Σχεδιασμός και υλοποίηση επεκτάσιμης αρχιτεκτονικής για την ευέλικτη και αποδοτική πρόσβαση σε απομακρυσμένους RFID αναγνώστες

Μάνος Κουτσουμπέλιας

για την απόκτηση του,
μεταπτυχιακού διπλώματος ειδίκευσης

Τριμελής Επιτροπή:
Σπύρος Λάλης, επιβλέπων
Πάνος Δημητρόπουλος,
Γιώργος Σταμούλης

Ιούλιος 2008

ΑΝΑΦΟΡΙΚΑ

Ευχαριστώ τον κκ Σπύρο Λάλη για τις πολύτιμες συμβουλές και την ουσιαστική βοήθειά του ως επιβλέπων καθηγητής αυτής της εργασίας. Επίσης ευχαριστώ τον κκ Πάνο Δημητρόπουλο για την ανεκτίμητη βοήθειά του και την άψογη συνεργασία. Τέλος ευχαριστώ τον κκ Γιώργο Σταμούλη για την συμβολή του στην ολοκλήρωση αυτής της εργασίας. Η εργασία αυτή χρηματοδοτήθηκε από τον Περιφερειακό Πόλο Καινοτομίας Θεσσαλίας έργο το οποίο συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση και το Ελληνικό Δημόσιο.

Περιληψη

Η παρούσα εργασία περιγράφει μία αρχιτεκτονική για την παροχή απομακρυσμένης πρόσβασης σε σταθερές και μετακινούμενες αποθήκες οι οποίες είναι εφοδιασμένες με απλούς RFID αναγνώστες. Το κύριο αντικείμενο του συστήματος είναι να παρέχει με ευέλικτο και αποδοτικό τρόπο την πρόσβαση σε χαμηλού κόστους αναγνώστες, οι οποίοι είναι υλοποιημένοι σε ενσωματωμένα συστήματα, στις υπηρεσίες ενδιάμεσου λογισμικού των εφαρμογών. Το σύστημα είναι υλοποιημένο στο λειτουργικό σύστημα Linux και επικοινωνεί με πρότυπους RFID αναγνώστες διαμέσου αργών ασύρματων συνδέσεων.

Πινακας Περιεχομενων

1 Εισαγωγή

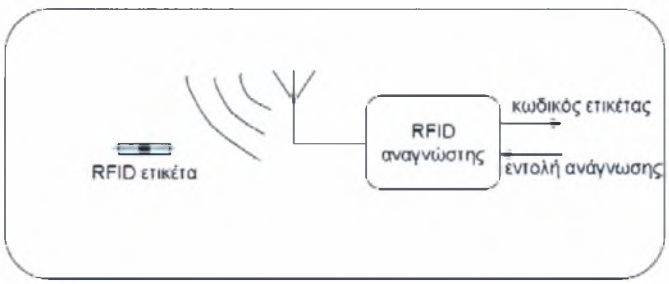
Η ιδέα για την απομακρυσμένη αναγνώριση της ταυτότητας των αντικειμένων μέσω ραδιοσυχνοτήτων είναι αρκετά παλιά. Οι ρίζες της βρίσκονται στις αρχές τις δεκαετίας του 1940 κατά την διάρκεια του 2ου παγκοσμίου πολέμου όταν οι εμπλεκόμενες χώρες όπως η Γερμανία και η Αγγλία χρειαζόταν ένα σύστημα για να αναγνωρίζουν την ταυτότητα των αεροσκαφών που εμφανιζόταν στις οθόνες των επίγειων RADAR. Οι τεχνολογίες της αναγνώρισης απομακρυσμένων αντικειμένων μέσω ραδιοσυχνοτήτων αναπτυχθήκαν αρκετά μέσα στις επόμενες δεκαετίες βρίσκοντας εφαρμογή σε όλο και περισσότερους τομείς και κυρίως αυτόν του εμπορίου.[1][1]

1.1 Η τεχνολογία RFID

Η ραδιοσυχνική αναγνώριση είναι η τεχνική της αναγνώρισης της ταυτότητας των αντικειμένων με την χρήση ραδιοσημάτων. Για να γίνει κατανοητή η διαδικασία αυτή δίνεται ένα απλό αλλά ενδεικτικό παράδειγμα ενός τέτοιου συστήματος. Το σύστημα αποτελείται από τρία μέρη τα οποία είναι μια κεραία, ένας αναγνώστης και μια ραδιοετικέτα. Η κεραία είναι συνδεδεμένη με τον αναγνώστη και η ραδιοετικέτα βρίσκεται στο πεδίο εκπομπής της κεραίας. Ο αναγνώστης μπορεί να δεχτεί εντολές ανάγνωσης και επιστρέφει είτε αριθμούς ετικετών είτε κενό. Η ραδιοετικέτα διαθέτει ένα

μοναδικό κωδικό τον οποίο έχει αποθηκεμένο σε μια περιοχή στη μνήμη της και είναι προσκολλημένη στο αντικείμενο που θέλουμε να αναγνωρίσουμε.

Στην απλή αυτή προσέγγιση μόλις ο αναγνώστης λάβει μια εντολή ανάγνωσης εκπέμπει ένα σήμα μέσω της κεραίας και περιμένει απάντηση. Η ραδιοετικέτα μόλις λάβει το σήμα του αναγνώστη απαντά εκπέμποντας ένα σήμα που περιέχει τον κωδικό της. Στη συνέχεια ο αναγνώστης λαμβάνει αυτό το σήμα το αποκωδικοποιεί ανακτά τον κωδικό και τον επιστρέφει.



Σχήμα 1-1: Τυπικό σχήμα ανάγνωσης RFID ετικετών

Αν αυτή την απλή περίπτωση τη σκεφτούμε σε μεγάλη κλίμακα, στα πλαίσια μιας πραγματικής εφαρμογής, με πολλούς αναγνώστες, αρκετές κεραίες και μεγάλο πλήθος ραδιοτετικετών αυτόματα δημιουργείται η ανάγκη για την παρουσία ενός συστήματος ελέγχου όλων αυτών των συσκευών. Επίσης είναι αναγκαία και η ύπαρξη ενός συστήματος συσχετισμού τον κωδικών που επιστρέφονται από τις συσκευές ανάγνωσης με φυσικά αντικείμενα. Ο σχεδιασμός αυτών των συστημάτων πρέπει να λαμβάνει υπ' όψη του την ετερογένια των τεχνολογιών δικτύωσης των συσκευών ανάγνωσης καθώς και την διαχείριση και προώθηση των αιτημάτων εγγραφής/ανάγνωσης. Επίσης πρέπει να συμπεριλάβει και την ανάγκη χρήσης φίλτρων διαχωρισμού των ετικετών που διαβάζονται ώστε να επιστρέφονται μόνο αυτές που αφορούν τους χρήστες κάθε φορά. Δεδομένου αυτών των αυξημένων απαιτήσεων το σύστημα αυτό δεν θα ήταν εύκολο και

αποδοτικό να υλοποιηθεί στα πλαίσια κάθε εφαρμογής ξεχωριστά. Αυτό μας οδηγεί στην ανάγκη ανάπτυξης ενδιάμεσου λογισμικού το οποίο θα επωμιστεί τον φόρτο που περιγράφικε παραπάνω και στο οποίο οι εφαρμογές θα θέτουν απλά τα αιτήματα τους προς διεκπεραίωση.

Εκτός από την διαχείριση των συσκευών των αναγνωστών θα πρέπει να υπάρχει και ένας μηχανισμός που συσχετίζει αυτά τα δεδομένα με τα πραγματικά προϊόντα. Οι συσχετίσεις αυτές μπορούν να είναι αρκετά απλές δηλαδή σαν αυτή μιας απλής αντιστοίχησης ετικέτας με αντικείμενο αλλά και αρκετά σύνθετες από την στιγμή που πιθανών θα πρέπει να ικανοποιούν κάποιο επιχειρηματικό μοντέλο. Οι συσχετίσεις των ετικετών με σύνθετο τρόπο μπορεί να απαιτούν αρκετή επεξεργασία και σύνθετους τρόπους αντιστοίχησης. Η ανάπτυξη ενδιάμεσου λογισμικού για την ικανοποίηση της διαχείρισης και της συσχέτισης σε μια οντότητα είναι πιθανό να οδηγούσε σε λύσεις που θα ήταν δύσκολο να κλιμακώσουν καθώς οι απαιτήσεις θα αυξάνονταν. Επομένως είναι σκόπιμο να διαχωρίσουμε δυο οντότητες ενδιάμεσου λογισμικού όπου η μία θα έχει σαν αρμοδιότητα την κατασκευή και την διατήρηση του δικτύου των αναγνωστών και ή άλλη την συσχέτιση και την προώθηση των δεδομένων των ραδιοετικέτών σε υψηλότερα επίπεδα λογισμικού[12]. Η εργασία αυτή προτείνει μια αρχιτεκτονική για το ενδιάμεσο λογισμικό της διαχείρισης των αναγνωστών.

1.2 Επισκόπηση κειμένου

Το υπόλοιπο του κειμένου είναι οργανωμένο ως εξής. Στο κεφάλαιο 2 παρουσιάζονται οι τεχνολογίες για RFID συστήματα από την πλευρά του υλικού όσο και ολοκληρωμένες εργασίες σχετικές με την παρούσα. Στο κεφάλαιο 3 παρουσιάζεται η αρχιτεκτονική του συστήματος μας. Στο κεφάλαιο 4 παρουσιάζεται η υλοποίηση και στο κεφάλαιο 5

περιγράφονται τα συμπεράσματα. Τέλος στο κεφάλαιο 6 αναπτύσσονται τα ζητήματα για μελλοντική μελέτη και υλοποίηση.

2 Σχετικές εργασίες

Στο παρών κεφάλαιο παρουσιάζονται οι τεχνολογίες τόσο για το υλικό αλλά και για το λογισμικό που είναι ευρέως διαδεδομένες σήμερα στις εφαρμογές της ραδιοσυχνικής αναγνώρισης.

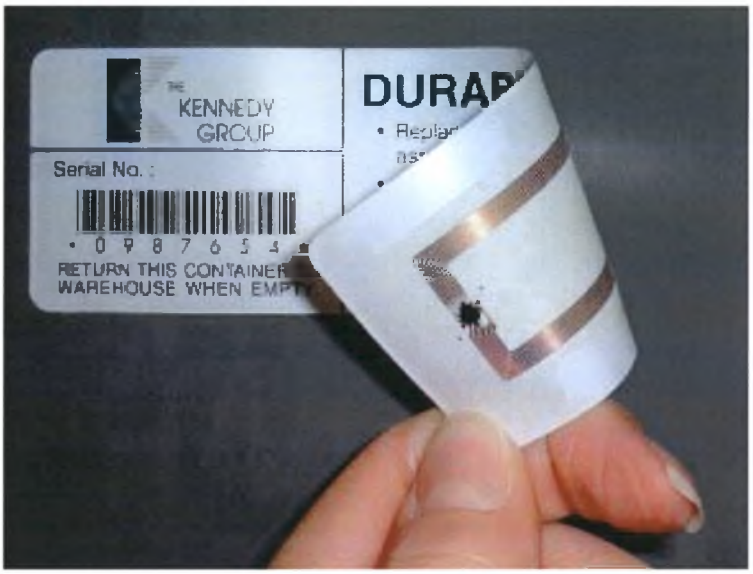
Η σημαντικότερη δουλεία που έχει γίνει μέχρι σήμερα στα συστήματα της ραδιοσυχνικής αναγνώρισης προέρχεται από μια κοινοπραξία ερευνητικών κέντρων και πανεπιστημίων που συνιστούν το Auto-ID Lab με έδρα το MIT. Από την κοινοπραξία αυτή έχουν προκύψει δύο διεπαφές για την προτυποποίηση της επικοινωνίας των ραδιοετικετών με τους αναγνώστες (Class 0 και Class 1) και η προτυποποίηση για ηλεκτρονικό κωδικό προϊόντος EPC, που είναι ένα σύστημα αριθμοδοσίας για τα προϊόντα. Επίσης έχει προταθεί και μια αρχιτεκτονική για την συσχέτιση των ραδιοετικετών με δεδομένα από το διαδίκτυο. Τα πρότυπα αυτά εφαρμόζονται σχεδόν σε όλες τις εμπορικές λύσεις τόσο στο επίπεδο του υλικού αλλά και στο ενδιάμεσο λογισμικό. Οι τεχνολογίες που παρουσιάζονται στη συνέχεια του κεφαλαίου είναι σε μεγάλο βαθμό συμβατές με τις προτυποποιήσεις αυτές [2][1].

2.1 Τεχνολογίες

Η τεχνολογίες για την ραδιοσυχνική αναγνώριση διαφοροποιούνται μεταξύ τους κυρίως ως προς τις συχνότητες ανάγνωσης και την συμπεριφορά των ραδιοετικετών που χρησιμοποιούνται από την κάθε μια. Οι ραδιοετικέτες μπορούν να κατηγοριοποιηθούν σε τρεις μεγάλες κατηγορίες, τις παθητικές τις ενεργές και τις ημι-παθητικές .

Παθητικές ραδιοετικέτες: Οι παθητικές ετικέτες δεν διαθέτουν καμία πηγή ενέργειας για να εκπέμψουν την απάντηση τους ή να διατηρήσουν τα δεδομένα στην μνήμη τους. Είναι σχεδιασμένες ειδικά ώστε οι κεραίες τους να απορροφούν την ενέργεια που προέρχεται από την εκπομπή του αναγνώστη ώστε να την χρησιμοποιήσουν για να εκπέμψουν το δικό τους σήμα. Οι ραδιοετικέτες είναι πιθανό να διαθέτουν μνήμη πολύ μεγαλύτερη από αυτή που χρειάζεται για την αποθήκευση του μοναδικού αριθμού τους οπότε είναι πιθανό να μπορούν να εκπέμπουν και άλλα δεδομένα κατ' απαίτηση του αναγνώστη. Οι παθητικές ραδιοετικέτες χρησιμοποιούνται κατά κύριο λόγω σε εφαρμογές μεγάλης κλίμακας όπως η διαχείριση εφοδιαστικής αλυσίδας, αφού έχουν μικρό κόστος και δεν απαιτείται κάποια περαιτέρω συντήρηση. Επίσης από την στιγμή που δεν υπάρχει μπαταρία το μέγεθός τους παραμένει εξαιρετικά μικρό με αποτέλεσμά να μπορούν να ενσωματωθούν σε αντικείμενα μεγέθους που δεν ξεπερνούν αυτό ενός κουτιού σπίρτων. Το βασικό τους μειονέκτημα είναι οι μικρές αποστάσεις ανάγνωσης που προσφέρουν αφού στις περισσότερες περιπτώσεις αυτές δεν ξεπερνάν τα 20 εκατοστά. Για τις αποστάσεις ανάγνωσης υπεύθυνες είναι κυρίως οι συχνότητες λειτουργίας. Στο εμπόριο κυκλοφορούν ραδιοετικέτες στις συχνότητες των 125 KHz, των 13,56 MHz καθώς και των UHF (EPC Class 1 Gen 2) 860 MHz. Στις δύο πρώτες περιπτώσεις οι αποστάσεις ανάγνωσης παραμένουν μικρές της τάξης των μερικών εκατοστών και για την επίτευξη μεγαλύτερων αποστάσεων απαιτούνται μεγάλες κεραίες αλλά αρκετά μεγάλη ισχύ εκπομπής από τους αναγνώστες. Για την επίτευξη αποστάσεων ανάγνωσης της

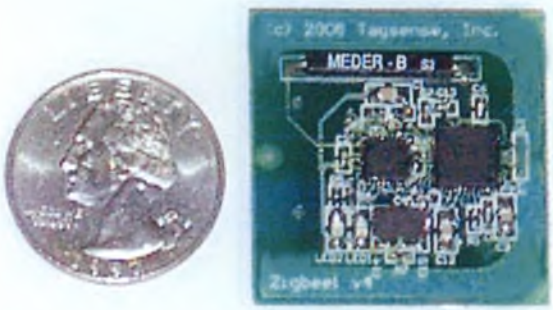
τάξεως των μερικών μέτρων με σχετικά μικρές κεραίες και χαμηλή ισχύ εκπομπής χρησιμοποιούνται οι συχνότητες UHF [1][2].



Εικόνα 2-1: παθητική ραδιοετικέτα

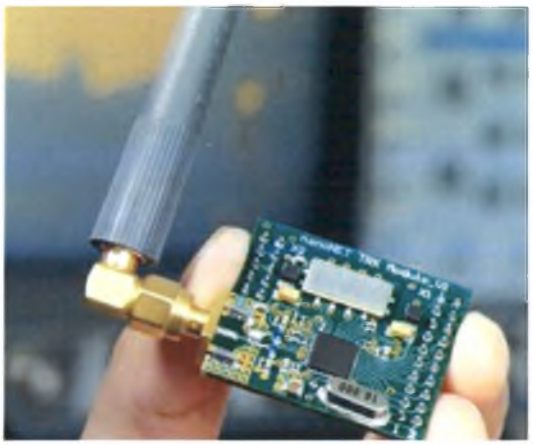
Ενεργές ραδιοετικέτες: Οι ενεργές ραδιοετικέτες διαθέτουν μια πηγή ενέργειας η οποία χρησιμοποιείται για την εκπομπή της απάντησης στο αίτημα του αναγνώστη. Σε αυτού του τύπου τις ετικέτες είναι συνηθισμένη η ύπαρξη μεγάλης μνήμης τις τάξεως τον μερικών Kb αλλά και η ύπαρξη υποσυστημάτων για την συλλογή δεδομένων από αισθητήρες. Λόγω τις τροφοδοσίας είναι δυνατό να επιτευχθούν αρκετά μεγάλες αποστάσεις ανάγνωσης τις τάξεως των αρκετών δεκάδων μέτρων χωρίς την προϋπόθεση μεγάλων διατάξεων κεραιών αλλά και με χαμηλή ενέργεια εκπομπής από τους αναγνώστες. Επίσης η τροφοδοσία εξασφαλίζει καλύτερες συνθήκες εκπομπής σε περιβάλλοντα με υψηλή ηλεκτρομαγνητική δραστηριότητα. Αυτές οι ετικέτες είναι αρκετά μεγαλύτερες από τις αντίστοιχες παθητικές αλλά και αρκετά ακριβότερες. Οι μπαταρίες είναι δυνατό να κρατούν από μερικές εβδομάδες εως και κάποια χρόνια και αυτός είναι συνήθως ο βασικός παράγοντας που καθορίζει το κόστος. Οι ραδιοετικέτες

αυτές είναι πιο συνηθισμένο να λειτουργούν σε υψηλές συχνότητες UHF για να διατηρούν το μέγεθος της κεραίας τους μικρό [2].



Εικόνα 2-2: Ενεργή ραδιοετικέτα

Ημι-ενεργές ετικέτες: Οι ημι-ενεργές ετικέτες είναι μια υβριδική τεχνολογία ετικετών και συνδυάζει τις δυο προαναφερθείσες τεχνολογίες. Σε αυτού του τύπου τις ετικέτες υπάρχει τροφοδοσία από μπαταρία αλλά αυτή δεν χρησιμοποιείται για την εκπομπή της απάντησης στον αναγνώστη. Η μπαταρία χρησιμοποιείται για την διατήρηση των περιεχομένων τις μνήμης αλλά και για την τροφοδοσία κυκλωμάτων για την συλλογή δεδομένων από αισθητήρες που πιθανά φέρει η ραδιοετικέτα. Η απάντηση της ετικέτας προς στον αναγνώστη εκπέμπετε χρησιμοποιώντας την ενέργεια του σήματος εκπομπής του αναγνώστη [1].



Εικόνα 2-3: Ημι-παθητική ραδιοετικέτα

Αναγνώστες : Η κύρια αρμοδιότητα των αναγνωστών είναι να δέχονται και να εκτελούν εντολές εγγραφής και ανάγνωσης στις ραδιοετικέτες. Στην αγορά υπάρχουν διαφόρων δυνατοτήτων αναγνώστες. Οι βασικοί άξονες διαφοροποίησης τους είναι ο αριθμός των πρωτοκόλλων ανάγνωσης/εγγραφής που υλοποιούν, ο αριθμός των κεραιών που διαχειρίζονται και η διεπαφή που παρέχουν στον χρήστη. Επίσης σημαντικό ρόλο παίζουν η ταχύτητα ανάγνωσης, ο αριθμός τον ετικετών που μπορούν να αναγνωριστούν σε ένα κύκλο ανάγνωσης αλλά και η μέγιστη απόσταση που μπορούν να γίνουν αυτές οι λειτουργίες.

Οι αναγνώστες χωρίζονται γενικά σε δύο μεγάλες κατηγορίες, τους σταθερούς και τους μετακινούμενους. Η πρώτη κατηγορία συνήθως προϋποθέτει την ύπαρξη σταθερής πηγής τροφοδοσίας για τον αναγνώστη καθώς και την ύπαρξη εγκατάστασης υποδομής για τις κεραίες του ενώ η δεύτερη περιορίζεται σε συσκευές χειρός. Και στις δυο κατηγορίες υπάρχουν συσκευές που προσφέρουν βασική λειτουργικότητα όπως εγγραφής / ανάγνωσης και χωρίς ιδιαίτερες δυνατότητες διαχείρισης αλλά και συσκευές που διαθέτουν δυνατότητες φιλτραρίσματος, χειρισμού πολλαπλών κεραιών καθώς και υποστήριξη υπηρεσιών και πρωτοκόλλων όπως http, mail, JVM.

2.2 Πρότυπα – Πρωτόκολλα

Υπάρχει μια αρκετά μεγάλη γκάμα προτύπων και πρωτοκόλλων που προδιαγράφουν τα συστήματα της ραδιοσυχνικής αναγνώρισης σε όλα τα επίπεδα. Για την κατακύρωση και την εξέλιξη αυτών των προτύπων υπεύθυνη είναι μια σύμπραξη ερευνητικών κέντρων και εταιριών με την ονομασία EPCglobal. Από την κοινοπραξία αυτή έχουν καθοριστεί τρεις βασικές ομάδες πρωτοκόλλων οι οποίες αφορούν την συλλογή των δεδομένων από τις ραδιοετικέτες, την διαχείριση των δεδομένων αυτών και την προώθηση τους στις υψηλότερες οντότητες λογισμικού [1][2].

Για την επικοινωνία των αναγνωστών με τις ραδιοετικέτες το EPCglobal έχει καθορίσει το Class1 Generation 2 UHF πρωτόκολλο. Αυτό το πρωτόκολλο καθορίζει δύο επίπεδα, το φυσικό επίπεδο επικοινωνίας και το επίπεδο αναγνώρισης τον ραδιοετικετών.

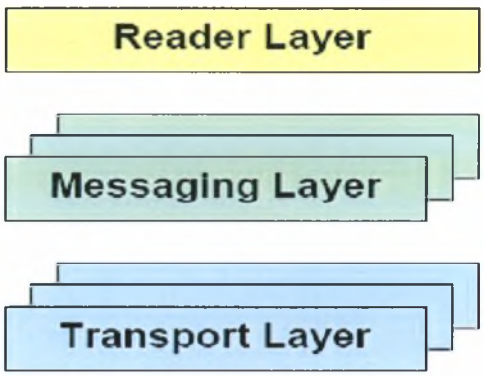
Φυσικό επίπεδο: Στο φυσικό επίπεδο ο αναγνώστης όταν θέλει να στείλει πληροφορία στις ραδιοετικέτες διαμορφώνει ένα υψίσυχνο φέρων ραδιοσήμα χρησιμοποιώντας την ψηφιακή διαμόρφωση πλάτους διπλής ζώνης DSB-ASK ή άλλες ψηφιακές διαμορφώσεις πλάτους όπως SSB-ASK και PR-ASK, χρησιμοποιώντας το πλαίσιο κωδικοποίησης PIE (pulse-interval encodig). Οι ραδιοετικέτες χρησιμοποιούν την ενέργεια αυτού του σήματος για να αποκωδικοποιήσουν αυτό το σήμα και να πάρουν τα δεδομένα που πιθανών θα εγγράψουν στην μνήμη τους. Όταν ο αναγνώστης θέλει να πάρει πληροφορία από της ραδιοετικέτες στέλνει ένα αδιαμόρφωτο υψίσυχνο σήμα και περιμένει να λάβει τις απαντήσεις. Οι ραδιοετικέτες χρησιμοποιώντας αυτό το σήμα διαμορφώνουν τα δεδομένα τους σε ένα υψίσυχνο σήμα χρησιμοποιώντας την διαμόρφωση Miller και τα στέλνουν στον αναγνώστη. Οι συχνότητες επικοινωνίας προδιαγράφονται στα 860MHZ [3].

Επίπεδο αναγνώρισης ετικετών: Ο αναγνώστης για να διαχειριστεί και να αναγνωρίσει τις ραδιοετικέτες εκτελεί τρεις βασικές λειτουργίες. Η πρώτη από αυτές είναι η *select* με την οποία επιλέγει την ομάδα των ραδιοετικετών που θέλει να ανταλλάξει δεδομένα. Η επιλογή αυτή μπορεί να γίνει με βάση τα δεδομένα που έχουν οι ραδιοετικέτες στην μνήμη τους ή με τους EPC κωδικούς τους. Η επόμενη λειτουργία είναι η *Inventory* όπου ο αναγνώστης στέλνει μια ακολουθία σημάτων ερώτησης σε όλες τις ραδιοετικέτες και περιμένει τις απαντήσεις. Στην πραγματικότητα αυτή η εντολή αποτελείται από ένα σύνολο εντολών και εκτελείται σε ένα η περισσότερους γύρους ανάγνωσης. Στο τέλος της εντολής αυτής ο αναγνώστης έχει μια εικόνα για την ραδιοετικέτες που υπάρχουν γύρο του. Τέλος η εντολή *Access* επιτρέπει στον αναγνώστη να γράψει και να διαβάσει την μνήμη των ραδιοετικετών [3].

Σε ότι αφορά την διαχείριση των αναγνωστών το EPCglobal έχουν καθοριστεί δυο πρωτόκολλα το Low Level Reader Protocol (LLRP) και το Electronic Product Code – Reader Protocol (EPC-RP).

Το LLRP πρωτόκολλο προδιαγράφει εντολές χαμηλού επιπέδου για την παραμετροποίηση των πρωτοκόλλων της επικοινωνίας των ραδιοετικετών με τους αναγνώστες. Αναφέρεται σε εντολές για την διαχείριση των χρόνων εκπομπής, τις χρήσεις πολλαπλών κεραιών αλλά και στο καθορισμό των εντολών του *inventory*, *select* και *access* [4].

Το EPC-RP πρωτόκολλο προδιαγράφει όλη την διεπαφή σε χαμηλό επίπεδο μεταξύ των αναγνωστών και των εφαρμογών πελάτη. Είναι σκόπιμο να αναφερθούμε λίγο πιο αναλυτικά σε αυτό από την στιγμή που στην παρούσα εργασία υλοποιείται ένα υποσύνολό του. Στο πρωτόκολλο καθορίζονται τρία επίπεδα τα οποία φαίνονται στο Σχήμα 2-1.



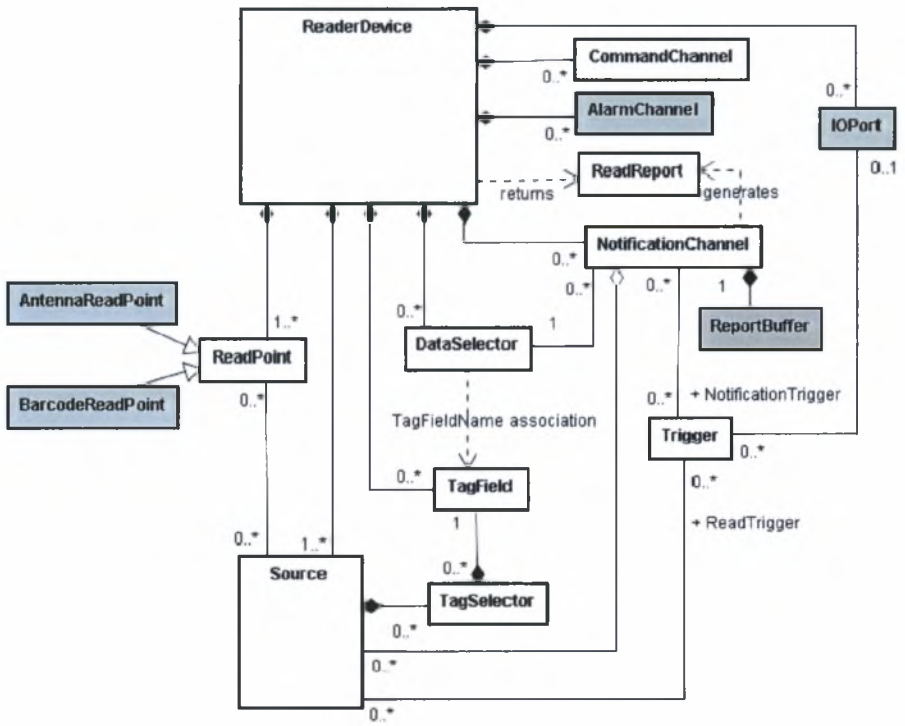
Σχήμα 2-1: Επίπεδα του EPC-RP [5]

Το επίπεδο αναγνώστη (reader layer) είναι υπεύθυνο για τον καθορισμό της σημασίας των μηνυμάτων που ανταλλάσσονται μεταξύ του αναγνώστη και των εφαρμογών. Αυτό το επίπεδο αποτελεί το σημαντικότερο κομμάτι του πρωτοκόλλου από την στιγμή που σε αυτό καθορίζονται ουσιαστικά οι λειτουργίες του αναγνώστη. Στο επίπεδο των μηνυμάτων (messaging layer) καθορίζεται η μορφή των μηνυμάτων, τα περιεχόμενα τους καθώς και η αλληλουχία αποστολής τους και τέλος στο επίπεδο μεταφοράς καθορίζονται οι τεχνολογίες δικτύου για την μεταφορά των μηνυμάτων.

Για την μεταφορά των μηνυμάτων από τις εφαρμογές-πελάτες στους αναγνώστες χρησιμοποιούνται δύο κανάλια επικοινωνίας, το κανάλι ελέγχου και το κανάλι των ειδοποιήσεων. Από το κανάλι ελέγχου στέλνονται οι εντολές που είναι σύγχρονες και συνήθως αφορούν την κατάστασή του αναγνώστη. Για την εντολές που είναι ασύγχρονες ορίζεται το κανάλι των ειδοποιήσεων στο οποίο ο αναγνώστης δέχεται εντολές και απαντά στην εφαρμογή πελάτη με ασύγχρονο τρόπο.

Οι εντολές που εκτελούνται στους αναγνώστες καθώς και οι συσχετίσεις μεταξύ τους δεν είναι εύκολο να μοντελοποιηθούν. Για τον λόγω αυτό το πρωτόκολλο αντιμετωπίζει αυτές τις συσχετίσεις με αντικειμενοστραφή τρόπο όπου κάθε λειτουργικό τμήμα του

αναγνώστη είναι ένα αντικείμενο και οι εντολές που εκτελεί είναι οι μέθοδοι αυτού του αντικειμένου. Στο Σχήμα 2-2 φαίνεται το UML διάγραμμα του πρωτοκόλλου με όλα τα αντικείμενα και τις συσχετίσεις. Για την υλοποίηση του πρωτοκόλλου δεν απαιτείται να υλοποιηθούν τα επιμέρους κομμάτια με αντικειμενοστραφή τρόπο απλά αρκεί να υλοποιείται η λειτουργικότητα τους. Οι μέθοδοι και οι λειτουργίες κάθε αντικειμένου είναι κωδικοποιημένες με την μορφή XML μηνυμάτων. Έτσι τελικά οι εφαρμογές πελάτη και οι αναγνώστες που θέλουν να υλοποιήσουν το πρότυπο αρκεί να ανταλλάσσουν και να εκτελούν τα XML μηνύματα της προδιαγραφής.



Σχήμα 2-2 : UML διάγραμμα το EPC-RP[5]

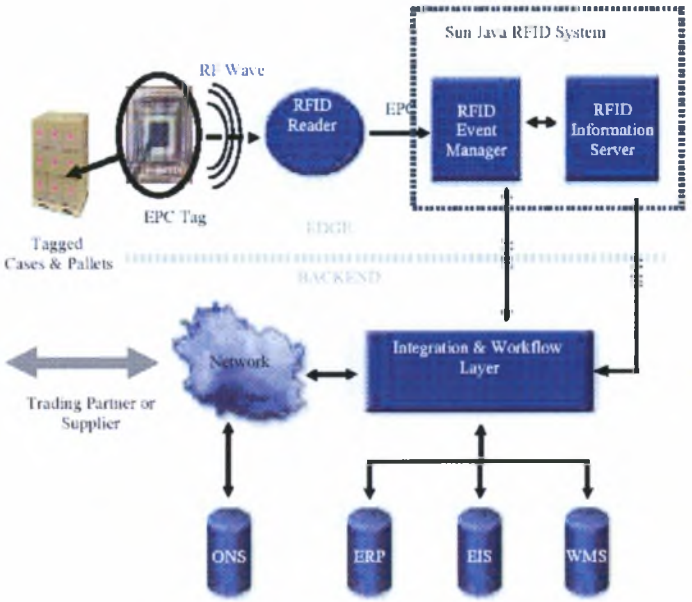
Για την διαχείριση των δεδομένων που προέρχονται από τα χαμηλότερα στρώματα λογισμικού από το EPCglobal έχει καθοριστεί το Application Level Events (ALE)

πρότυπο. Στο πρότυπο αυτό είναι προδιαγεγραμμένος ένας μηχανισμός όπου συλλέγει δεδομένα ραδιοετικετών από πολλαπλούς αναγνώστες τα επεξεργάζεται εφαρμόζοντας φίλτρα και τελικά παράγει γεγονότα που αφορούν τις εφαρμογές των πελατών. Τα γεγονότα που παράγονται είτε προωθούνται απ' ευθείας στης ενδιαφερόμενες εφαρμογές είτε καταχωρούνται σε διακομιστές διαχείρισης πληροφορίας EPC (EPC information services) [2][4][5].

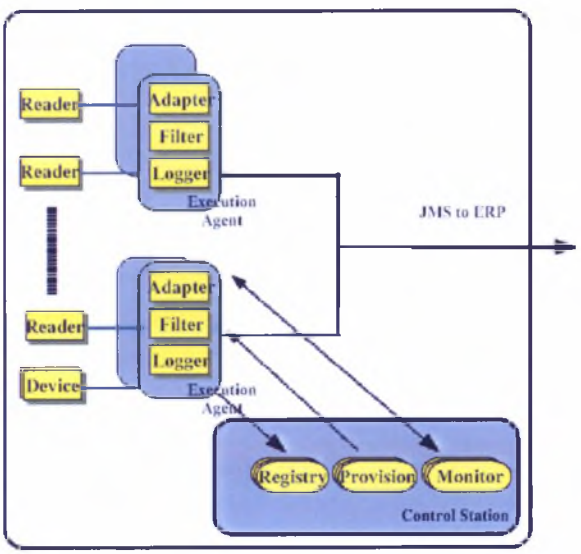
2.3 Χαρακτηριστικές εργασίες

Τυπικά συστήματα ενδιάμεσου λογισμικού παρόμοια με αυτό που προτείνεται σε αυτή την εργασία είναι τα παρακάτω.

Ένα τυπικό εμπορικό παράδειγμα προέρχεται από τη εταιρία SUN με την ονομασία Java system RFID software [2]. Η σχεδίαση της SUN είναι βασισμένη στα πρότυπα που καθορίζει το EPCglobal και αποτελεί μια ολοκληρωμένη λύση ραδιοσυχνικής ιχνηλασίας. Στόχος του συστήματος αυτού είναι η ανάγνωση των δεδομένων των ραδιοετικετών από πολλούς και διαφορετικής τεχνολογίας αναγνώστες το φιλτράρισμά τους καθώς και την δρομολόγηση τους στα πληροφοριακά συστήματα διαχείρισης πόρων (ERP). Το σύστημα αποτελείται από δύο βασικές οντότητες λογισμικού, τον διαχειριστή γεγονότων (RFID Event Manager) και τον εξυπηρέτη διαχείρισης πληροφορίας (RFID Information Server). Η κύρια αρμοδιότητα του διαχειριστή των γεγονότων είναι να συλλέγει και να φιλτράρει τα γεγονότα που δημιουργούνται από τους αναγνώστες καθώς και να τα προωθεί στον εξυπηρέτη διαχείρισης πληροφορίας. Ο εξυπηρέτης διαχείρισης της πληροφορίας έχει σαν κύρια αρμοδιότητα την διατήρηση των δεδομένων των ραδιοετικετών αλλά και την εξαγωγή συσχετίσεων υπό την μορφή γεγονότων προς τα υψηλότερα στρώματα λογισμικού με βάση το εμπορικό μοντέλο [10].Η επισκόπηση της αρχιτεκτονικής αυτού του συστήματος εμφανίζεται στο διάγραμμα της Εικόνα 2-4.



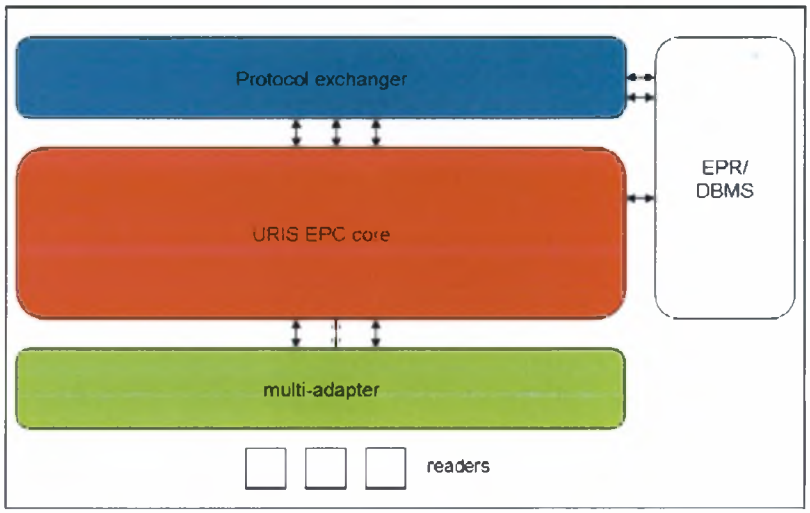
Εικόνα 2-4: αρχιτεκτονική του συστήματος Java system RFID software



Εικόνα 2-5: Σχηματικό διάγραμμα του διαχειριστή πληροφορίας

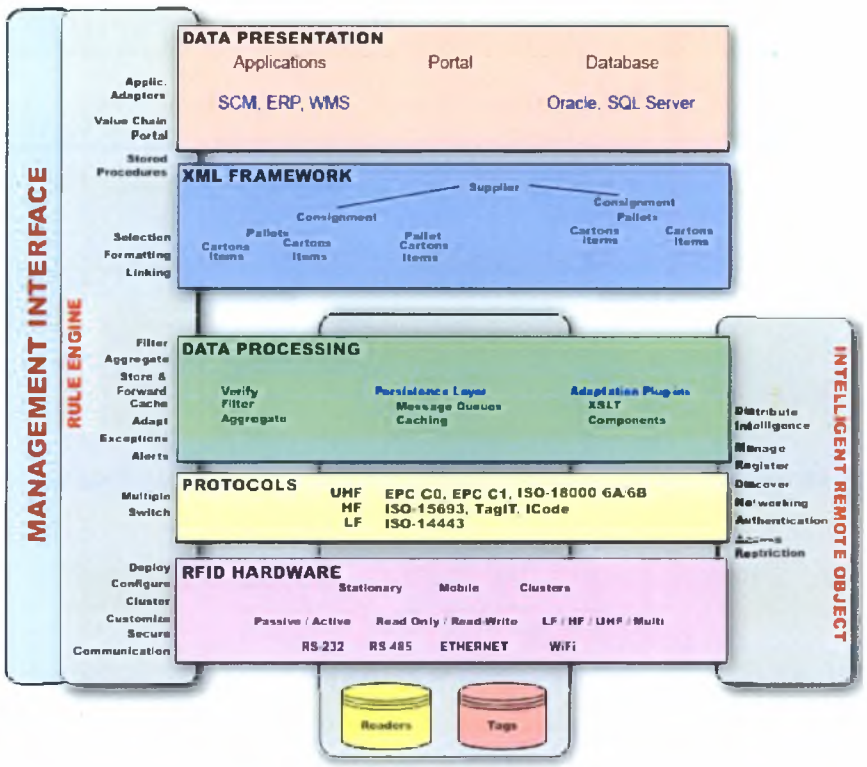
Αναλυτικότερα ο διαχειριστής πληροφορίας αποτελείται από δύο οντότητες, ένα σταθμό ελέγχου (control station) και ένα ή περισσότερους πράκτορες εκτέλεσης (Execution Agents). Οι πράκτορες καταχωρούνται στο σταθμό ελέγχου ο οποίος τους διαχειρίζεται και τους αναθέτει εργασίες που αφορούν την συλλογή και το φιλτράρισμα δεδομένων προερχόμενων από τους αναγνώστες. Οι πράκτορες συστήνονται με δυναμικό και ευέλικτο τρόπο συνδυάζοντας οντότητες προσαρμογέων, φίλτρων και καταγραφέων. Το σύστημα αυτό είναι υλοποιημένο σε Java κάνοντας χρήση των τεχνολογιών JINI και RIO για τον διαχειριστή γεγονότων και J2EE για τον διαχειριστή της πληροφορίας. Βεβαία ο σχεδιασμός αυτός προϋποθέτει την ύπαρξη αρκετά σύνθετων αναγνωστών από την στιγμή που πρέπει να υποστηρίζουν την υποδοχή πρακτόρων που εκτελούν σύνθετες ενέργειες. Το σχηματικό διάγραμμα του διαχειριστή γεγονότων είναι στην Εικόνα 2-5

Ακόμα ένα εμπορικό σύστημα είναι το προϊόν από την εταιρία Allixon URIS το οποίο ακολουθεί παρόμοιο σχεδιασμό με αυτόν του προηγούμενου συστήματος της SUN [6]. Αυτό αποτελείτε από τρία βασικά στοιχεία. Στο χαμηλότερο επίπεδο υπάρχει μια οντότητα πολύ-προσαρμογέα για την υποστήριξη αναγνωστών από διαφορετικές εταιρίες με διαφορετικές διεπαφές. Ο πολυ-προσαρμογέας τροφοδοτεί με τα δεδομένα από τις ραδιοετικέτες μια κεντρική οντότητα λογισμικού η οποία διενεργεί το φιλτράρισμα και την διανομή αυτής της πληροφορίας στις διάφορες εφαρμογές. Η τελευταία οντότητα του συστήματος της Allixon είναι αυτή η οποία φροντίζει για την προσαρμογή και την μεταφορά των δεδομένων στις υψηλότερες οντότητες λογισμικού των συστημάτων διαχείρισης πόρων (ERP) των εταιριών.



Εικόνα 2-6: URIS

Ένα ακόμα σύστημα παρόμοιο με τα προηγούμενα είναι το Το WinRFID [10]. Το σύστημα αυτό είναι ανεπτυγμένο με το .NET πλαίσιο ανάπτυξης εφαρμογών και αποτελείται από πέντε διαφορετικά στρώματα λογισμικού. Το πρώτο επίπεδο φροντίζει να παρέχει μέσω διεπαφών και μηχανισμών, όπως αυτών των προσαρμογέων, μια αφαίρεση για τους αναγνώστες. Επίσης σε αυτό το επίπεδο μοντελοποιούνται με αφαιρετικό τρόπο και οι ραδιοετικέτες. Το δεύτερο επίπεδο είναι υπεύθυνο για την υλοποίηση των διαφόρων πρωτοκόλλων για την επικοινωνία με τους αναγνώστες και το τρίτο φροντίζει για την αξιόπιστη μεταφορά και το πρωταρχικό φιλτράρισμα τον δεδομένων που προέρχονται από τους αναγνώστες. Το τέταρτο επίπεδο ενσωματώνει στα δεδομένα περιγραφές XML για την παραλαβή τους από τις υψηλότερες οντότητες λογισμικού και τα προωθεί στο πέμπτο επίπεδο για την παράδοση σε αυτές. Στο σύστημα αυτό όλα τα επίπεδα επικοινωνούν μεταξύ τους διαμέσου διεπαφών πράγμα που καθιστά εύκολη την προσάρτηση νέων επιπέδων με εμπλουτισμένη λειτουργικότητα. Αυτό επιτυγχάνεται σχετικά απλά αρκεί το κάθε νέο επίπεδο να υλοποιεί την προκαθορισμένη διεπαφή.



Εικόνα 2-7: WinRFID

Ακόμα ένα σύστημα βασισμένο στην τεχνολογία των πρακτόρων είναι το σύστημα της TagCentric [12]. Το σύστημα αυτό είναι μια συλλογή από Java κλάσεις που ενσωματώνουν και αλληλεπιδρούν με τους αναγνώστες. Με συνδυασμό αυτών των κλάσεων δημιουργούνται οντότητες πρακτόρων οι οποίες επικοινωνούν μεταξύ τους ανταλλάσοντας XML μηνύματα. Το σύστημα αυτό είναι υλοποιημένο πάνω από την αρχιτεκτονική του συστήματος Ubiquity το οποίο είναι μια γενικής χρήσης πλατφόρμα για την ανταλλαγή XML μηνυμάτων μεταξύ πρακτόρων. Το σύστημα της TagCentric στην πλειονότητα του αποτελεί μια επέκταση του συστήματος Ubiquity.

Τέλος μια όχι τόσο εμπορική αλλά περισσότερο ερευνητική εργασία αποτελεί το σύστημα RFIDstack [8]. Το σύστημα αυτό αποτελείται από ένα μηχανισμό ανταλλαγής μηνυμάτων και από ένα μηχανισμό συνδρομητικής επικοινωνίας. Οι αναγνώστες διαβάζοντας τις ραδιοετικέτες παράγουν γεγονότα τα οποία μέσω ανταλλαγής μηνυμάτων τροφοδοτούν μια συνδρομητική υπηρεσία. Οι εφαρμογές εγγράφονται στην συνδρομητική υπηρεσία δηλώνοντας το ενδιαφέρων τους για ένα ή περισσότερα γεγονότα. Η κεντρική υπηρεσία αναλαμβάνει να ενημερώσει τις εφαρμογές όταν συμβεί κάποιο γεγονός που τους ενδιαφέρει αλλά παράλληλα διαχειρίζεται και απενεργοποιεί αναγνώστες που παράγουν γεγονότα που κανείς δεν ενδιαφέρεται να λάβει.

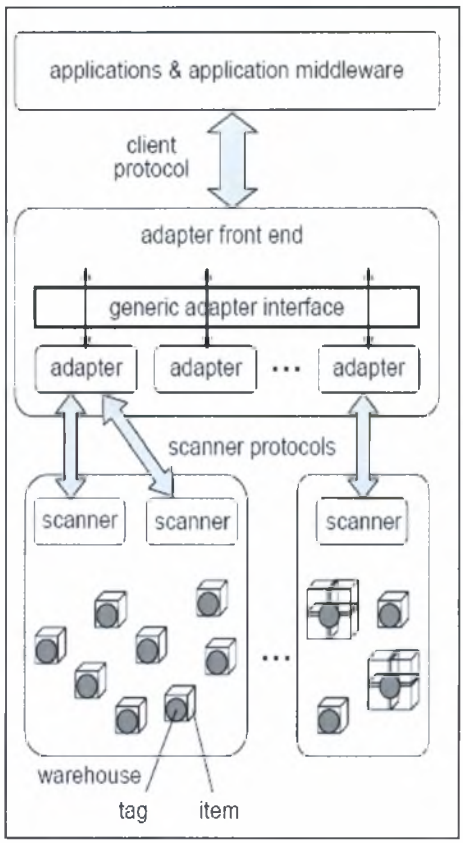
Οι προηγούμενες υλοποιήσεις που παρουσιάστηκαν παραπάνω αντιμετωπίζουν τους αναγνώστες σαν συσκευές με αυξημένες δυνατότητες σε επεξεργαστική ισχύ και επικοινωνία από την στιγμή που ανταλλάσσουν με αυτούς μηνύματα XML μέσω TCP/IP συνδέσεων. Σε πολλές από τις παραπάνω εφαρμογές γίνεται χρήση οντοτήτων λογισμικού προσαρμογέων οι οποίοι όμως χρησιμοποιούνται κυρίως για την μετατροπή των μηνυμάτων από μια τεχνολογία σε μια άλλη και όχι τόσο για την αποδοτική επικοινωνία. Στην υλοποίηση που παρουσιάζεται οι προσαρμογείς φροντίζουν για την αποδοτική επικοινωνία αλλά και για την απόκρυψη των τεχνικών λεπτομερειών δικτύωσης καθώς η επικοινωνία δεν γίνεται μόνο μέσω συνδέσεων TCP/IP.

3 Αρχιτεκτονική συστήματος

Στο παρόν κεφάλαιο θα παρουσιαστεί η αρχιτεκτονική του συστήματος που αναπτύχθηκε. Αρχικά θα δοθεί μια γενική εικόνα του συστήματος και στην συνέχεια το κείμενο θα επικεντρωθεί αναλυτικότερα στα επιμέρους στοιχεία του.

3.1 Επισκόπηση του συστήματος

Το σύστημα που περιγράφεται παρακάτω αναφέρεται στο χαμηλό επίπεδο του ενδιάμεσου λογισμικού RFID. Ο βασικός του στόχος είναι η υποστήριξη απλών συσκευών ανάγνωσης RFID οι οποίες βρίσκονται εγκατεστημένες σε απομακρυσμένες τοποθεσίες, που ενδεχομένως είναι κινητές, και η σύνδεσή τους με το κεντρικό σύστημα γίνεται πάνω από αργές συνδέσεις περιορισμένης χωρητικότητας. Το σύστημα μας κατά κύριο λόγο αποκρύπτει από το ενδιάμεσο λογισμικό της εφαρμογής τις ιδιαιτερότητες της απομακρυσμένης επικοινωνίας όπως και τις τεχνικές ιδιαιτερότητες της εκάστοτε συσκευής ανάγνωσης. Ο σχεδιασμός του συστήματος έγινε με βάση την υπόθεση ότι οι απομακρυσμένες συσκευές είναι χαμηλού κόστους και περιορισμένων δυνατοτήτων από άποψη επεξεργαστικής ισχύος, μνήμης και ενέργειας. Στο Σχήμα 3-1 φαίνεται μια συνολική άποψη του συστήματος με τα επιμέρους υποσυστήματα τα οποία θα εξηγηθούν αναλυτικά στη συνέχεια.



Σχήμα 3-1. Γενικότερη επισκόπηση του συστήματος

Τα βασικά στοιχεία που αποτελούν το σύστημα επιγραμματικά είναι τα εξής:

Στοιχείο (Item): Το αντικείμενο το οποίο βρίσκεται υπό παρακολούθηση. Αυτό μπορεί να είναι ένα φυσικό αντικείμενο όπως για παράδειγμα ένα μπουκάλι κρασί ή μία ομάδα από πολλά αντικείμενα όπως μια παλέτα με κουτιά που περιέχουν για παράδειγμα γιαούρτι.

Ετικέτα (Tag): Η RFID ετικέτα που είναι προσαρτημένη σε ένα στοιχείο. Κάθε ετικέτα υποθέτουμε ότι μπορεί να διενεργεί μετρήσεις για το περιβάλλον που βρίσκεται καθώς και για την κατάσταση του αντικειμένου στο οποίο είναι προσκολλημένη. Επίσης κάθε

ετικέτα είναι χαρακτηρισμένη μοναδικά με αποτέλεσμα τα υψηλότερου επιπέδου στοιχεία λογισμικού να είναι σε θέση να τη συσχετίσουν με ασφάλεια σε κάποιο φυσικό προϊόν.

Αποθήκη (Warehouse): Είναι το μέρος στο οποίο είναι αποθηκευμένα τα στοιχεία με τις ετικέτες. Κάθε αποθήκη περιέχει τουλάχιστο ένα σαρωτή. Η αποθήκη μπορεί να είναι μετακινούμενη η σταθερή.

Σαρωτής (Scanner): Είναι το στοιχείο το οποίο διενεργεί την ανάγνωση των ετικετών RFID που βρίσκονται σε ένα αποθηκευτικό χώρο. Ο σαρωτής επικοινωνεί με το υπόλοιπο σύστημα για να δεχτεί αιτήσεις και να μεταφέρει απαντήσεις.

Προσαρμογέας (Adapter): Το στοιχείο αυτό αναλαμβάνει να επικοινωνήσει με τους σαρωτές διαφόρων τύπων μέσα από διάφορες τεχνολογίες δικτύου.

Κύρια διεπαφή Προσαρμογέα (Adapter Front-end): Είναι το το πιο σημαντικό τμήμα του ενδιάμεσου λογισμικού το οποίο επικοινωνεί με τους πελάτες-εφαρμογές και με οντότητες υψηλότερου επιπέδου στις οποίες επιτρέπει να επικοινωνήσουν και να ανταλλάξουν πληροφορίες με τους απομακρυσμένους σαρωτές.

Μια τυπική ροή δεδομένων είναι η ακόλουθη. Μια εφαρμογή πελάτης στέλνει ένα αίτημα στην κύρια διεπαφή των προσαρμογέων η οποία το προωθεί στους σαρωτές μέσω των αντιστοίχων προσαρμογέων. Κάθε σαρωτής εκτελεί το αίτημα και μετά από κάποια προεπεξεργασία και φιλτράρισμα των δεδομένων στέλνει τα αποτελέσματα μέσω των προσαρμογέων στη κύρια διεπαφή η οποία με την σειρά της ειδοποιεί την εφαρμογή που έθεσε το αίτημα.

Η κύρια διεπαφή των προσαρμογέων καλεί τους προσαρμογείς μέσω μιας άλλης διεπαφής που ονομάζεται γενική διεπαφή προσαρμογέων (generic adapter interface) της οποίας η

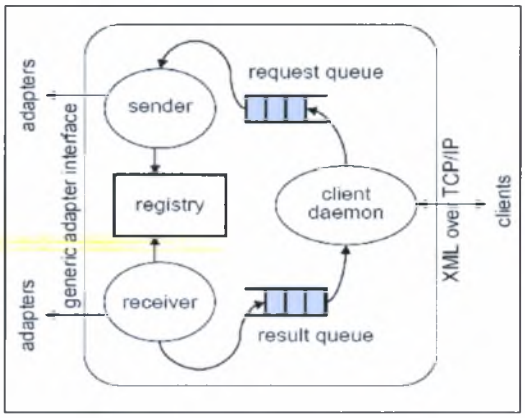
λειτουργικότητα θα εξηγηθεί αναλυτικά παρακάτω. Η αλληλεπίδραση της κύριας διεπαφής με τις εφαρμογές γίνεται μέσω του πρωτοκόλλου πελάτη (client protocol). Για να πετύχουμε επικοινωνία με μεγάλο εύρος εφαρμογών οι οποίες είναι γραμμένες σε διάφορες γλώσσες προγραμματισμού και εκτελούνται σε διάφορα λειτουργικά συστήματα το πρωτόκολλο πελάτη εστιάζεται στην ανταλλαγή XML μηνυμάτων πάνω από συνδέσεις TCP/IP. Ο εφαρμογή-πελάτης έχει την δυνατότητα να στείλει περισσότερα από ένα αιτήματα μέσα από την ίδια σύνδεση οπού το κάθε ένα έχει το δικό του αναγνωριστικό. Τα XML μηνύματα που ανταλλάσουν οι εφαρμογές πελάτες με την κύρια διεπαφή των προσαρμογέων είναι κατά κανόνα ένα τροποποιημένο υποσύνολο του EPC-RP πρωτοκόλλου. Δεδομένου ότι η επεξεργασία XML μηνυμάτων δεν είναι εύκολη υπόθεση για συστήματα με περιορισμένες δυνατότητες ακολουθήσαμε την κατανομή της λειτουργικότητας του EPC reader, όπως αυτή περιγράφεται στο EPC-RP, μεταξύ του σαρωτή και του προσαρμογέα σε δύο διαφορετικές οντότητες οι οποίες επικοινωνούν μεταξύ τους με ένα εσωτερικό πρωτόκολλο.

3.2 Ενδιάμεσο λογισμικό

3.2.1 Κύρια διεπαφή προσαρμογέα

Η κύρια διεπαφή προσαρμογέα αποτελεί τον πυρήνα του συστήματος του ενδιάμεσου λογισμικού και έχει τρεις βασικές αρμοδιότητες. Η πρώτη από αυτές είναι να "φορτώνει" και να "ξεφορτώνει" δυναμικά τους προσαρμογείς που είναι υπεύθυνοι για την επικοινωνία με τους σαρωτές αποκρύπτοντας τις διάφορες τεχνολογίες δικτύου. Η δεύτερη αρμοδιότητα της κύριας διεπαφής είναι η διαχείριση και η διεκπεραίωση των αιτήσεων των εφαρμογών-πελατών. Αυτή η διαδικασία περιλαμβάνει την παραλαβή των XML αιτήσεων καθώς και την μετατροπή τους στο εσωτερικό πρωτόκολλο επικοινωνίας με τους σαρωτές. Τέλος η κύρια διεπαφή προωθεί τα αιτήματα των εφαρμογών στους σαρωτές μέσω των αντίστοιχων προσαρμογέων και αντίστροφα προωθεί τα

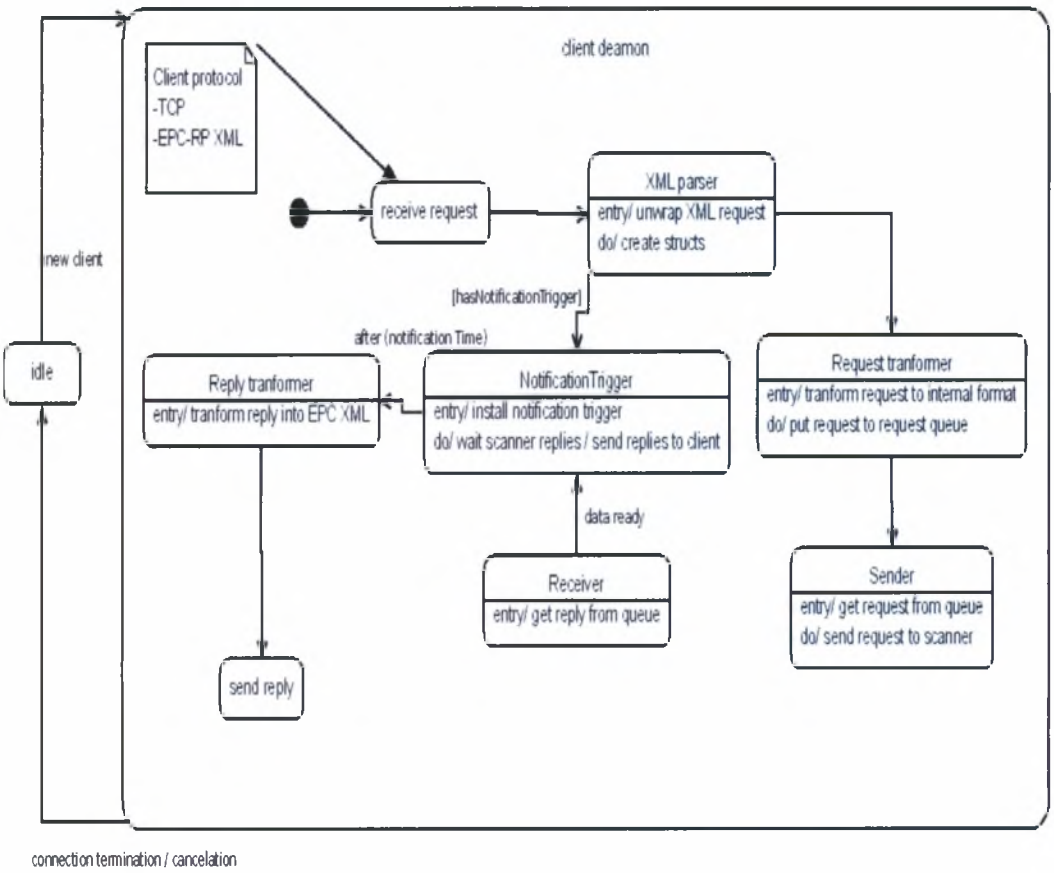
αποτελέσματα από τους σαρωτές στις εφαρμογές. Στο Σχήμα 3-2 φαίνεται η εσωτερική οργάνωση της κύριας διεπαφής με την μορφή των διεργασιών που εκτελούνται



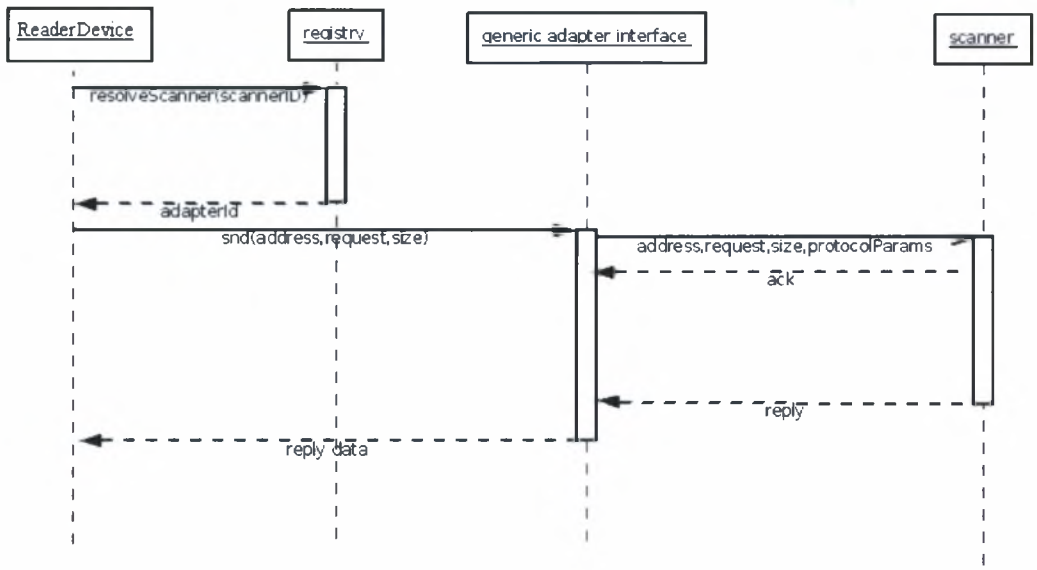
Σχήμα 3-2. Εσωτερική οργάνωση κύριας διεπαφής προσαρμογέα

Οι εφαρμογές-πελάτες εξυπηρετούνται συνολικά από μια διεργασία πελάτη (client daemon) η οποία παραλαμβάνει και στέλνει αιτήματα και απαντήσεις μέσω TCP/IP. Η διεργασία αυτή μπλοκάρει αν δεν υπάρχει αίτημα ή απάντηση να εξυπηρετήσει ή να προωθήσει αντίστοιχα. Όταν η διεργασία πελάτη παραλάβει ένα νέο αίτημα αφαιρεί την XML περιγραφή του και το καταχωρεί στην ουρά των αιτημάτων. Κατά την αντίθετη κατεύθυνση η διεργασία πελάτης αφαιρεί από την ουρά των αποτελεσμάτων τα δεδομένα και τα ενσωματώνει σε XML μηνύματα. Στη συνέχεια τα αποτελέσματα, σε XML μορφή, στέλνονται στην εφαρμογή που ενδιαφέρεται γι' αυτά μέσω της ίδιας TCP/IP σύνδεσης από την οποία προήλθαν τα αιτήματα. Αν η εφαρμογή πελάτης τερματίσει την σύνδεση τότε ακυρώνονται όλα τα ερωτήματα που είχαν σταλεί από την συγκεκριμένη εφαρμογή. Μια δεύτερη διεργασία αποστολέας (sender) αναλαμβάνει να προωθήσει τα αιτήματα στους σαρωτές. Ένας βρόχος διαβάζει και αφαιρεί τα αιτήματα από την ουρά των αιτημάτων και τα προωθεί στον αντίστοιχο προσαρμογέα. Τέλος μια τρίτη διεργασία αναλαμβάνει να παραλάβει τα δεδομένα από τους προσαρμογείς και να τα τοποθετήσει

στην ουρά των αποτελεσμάτων. Η παραπάνω περιγραφή φαίνεται αναλυτικότερα στο Διάγραμμα 3-1.



Διάγραμμα 3-1. Αναλυτική απεικόνιση του κύριας διεπαφής



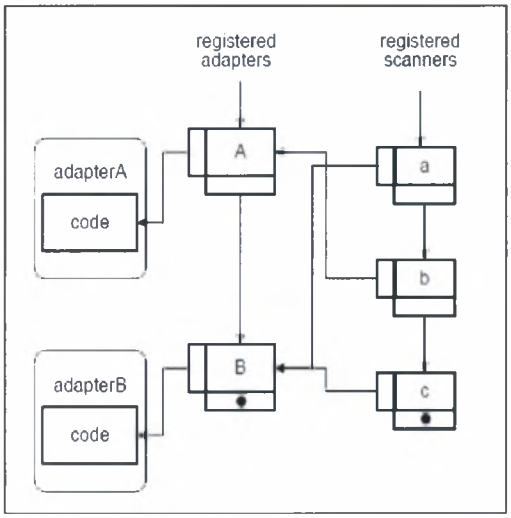
Διάγραμμα 3-2. Τυπική ανταλλαγή μηνυμάτων

3.2.2 Πολυπλεξία / Διευθυνσιοδότηση

Κάθε σαρωτής αναγνωρίζεται με ένα μοναδικό αριθμό. Οι σαρωτές είναι καταχωρημένοι στην κύρια διεπαφή προσαρμογέων μέσω μιας διαδικασίας που συσχετίζει το αναγνωριστικό και τη διεύθυνση του σαρωτή με τον αντίστοιχο προσαρμογέα. Αυτή η δομή συνιστά ένα μητρώο εγγραφών προσαρμογέων. Η καταχώρηση των σαρωτών στο μητρώο γίνεται είτε χειροκίνητα από τον διαχειριστή είτε αν αυτό δεν είναι δυνατό ο σαρωτής αναλαμβάνει να ενημερώσει την κύρια διεπαφή στέλνοντας ένα μήνυμα εγγραφής κατά την εκκίνηση του (Παράρτημα Α). Αν κάποιος σαρωτής επανεκκινηθεί τότε το μήνυμα εγγραφής αγνοείτε.

Το μητρώο στην κύρια διεπαφή προσαρμογέων αποτελείτε από δύο κύριες λίστες: μια που περιέχει τους καταχωρημένους προσαρμογείς και μία που περιέχει τους καταχωρημένους σαρωτές. Επιπρόσθετα κάθε σαρωτής δείχνει στην αντίστοιχή

καταχώρηση του προσαρμογέα που με την σειρά του περιέχει τον αντίστοιχο δείκτη στην γενική διεπαφή προσαρμογέων. Αν κάποια εφαρμογή-πελάτης θέλει να στείλει ένα αίτημα σε κάποιο σαρωτή θα πρέπει να επισυνάψει και το αναγνωριστικό του στην κύρια διεπαφή προσαρμόσεων απ' οπού στη συνέχεια ο προσαρμογέας θα αναγνωριστεί μέσω του μητρώου. Αν δεν υπάρχει αντίστοιχη εγγραφή στο μητρώο τότε ένα μήνυμα λάθους τοποθετείτε στην ουρά τον αποτελεσμάτων.



Σχήμα 3-3. Κατάσταση του μητρώου

Στο Σχήμα 3-3 φαίνεται η κατάσταση του μητρώου για μια περίπτωση με δυο προσαρμογείς και τρεις σαρωτές όπου οι σαρωτές a και c μπορούν να προσπελαστούν μέσω του προσαρμογέα B ενώ ο σαρωτής b μπορεί να προσπελαστεί μέσω του προσαρμογέα A. Η κύρια διεπαφή προσαρμογέα περιοδικά ελέγχει τους σαρωτές για να διαπιστώσει δυσλειτουργίες. Αν ο σαρωτής δεν απαντήσει μετά από ένα αριθμό ελέγχων τότε η κύρια διεπαφή ενημερώνει όλες τις εφαρμογές που έχουν θέσει αιτήματα σε αυτόν για το πρόβλημα. Εκτός του ότι παρέχει διαφάνεια στην επικοινωνία η κύρια διεπαφή προσαρμογέα λειτουργεί και σαν πολυπλέκτης από την στιγμή που παρέχει μια μοναδική είσοδο στο σύστημα για όλους τους σαρωτές. Αυτό δίνει τη δυνατότητα στην εφαρμογή

πελάτη να παρέχει όλα τα αιτήματα της σε μια μοναδική οντότητα λογισμικού ανεξάρτητα αν οι σαρωτές είναι "γνωστοί" στην εφαρμογή κάθε φορά. Επιπρόσθετα οι εφαρμογές πελάτες έχουν την δυνατότητα να στείλουν προς όλους τους σαρωτές ένα αίτημα, χρησιμοποιώντας ένα ειδικού τύπου αναγνωριστικό πολυεκπομπής. Με αυτό τον τρόπο οι εφαρμογές μπορούν να επαναπροσδιορίζουν δυναμικά τα αιτήματα τους. Για παράδειγμα εάν μια εφαρμογή-πελάτης επιθυμεί να παρακολουθήσει ένα προϊόν το οποίο μετακινείται και δεν γνωρίζει τη θέση του τότε μπορεί να θέσει ένα αίτημα σε όλους τους σαρωτές και στην συνέχεια να επαναπροσδιορίσει το αίτημα της μόνο για τον σαρωτή που απάντησε για το συγκεκριμένο προϊόν.

3.2.3 Προσαρμογείς

Οι προσαρμογείς είναι ο μηχανισμός ο οποίος αποκρύπτει τις λεπτομέρειες των τεχνολογιών του δικτύου που παρεμβάλλεται μεταξύ των σαρωτών και της κύριας διεπαφής προσαρμογέα. Με αυτή την προσέγγιση η κύρια διεπαφή προωθεί και παραλαμβάνει μηνύματα από και προς τους σαρρωτές με ένα ενιαίο τρόπο ανεξάρτητα τον τρόπο που οι σαρωτές επικοινωνούν με το ενδιάμεσο λογισμικό. Ο τρόπος που το επιτυγχάνει αυτό είναι η χρήση μιας γενικής διεπαφής (generic adapter interface) την οποία είναι υποχρεωμένοι να υλοποιούν όλοι οι προσαρμογείς.

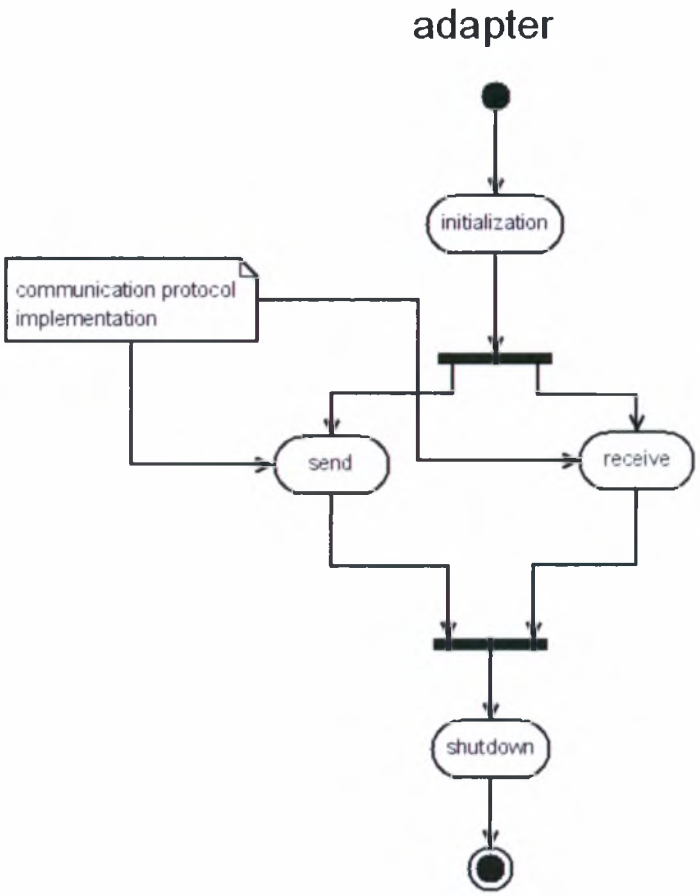
Η γενική διεπαφή των προσαρμογέων εμφανίζεται στο Τμήμα κώδικα 3-1. Σχεδιαστικά έχει επιλεγεί ότι κάθε προσαρμογέας έχει το δικό του νήμα ελέγχου για την αποστολή και την παραλαβή δεδομένων από τους σαρωτές που είναι συσχετισμένοι μαζί του.

```
void (*getinfo) (char **infostr);
void (*init) (struct sem *avldata, char *configfile);
void (*snd) (char * addr, char *request, int size);
int (*rcv) (char **addr, char **reply, int size);
void (*shutdown) (void);
```

Τμήμα κώδικα 3-1: Γενική διεπαφή προσαρμογέων

Αυτό δημιουργεί ανεξαρτητοποίηση της κύρια διεπαφής και τη εσωτερικής υλοποίησης του κάθε προσαρμογέα δίνοντας την δυνατότητα υλοποίησης ασύγχρονων σχημάτων επικοινωνίας. Επιπρόσθετα κάθε προσαρμογέας ειδοποιεί την διεργασία παραλαβής των δεδομένων η οποία βρίσκεται σε αναμονή, μέσω ενός γενικού σημαφόρου. Όταν η διεργασία αφυπνιστεί τότε μαζεύει τα δεδομένα από τους προσαρμογείς που την ειδοποίησαν μέσω μια κλήσεις συνάρτησης. Το διάγραμμα 3-3 παρουσιάζει τον κύκλο ζωής ενός προσαρμογέα

Η κύρια διεπαφή προσαρμογέων υποστηρίζει την προσθήκη και την απομάκρυνση προσαρμογέων χωρίς να χρειάζεται να επανεκινηθεί το σύστημα. Ένας νέος προσαρμογέας μπορεί να προστεθεί στο σύστημα μέσω μια εντολής της κύριας διεπαφής η οποία "φορτώνει" το αντίστοιχο άρθρωμα (module) του προσαρμογέα. Στην συνέχεια εξάγονται τα σύμβολα των συναρτήσεων και δημιουργείται μια νέα εγγραφή στο μητρώο της κύριας διεπαφής. Τέλος καλείται η init συνάρτηση του προσαρμογέα για να ολοκληρωθεί η αρχικοποίηση του. Οι όποιες ιδιαιτερότητες του προσαρμογέα καθορίζονται από το αρχείο ρυθμίσεων του οποίου το όνομα και η διαδρομή είναι παράμετρος στην init συνάρτηση. Αντίθετα για να καταργηθεί κάποιος προσαρμογέας καλείται η shutdown συνάρτηση η οποία διαγράφει την εγγραφή του προσαρμογέα από το μητρώο και ξεφορτώνει το αντίστοιχο άρθωμα.



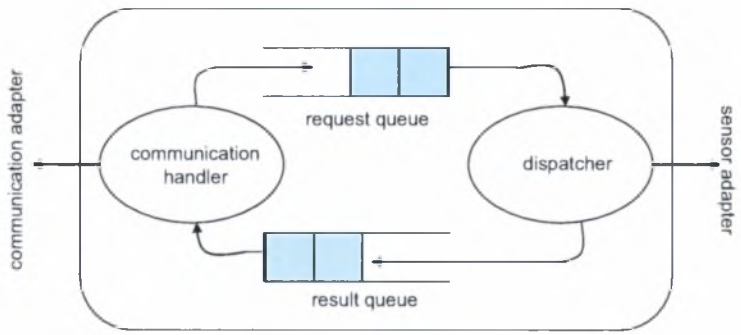
Διάγραμμα 3-4. Κύκλος ζωής προσαρμογέα

3.3 Σαρωτές

Οι σαρωτές είναι συσκευές που εκτελούν το υποσύνολο των EPC-RP εντολών που έχουμε προδιαγράψει. Είναι επιφορτισμένοι με την αποκωδικοποίηση και εκτέλεση των εσωτερικών εντολών που προέρχονται από την κύρια διεπαφή προσαρμογέα.

3.3.1 Αρχιτεκτονική σαρωτών

Μια επισκόπηση της αρχιτεκτονικής του σαρωτή φαίνεται στο Σχήμα 3-4.



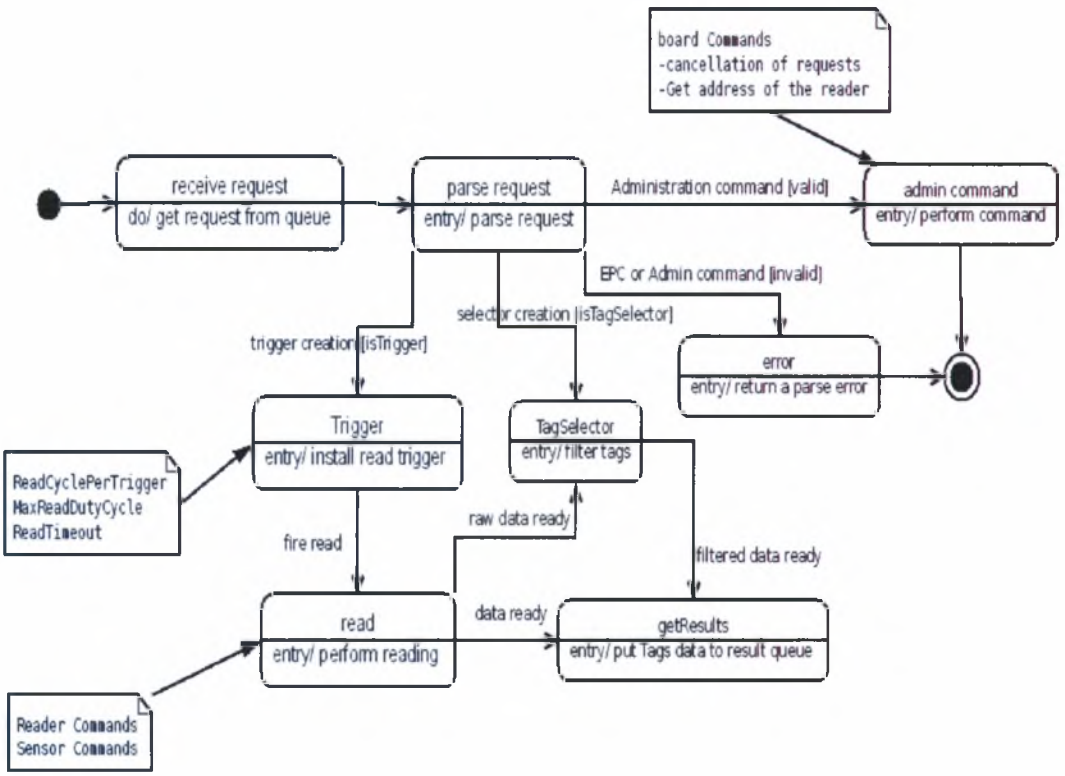
Σχήμα 3-4

Στον σαρωτή υπάρχουν δυο διεργασίες. Η μια φροντίζει για την επικοινωνία με την κύρια διεπαφή προσαρμογέα, και η άλλη φροντίζει για την αποκωδικοποίηση και εκτέλεση των αιτημάτων. Όπως φαίνεται στο σχήμα υπάρχουν τα αρθώματα επικοινωνίας τα οποία φορτώνονται κατά την εκκίνηση της εφαρμογής, οι ενδιάμεσες αποθήκες καθώς και τα αρθώματα των αισθητήρων.

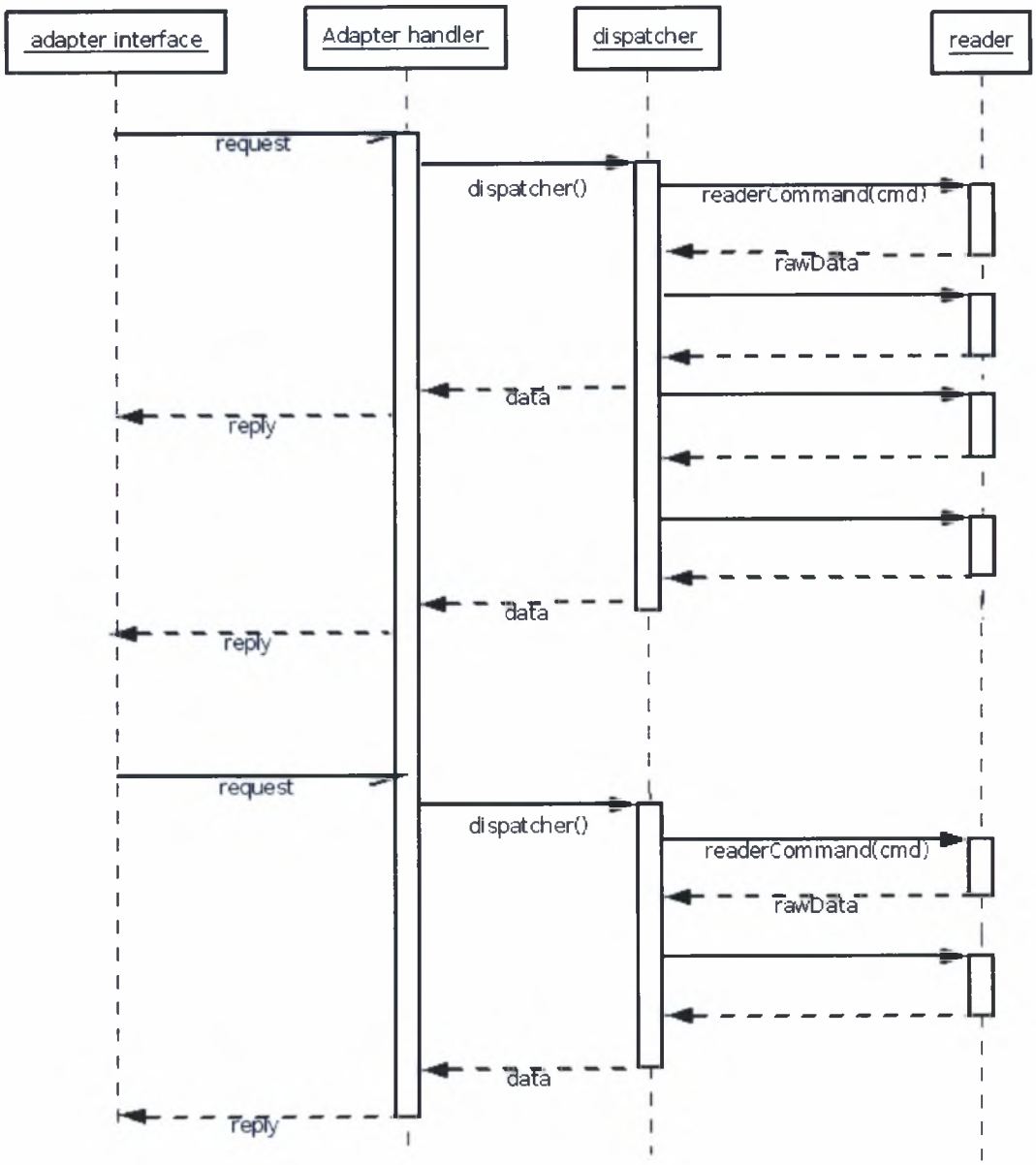
Η τυπική ροή των δεδομένων είναι η εξής. Η κύρια διεπαφή προσαρμογέα στέλνει ένα αίτημα στον σαρωτή. Η διεργασία που διαχειρίζεται το άρθρωμα της επικοινωνίας παραλαμβάνει το αίτημα και το καταχωρεί σε μια ουρά εισόδου που περιέχει τα ερωτήματα. Στη συνέχεια η διεργασία της διεκπεραίωσης παραλαμβάνει από την ουρά το αίτημα το αποκωδικοποιεί και στη συνέχεια το εκτελεί. Η διεργασία της διεκπεραίωσης αναλαμβάνει αφού εκτελεστεί το αίτημα να τοποθετήσει τα αποτελέσματα στην ουρά εξόδου και να ειδοποιήσει την διεργασία που διαχειρίζεται την επικοινωνία, μέσω ενός γενικού σημαφόρου, ώστε να στείλει τα αποτελέσματα.

Για τις λειτουργίες του αναγνώστη το μόνο που έχουμε είναι η ανάγνωση και εγγραφή των περιοχών της μνήμης των ετικετών καθώς και της διαδικασίας για την αναγνώριση

των ετικετών που είναι στην περιοχή του. Οι εντολές του αναγνώστη είναι σε ASCII μορφή και το σύνολο αυτών που χρησιμοποιούνται δίνεται στον πίνακα του παραρτήματος Β. Οι υπόλοιπες EPC εντολές εκτελούνται στο ενσωματωμένο σύστημα του σαρωτή και εγκαθίστανται από την διεργασία διεκπεραίωσης. Στο Διάγραμμα 3-5 που ακολουθεί φαίνεται αναλυτικά η λειτουργία και ο κύκλος ζωής του διεκπεραιωτή



Διάγραμμα 3-5. Διάγραμμα διεργασιών του διεκπεραιωτή



Σχήμα 3-5: Διάγραμμα εκτέλεσης εντολών στο σαρωτή

4 Υλοποίηση

Σε αυτό το κεφάλαιο θα επικεντρωθούμε στην διαδικασία υλοποίησης της παραπάνω αρχιτεκτονικής. Αρχικά θα παρουσιαστούν τα στάδια της ανάπτυξης και η πλατφόρμα υλοποίησης. Στη συνέχεια θα δοθούν οι υποστηριζόμενες εντολές και τα πρωτόκολλα που υλοποιήθηκαν.

4.1 Διαδικασία ανάπτυξης

Η διαδικασία υλοποίησης της παραπάνω αρχιτεκτονικής έγινε σε δύο βασικά στάδια. Το πρώτο από αυτά περιλάμβανε την επιλογή των εξαρτημάτων για την κατασκευή του πρωτοτύπου απομακρυσμένου αναγνώστη, την συναρμολόγηση του καθώς και την υλοποίηση μιας βασικής λειτουργικότητας. Η βασική λειτουργικότητα που αναπτύχθηκε στην πρώτη φάση εκτός από την υλοποίηση του κύριου κορμού της αρχιτεκτονικής συμπεριλάμβανε την ανάπτυξη δυο προσαρμογέων για την υποστήριξη σειριακών και dial-up συνδέσεων καθώς και την υποστήριξη βασικών εντολών ανάγνωσης με βάση την περιγραφή του EPC-RP πρωτοκόλλου.

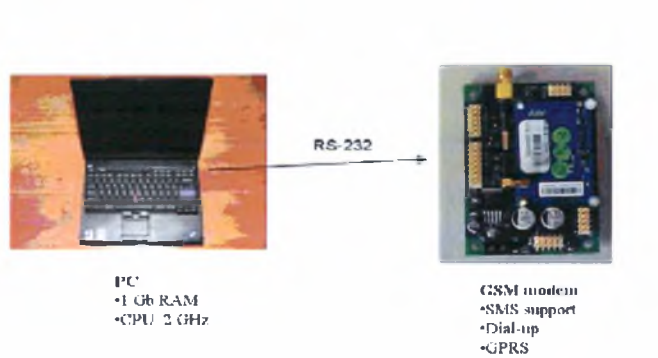
Στο δεύτερο στάδιο ανάπτυξης υποστηρίχθηκαν περισσότερες εντολές του EPC-RP πρωτοκόλλου στον απομακρυσμένο αναγνώστη αλλά υλοποιήθηκε και ένας

προσαρμογέας για την ανταλλαγή των αιτημάτων / απαντήσεων χρησιμοποιώντας ως μέσω την ανταλλαγή μηνυμάτων SMS.

4.2 Πλατφόρμα ανάπτυξης

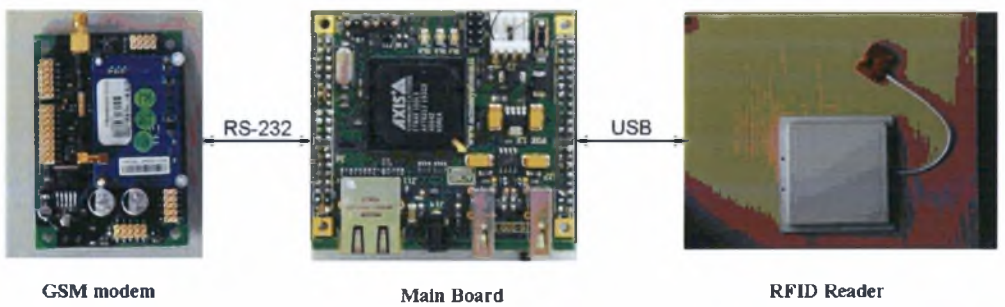
Η διαδικασία τις επιλογής των εξαρτημάτων για την κατασκευή του πρωτοτύπου του απομακρυσμένου αναγνώστη βασίστηκε σε δύο παράλληλους άξονες κριτηρίων. Ο πρώτος άξονας περιείχε το κόστος, το οποίο θα έπρεπε να ήταν σχετικά μικρό, αλλά και την επιθυμία για περιορισμένες δυνατότητες σε υπολογιστική ισχύ και μνήμη έτσι ώστε ή υλοποίηση να αποδεικνύει την χρησιμότητα της προτεινόμενης αρχιτεκτονικής. Ο δεύτερος άξονας έθετε σαν κριτήριο την ευκολία ανάπτυξης εφαρμογών στην πλατφόρμα καθώς και την δυνατότητα για περαιτέρω επεκτάσεις.

Η κύρια διεπαφή προσαρμογέα είναι υλοποιημένη σε γλώσσα C σε ένα τυπικό σύστημα PC αρχιτεκτονικής x86 με εγκατεστημένο το λειτουργικό σύστημα Linux. Η επιλογή αυτή έγινε λόγω του ότι η κύρια διεπαφή προσαρμογέα έχει αρκετές απαιτήσεις σε υπολογιστική ισχύ, αλλά και μνήμη. Το σύστημα διαθέτει σειριακές θύρες και δια μέσου αυτών είναι συνδεδεμένο ένα GSM/GPRS modem που χρησιμοποιείται για την επικοινωνία της κύριας διεπαφής προσαρμογέα με τους σαρωτές.



Εικόνα 4-1

Το σύστημα του απομακρυσμένου αναγνώστη αποτελείται από τρία μέρη. Η καρδιά του συστήματος είναι μια επεξεργαστική μονάδα στην οποία είναι συνδεδεμένα ένα GSM/GPRS modem και ένας RFID αναγνώστης με την κεραία του. Η κύρια επεξεργαστική μονάδα είναι ένα ενσωματωμένο σύστημα το οποίο είναι εφοδιασμένο με ένα επεξεργαστή Axis 100MHz με 16 MB RAM στην οποία τρέχει το λειτουργικό σύστημα Linux. Το σύστημα αυτό διαθέτει 3 σειριακές θύρες, 2 θύρες USB host καθώς και 30 ψηφιακές εισόδους / εξόδους γενικής χρήσης. Το GSM/GPRS modem είναι συνδεδεμένο σειριακά με το κεντρικό σύστημα ενώ ο RFID αναγνώστης είναι συνδεδεμένος μέσω της θύρας USB.

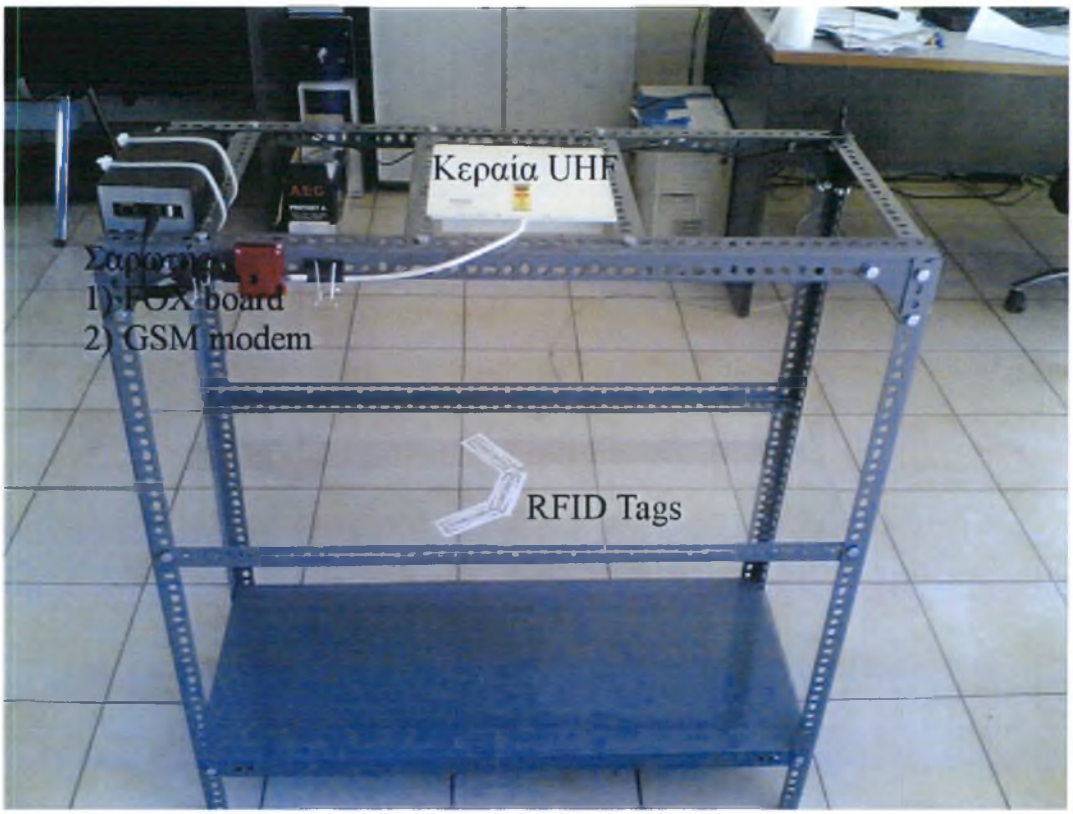


Εικόνα 4-2

Η επιλογή αυτού του συστήματος έγινε με βάση το χαμηλό κόστος του αλλά και τις ευελιξίας που παρέχει στην ανάπτυξη εφαρμογών. Επιπρόσθετα είναι αρκετά επεκτάσιμο αφού διαθέτει αρκετές θύρες και είναι σε μικρό μέγεθος. Το όλο σύστημα τροφοδοτείται με τάση 5V και η κατανάλωση του χωρίς τα περιφερειακά είναι 200mA.

Ο αναγνώστης RFID που χρησιμοποιούμε λειτουργεί στην συχνότητα των 815 MHz και χρησιμοποιεί το EPC Gen 2 πρότυπο για την επικοινωνία με τις ραδιοετικέτες. Οι εντολές

που υποστηρίζει είναι ανάγνωσης και εγγραφής των ραδιοετικετών. Το πρωτόκολλο επικοινωνίας με τον αναγνώστη είναι ανοιχτό από τον κατασκευαστή και είναι ASCII. Στο παράρτημα Γ παρατίθεται το σύνολο των εντολών του αναγνώστη.



Εικόνα 4-3: Ο σαρωτής εγκατεστημένος σε φορητό πλαίσιο

4.3 Λειτουργικότητα

Η λειτουργικότητα που υποστηρίζεται μέσω της αρχιτεκτονικής που παρουσιάστηκε είναι ένα υποσύνολο του EPC-RP πρωτοκόλλου. Η εντολές που είναι υλοποιημένες αποτελούν την βάση για την κατασκευή ολόκληρου του πρωτοκόλλου με κατάλληλο συνδυασμό τους.

4.3.1 Υποστηριζόμενες εντολές

Οι υποστηριζόμενες εντολές που φαίνονται στον Πίνακας 4-1 είναι υπό την μορφή μεθόδων EPC αντικειμένων που περιγράφει το EPC-RP πρότυπο.

	Source
1	`SelectReadPoints(readPoints: ReadPoint[]): void`
2	`deSelectReadPoints(readPoints: ReadPoint[]): void`
3	`addTagSelectors(selectors: TagSelector[]): void`
4	`removeTagSelectors(selectors: TagSelector[]): void`
5	`getAllTagSelectors(void): TagSelector[]`
6	`read(dataSelector: MT_DataSelector, selectors: TagSelector[], triggers: Trigger[]): MT_ReadReport`
7	`setReadCyclesPerTrigger(cycles : integer): void`
8	`setMaxReadDutyCycle(dutyCycle:integer): void`
9	`setReadTimeout(timeout: integer): void`
10	`setSession(session: integer): void`
11	`MT_addTagFields(tagfields: TagField[]): void`
12	`MT_removeTagFields(tagfields: TagField[]): void`
13	`MT_getAllTagFields(void): TagField[]`

Πίνακας 4-1: Πίνακας εντολών EPC-RP

Οι μέθοδοι αυτές με βάση την λειτουργία τους είναι περιγεγραμένες με την μορφή XML μηνυμάτων. Με βάση αυτή την περιγραφή παράγονται τα μηνύματα των εφαρμογών-πελάτη προς την κύρια διεπαφή προσαρμογέα και κατ' επέκταση προς τους σαρωτές. Τυπικά παράδειγμα αυτών των ερωτημάτων υπάρχουν στο παράρτημα Γ. Η σημασία των εντολών αυτών περιγράφεται παρακάτω.

`SelectReadPoints:` Σε αυτή την εντολή καθορίζονται οι κεραίες που θα ενεργοποιηθούν για την ανάγνωση. Η εντολή αυτή είναι σύγχρονη.

`deSelectReadPoints:` Σε αυτή την εντολή απομακρύνονται οι κεραίες που έχουν επιλεγεί για την ανάγνωση. Η εντολή αυτή είναι σύγχρονη.

addTagSelectors: Με αυτή την εντολή καθορίζεται ένα πρωταρχικό φίλτρο για την επιστροφή των αποτελεσμάτων από την ανάγνωση των ραδιοετικετών. Η εντολή αυτή είναι σύγχρονη.

removeTagSelectors: Αυτή η εντολή καθαρίζει τα φίλτρα που έχουν τεθεί στην ανάγνωση. Η εντολή αυτή είναι σύγχρονη.

getAllTagSelectors: Με αυτή την εντολή επιστρέφονται τα φίλτρα που έχουν καθοριστεί στους σαρωτές υπό της μορφή αναφορών. Η εντολή αυτή είναι σύγχρονη.

read: Διαβάζει και επιστρέφει τις ραδιοετικέτες που βρίσκονται στην ακτίνα ανάγνωσης του αναγνώστη με βάση τα κριτήρια που έχουν τεθεί. Επίσης εγκαθιστά σκανδαλιστές (triggers) που προκαλούν την ανάγνωση αυτόματα ανά τακτά χρονικά διαστήματα. Τα δεδομένα επιστρέφονται στο σύστημα ασύγχρονα

setReadCyclesPerTrigger: Καθορίζει των αριθμό τον κύκλων ανάγνωσης πριν την επιστροφή των αποτελεσμάτων. Η εντολή αυτή είναι σύγχρονη.

setMaxReadDutyCycle: Καθορίζει την χρονική διάρκεια του κάθε κύκλο ανάγνωσης. Η εντολή αυτή είναι σύγχρονη.

setReadTimeout: Καθορίζει το άνω χρονικό όριο για την αναμονή στην απαντήσεις των ετικετών. Η εντολή αυτή είναι σύγχρονη.
MT addTagFields: Θέτει τις περιοχές στην μνήμη των ραδιοετικετών που θα πρέπει να διαβάσει ο αναγνώστης. Η εντολή αυτή είναι σύγχρονη.

MT removeTagFields: Αφαιρεί τις περιγραφές για τις περιοχές μνήμης των ραδιοετικετών. Η εντολή αυτή είναι σύγχρονη.

`MT getAllTagFields:` Επιστρέφει όλες τις περιγραφές για τις περιοχές μνήμης των ραδιοετικετών. Η εντολή αυτή είναι σύγχρονη.

4.3.2 Εσωτερικό πρωτόκολλο

Η διαδικασία της αντιστοίχησης των εντολών EPC σε αυτή της εσωτερικής κωδικοποίησης ορίζεται στον Πίνακας 4-2. Οι εντολές-μέθοδοι του EPC δέχονται περισσότερες από μια παραμέτρους και κάθε μια από αυτές ενδέχεται να μην είναι μονοδιάστατη. Επιπρόσθετα κάθε παράμετρος μπορεί να είναι σύνθετος τύπος δεδομένων με αποτέλεσμα να υπάρχει η ανάγκη να οριστούν συμβάσεις τόσο για το πέρασμα όσο και για την επιστροφή των αποτελεσμάτων. Η προσέγγιση που ακολουθείται και περιγράφεται αναλυτικά στη συνέχεια κάνει χρήση τριών διαχωριστικών. Αρχικά υπάρχει η ανάγκη ενός διαχωριστικού όπου θα διαχωρίσει τις παραμέτρους μεταξύ τους. Το διαχωριστικό των παραμέτρων αναφέρεται στον αναλυτικό πίνακα αντιστοίχησης με το συμβολισμό OSP(0xDD). Επίσης λόγω του ότι οι παράμετροι ενδέχεται να είναι σύνθετες δομές που αποτελούνται από μεταβλητού μήκους πεδία εισάγεται ακόμα ένα διαχωριστικό το PSP(0xEE) για να διαχωρίσει το κάθε πεδίο. Τέλος το SP(0xFF) διαχωρίζει τα στοιχεία ενός πίνακα στην περίπτωση που κάποια από τις παραμέτρους ορίζεται κατ' αυτόν τον τρόπο.

Περιγραφή	Αντιστοίχηση

Obj {
  Type1 A;
  Type2 B;
  Type3 C;
}
	A PSP B PSP C

Obj param1[2];	A PSP B PSP C SP A PSP B PSP C
Cmd(Obj[] param1, Obj param2)	A PSP B PSP C SP A PSP B PSP C ,…,OSP A PSP B PSP C

Πίνακας 4-2: Πίνακας αντιστοίχησης εντολών

4.3.3 Πρωτόκολλο SMS

Το μέσο που επιλέχθηκε να για την επικοινωνία μεταξύ της κύριας διεπαφής προσαρμογέα και σαρωτών είναι η ανταλλαγή μηνυμάτων SMS μέσω του GSM δικτύου. Η επιλογή αυτή έγινε λόγω του χαμηλού κόστους ανά μήνυμα. Για την διεξαγωγή αυτής της επικοινωνίας έχει αναπτυχθεί ένας προσαρμογέας ο οποίος είναι επιφορτισμένος με την συνεπή αποστολή και παραλαβή των μηνυμάτων.

Επίπεδο μεταφοράς: Δεδομένου ότι το μέσο μεταφοράς είναι το SMS για το οποίο η παράδοση των μηνυμάτων από τον αποστολέα στον παραλήπτη στην χειρότερη περίπτωση είναι best effort[1] θα πρέπει να ορίσουμε ένα επίπεδο μεταφοράς. Παρακάτω (Πίνακας 4-3) περιγράφονται τα πλαίσια που μεταφέρουν μηνύματα καθώς και τα πλαίσια που μεταφέρουν τις επιβεβαιώσεις.

[1] Εξαρτάτε από το αν το κέντρο διαχείρισης (SMSC) υποστηρίζει την επανεκπομπή μνημάτων για αξιόπιστη μετάδοση.

msg-bit	msg-id	frag-nr	eom-bit	dlen	data	crc

msg-bit :Δηλώνει ότι το πλαίσιο που ακολουθεί είναι πλαίσιο μηνύματος
msg-id :Αναγνωριστικός αριθμός του μηνύματος της εφαρμογής που στέλνεται.
frag-nr :Αναγνωριστικός αριθμός τμήματος του μηνύματος
eom-bit :Δηλώνει ότι το πλαίσιο αυτό είναι το τελευταίο πλαίσιο του μηνύματος
dlen :Μέγεθος δεδομένων που ακολουθούν
data :Δεδομένα
crc :

Πίνακας 4-3 Πλαίσιο μηνύματος

ack-bit	msg-id	nof-frag-nrs	{frag-nr}	crc

ack-bit :Δηλώνει ότι το πλαίσιο που ακολουθεί είναι πλαίσιο επιβεβαίωσης
msg-id :Αναγνωριστικός αριθμός του μηνύματος της εφαρμογής
nof-frag-nrs :Αριθμός τμημάτων του μηνύματος που επιβεβαιώνονται
frag-nr :Λίστα αριθμών των τμημάτων του μηνύματος που επιβεβαιώνονται
crc :

Πίνακας 4-4 Πλαίσιο επιβεβαιώσεων

Το κάθε μήνυμα σε επίπεδο εφαρμογής έχει ένα αναγνωριστικό αριθμό ο οποίος ανακυκλώνεται. Η σύμβαση για το πως ανανεώνονται τα αναγνωριστικά των μηνυμάτων περιγράφεται αναλυτικά στην σύμβαση αποστολέα παραλήπτη. Το κάθε μήνυμα σε επίπεδο εφαρμογής ενδέχεται να χωρίζεται σε τμήματα λόγω του περιορισμένου όγκου δεδομένων που μπορεί να μεταφέρει ένα SMS. Για το λόγο αυτό σε κάθε πλαίσιο μηνύματος μεταφέρεται ο αριθμός του τμήματος του μηνύματος και ένα bit το οποίο καθορίζει αν το τμήμα του μηνύματος που μεταφέρεται είναι το τελευταίο ή ακολουθούν και άλλα. Τέλος το κάθε πλαίσιο έχει και ένα crc κωδικό για να εντοπίσει πιθανά λάθη που υπήρξαν στην μετάδοση.

Σύμβαση αποστολέα παραλήπτη: Ο αποστολέας έχει τη δυνατότητα να στείλει όλα τα τμήματα ενός μηνύματος χωρίς να περιμένει επιβεβαίωση για το κάθε τμήμα ξεχωριστά. Ο αποστολέας δεσμεύει το αναγνωριστικό του μηνύματος για να μην επαναχρησιμοποιηθεί και δημιουργεί ένα πίνακα με τις επιβεβαιώσεις που περιμένει. Για κάθε επιβεβαίωση που περιμένει ορίζει ένα άνω χρονικό όριο μετά το πέρας του οποίου στέλνει εκ νέου το τμήμα του μηνύματος για το οποίο δεν ήρθε επιβεβαίωση. Στην περίπτωση που κάποια επιβεβαίωση αργήσει ή χαθεί και ο παραλήπτης λάβει ένα τμήμα περισσότερες από μια φορές τότε αγνοεί και στέλνει επιβεβαίωση γι' αυτό. Ο αποστολέας πριν στείλει ένα μήνυμα αναθέτει σε αυτό ένα αναγνωριστικό και ελέγχει τον πίνακα των επιβεβαιώσεων για να δει αν το αναγνωριστικό είναι διαθέσιμο. Στην περίπτωση που το αναγνωριστικό που επέλεξε χρησιμοποιείτε τότε επιλέγει το αμέσως επόμενο και αν δεν υπάρχει κανένα διαθέσιμο τότε καθυστερεί την αποστολή του μηνύματος μέχρι να υπάρξει διαθεσιμότητα αναγνωριστικού.

Επίπεδο Δεδομένων: Παρακάτω καθορίζονται τα μηνύματα που ανταλλάσσονται σε επίπεδο εφαρμογής.

request-bit	type	req-id	body

request-bit : Καθορίζει το είδος του μηνύματος (0 request)
type : καθορίζει την εντολή που πρόκειται να εκτελεστεί
req-id : αναγνωριστικό του μηνύματος
body : Παράμετροι των εντολών

Πίνακας 4-5: Πλαίσιο αιτήματος

reply-bit	type	req-id	status-bit	Reply

reply-bit : Καθορίζει το είδος του μηνύματος (1 reply)
type : καθορίζει την εντολή για την οποία επιστρέφει τα αποτελέσματα
req-id : αναγνωριστικό του μηνύματος
status-bit : 0 reply 1 error
reply :Ανάλογα το status-bit το περιεχόμενο είτε περιέχει τα αποτελέσματα μιας εντολής είτε περιέχει τα λάθη

Πίνακας 4-6: Πλαίσιο απάντησης

Κάθε μήνυμα σε επίπεδο εφαρμογής έχει ένα αναγνωριστικό αριθμό ο οποίος δεν έχει σχέση με το αναγνωριστικό στο επίπεδο μεταφοράς. Ο αναγνωριστικός αριθμός είναι μοναδικός για κάθε μήνυμα και επιλέγεται από την εφαρμογή σε υψηλό επίπεδο. Για τα μηνύματα που ανταλλάσσονται σε επίπεδο εφαρμογής θα πρέπει να υπάρξει ένα σχήμα επιβεβαίωσης. Οι επιβεβαιώσεις σε επίπεδο μεταφοράς δε αρκούν για τα μηνύματα που ανταλλάσσονται σε επίπεδο εφαρμογής και κατά συνέπεια θα πρέπει να υπάρξει ένα σχήμα επιβεβαιώσεων. Οι επιβεβαιώσεις μεταφέρονται μέσα στο μήνυμα της απάντησης και για το λόγω αυτό κάθε εντολή που εκτελείτε επιστρέφει ένα αποτέλεσμα είτε έχει δεδομένα είτε όχι.

5 Συμπεράσματα

Από την υλοποίηση προέκυψαν αρκετά συμπεράσματα για την απόδοση και την διεκπεραιωτική ικανότητα του συστήματος.

Σε ότι αφορά την απόδοση παραθέτουμε το εξής. Ένα τυπικό ερώτημα από την εφαρμογή πελάτη για ανάγνωση χωρίς την χρήση φίλτρων είναι 1500 bytes. Με τον μετασχηματισμό του ερωτήματος αυτού στο εσωτερικό πρωτόκολλο αυτό μειώνεται στα 30 bytes. Από την πλευρά του δικτύου το αρχικό μήνυμα για να μεταδοθεί μέσω του SMS πρωτοκόλλου θα χρειαζόταν 10 μηνύματα SMS εκτός αυτά των επιβεβαιώσεων ενώ το μετασχηματισμένο θα χρειαζόταν μόνο ένα. Επίσης για την ανάγνωση και την αποκωδικοποίηση του XML μηνύματος του πελάτη στο PC απαιτούνται 710μs ενώ για το μεταφρασμένο μήνυμα στο σύστημα του σαρωτή η ίδια διαδικασία απαιτεί 450μs. Δεδομένου ότι το σύστημα του σαρωτή είναι 150 φορές πιο αργό από το κεντρικό σύστημα η συνολική επιτάχυνση είναι περιπου 1,6. Αυτό καθιστά από μόνο του αναγκαία την ύπαρξη του εσωτερικού πρωτοκόλλου. Μια τέτοια ανταλλαγή μηνυμάτων από μια τυπική εφαρμογή βρίσκεται στο παράρτημα Γ.

Για την τυπική απάντηση που περιέχει μόνο τον κωδικό μιας ραδιοετικέτας στο εσωτερικό πρωτόκολλο απαιτούνται 10 bytes για την αναπαράστασή της ενώ για την αναπαράσταση της με XML θα χρειαζόταν 240 bytes. Από την πλευρά του δικτύου για

την πρώτη περίπτωση χρειαζόμαστε 1 μήνυμα ενώ για την δεύτερη περίπτωση 2 μηνύματα.

Η συνολική διεκπεραιωτική ικανότητα του SMS πρωτοκόλλου είναι πάρα πολύ μικρή που χωρίς την χρήση επιβεβαιώσεων αγγίζει το 170 bytes/s ενώ με την πλήρη εφαρμογή του πρωτοκόλλου αυτή πέφτει περίπου στα μισά 80 bytes/s. Με την αντικατάσταση του SMS προσαρμογέα SMS με αυτόν του σειριακού προσαρμογέα η διεκπεραιωτική ικανότητα του μέσου ανεβαίνει στο 1440 bytes/s. Η εφαρμογή του SMS πρωτοκόλλου αποτρέπει την χρήση ερωτημάτων που εγκαθιστούν ερωτήματα τα οποία εγκαθιστούν σκανδαλιστές (triggers) που προκαλούν ανάγνωση και αποστολή με συχνότητες κάτω των 2 δευτερολέπτων. Αυτό όμως δεν καθιστά απαγορευτική την χρήση του από την στιγμή που ο χρόνος απόκρισης στα τυπικά ερωτήματα είναι της τάξεων των λεπτών.

Σε ότι αφορά την κατανάλωση ενέργειας όπως έχει προαναφερθεί είναι στα 200mA αυτό θα μπορούσε να μειωθεί αν στα μηνύματα του εσωτερικού πρωτοκόλλου ενσωματωθούν περιγραφές με τις οποίες ο σαρωτής απενεργοποιεί συσκευές που δεν χρησιμοποιεί.

6 Μελλοντικές επεκτάσεις

Τα στάδια της ανάπτυξης αλλά και η διαδικασία της αξιολόγησης του συστήματος έδωσαν μια εικόνα για μελλοντικές επεκτάσεις. Μια επέκταση η οποία θα είχε άμεσο αποτέλεσμα στην απόδοση της κύριας διεπαφής προτείνεται να είναι η πολυπλεξία όχι μόνο στο επίπεδο των σαρωτών αλλά και των αιτημάτων από τις εφαρμογές – πελάτη. Οι εφαρμογές - πελάτη θα στέλνουν αιτήματα τα οποία η κύρια διεπαφή θα λαμβάνει και θα τα ομαδοποιεί ανάλογα με την εγγύτητά τους. Στην συνέχεια αφού έχει συγκεντρώσει ικανό αριθμό μηνυμάτων θα τα στέλνει στους σαρωτές. Αυτό θα βελτίωνε την απόδοση του συστήματος σε υψηλό φόρτο.

Ακόμα μία προτεινόμενη επέκτασή είναι η υλοποίηση του σαρωτή σε ένα συστήμα χαμηλότερων δυνατοτήτων από αυτό που είναι υλοποιημένος. Παράδειγμα αποτελεί ένας σαρωτής με κεντρική μονάδα επεξεργασίας ένα μικροελεγκτή. Ο μικροελεγκτής θεωρούμε ότι είναι συνδεμένος απ' ευθείας με το κύκλωμα εκπομπής και εκτός του εσωτερικού πρωτοκόλλου υλοποιεί και το πρωτόκολλο επικοινωνίας με τις ραδιοετικέτες. Με αυτή την επέκταση η κατανάλωση ενέργειας στο σύστημα του σαρωτή θα μειωνόταν σε τέτοιο βαθμό ώστε να είναι δυνατή η λειτουργία του αποκλειστικά με μπαταρίες για αρκετά μεγάλο χρονικό διάστημα. Αυτό μπορεί βέβαια να μείωνε τις αποστάσεις ανάγνωσης αλλά το σύστημα θα ήταν αρκετά φθηνό ώστε οι απαιτήσεις της ανάγνωσης μεγάλων περιοχών να εξυπηρετούταν από πολλές τέτοιες συσκευές. Επίσης η χρήση

μπαταριών θα έκανε δυνατή την παρακολούθηση αποθηκών που δεν διαθέτουν σταθερή πηγή τροφοδοσίας. Το σύστημά μας θα μπορούσε να διαχειριστεί χωρίς τροποποιήσεις το πλήθος των σαρωτών μιας τέτοιας επέκτασης.

Επίσης θα ήταν σκόπιμο το κομμάτι τη κύριας διεπαφής προσαρμογέα που υλοποιεί την αποκωδικοποίηση των μηνυμάτων να υλοποιηθεί σε μια διαφορετική ανεξάρτητη υπηρεσία. Με αυτή την επέκταση θα χρειαζόταν μόνο να οριστεί μια νέα διεπαφή με αυτή την υπηρεσία με αποτέλεσμα να είναι εύκολη η υποστήριξη πολλαπλών πρωτοκόλλων πελάτη. Τέλος θα ήταν χρήσιμο να επεκταθούν οι εντολές του σαρωτή έτσι ώστε να υποστηρίζουν και την ύπαρξη αισθητήρων αλλά και ενεργοποιητών πράγμα που θα καθιστούσε την πλατφόρμα αυτή μιας γενικής χρήσης πλατφόρμα αισθητήρων.

Αναφορές

[1] History of RFID http://www.rfidjournal.com/article/view/1338/1/129
[2] Radio-frequency identification http://en.wikipedia.org/wiki/RFID
[3] EPCglobal http://www.epcglobalinc.org/
[4] EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.0.9, EPCglobal inc 2005
[5] Low Level Reader Protocol (LLRP), Version 1.0.1, EPCglobal inc 2007
[6] Reader Protocol Standard, Version 1.1, EPCglobal inc 2006
[7] Allixon URIS. http://www.allixon.com/product_uris.html
[8] C. Floerkemeier and M. Lampe, "RFID Middleware Design – Addressing Application Requirements and RFID Constrains", Joint Conference on Smart Objects and Ambient Intelligence, Grenoble, October 2005.
[9] J.E. Hoag and C.W. Thompson, "Architecting RFID Middle-ware", IEEE Internet Computing, 10(5), September / October 2006
[10] B.S. Prabhu, X. Su, C. Qiu, H. Ramamurthy, P. Chu, R. Gadh, "WinRFID-Middleware for Distributed RFID Infrastructure", International Workshop on RFID Radio Identification and Wireless Sensors, Kanpur, India, November 2005.
[11] Sun Java System RFID Software Architecture, technical white paper, Sun Microsystems, 2005. Published online at www.sun.com/software/solutions/rfid/EPCNetArch_wp.pdf
[12] TagCentric System. http://tag-centric.sourceforge.net
[13] S.F. Wamba, L.A. Lefebvre, E. Lefebre, "Enabling Intelligent B-to-B eCommerce Supply Chain Management using RFID and the EPC Network: A Case Study in the Retail Industry", International Conference on Electronic Commerce, Fredericton, Canada, August 2006.

Παράρτημα-Α

EPC command	type	Parameters

SelectReadPoints	0x10	Request
		body = name SP name SP ...
		name = BYTE //προκαθορισμένο απο πριν
		SP = 0xFF
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		err = name SP name SP …
		name = BYTE /*επιστρέφει τους κωδικούς που δεν αντιστοιχούν σε ReadPoints */
		
		SP = 0xFF

EPC command	type	Parameters

deSelectReadPoints	0x11	Request
		body = name SP name SP ...
		name = BYTE //προκαθορισμένο απο πριν
		SP = 0xFF
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		err = name SP name SP …
		name = BYTE /*επιστρέφει τους κωδικούς που δεν αντιστοιχούν σε ReadPoints */
		SP = 0xFF

EPC command	type	Parameters

addTagSelectors	0x12	Request
		Body = name PSP tfName PSP value PSP mask PSP inclusiveFlag SP
		name = ASCII
		tfName = ASCII
		value = όπως καθορίζεται στο EPC-RP
		mask = όπως καθορίζεται στο EPC-RP
		inclusiveFlag = όπως καθορίζεται στο EPC-RP
		PSP=0xEE
		SP=0xFF
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

EPC command	type	Parameters
removeTagSelectors	0x13	Request body = name SP name SP ... Name = ASCII SP = 0xFF Reply body = 0x20 0x20 //χωρίς λάθος Err err = name SP name SP … name = BYTE /*επιστρέφει τους κωδικούς που δεν αντιστοιχούν σε TagSelectors*/ SP = 0xFF

EPC command	type	Parameters
getAllTagSelectors	0x14	Request Body = 0x00 Reply body = name SP name SP ... Name = ASCII SP = 0xFF Err body = 0xFF 0xFF 0xFF 0xFF //απροσδιόριστο

EPC command	type	Parameters

read	0x15	Request
		body = Tn PSP Tf PSP Tl PSP TagField OSP TagSelector OSP TriggerType OSP TriggerValue
		Tn = INTEGER
		Tf = INTEGER
		Tl = INTEGER
		TagField = ASCII //αναφορά με όνομα
		TagSelector = ASCII /*αναφορά με
		
		όνομα */
		TriggerType = BYTE
		TriggerValue = INTEGER
		PSP = 0xEE
		OSP = 0xDD
		PS = 0xFF
		
		Reply
		body = Tn PSP Tf PSP Tl PSP DATA SP …
		Tn = INTEGER
		Tf = INTEGER
		Tl = INTEGER
		DATA{} = BYTE list /*το περιεχόμενο
		του
		TagField */
		PSP = 0xEE
		SP = 0xFF
		
		Err
		err = name PS name …
		name = ASCII
		SP = 0xFF

EPC command	type	Parameters
		/*επιστρέφει μόνο τα ονόματα των μεταβλητών που δεν υπάρχουν */

EPC command	type	Parameters

setReadCyclesPerTrigger	0x17	Request
		body = NUM
		NUM = INTEGER
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

EPC command	type	Parameters

setMaxReadDutyCycle	0x18	Request
		body = NUM
		NUM = INTEGER
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

EPC command	type	Parameters

setReadTimeout	0x19	Request
		body = NUM
		NUM = INTEGER
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

EPC command	type	Parameters

setSession	0x1a	Request
		body = NUM
		NUM = INTEGER
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

EPC command	type	Parameters

MT_addTagFields	0x1b	Request
		body = name PSP TagFieldName PSP MemoryBank PSP Offset PSP Length SP …
		Name = ASCII
		TagFieldName = vendor dep.
		MemoryBank = vendor dep.
		Offset = vendor dep.
		Length = vedor dep.
		PSP = 0xEE
		SP = 0xFF
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

EPC command	type	Parameters
MT_removeTagFields	0x1c	Request body = name SP name SP …. Name = ASCII SP = 0xFF Reply body = 0x20 0x20 //χωρίς λάθος Err body = name SP name SP …. /*επιστρέφει μόνο τα ονόματα των μεταβλητών που δεν υπάρχουν */

EPC command	type	Parameters
MT_getAllTagFields	0x1d	Request body = 0x00 Reply body = name SP name SP …. Name = ASCII SP = 0xFF Err body = 0xFF 0xFF 0xFF 0xFF //απροσδιόριστο

EPC command	type	Parameters

register_to_host	0x1c	Request
		body = UniqueID SP Address
		UniqueID = ASCII
		Address = ASCII
		SP = 0xFF
		
		Reply
		body = 0x20 0x20
		//χωρίς λάθος
		
		Err
		body = 0xFF 0xFF 0xFF 0xFF
		//απροσδιόριστο

Παράρτημα-Β

Εντολή	Περιγραφή
k	ενεργοποιεί το RF του αναγνωστή και προκαλεί συνεχόμενη σάρωση
K	Σταματά την συνεχόμενη σάρωση και απενεργοποιεί το RF
Z	Ο αναγνώστης εκτελεί μια και μοναδική λειτουργία για ανάγνωση των ετικετών που είναι στην περιοχή ανάγνωσης
W	εγγράφει το 12 byte ID στην EPC ετικέτα χωρίς να την κλειδώνει
Y	καθορίζει τον αριθμό των γύρων αποφυγής συγκρούσεων που επιχειρούνται σε κάθε εντολή ανάγνωσης
u	διαβάζει μια συγκεκριμένη περιοχή της μνήμης της ετικέτας
U	Εγγράφει μια συγκεκριμένη περιοχή της μνήμης της ετικέτας

Παράρτημα-Γ

XML αίτημα εφαρμογής-πελάτη

```
<?xml version="1.0" encoding="UTF-8" ?>
<command>
<id>1</id>
<targetName>ReaderDevice</targetName>
<objectType>
<objectTypeName>Trigger</objectTypeName>
<objectTypeCommand>
<commandName>create</commandName>
<parameter>
<parameterName>Name</parameterName>
<parameterValue>notificationTrigger</parameterValue>
</parameter>
<parameter>
<parameterName>Type</parameterName>
<parameterValue>tipota</parameterValue>
</parameter>
<parameter>
<parameterName>Value</parameterName>
<parameterValue>12</parameterValue>
</parameter>
</objectTypeCommand>
</objectType>
<objectType>
<objectTypeName>NotificationChannel</objectTypeName>
<objectTypeCommand>
<commandName>addNotificationTriggers</commandName>
<parameter>
<parameterName>triggers</parameterName>
<parameterValue>notificationTrigger</parameterValue>
</parameter>
</objectTypeCommand>
</objectType>
<objectType>
<objectTypeName>Trigger</objectTypeName>
<objectTypeCommand>
<commandName>create</commandName>
<parameter>
  <parameterName>Name</parameterName>
  <parameterValue>readTrigger</parameterValue>
```

```
  </parameter>
<parameter>
<parameterName>Type</parameterName>
<parameterValue>timer</parameterValue>
</parameter>
<parameter>
<parameterName>Value</parameterName>
<parameterValue>10</parameterValue>
</parameter>
</objectTypeCommand>
</objectType>
<objectType>
<objectTypeName>Source</objectTypeName>
<objectTypeCommand>
<commandName>setReadCyclesPerTrigger</commandName>
<parameter>
<parameterName>cycles</parameterName>
<parameterValue>5</parameterValue>
</parameter>
</objectTypeCommand>
<objectTypeCommand>
<commandName>addReadTriggers</commandName>
<parameter>
<parameterName>triggers</parameterName>
<parameterValue>readTrigger</parameterValue>
</parameter>
</objectTypeCommand>
</objectType>
</command>
```

XML απάντηση αιτήματος εφαρμογής πελάτη

```
<?xml version="1.0" encoding="UTF-8" ?>
<notification>
<id>1</id>
<readPoint> 1</readPoint>
<readReport>
<tag>
<tagID>1213452542543252545</tagID>
</tag>
</readReport>
<goodBye />
</notification>
```

Εσωτερική αντιστοίχηση αιτήματος εφαρμογής πελάτη

1PSP0PSP1PSP0OSP0OSOSP10

Εσωτερική αντιστοίχηση απάντησης σαρωτή

1PSP0PSP0PSP12134525425432525455P