της εργασίας αυτής και ακολουθεί η περιγραφή τους.

i2c.c: Στο αρχείο αυτό υλοποιείται το I2C (ή TWI). Στον Atmega υπάρχει ένα ξεχωριστό module που χρησιμοποιείται για την επικοινωνία αυτήν το TWI module

το οποίο ελέγχεται από τέσσερις register, τον TWBR (Two Wire Bit Register) που καθορίζει το bit rate της επικοινωνίας, τον TWCR(Two Wire Control Register) που είναι ο κύριος control register και από αυτόν καθορίζονται οι κύριες παράμετροι της επικοινωνίας, τον TWAR (Two Wire Address Register) που περιέχει την διεύθυνση επικοινωνίας και τον TWDR (Two Wire Data Register) που περιέχει τα τελευταία byte που υπήρχε στο Bus, είτε το έχει σταλεί από τον επεξεργαστή (Transmit Mode) ,είτε το έχει λάβει ο επεξεργαστής (Receive Mode).

Διαβάζοντας το datasheet ([3.1] σελ 214) βλέπουμε ότι ο Bit Rate Register έχει που καθορίζει την συχνότητα του σήματος SCK υπολογίζεται από την εξής εξίσωση:

$$SCK\ freq = \frac{CPU\ Clock\ freq}{16+2*(TWBR)*4^{TWPS}}$$

όπου CPU Clock Freq ή συχνότητα του κρυστάλλου του επεξεργαστή TWBR η τιμή του Bit Rate Register και TWPS η τιμή του Prescale Register. Λύνοντας ως προς TWBR και θέτοντας το Prescale ίσο με 0, οπότε $4^{TWPS} = 1$, παίρνουμε για το TWBR

$$TWBR = \left(\frac{Cpu\ Clock\ Freq}{SCK\ Freq} - 16 \right) * (1/2)$$

Την τιμή την δίνουμε στην συνάρτηση `initI2C` όπου γίνονται οι απαραίτητες αρχικοποιήσεις για τον TWBR αλλά και τον TWCR. Στην συνέχεια έχουμε την σύναρτηση `i2cWaitForComplete` όπου χρησιμοποιείται για να περιμένουμε το Interrupt Flag να τεθεί ίσο με 1. Το flag αυτό γίνεται 1 στην περίπτωση που έχουμε στείλει δεδομένα και έχουν ληφθεί είτε ACK είτε NACK. Να σημειωθεί εδώ ότι πειραματικά ο γράφων εντόπισε την περίπτωση να χάνεται κάποιες λίγες φορές είτε τα δεδομένα αποστολή είτε η απάντηση επιβεβαίωσης (ίσως λόγο της ασταθής συμπεριφοράς των συνδέσεων του breadboard) γεγονός που είχε σαν αποτέλεσμα να μπλοκάρει όλη η εφαρμογή στο σημείο αυτό περιμένοντας ένα flag που δεν θα ερχόταν ποτέ. Για τον λόγο αυτό προστέθηκε ο έλεγχος μέγιστων κύκλων αναμονής (10000 που είναι υπερ-αρκετοί) για να μπορούμε να καταλαβαίνουμε ότι χάθηκε η απάντηση οπότε πρέπει η διαδικασία να ξεκινήσει από την αρχή.

Οι υπόλοιπες συναρτήσεις χειρίζονται την επικοινωνία του bus: οι `i2cstart/stop` στέλνουν αντίστοιχα τα σήματα start ή stop στο bus οι `i2cReadACK/ReadNoACK` διαβάζουν τα δεδομένα του bus όταν έχουμε θετική ή αρνητική επιβεβαίωση αντίστοιχα. Η `i2csend` χρησιμοποιείται για να αποσταλούν δεδομένα ενώ τέλος υπάρχει και η συνάρτηση `reset_bus` που αφού μηδενίσει τον TWCR τον επαναρχικοποιεί. Η συνάρτηση αυτή δημιουργήθηκε γιατί πειραματικά παρατηρήθηκε το φαινόμενο ο αισθητήρας διοξειδίου μετά από αρκετές

ολοκληρωμένες επικοινωνίες να “κολλάει” και να μην μπορεί να απαντήσει στα request που του στέλνονται. Με την επαναρχικο-ποίηση του bus όμως το πρόβλημα αυτό λυνόταν.

TM75.c: Στο αρχείο αυτό γίνεται η επικοινωνία με τον TMP75 τον αισθητήρα θερμοκρασίας. Στην συνάρτηση TM_Resolution καθορίζεται η ανάλυση που θέλουμε για την ακρίβεια των μετρήσεων σαν όρισμα από 9 μέχρι 12 bit. Η διεύθυνση του Configuration Register είναι 0x“01” (όπως βλέπουμε από το datasheet [4.9]) οπότε αφού σταλεί Start στο bus, στέλνεται στην συνέχεια η διεύθυνση του TM75 μαζί με το Read/Write bit 0 που σηματοδοτεί εγγραφή και κατόπιν η διεύθυνση του Configuration Register και στην συνέχεια την τιμή που επιθυμούμε να πάρει ο Register αυτός για να αντιστοιχεί στην ακρίβεια που θέλουμε. Από το datasheet βλέπουμε ότι τα bit ακρίβειας είναι 2 που επιτρέπει 4 συνδυασμούς ακρίβειας μετρήσεων, από 9 μέχρι 12 bit ακρίβειας. Στην πράξη χρησιμοποιήθηκε η μέγιστη ανάλυση για την θερμοκρασία που επιτρέπει ακρίβεια μέχρι τρίτου δεκαδικού ψηφίου.

Η πιο σημαντική συνάρτηση είναι η Read_Temp που διαβάζει την τιμή του αισθητήρα. Αφού σταλεί Start, στέλνεται η διεύθυνση του αισθητήρα και στην συνέχεια περιμένουμε να διαβάσουμε 2 byte πληροφορίας όπου για το πρώτο περιμένουμε θετική επιβεβαίωση και στο δεύτερο αρνητική που σημαίνει ότι δεν υπάρχει επόμενο byte πληροφορίας. Στην συνέχεια τα δύο byte αυτά μεταφράζονται στην αντίστοιχη τιμή τους στο δεκαδικό και επιστρέφονται σαν float.

CO2.c: Ο K-20 είναι ένας αισθητήρας αρκετά πολύπλοκος με αρκετές δυνατότητες για αυτό για την επικοινωνία μαζί του χρησιμοποιήθηκε σαν υπόδειγμα ο οδηγός που δίνει η ίδια η εταιρία παραγωγής του που βρίσκεται στο [4.13]. Για να έχουμε μια επιτυχημένη επικοινωνία στέλνεται αρχικά το σήμα start για να ξεκινήσει η επικοινωνία στο bus. Κατόπιν στέλνεται η διεύθυνση του αισθητήρα (0b1101000) μαζί με το Read/Write bit (0 για εγγραφή), μαζί με 0x22 που δηλώνει εντολή τύπου 2 που είναι Read RAM όπως διαβάζουμε στο [4.13] και 2 byte πληροφορίας για να ανάγνωση. Στην συνέχεια στέλνεται το Checksum byte των byte που έχουν ήδη σταλεί για να έχουμε δυνατότητα ελέγχου του δεδομένων. Κατόπιν περιμένουμε 20 ms για να βεβαιωθούμε ότι έχει δοθεί αρκετός χρόνος για να διαβαστεί η τιμή από την μνήμη του αισθητήρα.

Στην συνέχεια διαβάζουμε τέσσερα byte από τον αισθητήρα, τα τρία πρώτα περιμένουν θετική επιβεβαίωση και το τελευταίο το τέταρτο έχει αρνητική επιβεβαίωση που δηλώνει ότι είναι το τελευταίο. Το πρώτο byte είναι το operation status του αισθητήρα που για επιτυχημένη εκτέλεση της εντολής που στάλθηκε είναι 0. Ακολουθούν τα δυο byte που είναι η τιμή της συγκέντρωσης του διοξειδίου και τα τελευταία που είναι το Checksum για έλεγχο της ακεραιότητας των

δεδομένων. Αφού υπολογιστεί το checksum από τα byte που έφτασαν και συγκριθεί με το Checksum που έχει σταλεί και βρεθεί το ίδιο οπότε και έχουμε επικοινωνία χωρίς λάθος επιστρέφεται η τιμή της συγκέντρωσης του διοξειδίου σαν ακέραιος-Integer.

bacnet.c: Το αρχείο αυτό το είδαμε και στην περιγραφή της βιβλιοθήκης αρχικά. Η συνάρτηση `bacnet_task` έχει αλλάξει ώστε να ανταποκρίνεται στις εισόδους που έχουμε. Έτσι έχουμε ένα αντικείμενο Analog Input (AI-0) που αντιστοιχεί στις τιμές που έρχονται από τον αισθητήρα υγρασίας και δυο αντικείμενα Analog Value (AV0, AV1) που αφορούν τα δεδομένα που έρχονται από τους άλλους δύο αισθητήρες (θερμοκρασίας, συγκέντρωσης CO₂). Τα αντικείμενα αυτά ενημερώνουν το property present value που χρησιμοποιείται από την το service Read Property για την ανάγνωση των δεδομένων των αισθητήρων. Η ενημέρωση γίνεται με τις συναρτήσεις `Analog_Value/Analog_Input_Present_Value_Set` οι οποίες καλούνται περιοδικά. Για να εξασφαλιστεί ότι δεν υπάρχει περίπτωση ταυτόχρονης μετάδοσης στο bus χρησιμοποιείται ένας timer που επιτρέπει την ενημέρωση των τιμών κάθε ένα δευτερόλεπτο πρώτα για τους αισθητήρες υγρασίας θερμοκρασίας και έπειτα για τον αισθητήρα διοξειδίου.

test.c: Όπως έχει αναφερθεί ήδη στο `test.c` περιέχονται συναρτήσεις που στέλνουν μέσω της δεύτερης σειριακής θύρας δεδομένα πληροφοριακού χαρακτήρα και όχι BACnet frame. Έτσι έγιναν κάποιες αλλαγές έτσι ώστε πέρα από την MAC address να στέλνει και τις τρέχουσες τιμές των αισθητήρων. Οι τιμές αυτές προέρχονται από static μεταβλητές που υπάρχουν στο αρχείο `bacnet.c` και ενημερώνονται ταυτόχρονα με τα BACnet Objects και χρησιμοποιήθηκε η δυνατότητα αυτή, κυρίως για debugging.

serial.c: Στο αρχείο αυτό προστέθηκαν κάποιες συναρτήσεις για να έχουμε πιο εύκολη αποστολή μηνυμάτων μέσω της σειριακής θύρας όπως η `serialWriteInt` που εκτυπώνει ένα ακέραιο μέσω της σειριακής και η `serialwriteFloat` που εκτυπώνει έναν Float αριθμό με ακρίβεια μέχρι 5 δεκαδικά.

Κεφάλαιο 5

Πειραματικά αποτελέσματα

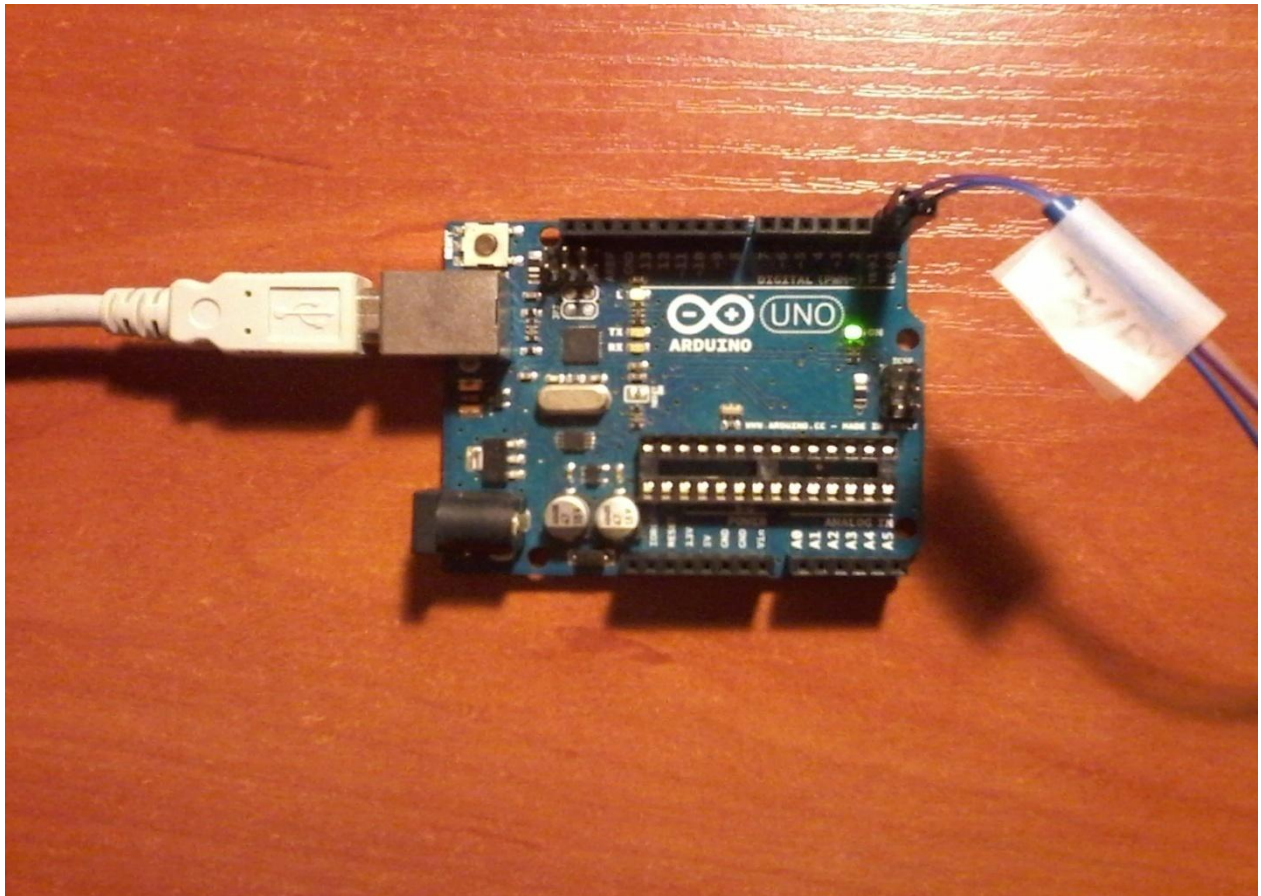
5.1 Εισαγωγή

Στο κεφάλαιο 4 περιγράφηκαν αναλυτικά η λειτουργικότητα του stack αλλά και οι αλλαγές που έγιναν για να λειτουργήσει το ενσωματωμένο σύστημα με τις απαιτήσεις που περιγράφηκαν στο κεφάλαιο 1. Περιγράφηκε ο εξοπλισμός όσον αφορά τους αισθητήρες και την συνδεσμολογία που είχαμε μεταξύ του συστήματος και των αισθητήρων και οι αλλαγές στο επίπεδο εφαρμογής (Application Layer) που χρειάστηκαν να γίνουν για να έχουμε επικοινωνία. Στο κεφάλαιο αυτό θα ασχοληθούμε με τα αποτελέσματα της επικοινωνίας αυτής.

5.2 Επικοινωνία με τον Υπολογιστή

Όπως έχει ήδη αναφερθεί στόχος είναι να το ενσωματωμένο να λειτουργεί σαν server σε request που του γίνονται από έναν BACnet server. Το ρόλο του server μπορεί τον αναλάβει ένας οποιοδήποτε υπολογιστής όπου έχει τουλάχιστον μια USB θύρα (μιας και οι RS-232 θύρες πλέον είναι σπάνιες στους καινούργιους υπολογιστές) και τρέχει τις υλοποιήσεις των services που συνοδεύουν το BACnet Stack. Στην USB θύρα μπορεί να ένας κοινός μετατροπέας από USB είτε σε RS-232 είτε σε RS-485 είτε ακόμα ένας απλός USB-TTL converter ώστε να συνδεθεί απευθείας στο επεξεργαστή. Στην εργασία αυτή χρησιμοποιήθηκε η τρίτη επιλογή και χρησιμοποιώντας ένα Arduino UNO [5.1].

Το Arduino είναι ένα μια πολύ δημοφιλής open source πλατφόρμα ανάπτυξης που βασίζεται σε έναν μόνο επεξεργαστή. Χρησιμοποιείται σε πληθώρα εφαρμογών ενώ μπορεί να λειτουργήσει σαν TTL-USB converter χωρίς καμία παραμετροποίηση απλά σε reset mode. Σε reset mode μπορεί να λειτουργήσει είτε απλώς αφαιρώντας τον επεξεργαστή του είτε απλά συνδέοντας την τροφοδοσία με την reset είσοδο. (για περισσότερες πληροφορίες βλέπε [5.2])



Εικόνα 5.1 Arduino σαν TTL-USB converter εν ώρα λειτουργίας

Στην εικόνα 5.1 φαίνεται το Arduino που χρησιμοποιήθηκε σαν TTL-USB converter. Από την μία είναι συνδεδεμένο με USB type B βύσμα με τον Υπολογιστή και από την άλλη είναι εμφανή τα καλώδια TX/RX που χρησιμοποιούνται για την σειριακή επικοινωνία με το σύστημα.

Αφού λοιπόν εγκατάσταθηκε η επικοινωνία με τον Υπολογιστή το επόμενο βήμα ήταν εκτελεστούν τα Request από μεριάς του Η/Υ ώστε να δρα σαν client στην επικοινωνία αυτήν. Στο φάκελο Stack/demo/ υπάρχει ο πηγαίος κώδικας των services που είναι υλοποιημένα. Για παράδειγμα στον φάκελο Stack/demo/readprop βρίσκεται ο πηγαίος κώδικας για το Read Property Service. Τα αρχεία αυτά πρέπει να γίνουν compile με κάποιον c compiler για να παράγουν τα εκτελέσιμα αρχεία.

Σημαντικό ρόλο παίζει εδώ το λειτουργικό σύστημα. Θα αναφερθώ ξεχωριστά στις δύο διαθέσιμες επιλογές Windows και Linux

5.2.1 Windows

Στα Windows στο κεντρικό path (stack/) υπάρχει ένα αρχείο τύπου Windows Batch File (.bat) το οποίο εκτελεί τον κώδικα που είναι απαραίτητος με τα

κατάλληλα flags για το compile, προυποθέτοντας την ύπαρξη δύο εργαλείων ανάπτυξης λογισμικού: του MinGW (Minimalist GNU for Windows) και του MSYS (Minimal System). Το MinGW (link όπως και για το MSYS στο [5.3]) είναι ένα περιβάλλον ανάπτυξης εφαρμογών για Microsoft Windows, δηλαδή πρόκειται για μια συλλογή εργαλείων που είναι χρήσιμες για ανάπτυξη MS-Windows εφαρμογών, όπως μια συλλογή από C compilers αλλά και assemblers, linkers και άλλα εργαλεία. Το MSYS από την άλλη είναι πυρήνας (kernel) λειτουργικού που προσφέρει μια εναλλακτική σύστημα διερμηνευτή (interpreter) από το κλασσικό Command Line των Windows. Το MSYS είναι απαραίτητο για την λειτουργία του MinGW.

Αφού γίνει η εγκατάσταση των δύο αυτών εργαλείων (βλέπε [5.4]) εκτελείται το αρχείο build.bat μέσα από το command line αφού έχουν γίνει οι εξής διορθώσεις. Πρέπει να προστεθεί η εντολή

```
set PATH=C:[\..\]MinGW\msys\1.0\bin; C:[\..\]MinGW\bin
```

όπου εντός των αγκυλών είναι το path του directory εγκατάστασης των δυο αυτών εργαλείων ενώ εάν θέλουμε το datalink επίπεδο να έχει χρησιμοποιήσει το MS/TP (και όχι να τρέχουν οι εφαρμογές το BACnet/IP που είναι η default ρύθμιση) πρέπει η γραμμή

```
make BACNET_PORT=win32 BUILD=release clean all
```

να αντικατασταθεί με την εντολή

```
make BACNET_PORT=win32 BACDL_DEFINE=-DBACDL_MSTP=1 clean all
```

Το αποτέλεσμα του compile, τα εκτελέσιμα αρχεία αποθηκεύονται στο /stack/bin/.

Στο readme που περιέχεται στον ίδιο φάκελο δίνονται κάποια παραδείγματα λειτουργίας των εφαρμογών αυτών. Για παράδειγμα χρησιμοποιώντας το BACnet/IP services τρέχουμε διαδοχικά:



```
C:\Users\nikos\bin>bacserv 1234
BACnet Server Demo
BACnet Stack Version 0.8.2
BACnet Device ID: 1234
Max APDU: 1476
```

```
C:\Users\nikos\bin>bacwi
Registering with BBMD at 192.168.12.1:47808 for 60000 seconds
Received I-Am Request from 1234, MAC = 192.168.12.1
;Device    MAC (hex)      SNET  SADR (hex)      APDU
-----
; 1234     C0:A8:0C:01:BA:C0    0      00              1476
;
; Total Devices: 1
C:\Users\nikos\bin>
```

Εικόνα 5.2 BACnet/IP Demo server

bacserv -1234 που δημιουργεί έναν BACnet server με device id 1234
bvlc.bat [*.*.*.*] το service που ορίζει το default IP που κατευθύνονται τα services (τα * στο παράδειγμα μας ήταν 192.168.12.1 που είναι η IPV4 διεύθυνση όπως την διαβάζουμε από την *ipconfig*)

bacwi που στέλνει ένα Who-Is service στην IP που έχει οριστεί για όλα τα device id.

Το αποτέλεσμα φαίνεται στην εικόνα 5.2 όπου το πρώτο cmd (το command interpreter των windows) τρέχει το server με device id 1234 και το δεύτερο έχει τα αποτελέσματα του Who-Is request όπου βλέπουμε το Device ID (1234) το MAC address (C0:A8:0C:01:BA:C0) και την IP (192.168.12.1) της συσκευής που μας απάντησε.

Στην συνέχεια έγινε προσπάθεια να χρησιμοποιηθούν τα BACnet/MSTP services για επικοινωνία με τον Υπολογιστή. Δυστυχώς η προσπάθεια αυτή δεν στέφθηκε με επιτυχία. Ενώ οι συνδέσεις ήταν σωστές η προσπάθεια να σταλούν BACnet Frame από τον υπολογιστή στο ενσωματωμένο αποδείχτηκαν άκαρπες. Για την ακρίβεια ενώ BACnet frame μπορούσαν να σταλούν από το ενσωματωμένο στον υπολογιστή η αντίστροφη επικοινωνία δεν είχε αποτέλεσμα.

Τελικά μετά από αρκετές προσπάθειες η αιτία του προβλήματος εντοπίστηκε στην δυσκολία να ανταποκριθεί το λειτουργικό σύστημα στα real-time constraints του ενσωματωμένου. Το πρόβλημα επιγραμματικά ήταν το εξής: Όταν η συσκευή αρχικοποιούταν είχε το frame Token οπότε μπορούσε να στείλει οποιοδήποτε frame επιθυμούσε άρα η επικοινωνία από το ενσωματωμένο στον Υπολογιστή λειτουργούσε ομαλά. Στην συνέχεια το ενσωματωμένο πρέπει να στείλει το token στον επόμενο γειτονικό του κόμβο. Για να βρει ποιος είναι αυτός ο γειτονικός κόμβος, όπως έχουμε ήδη δει, στέλνει το frame Poll for Master σε όλες τις πιθανές διευθύνσεις του δικτύου (0-127) και περιμένει σαν απάντηση ένα Reply-To-Poll-For-Master frame προτού λήξει όμως το Timeout που έχει για την χρήση του Token. Το Timeout αυτό είναι συνήθως 20ms ενώ στην συγκεκριμένη υλοποίηση ήταν 60 ms (ορίζεται από την μεταβλητή *Tusage_timeout* που βρίσκεται στο αρχείο *d1mst.c*).

Ο χρόνος αυτός είναι πρακτικά απαγορευτικός για τα Windows γιατί είτε δεν προλαβαίνουν να απαντήσουν με Reply-To-Poll-For-Master είτε εάν καταφέρουν να απαντήσουν δεν προλαβαίνουν να στείλουν ένα BACnet Frame εντός του Token Timeout. Η αιτία που συμβαίνει αυτό είναι προφανώς ότι δεν μπορεί ένα λειτουργικό σύστημα να ανταποκρίνεται αξιόπιστα σε τόσο μικρούς χρονικούς περιορισμούς (παρόλο που τρέχει σε πολλαπλάσιο γρηγορότερο hardware) καθώς μόνο ένα process swap είναι συγκρίσιμο με τα 20 ms. Η λύση επήλθε με την αύξηση του Timeout σε πάνω από 100 ms (120 για μια στοιχειώδη επικοινωνία)

Στην εικόνα 5.3 βλέπουμε τα αποτελέσματα μιας τέτοιας επικοινωνίας.

```
C:\Users\nikos\Desktop\b\bin>bacepics -v 0
Adjusted interface name to \\.\COM3
RS485 Interface: \\.\COM3
MS/TP MAC: 7F
MS/TP Max_Master: 7F
MS/TP Max_Info_Frames: 1
PICS 0
BACnet Protocol Implementation Conformance Statement

--
--
-- Generated by BACnet Protocol Stack library EPICS tool
-- BACnet/IP Interface for BACnet-stack Devices
-- http://sourceforge.net/projects/bacnet/
--
--
Vendor Name: "BACnet Stack at SourceForge"
Product Name: "bdk-atxx4-mstp"
Product Model Number: "bdk-atxx4-mstp"
Product Description: "BACnet Development Kit"

BIBBs Supported:
<
DS-RP-B
-- possible BIBBs in this device
-- DS-RPM-B
-- DS-WP-B
-- DM-DOB-B
-- DM-DOB-B
-- DM-DCC-B
-- DM-RD-B
-- DS-COU-A
```

Εικόνα 5.3 Επικοινωνία με το ενσωματωμένο σε Windows.

Χρησιμοποιώντας το demo application bacepics που δημιουργεί ένα ηλεκτρονικό PICS (Protocol Implementation Conformance Statement) που μας δείχνει όλες τις πληροφορίες (BIBBS που έχουν υλοποιηθεί, λίστες με τα αντικείμενα κτλ) για την υλοποίηση σε textual μορφή.

Γενικά η επικοινωνία με Windows σε οποιαδήποτε έκδοση δοκιμάστηκε (Windows 8 και 7) ήταν ασταθής και χαρακτηριζόταν από μεγάλους χρόνους για την επικοινωνία για αυτό και δεν προτιμήθηκε σαν τελική λύση και δεν ενδείκνυται για οποιονδήποτε θέλει να δοκιμάσει το BACnet stack.

5.2.1 Linux

Αντιθέτως από ότι περιγράφηκε μέχρι τώρα το stack δουλεύει εξαιρετικά αξιόπιστα σε Linux. Οι δοκιμές έγιναν στην τελευταία έκδοση Ubuntu (14.04). Αρχικά τα demo application πρέπει να γίνουν compile. Το compile έγινε χρησιμοποιώντας gcc compiler δίνοντας όμως τα κατάλληλα flag για το MSTP. Στο terminal του Ubuntu πρέπει να δοθεί η εντολή

```
make BACDL_DEFINE=-DBACDL_MSTP=1 clean all
```

Τα αποτελέσματα του compile αποθηκεύονται στον φάκελο stack/bin/. Τα demo application όπως έχουμε ήδη αναφέρει είτε υλοποιούν κάποιο BACnet service είτε ένα χρήσιμο εργαλείο όπως το mstrcar που αποθηκεύει την εισερχόμενη κίνηση για να μπορεί να εξεταστεί με έναν αναλυτή πρωτοκόλλου. Στις εικόνες 5.4 και 5.5 φαίνεται το αποτέλεσμα μιας τέτοιας επικοινωνίας. Στην εικόνα 5.4 βλέπουμε το αποτέλεσμα ενός Who-Is Request. Ο sever (ενσωματωμένο) απαντάει με ένα I-am service και εκτυπώνονται στο terminal το MAC address, το Device Instance και το μέγεθος του APDU

```
Sending Who-Is
MS/TP Interface: /dev/ttyACM0
RS485: Initializing /dev/ttyACM0=success!
MS/TP MAC: 7F
MS/TP Max_Master: 7F
MS/TP Max_Info_Frames: 1
Received I-Am Request from 0, MAC = 1.0.0.0.0.0
;Device    MAC (hex)          SNET    SADR (hex)          APDU
;-----
; 0         01                0       00                  128
;
; Total Devices: 1
nick@ubuntu:~/Desktop/AllIncluded$
```

Εικόνα 5.4 Who-is service μαζί με τις πληροφορίες που ανακτήθηκαν από την απάντηση

Αντίστοιχα έχουμε την απάντηση σε Read Property Request στην εικόνα 5.5. Διαβάζεται το BACnet Object Analog Value 1 που περιέχει την τρέχουσα τιμή από τον αισθητήρα θερμοκρασίας TMP75. Η τιμή του Present Value του αντικειμένου είναι 29.9375 C.°

```
Read Property Analog Value 1 (Temp)
MS/TP Interface: /dev/ttyACM0
RS485: Initializing /dev/ttyACM0=success!
MS/TP MAC: 7F
MS/TP Max_Master: 7F
MS/TP Max_Info_Frames: 1
20.937500
nick@ubuntu:~/Desktop/AllIncluded$
```

Εικόνα 5.5 Απάντηση σε Read Property Request για το αντικείμενο Analog Value 1.

Αναφέρθηκε προηγουμένως το εργαλείο mstrcar το οποίο μας επιτρέπει να αποθηκεύουμε τα frame του MS/TP σε ένα αρχείο τύπου pcap κατάλληλο για έναν επεξεργασία με έναν αναλυτή πρωτοκόλλων. Στην εικόνα 5.5 φαίνεται η δομή των πακέτων και η ενθυλάκωση των πρωτοκόλλων όπως εμφανίζεται στον ευρέως χρησιμοποιούμενο αναλυτή πρωτοκόλλων WireShark.

132	22.025631	0x01	0x02	BACnet	8 BACnet MS/TP Poll For Master
133	22.184529	0x01	0x03	BACnet	8 BACnet MS/TP Poll For Master
134	22.335927	0x01	0x04	BACnet	8 BACnet MS/TP Poll For Master
135	22.493561	0x01	0x05	BACnet	8 BACnet MS/TP Poll For Master
136	22.648141	0x01	0x06	BACnet	8 BACnet MS/TP Poll For Master
137	22.799573	0x01	0x07	BACnet	8 BACnet MS/TP Poll For Master
138	22.957458	0x01	0x08	BACnet	8 BACnet MS/TP Poll For Master
139	23.143154	0x03	0xff	BACnet-AP	30 Unconfirmed-REQ i-Am device,0
140	23.167083	0x03	0x09	BACnet	8 BACnet MS/TP Poll For Master
141	23.272656	0x03	0x0a	BACnet	8 BACnet MS/TP Poll For Master
142	23.426144	0x03	0x0b	BACnet	8 BACnet MS/TP Poll For Master
143	23.579918	0x03	0x0c	BACnet	8 BACnet MS/TP Poll For Master
144	23.739606	0x03	0x0d	BACnet	8 BACnet MS/TP Poll For Master

► Frame 139: 30 bytes on wire (240 bits), 30 bytes captured (240 bits)
▼ BACnet MS/TP, Src (3), Dst (255), BACnet Data Not Expecting Reply
Preamble 55: 0x55
Preamble FF: 0xff
Frame Type: BACnet Data Not Expecting Reply (6)
Destination Address: 255
Source Address: 3
Length: 20
► Header CRC: 0x43 [correct]

Εικόνα 5.6 Screenshot από τον αναλυτή πρωτοκόλλων WireShark

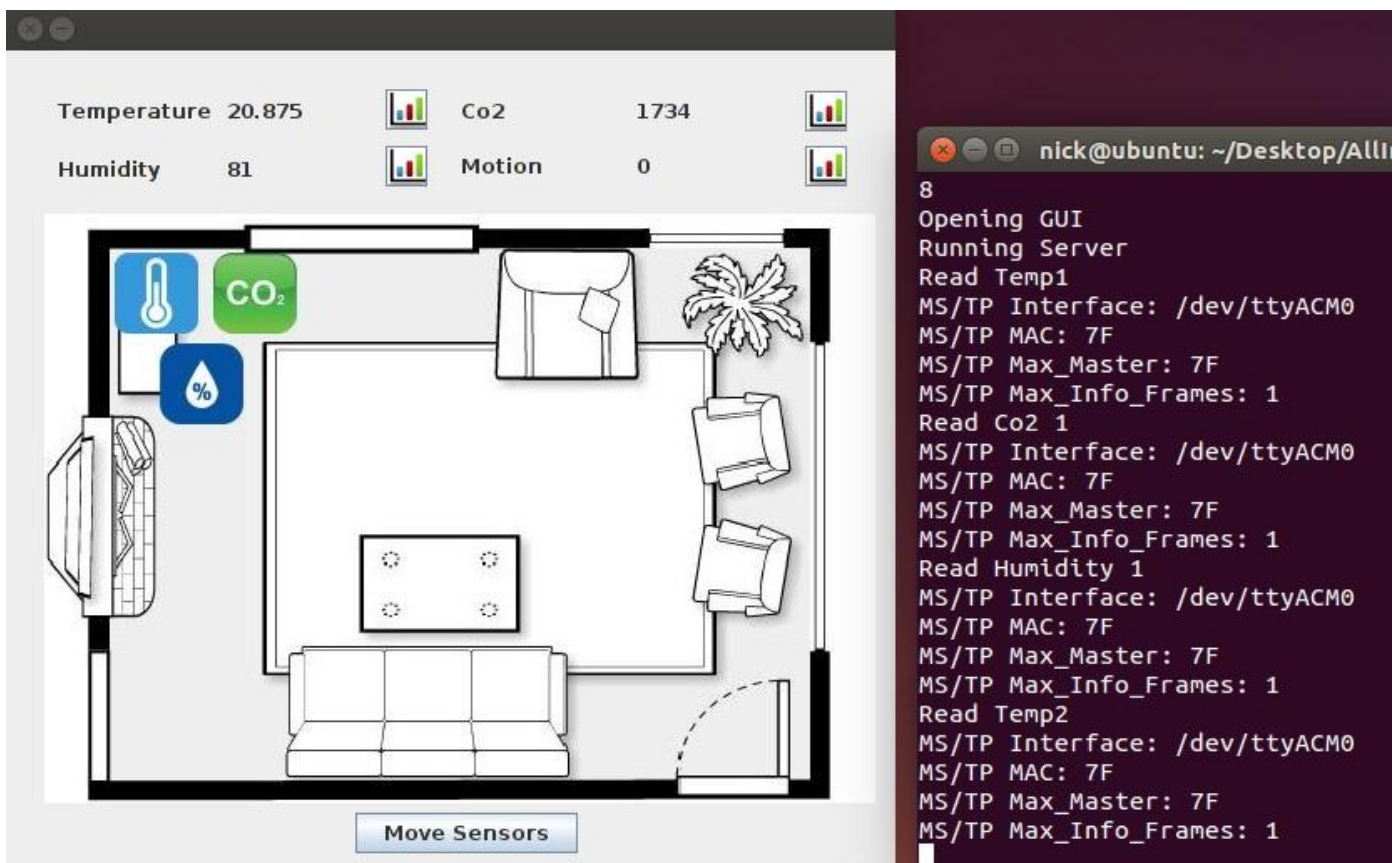
Στην εικόνα 5.6 φαίνονται τα frame που στέλνει το ενσωματωμένο στον Υπολογιστή. Στην εικόνα φαίνεται επιλεγμένο το frame 139. Όπως φαίνεται δίπλα στις πληροφορίες του πακέτου είναι Unconfirmed Service Request τύπου I-am. Παρατηρούμε επίσης ότι μεταξύ 138 και 139 frame το MAC address άλλαξε (από 0x"01" έγινε 0x"03") οπότε έπρεπε να σταλεί I-am service γιατί στις αλλαγές MAC address πάντα πρέπει να ενημερώνεται το δίκτυο. Στις πληροφορίες ακόμα του πακέτου στο κάτω μέρος τις εικόνας φαίνεται σαν source το νέο MAC (0x"03") και σαν destination το 255 (0x"FF") πού όπως έχουμε ήδη δει είναι η διεύθυνση για broadcast στο δίκτυο.

Ένα τελευταίο ενδιαφέρον σημείο στην εικόνα είναι ότι τα πακέτα πριν και μετά το 139 είναι όλα Poll-For-Master το οποία στέλνονται για να ενημερώσουν τυχόν άλλες συσκευές στο δίκτυο ότι υπάρχει ένας master στο συγκεκριμένο MAC. Το ενδιαφέρον εδώ είναι ότι δεν στέλνεται σε broadcast αλλά σε όλες τις πιθανές διευθύνσεις συσκευών στο δίκτυο, από 0 μέχρι 127 (εκτός από την διεύθυνση της συσκευής φυσικά, στο screenshot φαίνεται η αποστολή από το 0x"01" μέχρι 0x"0d"). Αυτό γίνεται για να αποφευχθεί το φαινόμενο της "πλημύρας" στο δίκτυο

(flood) καθώς λόγω του τρόπου που λειτουργεί το broadcast (ένας κόμβος βλέπει ότι τα πακέτα είναι Broadcast το κρατάει για τον εαυτό του το αντιγράφει και το στέλνει στον επόμενο γείτονα του) θα είχαμε μια εκρηκτική αύξηση του αριθμού των πακέτων που θα έπρεπε να αποσταλούν με αποτέλεσμα όλοι οι πόροι του δικτύου να ασχολούνται με αυτό και να μην υπάρχουν πόροι για την επικοινωνία.

5.3 Πειραματικά αποτελέσματα αισθητήρων

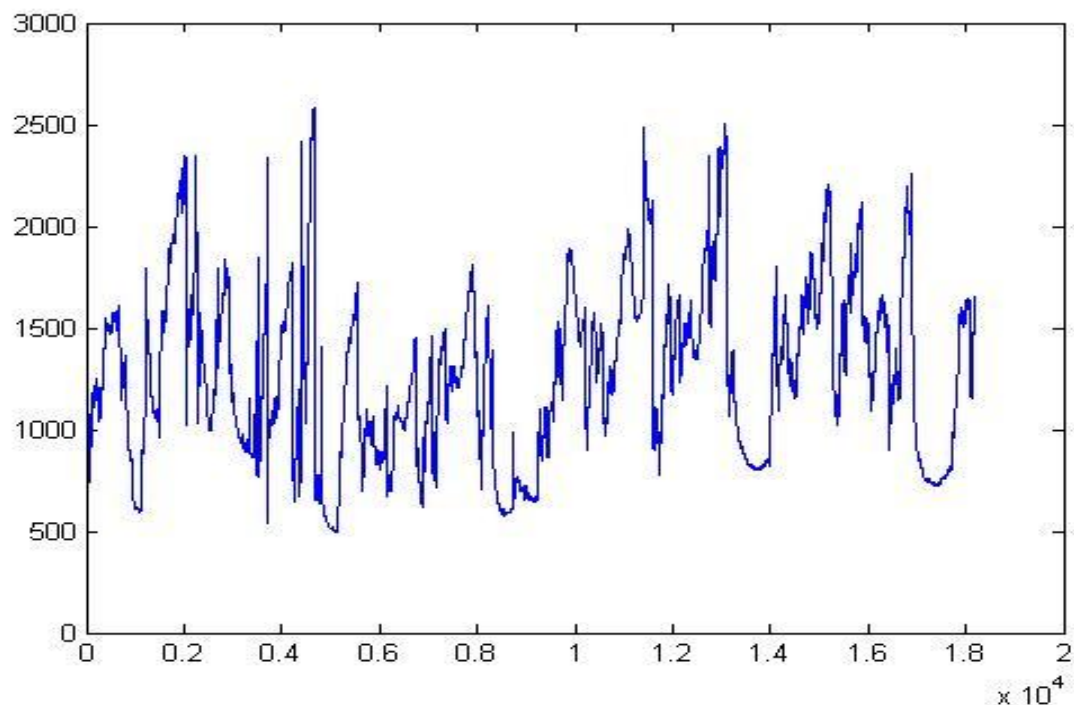
Αφού λύθηκε το πρόβλημα τις επιτυχημένης επικοινωνία με τον Υπολογιστή ξεκίνησε η καταγραφή των δεδομένων των αισθητήρων για την καλύτερη απεικόνιση των δεδομένων δημιουργήθηκε ένα GUI(Graphical User Interface) σε JAVA όπου φαίνεται σχηματικά η θέση των αισθητήρων στο χώρο (σε δισδιάστατη κάτοψη) οι τρέχουσες τιμές των αισθητήρων και υπάρχει η δυνατότητα εμφάνισης των γραφικών παραστάσεων του συνόλου των μετρήσεων μέχρι την στιγμή εκείνη. Στην εικόνα 5.7 φαίνεται ένα screenshot του GUI.



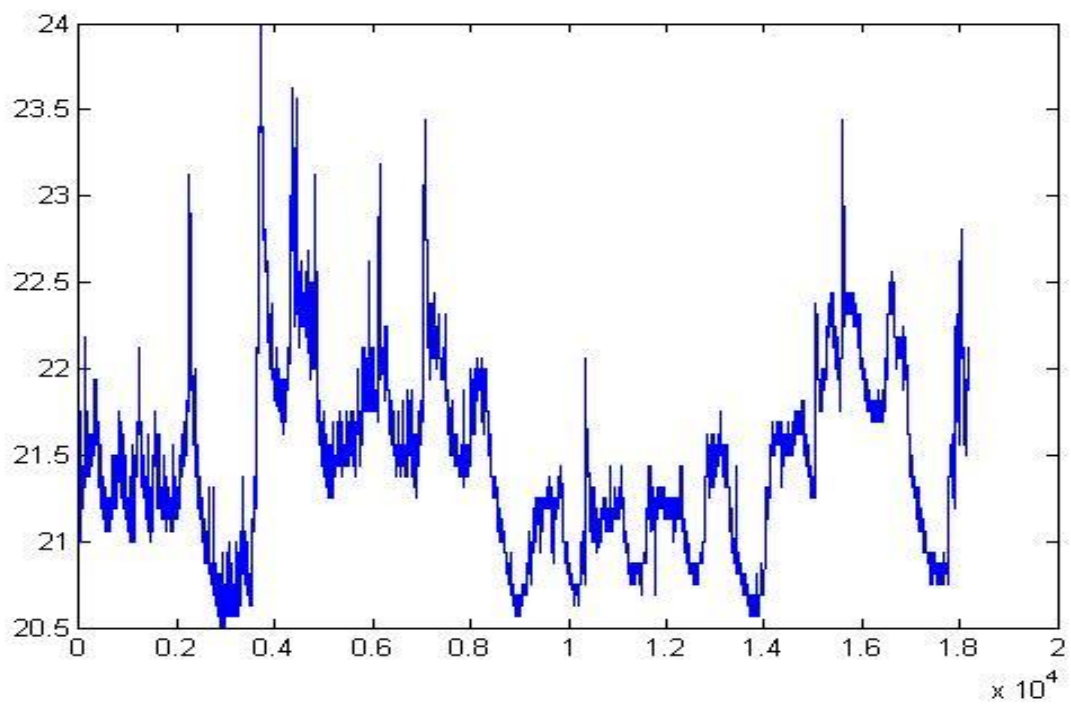
Εικόνα 5.7 Screenshot του GUI που δημιουργήθηκε για την απεικόνιση των Real-Time μετρήσεων

Στην συνέχεια ακολουθούν οι γραφικές παραστάσεις των μετρήσεων συγκέντρωσης διοξειδίου του άνθρακα, θερμοκρασίας και υγρασίας όπως πάρθηκαν από το σύστημα για δεκαπέντε μέρες από 6/11 μέχρι 21/11. Μια ολοκληρωμένη επικοινωνία Υπολογιστή-Ενσωματωμένου διαρκεί 20 sec κατά μέσο όρο οπότε για να διαβαστούν και οι τρεις αισθητήρες χρειαζόμαστε 60 sec πάλι κατά μέσο όρο. Συνεπώς το κάθε δείγμα απέχει χρονικά 1 λεπτό της ώρας από το επόμενο. Το αρχείο με τις μετρήσεις (περίπου 18.200 μετρήσεις) και το χρονικό τους στίγμα υπάρχει στα συνοδευτικά αρχεία της εργασία στο path Files/Measurements

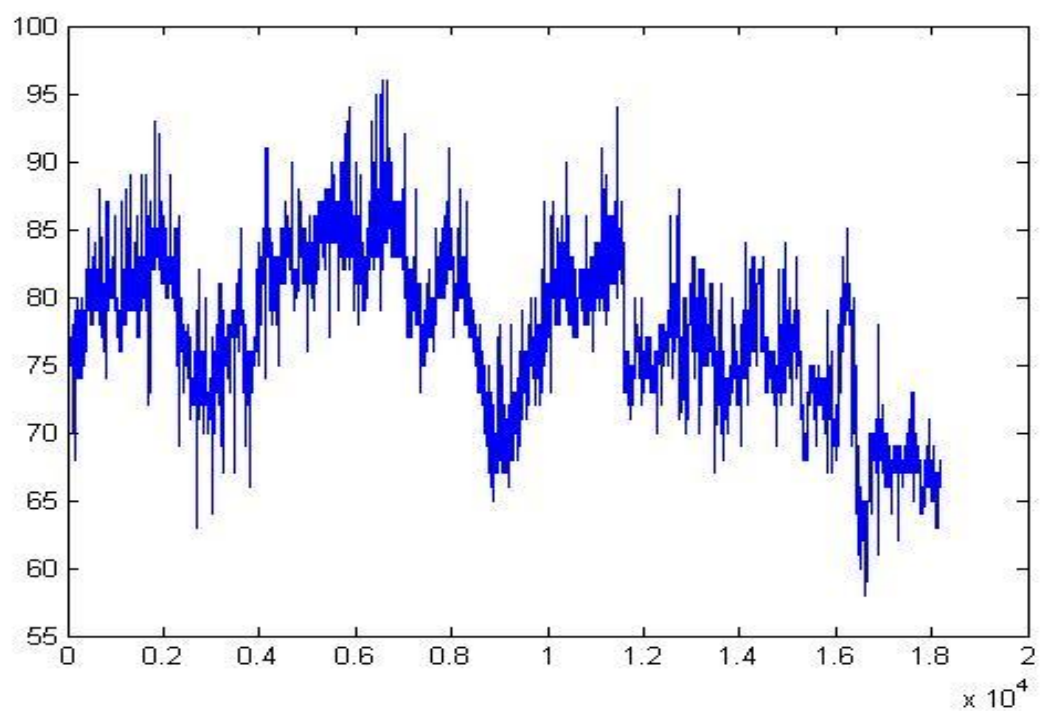
Ακολουθούν οι γραφικές παραστάσεις των μετρήσεων αυτών



Εικόνα 5.8 Μετρήσεις Co2 για 15 ημέρες



Εικόνα 5.9 Μετρήσεις θερμοκρασίας για 15 ημέρες



Εικόνα 5.10 Μετρήσεις υγρασίας για 15 ημέρες

Κεφάλαιο 6

Συμπεράσματα - Προτάσεις για μελλοντική έρευνα

Η εργασία όπως αναφέρθηκε στην παράγραφο 1.1 είχε σαν στόχο την δημιουργία μιας πλατφόρμας ανάπτυξης εφαρμογών στηριγμένων στο BACnet. Ο στόχος αυτός καλύφθηκε πλήρως. Από εκεί και πέρα, μένει να χρησιμοποιηθεί σαν βάση για μια πραγματική BACnet εγκατάσταση σε κάποια κτιριακή υποδομή.

Επιγραμματικά η εργασία αυτή μπορεί να χρησιμοποιηθεί σαν οδηγός για να αναπτυχθεί μια πραγματική BACnet εγκατάσταση που να προσφέρει τα οφέλη των αυτοματισμών όσον αφορά την ενεργειακή κατανάλωση.

Μια τέτοια προσπάθεια σίγουρα θα είχε το πλεονέκτημα ότι ξεκινώντας από ένα δοκιμασμένο επίπεδο λειτουργικότητας, όπως περιγράφεται στην εργασία θα μπορούσε να επεκτείνει της δυνατότητες τις συσκευής υλοποιώντας για παράδειγμα λειτουργίες που περιγράφονται στο πρωτόκολλο αλλά δεν δοκιμάστηκαν εδώ. Με την επιπλέον λειτουργικότητα θα μπορούσαν να δημιουργηθούν συσκευές με πληρέστερο Device-Profile από αυτό που έχει τελικά το ενσωματωμένο σύστημα γεγονός που θα τις κάνει πιο λειτουργικές και με μεγαλύτερο φάσμα εφαρμογών.

Τέλος παρόλο που η εργασία αυτή ασχολήθηκε με δίκτυα MS/TP ίσως θα ήταν χρήσιμο να δοκιμαστούν και άλλες τεχνολογίες επιπέδου ζεύξης δεδομένων όπως το ZigBee για ασύρματη επικοινωνία αλλά και το TCP/IP. Ειδικά η επιλογή δημιουργίας μιας BACnet/IP συσκευής διαφαίνεται ιδιαίτερα επίκαιρη καθώς με την υιοθέτηση της 6^{ης} έκδοσης του πρωτοκόλλου IP (IPv6), εκτινάσσεται ο διαθέσιμος χώρος διευθύνσεων του Internet οπότε πλέον η νέα γενιά ενεργοβόρων συσκευών αιχμής προσανατολίζεται στο να μπορεί να συνδέεται με το Internet απευθείας για σκοπούς διαχείρισης και καταγραφής (Internet of things).

Βιβλιογραφία, Ηλεκτρονικοί Σύνδεσμοι

Κεφάλαιο 1

[1.1] Experimental Case Study of a BACnet-Based Lighting Control System Tae-Jin Park Seung-Ho Hong - Automation Science and Engineering, IEEE Transactions on (Volume:6 , Issue: 2, 2009)

[1.2] BACnet–EnOcean Smart Grid Gateway and its application to demand response in buildings Chang Lin, Jo Hong- Energy And Buildings Vol.78 August 2014

[1.3] Experimental evaluation of BZ-GW (BACnet-ZigBee smart grid gateway) for demand response in building -S.H. Hong, S.H Kim, G.M Kim, H.L Kim - Energy Volume 65, 1 February 2014, Pages 62–70

[1.4] A reference model of fire detection and monitoring system using BACnet –Seon Song, Ho Hong Building And Environment Vol.42 February 2007 p.1000-1010.

[1.5] Performance evaluation of BACnet MS/TP protocol using experimental model - Industrial Technology, 2005. ICIT 2005. IEEE International Conference on December of 2005 - T.JPark- Young-Chan Kwon Seung-Ho Hong

Κεφάλαιο 2

[2.1] BACnet The Global Standard for Building Automation and Control Networks -Michael H. Newman - Momentum Press 2013

[2.2] Building Automation Communication systems with EIBKNX, LON and BACnet Συλλογικό έργο – Springer 2009 [1.2]

[2.3] ANSI/ASHRAE Standard 135-2012, BACnet-A Data Communication Protocol for Building Automation and Control Networks –ANSI/ASHRAE 2012

[2.4] Δίκτυα επικοινωνιών Jean Walrand -Παπασωτηρίου 1997 σελ.54

[2.5] Δικτύωση Υπολογιστών Προσέγγιση από πάνω προς τα κάτω Kurose,Ross -Γκιούρδας 2013 σελ 680

[2.6] Δικτύωση Υπολογιστών Προσέγγιση από πάνω προς τα κάτω Kurose,Ross -Γκιούρδας 2013 σελ 399

- [2.7] CRC Cyclic Redundancy Check Analysing and Correcting Errors W. Kowalk-Fachbereich Informatik 2006 σελ 9
- [2.8] Δίκτυα επικοινωνιών Jean Walrand -Παπασωτηρίου 1997 σελ.166
- [2.9] American Standard Code for Information Interchange, ASA X3.4-1963 American Standards Association. June 17, 1963.

Κεφάλαιο 3

- [3.1] <http://bacnetdevelopmentkit.com/>
- [3.2] <http://bacnet.sourceforge.net/>
- [3.3] <http://www.atmel.com/devices/ATMEGA644P.aspx?tab=documents>
- [3.4] <http://ww1.microchip.com/downloads/en/devicedoc/21081G.pdf>
- [3.5] <http://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>
- [3.6] <http://pdf.datasheetcatalog.com/datasheet2/8/0ie0r0j2cotgc6902k5g19rl5fcy.pdf>
- [3.7] <http://ww1.microchip.com/downloads/en/DeviceDoc/22228B.pdf>
- [3.8] <http://www.alldatasheet.com/datasheet-pdf/pdf/169267/NSC/LM7800.html>
- [3.9] <http://www.labcenter.com/index.cfm>
- [3.10] http://en.wikipedia.org/wiki/Pull-up_resistor
- [3.11] <http://www.radio-electronics.com/info/rf-technology-design/rf-filters/simple-lc-lowpass-filter-design.php>
- [3.12] <http://bacnetdevelopmentkit.com/pictures/bdk-atxx4-mstp-sch-v3.png>
- [3.13] Make: AVR Programming Learning to Write Software for Hardware Elliot Williams Maker Media 2014 σελίδα 128
- [3.14] <http://www.seattlerobotics.org/encoder/jun97/basics.html>
- [3.15] http://en.wikipedia.org/wiki/Through-hole_technology
- [3.16] http://en.wikipedia.org/wiki/Surface-mount_technology

Κεφάλαιο 4

- [4.1] <https://gnu.org/licenses/gpl.html>
- [4.2] <http://opensource.org/licenses/MIT>
- [4.3] http://en.wikipedia.org/wiki/Atmel_AVR

[4.4] Οργάνωση και σχεδίαση Υπολογιστών Patterson Hennessy Κλειδάριθμος 2009 Τόμος Α' σελ 140

[4.5] <http://www.atmel.com/tools/atmelstudio.aspx>

[4.6] Make: AVR Programming Learning to Write Software for Hardware Elliot Williams Maker Media 2014 σελ 68-74

[4.7] Οργάνωση και σχεδίαση Υπολογιστών Patterson Hennessy Κλειδάριθμος 2009 Τόμος Α' σελ 140 σελ 450,492

[4.8] BACnet The Global Standard for Building Automation and Control Networks -Michael H. Newman - Momentum Press 2013 σελ 104

[4.9] <http://www.ti.com/lit/ds/symlink/tmp75.pdf>

[4.10] <https://www.sparkfun.com/datasheets/Sensors/Weather/SEN-09569-HIH-4030-datasheet.pdf>

[4.11] <http://co2meters.com/Documentation/Datasheets/DS30-01%20-%20K30.pdf>

[4.12] <https://cdn.shopify.com/s/files/1/0019/5952/files/Senseair-Arduino.pdf>

[4.13] https://cdn.shopify.com/s/files/1/0019/5952/files/I2C_comm_guide_2_1031.pdf

Κεφάλαιο 5

[5.1] <http://www.arduino.cc/>

[5.2] <http://www.instructables.com/id/Turn-Arduino-into-USB-to-TTL-Adapter-with-1-wire/>

[5.3] <http://www.mingw.org/>

[5.4] http://www.mingw.org/wiki/Getting_Started