ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΙΤΛΟΣ ΕΡΓΑΣΙΑΣ:

«Σχεδίαση και κατασκευή access control μέσω δακτυλικού αποτυπώματος αλλά και μέσω κάρτας RF-ID»

ΦΟΙΤΗΤΕΣ:

ΚΟΥΤΣΟΥΠΑΚΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΚΑΣ: 513094

ΛΟΛΑΣ ΒΑΓΓΕΛΗΣ

ΚΑΣ: 513108

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΓΙΑΚΟΥΜΗΣ ΑΓΓΕΛΟΣ

ΘΕΣΣΑΛΟΝΙΚΗ, Σεπτέμβριος 2020

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον επιβλέπων καθηγητή κ. Άγγελο Γιακουμή για τη συνεργασία και την πολύτιμη συμβολή του, στην ολοκλήρωση της πτυχιακής εργασίας.

Περίληψη

Η πτυχιακή εργασία, όπως διακρίνεται από τον τίτλο της, πραγματεύεται την υλοποίηση και τη σχεδίαση κυκλώματος Access Control, όπου θα δίνεται η δυνατότητα ελέγχου πρόσβασης του χρήστη σε χώρους ασφαλείας ή σε συστήματα προστασίας δεδομένων-εγγράφων, μέσω δύο διαδεδομένων συστημάτων ταυτοποίησης. Τα συστήματα ταυτοποίησης που θα χρησιμοποιηθούν ανήκουν στη κατηγορία των RFID ετικετών (NFC / RFID - MFRC522 13.56MHz) και της ταυτοποίησης μέσω αισθητήρα δακτυλικού αποτυπώματος (Adafruit Optical Fingerprint Sensor). Τα πρώτα βήματα που ακολουθήσαμε για να μπορέσουμε να επιτύχουμε τον παραπάνω σκοπό, αφορούσαν τις έρευνες που πραγματοποιήσαμε πάνω στους αισθητήρες των δακτυλικών αποτυπωμάτων και στην RFID τεχνολογία. Έπειτα, αφού κατανοήσαμε και αναλύσαμε την λειτουργία των δύο τεχνολογιών, σειρά είχε η έρευνα και η επιλογή του κατάλληλου μικροελεγκτή (ATmega328P), ο οποίος αποτέλεσε τον <<εγκέφαλο>> του κυκλώματος στη συγκεκριμένη εργασία, καθώς συνδύασε τις δύο παραπάνω τεχνολογίες με τον πλέον αποδοτικό τρόπο λειτουργίας. Στο σημείο αυτό, από τη στιγμή που συλλέξαμε όλες τις απαραίτητες μονάδες της εργασίας, επικεντρωθήκαμε στην σωστή κατασκευή μιας λειτουργικής πλακέτας (hardware), μετέπειτα στον προγραμματισμό του μικροελεγκτή, επιλέγοντας το κατάλληλο πρόγραμμα προγραμματισμού (Arduino IDE), όπως και της κατάλληλης γλώσσας προγραμματισμού (γλώσσα C). Τέλος, στη πτυχιακή εργασία δεν θα μπορούσαν να παραλείπονται κάποιες από τις παρατηρήσεις που σχηματίστηκαν κατά την υλοποίηση του κυκλώματος, και σκέψεις σχετικά με μελλοντικές βελτιώσεις ή παραλλαγές της εργασίας, καθώς και τα συμπεράσματα που προέκυψαν με το πέρας αυτής.

Abstract

The thesis, as defined by the title, researches the design and development of an Access Control circuit, that enables users with access control capabilities into security databases or data-files security systems, through two different identification protocols. The identification systems utilized, belong into the category of RFID tag systems identification sensors (Adafruit Optical Fingerprint Sensor). First steps taken for the fulfilment of the project, were about the research of fingerprint identification sensors and RFID technologies. After understanding the functionality of both technologies, we proceeded by researching and selecting a suitable microcontroller for the task (Atmega328P), which utilized both technologies simultaneously in an efficient manner. After gathering and processing the required data, we focused on the construction of a functional circuit board and the programming of the microcontroller, while using a suitable programming environment (Arduino IDE) and programming language (C language) . Finally, we list a number of observations throughout tha development of the project and thoughts on future work or alteration of it, as well as our conclusions on its results.

Κατάλογος περιεχομένων

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Εισαγωγή

Στην εποχή μας, οι ραγδαίοι ρυθμοί ανάπτυξης και εξέλιξης της τεχνολογίας, έχουν επηρεάσει σημαντικά το τρόπο ζωής των ανθρώπων. Πλέον, οι ανάγκες τους και η καθημερινότητά τους εξαρτώνται απόλυτα από την τεχνολογία και δη περισσότερο από το ψηφιακό κόσμο. Με τη ψηφιοποίηση και καταγραφή (προσωπικών δεδομένων, πληροφοριών, προϊόντων κτλ.), διευκολύνεται και βελτιώνεται σε μεγάλο βαθμό η καθημερινότητα τους, όμως παράλληλα αυξάνεται η ανάγκη για καλύτερη προστασία αυτών.

Η πτυχιακή εργασία που επιλέξαμε, σχετίζεται με την έρευνα και τη κατασκευή συστήματος Access control με τη χρήση αισθητήρα δακτυλικού αποτυπώματος όπως και τη χρήση RF-ID Card.

Στο πρώτο κεφάλαιο της εργασίας, παρουσιάζουμε τον λεγόμενο "εγκέφαλο" του συστήματος, σύμφωνα με τον οποίο καταφέραμε να συνδυάσουμε τις δυο τεχνολογίες ταυτοποίησης αλλά και να προγραμματίσουμε τη λειτουργία του. Περιγράφουμε αναλυτικά τη λειτουργία του μικροελεγκτή ATmega328P , όπως και τη λειτουργία των διατάξεων που βρίσκονται ενσωματωμένοι σε αυτόν.

Στο επόμενο κεφάλαιο, αρχικά δίνουμε τον ορισμό της RFID τεχνολογίας και αναλύουμε τον τρόπο λειτουργίας της. Στη συνέχεια αναλύουμε τα διάφορα μέρη του RFID συστήματος και μετέπειτα αναφέρουμε τις κατηγορίες και τα είδη που προκύπτουν. Τέλος, συγκρίνουμε την RFID τεχνολογία με το BAR-CODE και αναφέρουμε κάποιες από τις εφαρμογές της.

Στο τρίτο κεφάλαιο, συναντούμε τη δεύτερη τεχνολογία ταυτοποίησης ,τον αισθητήρα δακτυλικού αποτυπώματος. Όπως και στο δεύτερο κεφάλαιο, αναλύουμε αρχικά τον όρο του δακτυλικού αποτυπώματος και στη συνέχεια περιγράφουμε τη λειτουργία του αισθήτηρα. Τέλος, κατηγοριοποιούμε τα είδη του αισθητήρα και αναφερόμαστε στον αισθητήρα που χρησιμοποιήσαμε στην εργασία.

Στα υπόλοιπο κεφάλαια της εργασίας, περιγράφουμε αναλυτικά, τον τρόπο σχεδίασης του κυκλώματος, την κατασκευή της πλακέτας, όπως και τη λειτουργία του κυκλώματος, ενώ τέλος σχολιάζουμε τα πιο σημαντικά σημεία του κώδικα.

Κεφάλαιο 1ο

Μικροελεγκτής ATmega 328P
1.1. Ιστορική αναδρομή

Οι AVR μικροελεγκτές (microcontrollers), αποτελούν προϊόν της εταιρίας Atmel Corporation, η οποία ιδρύθηκε το 1984, από τον George Perlegos. Το όνομα << Atmel >> της εταιρίας, συμβολίζει τα αρχικά γράμματα της πρότασης << Advanced Technology for Memory and Logic >>. Όπως γίνεται αντιληπτό, η Atmel αποτέλεσε εταιρία παραγωγής ηλεκτρονικών στοιχείων, που στόχευε στην ανάπτυξη και εξέλιξη ενσωματωμένων συστημάτων σε μικροελεγκτές. Για την εποχή εκείνη, θα μπορούσαμε να πούμε, πως η Atmel, παρουσίασε για πρώτη φορά στην αγορά, ηλεκτρονικές διατάξεις χαμηλής κατανάλωσεις, σε σχέση με τα προϊόντα άλλων εταιρίων. Παρά την ίδρυσή της στα μέσα της δεκαετίας του '80, η Atmel άρχισε να δραστηριοποιείται στον χώρο των μικροελεκτών δέκα χρόνια αργότερα. Το 1996, παρουσίασε την πρώτη οικεγένεια μικροελεγκτών βασισμένη στην αρχιτεκτονική Harvard, με την ονομασία AVR. Αν και δεν υπάρχει επίσημη εξήγηση για το τι συμβολίζουν τα αρχικά AVR, είναι κοινώς αποδεκτή στην επιστημονική κοινότητα, η αντίληψη ότι τα αρχικά προκύπτουν από το Alf and Vegard's Risc processor. Ο Alf-Egil Bogen και ο Vegard Wollan, ήταν δύο φοιτητές του Νορβηγικού Ινστιτούτου Τεχνολογίας (Norwegian Institute of Technology), οι οποίοι αποτελούσαν τους εφευρέτες των μικροελεκγτών AVR.
Οι πρώτοι μικροελεγκτές AVR, των 8 bit έκαναν την εμφάνισή τους το 1997, όπου αξίζει να σημειωθεί, ότι μέχρι το 2003 πουλήθηκαν κοντά στα 500 εκατομμύρια μικροελεγκτές.

Οι κυριότερες οικογένειες μικροελεγκτών AVR είναι οι εξής:

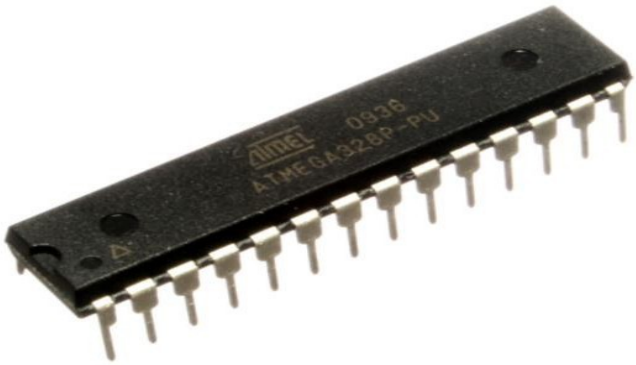
- tinyAVR
- megaAVR
- XMEGA
- Application-specific AVR
- FPSLIC
- 32-bit AVR

Το 2008, η εταιρία Microchip Technology, προσέφερε το ποσό των 2.3 εκατομμυρίων, προσπαθώντας να εξαγοράσει την Atmel. Η πρόταση απορρίφθηκε από την δεύτερη, ωστόσο το 2016 σε μία ακόμη προσπάθεια να αγοράσει τα δικαιώματα της Atmel, προσέφερε το ποσό των 3.6 εκατομμυρίων, γεγονώς που οδήγησε στην ενσωμάτωση της Atmel στην εταιρία Microchip Technology.

1.2. Γενικές πληροφορίες

Ο AVR Atmega328p αποτελεί μέλος της οικογένειας των 8 – bit μικροελεγκτών της κατασκευαστικής εταιρίας Atmel, όπου λειτουργεί βάση της βελτιωμένης αρχιτεκτονικής RISC. Είναι υψηλής απόδοσης μικροεπεξεργαστής με χαμηλή κατανάλωση ενέργειας. Για την εκτέλεση εντολών σε έναν απλό κύκλο ρολογιού, η διαμεταγωγή που επιτυγχάνει ο Atmega328p φτάνει το 1 MIPS ανά MHz.

Ένας μικροελεγκτής AVR Atmega328p περιλαμβάνει 32KB ISP (In- System Programming) Flash μνήμη προγράμματος με τη δυνατότητα read – while – write (RWW) προγραμματισμού, 1024B EEPROM μνήμη, 2KB SRAM μνήμη, 23 γραμμές εισόδου-εξόδου (I/O) γενικής χρήσεως, 32 γενικούς καταχωρητές, 3 Timer / Counters με λειτουργίες σύγκρισης και PWM, εσωτερικούς και εξωτερικούς διακόπτες, ένα κύκλωμα USART (Universal Synchronous and Asynchronous receiver-transmitter), μία σειριακή διεπαφή προσανατολισμένη σε byte δύο καλωδίων, σειριακή θύρα SPI, προγραμματιζόμενος χρονοδιακόπτης παρακολούθησης με εσωτερικό ταλαντωτή, έναν 6κάναλο μετατροπέα σημάτων ADC των 10 bits και πέντε λειτουργίες εξοικονόμησης ενέργειας [1].



Εικόνα 1.1: Μικροελεγκτής ATmega328P.

1.3. Ανάλυση μικροελεγκτή ATmega328P

Ο μικροελεγκτής (microcontroller) αποτελεί ένα μικρό υπολογιστικό σύστημα, χαμηλού κόστους, το οποίο χρησιμοποιείται για την εκτέλεση ειδικών καθηκόντων. Συνήθως καλείται ως ενσωματωμένος ελεγκτής (embedded controller), καθώς ενσωματώνεται στη συσκευή για να μπορεί να ελέγχει τα χαρακτηριστικά ή τις ενέργειες του προϊόντος. Τα κύρια στοιχεία από τα οποία αποτελείται ένας μικροελεγκτής είναι ο επεξεργαστής (CPU), οι μνήμες (RAM, ROM, EEPROM), σειριακές θύρες, όπως και οι χρονιστές-μετρητές.
Οι μικροελεγκτές διαχωρίζονται σε διάφορες κατηγορίες ανάλογα την μνήμη, την αρχιτεκτονική, τη διαμόρφωση των bits και το σύστημα εντολών [1].

- Μνήμη

Βάση των διαφορών στις μνήμες, ο μικροελεγκτής διαχωρίζεται σε δύο κύριες κατηγορίες:

1) Εξωτερική μνήμη μικροελεγκτή: Αυτού του είδους ελεγκτών δεν περιλαμβάνει ενσωματωμένη μνήμη στο τσιπ. Σε αυτό το είδος μικροελεγκτή ανήκει το μοντέλο Intel 8031.

2) Ενσωματωμένη μνήμη μικροελεγκτή: Σε αυτή τη περίπτωση ο μικροελεγκτής περιλαμβάνει όλα τα προαναφερόμενα στοιχεία. Ο μικροελεγκτής Intel 8051 ανήκει σε αυτή τη κατηγορία.

- Bits

Βάση της διαφορετικής διαμόρφωσης των bit, διαχωρίζουμε επιπλέον τον μικροελεγκτή σε τρείς κατηγορίες:

1) 8-bit μικροελεγκτής: Αυτό το είδος μικροελεγκτών χρησιμοποιείται για την πραγματοποίηση αριθμητικών και λογικών υπολογισμών, όπως πρόσθεση, αφαίρεση, διαίρεση και πολλαπλασιασμό. Τα μοντέλα Intel 8031 και 8051 ανήκουν σε αυτή τη κατηγορία.
2) 16-bit μικροελεγκτής: Πραγματοποιεί την ίδια λειτουργία με τον 8-bit ελεγκτή, με μοναδική διαφορά τη μεγαλύτερη ακρίβεια και απόδοσή του. Το μοντέλο Intel 8096 ανήκει σε αυτή τη καηγορία.
3) 32-bit μικροελεγκτής: Χρησιμοποιείται σε αυτόματα ελεγχόμενες συσκεύες. Μεγαλύτερη απόδοση και ταχύτητα ανάγνωσης.

- Αρχιτεκτονική

Με βάση την αρχιτεκτονική τους, οι μικροελεγκτές διαχωρίζονται σε δύο σημαντικές κατηγορίες τους :
2) Αρχιτεκτονική CISC: Χρησιμοποιείται ως υπολογιστής συνθέτου συνόλου εντολών. Επιτρέπει στον χρήστη να εισάγει μία μοναδική εντολή ως εναλλακτική, σε πολλές απλές εντολές.
3) Αρχιτεκτονική RISC: Χρησιμοποιείται ως υπολογιστής περιορισμένου συνόλου εντολών.

1.4. Τεχνικά Χαρακτηριστικά

Καθίσταται σαφές ότι για να διευκολύνουμε την διαδικασία προγραμματισμού (η οποία περιγράφεται λεπτομερώς σε επόμενο κεφάλαιο), απαιτείται η όσο το δυνατόν καλύτερη κατανόηση της λειτουργίας του μικροελεγκτή, της οποίας βάση αποτελούν τα τεχνικά χαρακτηριστικά του μικροελεγκτή. Ορισμένα από τα ποιό σημαντικά και απαραίτητα τεχνικά χαρακτηριστικά του μικροελεγκτή που χρησιμοποιήσαμε αναφέρονται παρακάτω [2]:

Περιγραφή ακροδεκτών (pins) Atmega 328p

- VCC
  Ψηφιακή παροχή τάσης τροφοδοσίας.

- GND
  Γείωση.

- PC6 (RESET) / PORT C: Χρησιμοποιείται ως ακροδέκτης επαναφοράς. Μπορεί να χρησιμοποιηθεί ως ακροδέκτης I/O μόνο όταν η ασφάλεια RSTDIBL είναι προγραμματισμένη.

- PD0 (RXD) / PORT D: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως είσοδος για τη σειριακή επικοινωνία με τη βαθμίδα USART.

- PD1 (TXD) / PORT D: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως έξοδος της βαθμίδας USART.

- PD2 (INT0) / PORT D: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως εξωτερικό interrupt 0.

- PD3 (INT1) / PORT D: Ψηφιακός ακροδέκτης (PWM), χρησιμοποιείται ως εξωτερικό interrupt 1.

- PD4 (XCK/T0) / PORT D: Ψηφιακός ακροδέκτης χρησιμοποιείται ως εξωτερικός counter εισόδου ή ως USART εξωτερικός χρονομετρητής (clock) I/O.

- PB6 (XTAL1/TOSC1) / PORT B: Χρησιμοποιείται για την σύνδεση του κρυσταλλικού ταλαντωτή.

- PB7 (XTAL2/TOSC2) / PORT B: Χρησιμοποιείται για την σύνδεση του κρυσταλλικού ταλαντωτή.

- PD5 (T1/OC0B) / PORT D: Ψηφιακός ακροδέκτης (PWM), χρησιμοποιείται ως TIMER1.

- PD6 (AIN0/OC0A) / PORT D: Ψηφιακός ακροδέκτης (PWM), χρησιμοποιείται ως θετικός αναλογικός συγκριτής I/P.

- PD7 (AIN1) / PORT D: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως αρνητικός αναλογικός συγκριτής I/P.

- PB0 (ICP1/CLKO) / PORT B: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως counter ή timer πηγής εισόδου.

- PB1 (OCIA) / PORT B: Ψηφιακός ακροδέκτης (PWM), χρησιμοποιείται ως counter ή timer συγκριτής match A.

- PB2 (SS/OC1B) / PORT B: Ψηφιακός ακροδέκτης (PWM), λειτουργεί ως “slave choice”.

- PB3 (MOSI/OC2A) / PORT B: Ψηφιακός ακροδέκτης (PWM), χρησιμοποιείται ως έξοδος για τον “master” και είσοδος για τον “slave”. Όταν ο επεξεργαστής λειτουργεί ως “slave”, λαμβάνει τα δεδομένα μέσω αυτού του ακροδέκτη.

- PB4 (MISO) / PORT B: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως είσοδος για τον “master” και έξοδος για τον “slave”. Όταν ο επεξεργαστής λειτουργεί ως “slave”, στέλνει δεδομένα στον “master” μέσω αυτού του ακροδέκτη.

- PB5 (SCK) / PORT B: Ψηφιακός ακροδέκτης, χρησιμοποιείται ως χρονομετρητής μεταξύ του επεξεργαστή και διαφόρων συστημάτων για την ακριβή μεταφορά δεδομένων.

- AVCC: Τροφοδοσία του εσωτερικού ADC μετατροπέα.

- AREF: Αναλογική τάση αναφοράς για τον ADC μετατροπέα.

- PC0 (ADC0) / PORT C: Αναλογικός ακροδέκτης, χρησιμοποιείται ως ADC είσοδο κανάλι 0.

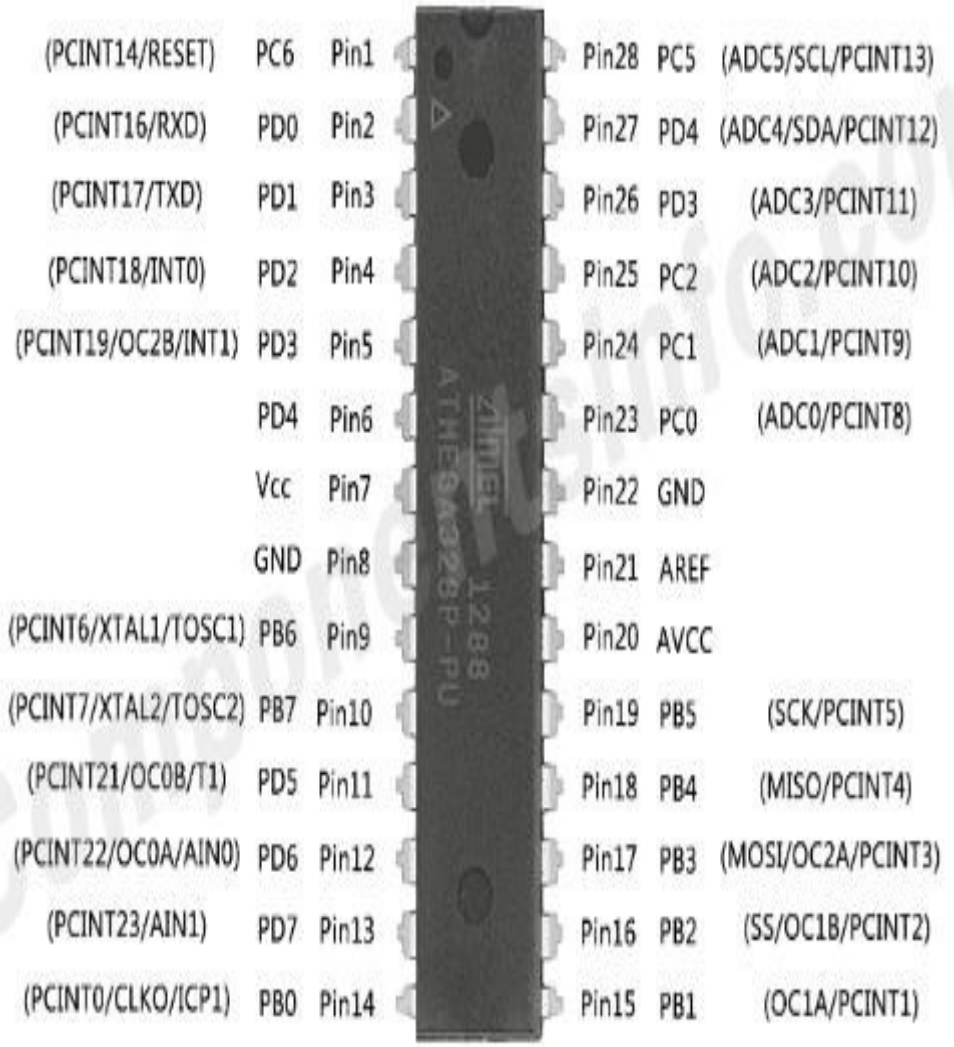
- PC1 (ADC1) / PORT C: Αναλογικός ακροδέκτης, χρησιμοποιείται ως ADC είσοδο κανάλι 1.

- PC2 (ADC2) / PORT C: Αναλογικός ακροδέκτης, χρησιμοποιείται ως ADC είσοδο κανάλι 2.

- PC3 (ADC3) / PORT C: Αναλογικός ακροδέκτης, χρησιμοποιείται ως ADC είσοδο κανάλι 3.

- PC4 (ADC4/SDA) / PORT C: Αναλογικός ακροδέκτης, χρησιμοποιείται ως ADC είσοδο κανάλι 4. Επίσης μπορεί να χρησιμοποιηθεί ως σειριακή διασύνδεση για δεδομένα.

- PC5 (ADC5/SDA) / PORT C: Αναλογικός ακροδέκτης, χρησιμοποιείται ως ADC είσοδο κανάλι 5. Επίσης μπορεί να χρησιμοποιηθεί ως σειριακή διασύνδεση για δεδομένα.



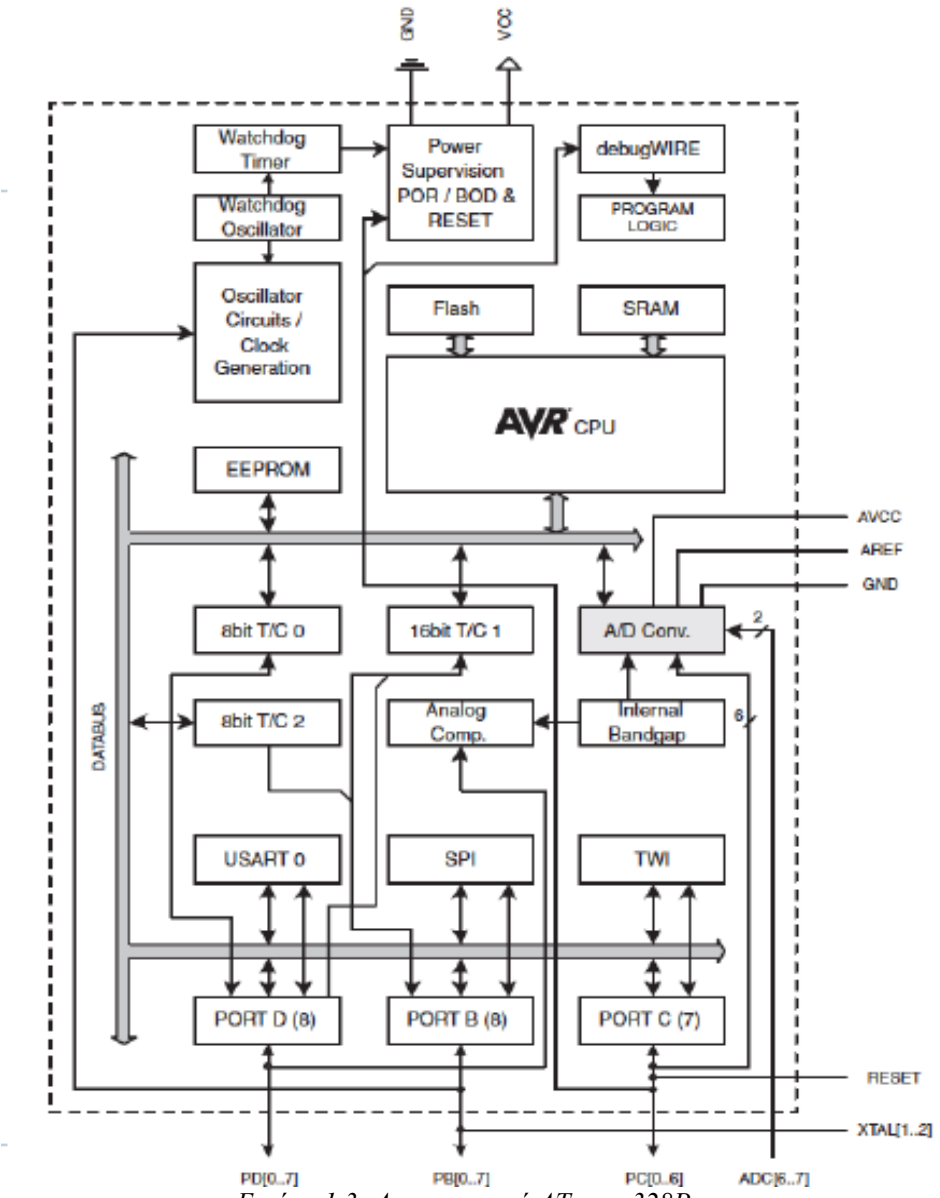
Εικόνα 1.2: Ακροδέκτες ATmega 328P.

15. AVR Επεξεργαστής (CPU)

Η κύρια λειτουργία του επεξεργαστή αφορά τη διασφάλιση της σωστής εκτέλεσης του προγράμματος. Για να μπορέσει, ο επεξεργαστής, να επιτύχει τον στόχο του, πρέπει να έχει πρόσβαση στις μνήμες (FLASH, SRAM, EEPROM), να μπορεί να ελέγχει περφερειακές λειτουργίες, να πραγματοποιεί υπολογισμούς και να διαχειρίζεται τις διακοπές (interrupts).

Όπως προαναφέραμε ο AVR βασίζεται στην ενισχυμένη αρχιτεκτονική RISC. Παράλληλα για τη βελτίωση της απόδοσης και της χρονικής επίδοσης της αντιστοιχίας, ο επεξεργαστής χρησιμοποιεί ξεχωριστές μνήμες και διαδρομές για το πρόγραμμα και τα δεδομένα (αρχιτεκτονική Harvard). Η εντολή στη μνήμη προγράμματος εκτελείται σε κάθε κύκλο ρολογιού.

Ο AVR περιέχει 32 καταχωρητές γενικής χρήσεως, άμεσα συνδεδεμένοι με την αριθμητική και λογική μονάδα (ALU), δίνοντας τη δυνατότητα σε δύο ανεξάρτητους καταχωρητές να είναι προσβάσιμοι σε μία μοναδική εντολή, που εκτελείται σε ένα κύκλο ρολογιού. Η (ALU) μονάδα είναι ψηφιακό κύκλωμα που χρησιμοποιείται για την εκτέλεση αριθμητικών και λογικών υπολογισμών και αντιπροσωπεύει τη θεμελιώδης δομή της κεντρικής μονάδας επεξεργασίας (CPU) ενός υπολογιστή.

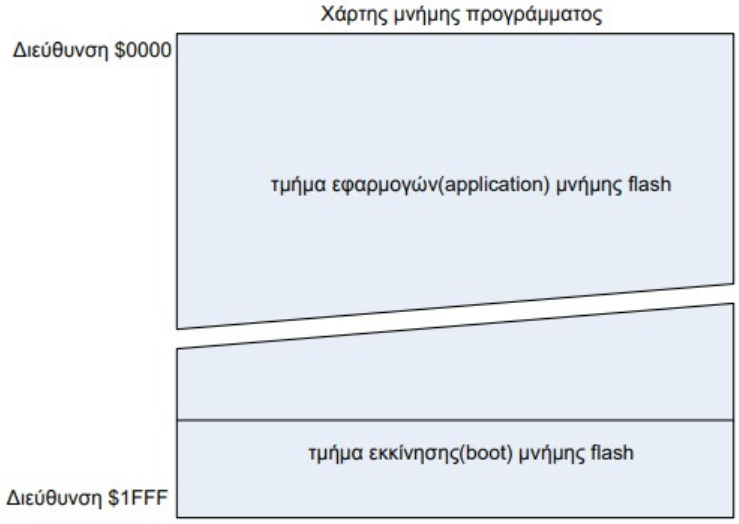


Εικόνα 1.3: Αρχιτεκτονική ATmega328P.

1.6. AVR μνήμες (memories)

Ο AVR μικροελεγκτής περιέχει δύο κύριους χώρους μνήμης, τον χώρο αποθήκης δεδομένων και το χώρο μνήμης προγράμματος. Επιπλέον, οι μικροεπεξεργαστές Atmega328p περιλαμβάνουν και μία τρίτη EEPROM μνήμη για την αποθήκευση δεδομένων.

➔ FLASH Memory : Ο μικροελεγκτής Atmega328p περιέχει 32Kbytes επαναπρογραμματιζόμενη Flash μνήμη ως αποθήκη προγραμμάτων. Οι κώδικες (οδηγίες) του προγράμματος αποθηκεύονται στη μνήμη Flash. Για λόγους ασφαλείας, η Flash μνήμη χωρίζεται σε δύο ενότητες, ενότητα εκκίνησης προγράμματος (Boot loader) και την ενότητα εφαρμογής προγράμματος (Application program). Τέλος, πρόκειται για μη πτητική μνήμη προγράμματος.



Εικόνα 1.4: Μνήμη FLASH.

➔ SRAM Memory : Η SRAM μνήμη αποτελεί στατική (static) RAM μνήμη 2Kbyte. Στη SRAM μνήμη οι δυαδικές τιμές αποθηκεύονται χρησιμοποιώντας τις κλασσικές διατάξεις λογικών πυλών των Flip- Flop [3]. Ενώ οι καταχωρητές χρησιμοποιούνται για την εκτέλεση υπολογισμών, η SRAM μνήμη χρησιμοποιείται για την αποθήκευση και τη διατήρηση προσωρινών δεδομένων, εφόσον τροφοδοτείται από ηλεκτρικό ρεύμα. Εάν διακοπεί η τροφοδοσία της τότε χάνεται και το περιεχόμενό της. Αυτού του είδους RAM χρησιμοποιούνται σε ειδικά κυκλώματα που απαιτούν μεγάλη ταχύτητα πρόσβασης.

Στον Atmega 328P υπάρχουν 2303 θέσεις μνήμης οι οποίες μοιράζονται στις εξής κατηγορίες: Οι πρώτες 32 θέσεις αφορούν τους καταχωρητές εργασίας (Register File), οι επόμενες 64 θέσεις τις standard εισόδους – εξόδους (I/O), μετέπειτα οι 160 θέσεις αντιστοιχούν στην εκτεταμένη μνήμη εισόδου - εξόδου (I/O), και τέλος οι 2048 θέσεις αφορούν την εσωτερική μνήμη δεδομένων SRAM.

➔EEPROM Memory : Στον Atmega328p συνήθως ενσωματώνεται EEPROM μνήμη 1KByte, η οποία χρησιμοποιείται για την αποθήκευση ρυθμίσεων και παραμέτρων του συστήματος. Η μνήμη αποθηκεύει και διαβάζει μεταβλητές κατά την εκτέλεση του προγράμματος και διατηρεί το περιεχόμενό της, εφόσον διακοπεί η τροφοδοσία της. Η EEPROM μνήμη μπορεί να διαγράψει το περιεχόμενο της και να επαναπρογραμματιστεί στο κύκλωμα, χωρίς να απομακρυνθεί από αυτή είτε να τοποθετηθεί σε ειδική συσκευή. Η χωρητικότητά της είναι ελάχιστη, με αποτέλεσμα να μην χρησιμοποιείται για την αποθήκευση κώδικα. Επίσης όπως κάθε EEPROM

μνήμη, μπορεί να προγραμματιστεί και να διαγραφεί εκατοντάδες χιλιάδες φορές (100.000 κύκλους εγγραφής/διαγραφής).

Η εγγραφή/ανάγνωση της μνήμης EEPROM πρέπει να γίνει, ακολουθώντας ένα ειδικό λογικό σύστημα. Επομένως οι λειτουργίες ανάγνωσης και εγγραφής πρέπει να γίνονται μέσω ειδικών καταχωρητών. Ο Atmega328p χρησιμοποιεί τους εξής καταχωρητές για να ελέγξει την EEPROM μνήμη : 16-bit EEAR (EEPROM καταχωρητής διεύθυνσης), 8-bit EEDR (EEPROM καταχωρητής δεδομένων), EECR (EEPROM καταχωρητής ελέγχου) [4].

1.7. Διάταξη USART

Ο μικροελεγκτής Atmega328P περιέχει τη διάταξη USART (Universal Synchronous and Asynchronous serial Reicever and Transmitter) ως σύνδεσμο σειριακής επικοινωνίας. Το κύκλωμα αυτό, αποτελεί ενσωματωμένο περιφερειακό του Atmega328P, όπου μετατρέπει τους εισερχόμενους και εξερχόμενους χαρακτήρες σε σειριακή μορφή. Το μεγάλο πλεονέκτημα της USART διάταξης , αφορά τη δυνατότητά του να λειτουργεί και με τους δύο τρόπους χρονισμού της σειριακής μετάδοσης (σύγχρονη – ασύγχρονη μετάδοση).

- ➔ Στην Ασύγχρονη μετάδοση, το πρώτο bit κάθε πλαισίου (Start Bit) ξεκινά τη σειριακή <<ροή>> και το Stop Bit (ένα ή δύο bit), δηλώνει το τέλος του πλαισίου. Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να επιλέξει την ισοτιμία (parity bit). Η ισοτιμία είναι μία διαδικασία ελέγχου σφαλμάτων που πιθανά έχουν προκύψει κατά τη μετάδοση των bits. Η δομή του πλαισίου της δυαδικής πληροφορίας στην ασύγχρονη σειριακή επικοινωνία καθώς και ο ρυθμός μετάδοσής του θα πρέπει να είναι συμφωνημένα μεταξύ του πομπού και του δέκτη και για το λόγο αυτό θα πρέπει να έχουν γίνει ακριβώς οι ίδιες ρυθμίσεις και στον ένα και στον άλλο. [5].
- ➔ Στη Σύγχρονη μετάδοση, τα δεδομένα ομαδοποιούνται σε μεγάλα πακέτα δεδομένων, χωρίς start και stop bits, στην αρχή και στο τέλος κάθε χαρακτήρα.[5]. Παράλληλα, ο συγχρονισμός μεταξύ του ρολογιού (Clock) του πομπού και του δέκτη επιτυγχάνεται, με την αποστολή ταυτόχρονου σήματος συγχρονισμού.

Οι ακροδέκτες του Atmega328P που χρησιμοποιούνται για την εκπομπή και λήψη των USART δεδομένων είναι αντίστοιχα οι TxD και RxD. Οι USART εκπομποί (TxD) και δέκτες (RxD) χρησιμοποιούν τους ίδιους ακροδέκτες ως εισόδους/εξόδους στις θύρες PD1 και PD0.

Κεφάλαιο 2ο:

Κάρτα RF-ID
2.1. Ορισμός

Ο τίτλος Radio Frequency Identification (Ταυτοποίηση Μέσω Ραδιοσυχνοτήτων) αποτελεί τα αρχικά του όρου RFID. Το σύστημα RFID ορίζεται ως η ασύρματη ανταλλαγή δεδομένων μεταξύ του RFID πομποδέκτη (transponder) και του αναγνώστη (reader), μέσω ραδιοσυχνοτήτων. Οι κύριες λειτουργίες είναι η αυτόματη ταυτοποίηση αντικειμένων, η παρακολούθηση αλλά και η άντληση πληροφοριών για αντικείμενα ή και ζωντανούς οργανισμούς που έχουν ενταχθεί σε ένα σύστημα RFID, σε πραγματικό χρόνο. Η RFID τεχνολογία αποτελεί τη πλέον σύγχρονη εφαρμογή ηλεκτρονικής ταυτοποίησης, καθώς θεωρείται ιδιαίτερα αποτελεσματική, και έχει βρει εφαρμογή σε διάφορους τομείς όπως τον βιομηχανικό και εμπορικό τομέα.

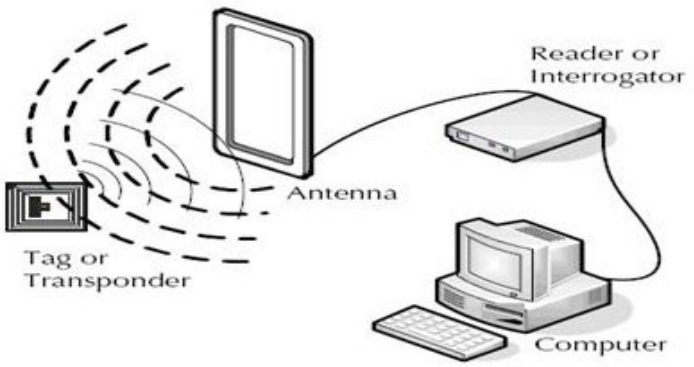
2.2. Ιστορική αναδρομή

Κατά τη περίοδο του Β΄ Παγκόσμιου πολέμου η ανακάλυψη του RADAR συνέβαλε στην εξέλιξη της τεχνολογίας ράδιο - ταυτοποίησης. Ο Βρετανός φυσικός Sir Robert Alexander Watson-Watt το 1935, ήταν αυτός που με τη τοποθέτηση ενός πομπού σε κάθε ένα από τα Βρετανικά μαχητικά αεροπλάνα, κατάφερε να προβαίνει στην αναγνώριση και τη διάκρισή τους σε φιλικά και εχθρικά αεροσκάφη [6]. Η ραγδαία τεχνολογική εξέλιξη που ακολούθησε τις επόμενες δεκαετίες , είχε ως αποτέλεσμα την εντατική έρευνα και μελέτη της ράδιο - ταυτοποίησης. Το 1960, είναι δεκαετία σταθμός για τη RFID τεχνολογία καθώς για πρώτη φορά μεγάλες εταιρίες προχωρούν στην ανάπτυξη ηλεκτρονικών συστημάτων προσδιορισμού αλλά και εφαρμογών παρακολούθησης για το κοινό και όχι για στρατιωτική χρήση. Πλέον, η τεχνολογία RFID βρίσκει εφαρμογή σε σημεία ελέγχου ως αντικλεπτικό σύστημα., για την αποτροπή κλοπής αντικειμένων, εγγράφων, βιβλίων από καταστήματα λιανικής, βιβλιοθήκες και εταιρίες. Τη δεκαετία του 70', συνεχίζεται η μελέτη και η ανάπτυξη ερευνητικών σχεδίων από πανεπιστήμια και κυβερνητικές υπηρεσίες. Νέες εφαρμογές εμφανίζονται, όπως η ηλεκτρονική συλλογή διοδίων, διάφοροι αυτοματισμοί σε εργοστάσια, ενώ αναπτύσσονται οι εφαρμογές παρακολούθησης ζώων και αυτοκινήτων. Τη δεκαετία του 80’ με την ανάπτυξη του προσωπικού υπολογιστή έχουμε τα πρώτα διαδεδομένα εμπορικά συστήματα RFID. Εμπορικές εφαρμογές του RFID επικρατούν παντού. Στη δεκαετία του 1990 χρησιμοποιούνται ευρέως και γίνεται μέρος της καθημερινότητας μας. Μέχρι και σήμερα όπου η RFID τεχνολογία συνεχώς εξελίσσεται και νέες εφαρμογές της σχεδιάζονται.

2.3. Γενικές πληροφορίες

Ένα σύστημα RF-ID αποτελείται από ορισμένα επιμέρους στοιχεία που όλα μαζί συμβάλλουν στην αποτελεσματική λειτουργία του όλου συστήματος. Τα βασικά αυτά στοιχεία διακρίνονται παρακάτω:

1. Πομποδέκτες (transponders) , γνωστοί ως “ετικέτες” (tags).
2. Κεραία (Antenna).
3. RFID αναγνώστες (readers ή interrogators).
4. Λογισμικό RFID (middleware).



Εικόνα 2.1: Βασικά στοιχεία RFID συστήματος.

- Τρόπος λειτουργίας

Ο RFID αναγνώστης εκπέμπει μέσω ραδιοσυχνοτήτων σήμα το οποίο λαμβάνεται από τις ετικέτες (tags) που βρίσκονται εντός του πεδίου εμβέλειάς του. Το πεδίο εμβέλειας επεκτείνεται από λίγα εκατοστά μέχρι και μέτρα. Εφόσον οι ετικέτες (tags) λάβουν το σήμα, ενεργοποιούνται και αυτόματα αποστέλλουν τα αποθηκευμένα δεδομένα τους (π.χ δεδομένα μνήμης), πίσω στον αναγνώστη. Ο αναγνώστης αποδιαμορφώνει τα δεδομένα που ανακτώνται από τις ετικέτες και τα αποκωδικοποιεί σε χρήσιμη πληροφορία. Σε περίπτωση που απαιτείται από την εφαρμογή, ο RFID αναγνώστης μπορεί ασύρματα να προσθέσει ή να αλλάξει νέες πληροφορίες σε συγκεκριμένες ετικέτες.

- RFID ετικέτες (tags)

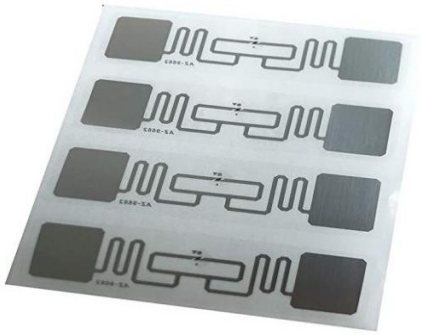
Η ετικέτα (tag) στη πιο απλοϊκή του μορφή, αποτελείται από μία κεραία που επιτρέπει την εκπομπή-λήψη ραδιοκυμάτων και ένα μικρό RFID microchip ή ολοκληρωμένο κύκλωμα (IC) που αποθηκεύει την ταυτότητα της ετικέτας (ID tag) και κάποιες επιπλέον πληροφορίες. Η RFID ετικέτα εκπέμπει δεδομένα μέσω ηλεκτρομαγνητικών κυμάτων προς τον αναγνώστη (reader). Συνήθως οι ετικέτες δεν έχουν εσωτερική τροφοδοσία (εκτός αν ανήκουν στη κατηγορία των ενεργών ή BAP ετικετών), αντ' αυτού η ετικέτα τροφοδοτείται μέσω των RF κυμάτων που εκπέμπονται από τον reader. Τα RF κύματα ενεργοποιούν το microchip, αυτό αποκωδικοποιεί το σήμα λαμβάνοντας την επιθυμητή πληροφορία και αναμεταδίδει ένα καινούργιο σήμα πίσω στη κεραία του reader. Σε κάθε microchip, υπάρχουν τέσσερις αποθήκες δεδομένων (EPC, TID, USER και Reserved) [7]. Κάθε μία από τις αποθήκες περιέχει πληροφορίες για το αντικείμενο που τοποθετήθηκε η ετικέτα ή για το ίδιο το tag. Ανάλογα με τις περιβαλλοντολογικές συνθήκες, την επιφάνεια που θα τοποθετηθεί (π.χ πλαστικό, ξύλο) και τις εφαρμογές του, συναντούμε RFID tags σε διαφορετικά σχέδια – μεγέθη ακόμη και υλικό κατασκευής ώστε να μπορούν να προσαρμοστούν σε κάθε συνθήκη. Επίσης, όπως προαναφέραμε, οι ετικέτες ανάλογα με τον τρόπο τροφοδοσίας τους διακρίνονται σε τρεις κατηγορίες :

1) Παθητικές (Passive).
2) Ενεργητικές (Active).
3) Ημι-παθητικές (Semi - Passive).

- Παθητικές (Passive)

Οι παθητικές ετικέτες δεν έχουν δική τους πηγή ενέργειας (μπαταρία), άρα ο μόνος τρόπος για να ενεργοποιηθούν είναι όταν βρεθούν εντός του ηλεκτρομαγνητικού πεδίου που εκπέμπει ο αναγνώστης. Η ετικέτα περιέχει ηλεκτρονικό κύκλωμα ικανό να απορροφήσει την ισχύ από τη κεραία του "αναγνώστη". Τότε η ενέργεια μεταφέρεται από την κεραία της ετικέτας στο ολοκληρωμένο κύκλωμα (IC), όπου ενεργοποιείται το microchip και αυτό με τη σειρά του αποστέλλει σήμα προς τον αναγνώστη. Η διαδικασία αυτή ονομάζεται οπισθοσκέδαση (backscatter). Μετά την οπισθοσκέδαση, δηλαδή την ανάκλαση κυμάτων στον αναγνώστη (reader) μέσω της κεραίας, συνεχίζεται η αποκωδικοποίηση της πληροφορίας που δέχθηκε [8].
Η βασική σχεδίαση του παθητικού RFID tag συστήματος αποτελείται από την κεραία λήψης – εκπομπής και του ολοκληρωμένου συστήματος (IC), όπου αναφέρεται ως RFID Inlay. Αν και υπάρχουν αρκετά είδη παθητικών RFID tags στην αγορά, τα πιο σημαντικά διακρίνονται σε δύο κατηγορίες: i) Inlays tags ii) Hard tags. Τα Hard RFID tags χρησιμοποιούνται σε ειδικές συνθήκες, γι αυτό σχεδιάζονται από ανθεκτικά υλικά όπως μέταλλο, πλαστικό ή καουτσούκ.

Εικόνα 2.2: Hard Tags.



Εικόνα 2.3: Inlay Tags.

Επειδή, οι παθητικές ετικέτες λειτουργούν χωρίς τροφοδοσία από δική τους πηγή, είναι αρκετά μικρές σε μέγεθος και το κόστος παραγωγής τους αρκετά φθηνό. Ωστόσο, η απουσία εσωτερικής πηγής περιορίζει την εμβέλεια λειτουργίας του όπως και το μέγεθος των δεδομένων που μπορεί να αποθηκεύσει αλλά και να αναμεταδώσει.
Οι παθητικές ετικέτες μπορούν να λειτουργήσουν σε συγκεκριμένες συχνότητες. Οι πιο κοινές συχνότητες που χρησιμοποιούνται είναι οι εξής :

➔125 – 134 KHz Low Frequency (LF)

Η χαμηλή μπάντα συχνοτήτων κυμαίνεται από 30 KHz έως 300 KHz. Οι LF RFID tags χρησιμοποιούν το μικρό εύρος ανάμεσα στα 125 – 134 KHz. Το εξαιρετικά μεγάλο μήκος κύματος, του επιτρέπει να διαπερνά εμπόδια όπως μεταλλικές
επιφάνειες ή αντικείμενα με υψηλή περιεκτικότητα νερού, όμως η ακτίνα ανάγνωσης περιορίζεται σε λίγα μόλις εκατοστά (1 εκ. - 50 εκ.). Επίσης η ταχύτητα ανάγνωσης είναι χαμηλή εξαιτίας του χαμηλού ρυθμού μετάδοσης δεδομένων (Bit/sec). Η χαμηλής συχνότητας ετικέτες χρησιμοποιούνται συχνά σε εφαρμογές παρακολούθησης ζώων όπως και σε εφαρμογές access control [9].

➔13.56 MHz High Frequency (HF)

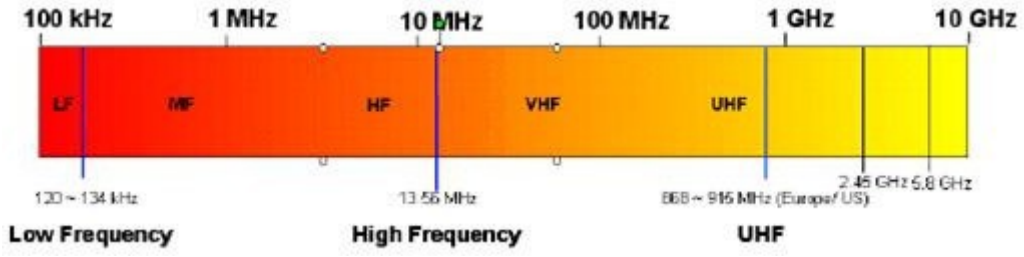
Η υψηλή μπάντα συχνοτήτων κυμαίνεται από 3 MHz έως 30 MHz. Οι HF RFID
tags χρησιμοποιούν τη συχνότητα 13.56 Mhz ως πρωτόκολλο επικοινωνίας στην οποία λειτουργεί και η τεχνολογία Near – Field Communications (NFC). Η ακτίνα ανάγνωσης ξεκινά από το 1 εκατοστό έως το 1 μέτρο. Το μήκος κύματός τους είναι μικρότερο σε σχέση με τις LF συχνότητες, ως συνέπεια τα HF κύματα να μπορούν να διαπεράσουν τις περισσότερες επιφάνειες, εκτός το παχύ μέταλο και αντικείμενα με υψηλή περιεκτικότητα νερού. Κυρίως χρησιμοποιούνται σε εφαρμογές μεταφοράς δεδομένων, “έξυπνες” διαφημίσεις, ελέγχου εισιτηρίων και ασφάλειας διαβατηρίων, χρεωστικές κάρτες, ιχνηλάτησης δεδομένων περουσιακών στοιχείων και αεροπορικών αποσκευών.

➔865 – 960 MHz Ultra High Frequency (UHF)

Η UHF μπάντα κυμαίνεται από 300 MHz έως 3 GHz. Οι UHF RFID tags χρησιμοποιούν το εύρος ανάμεσα στα 865 MHz έως 960 MHz. Επίσης, προσφέρουν μεγαλύτερη ακτίνα ανάγνωσης (3-6 μέτρα) και μπορούν να μεταφέρουν δεδομένα πολύ γρήγορα σε σχέση με τις LF και HF συχνότητες. Ακόμη, υπάρχουν μεγάλα συστήματα UHF tags που η ακτίνα ανάγνωσης μπορεί να φτάσει και τα 30 μέτρα σε ιδανικές συνθήκες. Αντίθετα, όσο μεγαλύτερη συχνότητα χρησιμοποιείται τόσο μικρότερο είναι το μήκος κύματος, με αποτέλεσμα να αντιμετοπίζει πολλές δυσκολίες όταν συναντά υλικά και επιφάνειες που είναι αρκετά σκληρές και με υψηλή περιεκτικότητα σε νερό [9]. Μπορούμε να τις συναντήσουμε σε εφαρμογές όπως , ιχνηλάτησης παλέτων, κιβωτίων, αναγνώρισεις και ελέγχου εργαλείων, χρονομέτρηση τρεξίματος και αγώνων.

➔2.45 GHz & 5.4 GHz Microwave Frequency

Η μικροκυματική μπάντα περιλαμβάνει συχνότητες από 1 έως 10 Ghz. Για RFID εφαρμογές χρησιμοποιούνται δύο συχνότητες (2.45 GHz & 5.8 GHz). Οι μικροκυματικές παθητικές ετικέτες είναι μικρότερες σε μέγεθος από τις UHF ετικέτες και έχουν σχεδόν την ίδια ακτίνα ανάγνωσης. Το κόστος αγοράς είναι αρκετά υψηλό σε σχέση με τις UHF ετικέτες. Επίσης, σε αυτές τις συχνότητες μπορούν να λειτουργήσουν ως ενεργές (active), παθητικές (passive) και ημι-παθητικές (semi- passive) ετικέτες [10]. Μία εφαρμογή του, που συναντάμε συχνά, είναι η αυτόματη πληρωμή διοδίων.



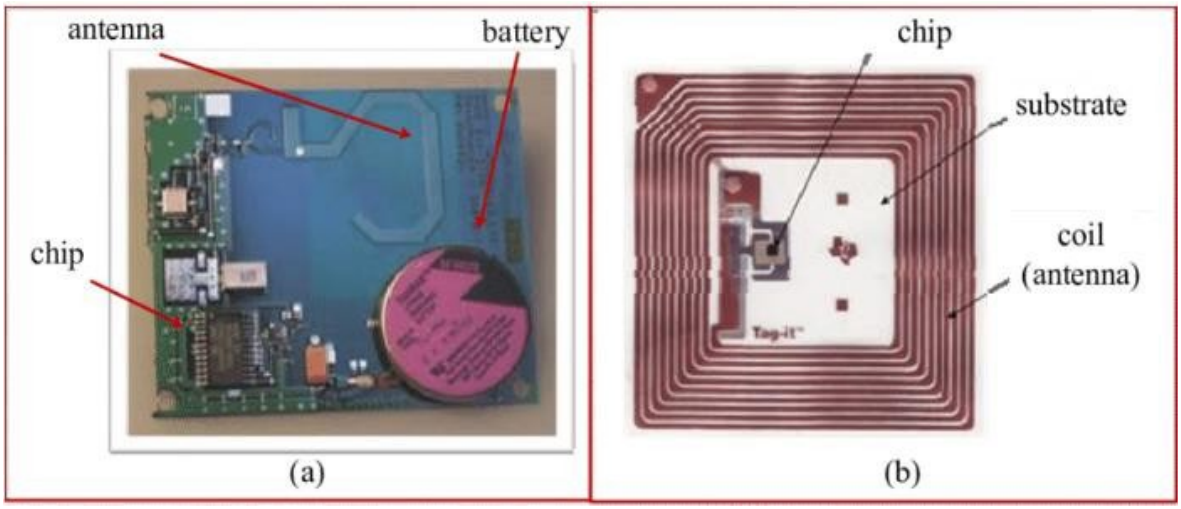
Εικόνα 2.4: Ζώνη συχνοτήτων.

- Ενεργητικές (Active).

Ένα ενεργό (Active) RFID σύστημα αποτελείται από έναν αναγνώστη (reader), μια ενεργή ετικέτα (tag) και μία κεραία εκπομπής – λήψης. Οι ενεργές ετικέτες σε σχέση με τις παθητικές , διαθέτουν εκτός της κεραίας και του microchip , εσωτερική πηγή τροφοδοσίας (μπαταρία). Η προσθήκη της εσωτερικής μπαταρίας τους δίνει τη δυνατότητα να αυξήσουν εξαιρετικά πολύ την ακτίνα ανάγνωσης (100 μέτρα), όπως και την μνήμη αποθήκευσης δεδομένων. Ακόμη, η μεγάλη διάρκεια ζωής της μπαταρίας (3 – 5 χρόνια) , επιτρέπει τη συνεχή και αδιάκοπη μετάδοση δεδομένων , ασχέτως αν βρίσκεται στο πεδίο του αναγνώστη. Επιπλέον, έχουν την ικανότητα να επικοινωνούν με οποιαδήποτε ετικέτα βρεθεί σε κοντινή απόσταση. Η δυνατότητα του αναγνώστη (reader) να διαβάζει τις ενεργές ετικέτες που βρίσκονται στο πεδίο του, σε πολλή μικρή χρονική διάρκεια, το καθιστά αρκετά αξιόπιστο και πιο ασφαλές σε σχέση με τις παθητικές ετικέτες. Οι κύριες συχνότητες που χρησιμοποιούν είναι τα 433 Mhz και 915 Mhz. Οι περισσότερες εταιρίες προτιμούν τη συχνότητα 433 Mhz, εξαιτίας του μεγάλου μήκους κύματος που εκπέμπουν και την ικανότητά τους να λειτουργούν σε μη- φιλικό περιβάλλον (Μέταλλο, Νερό) για το RFID σύστημα. Πιο σπάνια χρησιμοποιούνται active tags στη μικροκυματική συχνότητα 2.45 Ghz. Έχουν μεγάλο μέγεθος και η τιμή αγοράς τους είναι αρκετά υψηλή, σε αντίθεση με τους passive tags. Οι κυριότερες κατηγορίες ενεργών RFID tags είναι οι Αναμεταδότες (Transponders) και ο Ραδιοφάρος (Beacon) [8].

- ✓ Οι Αναμεταδότες (Transponders) tags επικοινωνούν με τον reader, εφόσον βρίσκονται κοντά στο πεδίο ανάγνωσής του. Ο αναγνώστης εκπέμπει πρώτος σήμα και στη συνέχεια η ενεργή ετικέτα αναμεταδίδει τη σχετική πληροφορία. Επίσης, ένα από τα πλεονέκτημά τους είναι ότι εξοικονομούν ισχύ μπαταρίας όταν απομακρύνονται από την ακτίνα του reader.
- ✓ Ο Ραδιοφάρος (Beacon) tags στέλνει κάθε 3-6 δευτερόλεπτα συγκεκριμένες πληροφορίες, χωρίς πρώτα να του σταλθεί σήμα από τον αναγνώστη.

Οι active RFID tags, λόγο της ενσωματομένης μπαταρίας , της κατασκευής τους από υλικά υψηλής αντοχής και της δυνατότητά τους να συνεργαστούν με τεχνολογίες όπως (αισθητήρες, GPS κτλ), χρησιμοποιούνται με αισθητήρες μέτρησης (θερμοκρασίας – υγρασίας) σε ψυγεία φορτηγών ή πειραματικών εργαστηρίων ώστε να συλλέγουν δεδομένα και να επικοινωνούν με τον reader σε πραγματικό χρόνο. Επίσης, χρησιμοποιούνται σε μεγάλες και βαριές βιομηχανίες, με σκοπό την εύκολη και ασφαλή συλλογή δεδομένων.



Εικόνα 2.5: α) Ενεργητική ετικέτα. β) Παθητική ετικέτα.

- Ημι-παθητικές (Semi – passive)

Οι ημι-παθητικές ετικέτες αποτελούνται από το ολοκληρωμένο κύκλωμα, μία κεραία και την εσωτερική πηγή (μπαταρία). Η μπαταρία θέτει σε συνεχή λειτουργία το microchip, όμως το κύκλωμα δεν μπορεί να εκπέμψει σήμα. Επομένως, οι semi- passive tags λειτουργούν όπως τους passive tags, καθώς πρέπει να βρίσκονται στην εμβέλεια του reader για να μπορέσουν να επικοινωνήσουν μαζί του. Εφόσον, συγκεντρώσουν αρκετή ισχύς, μεταδίδουν μέσω της ανάκλασης (backscatting) το σήμα πίσω στο reader. Ένα πλεονέκτημα της είναι ότι η ημι-παθητική ετικέτα έχει μεγαλύτερη ακτίνα ανάγνωσης σε σχέση με τη παθητική ετικέτα, όμως μειονεκτεί στην υψηλή τιμή αγοράς και στη μικρή διάρκεια ζωής της μπαταρίας [11].

- Κατηγορίες Ετικετών ανάλογα με το τύπο μνήμης (RO, RW ,WORM).

➔Read-Only :

Οι Read-Only ετικέτες περιέχουν μοναδικό σειριακό αριθμό και πληροφορίες λειτουργίας που προστέθηκαν κατά τη διάρκεια κατασκευής τους. Οι πληροφορίες και τα στοιχεία που έχουν αποθηκευτεί δεν μπορούν να τροποποιηθούν ή να ενημερωθούν [12]. Η μνήμη τους έχει μικρή χωρητικότητα και δεν μπορεί να αυξηθεί ο αποθηκευτικός τους χώρος. Χρησιμοποιούνται συνήθως σε απλοΐκές εφαρμογές αναγνώρισεις/ ταυτοποίησης αντικειμένων π.χ (μαγαζιά ρούχων).

➔ Read-Write:

Οι Read-Write ετικέτες, δίνουν τη δυνατότητα στο χρήστη να τροποποιήσει ή να ενημερώσει τις πληροφορίες που έχουν αποθηκευτεί, καθώς και το σειριακό αριθμό της ετικέτας, χιλιάδες φορές. Συνήθως, οι (RW) ετικέτες χρησιμοποιούν EEPROM μνήμη.

➔Write Once Read Many (WORM) :

Οι (WORM) ετικέτες είναι παρόμοιες με τις (RO). Η διαφορά τους είναι ότι ο χρήστης μπορεί να τροποποιήσει ή να ενημερώσει τις αποθηκευμένες πληροφορίες μόνο μία φορά [12]. Εφόσον συμβεί αυτό, η ετικέτα κλειδώνει και δεν επιτρέπει την επεξεργασία της πληροφορίας.

2.4. RFID Αναγνώστες (Readers)

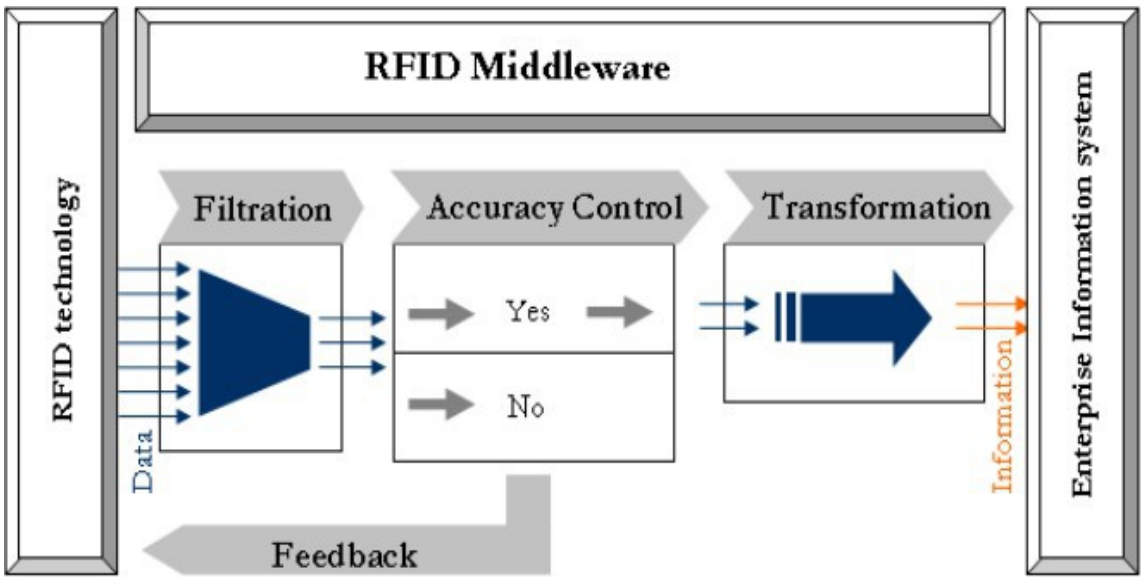
Ο RFID αναγνώστης αποτελεί σημαντικό μέρος του RFID συστήματος καθώς, χρησιμοποιείται ως γέφυρα ανάμεσα στις RFID ετικέτες (tags) και τον υπολογιστή (Middleware). Συγκεκριμένα, αναγνωρίζει την ID της ετικέτας, αναλύει τα δεδομένα και τα αναμεταδίδει πίσω στον υπολογιστή. Εφόσον, πρόκειται για μία "έξυπνη" ετικέτα, ο reader μπορεί να γράψει/προσθέσει νέα στοιχεία σε αυτή . Ο αναγνώστης έχει τη δυνατότητα να λειτουργεί σε μεγάλο εύρος συχνοτήτων, με την προϋπόθεση ότι χρησιμοποιεί μία συχνότητα την φορά [13]. Επίσης, όπως προαναφέραμε, οι αναγνώστες ενεργοποιούν τις παθητικές (Passive) και τις ημι-παθητικές (Semi- Passive) ετικέτες.
Τα βασικά μέρη του αναγνώστη είναι μία ή περισσότερες κεραίες, ένα ηλεκτρονικό σύστημα RF για την επικοινωνία(reader - tag) και η ηλεκτρονική μονάδα ελέγχου(microchip). Η μονάδα ελέγχου, πραγματοποιώντας μικρές διαφοροποιήσεις και μέτρα αποφυγής παρεμβολών, δίνει την δυνατότητα στον reader να επικοινωνήσει με την ετικέτα. Μετέπειτα αποκωδικοποιεί τα δεδομένα, τα επεξεργάζεται και τέλος τα αποθηκεύει σε μία κάρτα μνήμης.
Οι RFID αναγνώστες χωρίζονται σε Fixed RFID Readers και Mobile RFID Readers. Οι Fixed RFID Readers τοποθετούνται σε στάσιμα σημεία στο χώρο π.χ (γραφεία, τοίχους, δίπλα σε πόρτες), σε αντίθεση με τους Handheld RFID Readers που εντοπίζονται σε συσκευές χειρός.

Εικόνα 2.6: Handheld RFID Reader.

Εικόνα 2.7: Fixed RFID Reader.

2.5. RFID Λογισμικό (Middleware)

Το RFID Middleware αποτελεί εξειδικευμένο λογισμικό ενδιαμέσου , το οποίο επεξεργάζεται τα δεδομένα που μεταδίδονται από κάθε ετικέτα (tag), ώστε να μετατραπούν σε χρήσιμη και αξιόπιστη πληροφορία για τα εταιρικά πληροφοριακά συστήματα. Οι RFID τεχνολογίες έχουν τη δυνατότητα να παρέχουν τεράστιο αριθμό πληροφοριών. Ωστόσο τα εταιρικά πληροφοριακά συστήματα έχουν τη τάση να αποκτούν μόνο τις ακριβής και χρήσιμες πληροφορίες. Επομένως το RFID Middleware, είναι υπεύθυνο για τη διαχείριση της ροής δεδομένων ανάμεσα στην ετικέτα, τον αναγνώστη και των εταιρικών εφαρμογών, της ποιότητας του περιεχομένου καθώς και της χρηστικότητας της πληροφορίας. Το middleware λογισμικό, πιο αναλυτικά, φιλτράρει την περίσσεια και περιττή πληροφορία, διασφαλίζει την ακρίβεια των δεδομένων και τη μετατρέπει σε χρήσιμη πληροφορία, ώστε οι εταιρίες που χρησιμοποιούν αυτή τη τεχνολογία, να ωφεληθούν από τη καλύτερη διαχείριση των δεδομένων [14].



Εικόνα 2.8: Σύστημα RFID Middleware.

- Τρόποι λειτουργίας RFID Middleware.

Ο RFID Middleware είναι η διεπαφή μεταξύ του υλισμικού RFID (Hardware) και των RFID εφαρμογών. Το ενδιάμεσο λογισμικό αποσπά πληροφορίες από τον αναγνώστη (reader), τις φιλτράρει, τις συγκεντρώνει και μετέπειτα τις δρομολογεί σε πληροφοριακά συστήματα (IT) όπως, ενδοεπιχειρησιακού σχεδιασμού (ERP). Η συνεχή επικοινωνία των RFID συστημάτων με πληροφοριακά συστήματα (IT) και η πρόσβασή τους σε βάσεις δεδομένων, δίνει την δυνατότητα στις εταιρίες να διαχειρίζονται με μεγαλύτερη αξιοπιστία και ακρίβεια τις πληροφορίες που εκλαμβάνουν, με αποτέλεσμα τη βελτίωση των οργανωτικών διαδικασιών [14].

.

2.6. Εκτυπωτές RFID

Ο εκτυπωτής RFID αποτελεί επίσης, σημαντικό κομμάτι της τεχνολογίας RFID, καθώς χρησιμοποιείται για την εκτύπωση των έξυπνων ετικετών, αλλά και την εγγραφή δεδομένων στο RFID μικροτσίπ που βρίσκεται ενσωματωμένο στην ετικέτα. Ο εκτυπωτής περιλαμβάνει έναν κωδικοποιητή RF, ο οποίος προσθέτει δεδομένα στην ετικέτα. Στη συνέχεια ελέγχει εάν η κωδικοποίηση έγινε σωστά. Τέλος, εκτυπώνει την ετικέτα, στην οποία έχουν προστεθεί barcode, γραφήματα, ή οποιαδήποτε πληροφορία θεωρείται χρήσιμη για την ανάγνωση της ετικέτας.
Υπάρχουν τρία κύρια είδη RFID εκτυπωτών, (βιομηχανικοί, γραφείου και κινητοί), που καθορίζονται ανάλογα με την ανθεκτικότητα, το μέγεθος και τον ρυθμό παραγωγής των ετικετών.
Επίσης, οι εκτυπωτές μπορούν να εκτυπώσουν σε επικαλυμμένο ή μη επικαλυμμένο χαρτί, όπως και σε πλαστικές ή συνθετικές επιφάνειες ετικετών.

Εικόνα 2.9: Εκτυπωτής RFID.

2.7. Σύγκριση τεχνολογίας RFID – Barcode

Οι Barcode και RFID είναι δύο διαφορετικές μορφές τεχνολογίας που χρησιμοποιούνται για κοινό σκοπό, την ανάγνωση και τη συλλογή δεδομένων. Συνήθως τις συναντάμε σε εφαρμογές που αφορούν τη παρακολούθηση και τη διαχείριση προϊόντων ή εξοπλισμού που διαθέτουν οι διάφορες εταιρίες. Κάποιες από τις διαφορές των δύο τεχνολογιών είναι :

BARCODE

1. Διαβάζει την κάθε ετικέτα ξεχωριστά, με αποτέλεσμα η ανάγνωση των ετικετών να χρειάζεται περισσότερο χρόνο, σε σχέση με την RFID τεχνολογία.
2. Οι Barcode ετικέτες εκτυπώνονται σε χαρτί ή κολλητική ταινία, με αποτέλεσμα την εύκολη φθορά τους και τη δύσκολη ανάγνωσή τους.
3. Ο αναγνώστης πρέπει να βρίσκεται οπτικά στην ίδια ευθεία με την ετικέτα, με αποτέλεσμα το μικρό εύρως ανάγνωσης.
4. Οι Barcode ετικέτες μπορούν να αποθηκεύσουν ελάχιστο αριθμό δεδομένων (Χρησιμοποιούν κώδικα UPC).
5. Συνήθως έχουν χαμηλό κόστος.
6. Είναι πιο ελαφριές σε σχέση με τις RFID ετικέτες.
7. Χρησιμοποιούνται ως Read-Only ετικέτες.
8. Ανάγνωση των Barcode ετικετών από ανθρώπινο μάτι.

Διακρίνοντας τις διαφορές ανάμεσα στις δύο τεχνολογίες, κατανοούμε ότι η RFID τεχνολογία οδεύει προς την αντικατάσταση των Barcode ετικετών [15]. Όμως, η δυνατότητα των Barcodes ετικετών να διαβάζονται από το ανθρώπινο μάτι είναι πάντα απαραίτητη. Επομένως, ίσως ο συνδυασμός των δύο τεχνολογιών να αποτελεί τη χρυσή τομή στην εξέλιξή τους.

2.8. Εφαργμογές RFID Τεχνολογίας

Στην εποχή μας, η ραγδαία ανάπτυξη της RFID τεχνολογίας είναι γεγονός. Πλέον, οι RFID εφαρμογές αποτελούν σημαντικό μέρος της ζωής του σύγχρονου ανθρώπου καθώς μπορούμε να τις εντοπίσουμε σχεδόν σε όλους τους τομείς παραγωγής, στην οικονομία, στην ιατρική και στη καθημερινότητά του. Στους τομείς παραγωγής , η RFID τεχνολογία χρησιμοποιείται σε εφαρμογές που αφορούν τη ταυτοποίηση, οργάνωση, αποθήκευση και μεταφορά αντικειμένων – προϊόντων, σε μονάδες όπως τα εργοστάσια, αποθήκες εμπορευμάτων και τα κέντρα διανομής. Η δυνατότητα των RFID να αναγνωρίζουν και να διαβάζουν πλήθος ετικετών (tags) σε ελάχιστο χρόνο, διευκολύνει τη λειτουργία των μονάδων, ενώ παράλληλα αυξάνει και την παραγωγικότητά τους. Επίσης, νοσοκομειακές μονάδες και κλινικές, χρησιμοποιούν εφαρμογές όπως το RFID ιατρικό βραχιολάκι ή έξυπνες εφαρμογές ανίχνευσης αντικειμένων, ώστε το ιατρικό προσωπικό να μπορεί να ενημερώνεται για τα στοιχεία και την υγεία του ασθενή, καθώς και για τη προμήθεια φαρμάκων και ιατρικού εξοπλισμού. Ακόμη, η πλειονότητα των οικονομικών συναλλαγών (εμπορικές συναλλαγές, πληρωμή διοδίων) πραγματοποιείται με τη χρήση RFID Card. Τα μεγαλύτερα αεροδρόμια στο κόσμο, χρησιμοποιούν την τεχνολογία RFID σε συστήματα διαχείρισης και παρακολούθησης αποσκευών. Παράλληλα, ο έλεγχος πρόσβασης σε αεροδρόμια, τρένα, λεωφορεία διευκολύνεται με τη κατοχή κάρτας RFID.

2.9. Το RF-ID σύστημα της εργασίας

Το RFID System που θα χρησιμοποιήσουμε στην εργασία είναι το MFRC- 522. Αποτελείται από ένα NFC chip και μπορεί να ενσωματωθεί σχεδόν σε κάθε κινητό ή συσκευή που χρησιμοποιεί τη τεχνολογία Near-Field Communication. Είναι φθηνό σε κόστος και μπορεί να χρησιμοποιηθεί σε αρκετές εφαρμογές ελέγχου πρόσβασης , αυτόματης ταυτοποίησης , ρομποτικής , σύστημα πληρωμής κτλ [16].

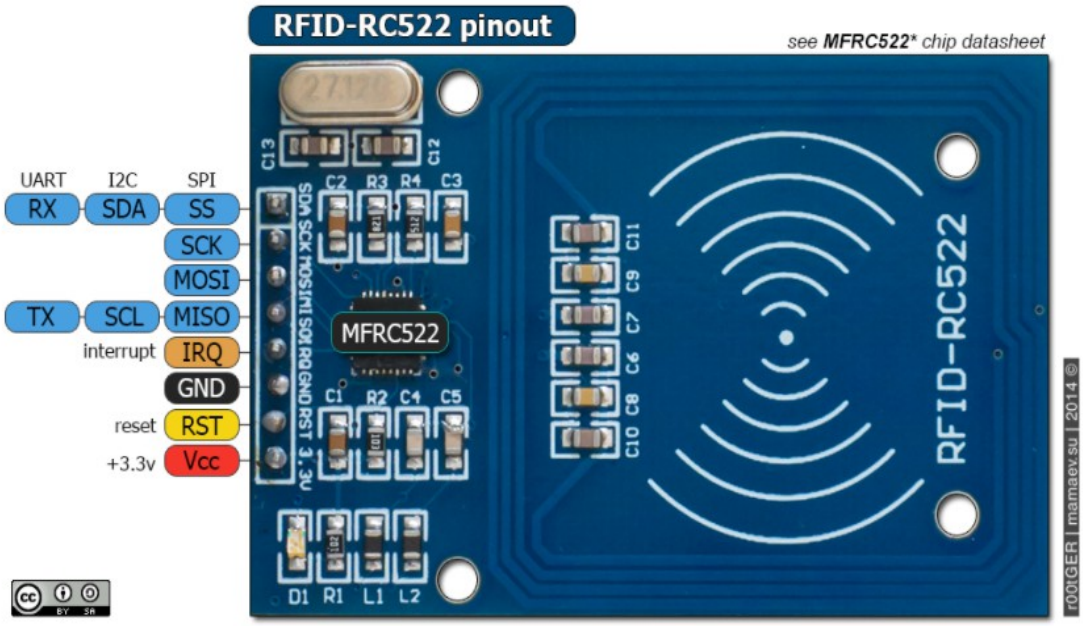
Χαρακτηριστικά NFC / RFID - MFRC522 13.56MHz :

- Τύπος Αισθητήρα: NFC / RFID
- Τάση Εισόδου: 3.3 VDC
- Ρεύμα Λειτουργίας: 25 mA
- Διασύνδεση: Ψηφιακή
- Πρωτόκολλο Επικοινωνίας : SPI
- Συχνότητα Λειτουργίας: 13.56 MHz
- Ακτίνα Ανάγνωσης: 0 – 60 mm
- Ταχύτητα Μετάδοσης Δεδομένων: Max 10 Mbit/s

Διαστάσεις NFC / RFID - MFRC522 13.56MHz :

- Μέγεθος αναγνώστη: 40 x 60 mm
- Μέγεθος Hard tag: 0.87 × 85.5 × 54 mm
- Μέγεθος κάρτας tag: 32 x 40.5 x 4.2 mm

Εικόνα 2.10: α) RFID Hard Ετικέτα. β) RFID Αναγνώστης. γ) RFID Κάρτα Ετικέτα.



Εικόνα 2.11: Επαφές αναγνώστη (reader).

Κεφάλαιο 3ο:

Αισθητήρας Δακτυλικών αποτυπωμάτων

3.1. Δακτυλοσκοπία - Ορισμός

Η δακτυλοσκοπία σαν έννοια είναι κατεξοχήν ελληνική καθώς προέρχεται από τον συνδυασμό των δυο ελληνικών λέξεων δάκτυλο και σκοπιά, και είναι η μέθοδος εξακρίβωσης της ταυτότητας ενός ατόμου με βάση τα δακτυλικά του αποτυπώματα. Η χρήση της έννοιας αυτής είναι αναγνωρισμένη διεθνώς καθώς μπορούμε να παρατηρήσουμε ότι και στην Αγγλική γλώσσα χρησιμοποιούν τον όρο dactyloscopy για να περιγράψουν την ίδια διαδικασία με αυτήν που αναφέρθηκε προ ολίγου.

3.2. Ιστορική αναδρομή

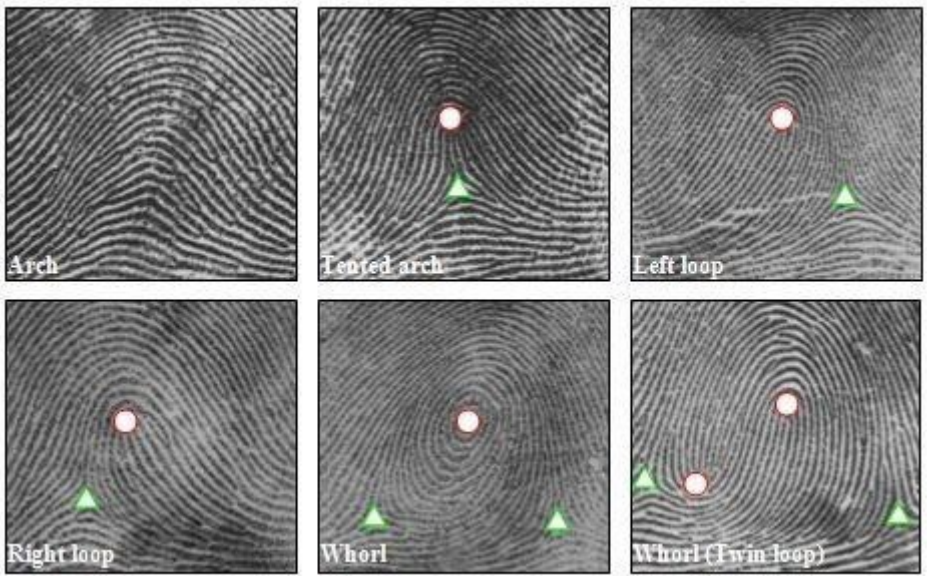
Από την αρχή της ιστορίας της μέχρι και σήμερα, παρατηρούμε πως η δακτυλοσκοπία χρησιμοποιήθηκε κυρίως για δύο πολύ σημαντικούς λόγους: ως απόδειξη για κατοχή ορισμένων αγαθών και ως μέσο πάταξης και ελαχιστοποίησης του επιπέδου της εγκληματικότητας.
Με την πάροδο των χρόνων, έχουν γίνει ουκ ολίγες έρευνες για την ιστορία και την χρήση των δακτυλικών αποτυπωμάτων φέρνοντας στο φως σημαντικές πληροφορίες για τους επιστήμονες. Όπως είναι λογικό λοιπόν, στα πρώτα βήματα της ιστορίας της, η δακτυλοσκοπία χρησιμοποιήθηκε κυρίως, ως αποδεικτικό στοιχείο κτήσης αντικειμένων. Η πρώτη εσκεμμένη χρήση του δακτυλικού αποτυπώματος λοιπόν, παρατηρείται στον 3ο αιώνα π.Χ. (μεταξύ 2600-2350 π.Χ.) στην Μεσοποταμία, καθώς από αρχαιολογικές έρευνες που έχουν γίνει, δακτυλικά αποτυπώματα παρατηρήθηκαν σε αρκετές πήλινες σφραγίδες. Ωστόσο, πολλοί ειδικοί στον χώρο της δακτυλοσκοπίας εκφράζουν τους ενδοιασμούς τους για την ακριβή χρονολογία που οι άνθρωποι στα αρχαία χρόνια άρχισαν να χρησιμοποιούν το δακτυλικό αποτύπωμα ως αποδεικτικό στοιχείο, καθώς δακτυλικά αποτυπώματα έχουν παρατηρηθεί σε αρχαιολογικά ευρήματα ακόμη πιο πριν από τον 3ο αιώνα π.Χ., χωρίς όμως να υπάρχουν ιστορικές πηγές που να κάνουν λόγο για κάποια συγκεκριμένη επιθυμητή χρήση αυτών. Εκτός από την Μεσοποταμία, παρατηρήθηκε ότι τα δακτυλικά αποτυπώματα χρησιμοποιήθηκαν ως αποδεικτικό στοιχείο και σε πήλινες πλάκες στην Βαβυλωνία μεταξύ 1913 και 1855 π.Χ, καθώς και στην Κίνα κατά την διάρκεια της δυναστείας Qin και της δυναστείας Han με την αρχή να γίνεται περίπου στο 246 π.Χ.. Στο σημείο αυτό, μπορούμε να αναφέρουμε πως η περίοδος αυτή στην Κίνα, αποτέλεσε σημείο σταθμό για την δακτυλοσκοπία καθώς μέχρι το 589 μ.Χ. Παρατηρείται μεγάλη εξέλιξη της χρήσης της δακτυλοσκοπίας για πολλαπλούς σκοπούς. Στο διάστημα αυτό λοιπόν, βλέπουμε το δακτυλικό αποτύπωμα να χρησιμοποιείται στην Κίνα για την ιδιωτικοποίηση υλικών αγαθών, όμως είναι αρκετά σημαντικό να αναφέρουμε ότι η ανακάλυψη του χαρτιού και του μεταξιού, οδήγησε στη χρήση του δακτυλικού αποτυπώματος σε έγγραφα γάμων και διαζυγίων, σε περιπτώσεις καταγραφής στρατιωτικού δυναμικού, σε έγγραφα αναγνώρισης χρέους, σε έγγραφα κτήσης οικοπέδου και κατοικίας, σε περιουσιακά έγγραφα και το σημαντικότερο, χρησιμοποιήθηκε ως αποδεικτικό στοιχείο σε εγκληματικές πράξεις, όπως και σε έγγραφα ομολογίας εγκλημάτων [19]. Στην νεότερη ιστορία της δακτυλοσκοπίας, τα πιο σημαντικά γεγονότα αρχίζουν στα μέσα του 19ου αιώνα, με τον William Herschel να γίνεται ο πρώτος ευρωπαίος πολίτης που αναγνωρίζει την δύναμη της δακτυλοσκοπίας. Ο Herschel, στέλεχος της διοικητικής οργάνωσης της Αγγλικής αυτοκρατορίας στις αγγλικές αποικίες της Ινδίας, εκτός των άλλων υπήρξε υπεύθυνος και για θέματα δικαιοσύνης κατά την διάρκεια της θητείας του στην Ινδία και κυρίως στην επαρχία της Βεγγάλης, αναγνώρισε την χρησιμότητα του συστήματος της δακτυλοσκοπίας που χρησιμοποιούσαν οι Ινδοί για χρόνια πριν από την άφιξη των Άγγλων, υιοθετώντας και ο ίδιος το σύστημα αυτό, για να φτάσει στο συμπέρασμα ότι τα δακτυλικά αποτυπώματα είναι μοναδικά από άνθρωπο σε άνθρωπο, ύστερα από μελέτη πολλών εγγράφων δακτυλικών αποτυπωμάτων στα οποία είχε πρόσβαση. Ένα

ακόμη σημαντικό πρόσωπο αυτής της εποχής για την δακτυλοσκοπία υπήρξε και ο Henry Faulds. Ο Faulds ήταν απεσταλμένος γιατρός σε αποστολές στην Ιαπωνία και στην Ινδία, όπου έκανε εξονυχιστική μελέτη πάνω στην δομή των δακτυλικών αποτυπωμάτων. Ο Faulds ήταν ο πρώτος ευρωπαίος που δημοσίευσε άρθρο σχετικά με τα δακτυλικά αποτυπώματα το 1880, και θέλησε να μοιραστεί και να ανταλλάξει απόψεις με τον ειδικό της τότε εποχής για την εξέλιξη του ανθρώπινου γένους Charles Darwin. Ωστόσο, ο δεύτερος, αναγνωρίζοντας οτι οι γνώσεις του, δεν επαρκούσαν για να βοηθήσουν τον Faulds, τον παρέπεμψε στον ξάδερφό του και ειδικό ανθρωπολόγο, Sir Francis Galton [20]. Ο Galton, έφτασε στην συγγραφή και την δημοσίευση του πρώτου βιβλίου για την δακτυλοσκοπία το 1892, με τίτλο “Finger Prints”, μελετώντας τις έρευνες που πραγματοποίησαν οι Herschel-Faulds, καθώς και της ιστορίας της δακτυλοσκοπίας στις Ασιατικές χώρες. Ο Galton χαρακτήρισε τα δακτυλικά αποτυπώματα ως μοναδικά και μη αλλοιώσιμα και περιέγραψε την λεπτή δομή των μοτίβων που ονομάζουμε μικρολεπτομέρειες ή λεπτομέρειες Galton, τα οποία είναι τα χαρακτηριστικά αναγνώρισης στις συγκρίσεις μεταξύ δακτυλικών αποτυπωμάτων. Τα τελευταία σημαντικά ονόματα στην ιστορία της δακτυλοσκοπίας πριν την ανακάλυψη των υπολογιστών, είναι αυτά του Alphonse Bertillon και του Juan Vucetich. Ο Bertillon, αν και ήταν επιστήμονας στον χώρο της ανθρωπολογίας, δεν είχε καμία σχέση με την δακτυλοσκοπία όμως η ανακάλυψη και η καθιέρωση ενός συστήματος σύγκρισης βιολογικών χαρακτηριστικών του ανθρώπινου σώματος στο εγκληματολογικό σύστημα, ήταν αρκετό για να αποδειχθεί εν τέλη ότι ο πιο αποτελεσματικός τρόπος για την εξιχνίαση ενός εγκλήματος είναι η αναγνώριση των δακτυλικών αποτυπωμάτων του δράστη. Αυτό συνέβη, καθώς το σύστημα του Bertillon το οποίο ονομάστηκε Bertillonage, παρουσίασε αρκετά κενά και προβλήματα κατά την εφαρμογή του, αναγκάζοντας πολλούς ειδικούς της τότε εποχής, να στραφούν στην δακτυλοσκοπία, η οποία βρισκόταν σε έξαρση. Την ίδια εποχή ο Vucetich, Κροάτης μετανάστης στην Αργεντινή, ο οποίος ασχολήθηκε με την στατιστική επιστήμη στο αστυνομικό τμήμα του Μπουένος Άιρες, έφτασε στην ανακάλυψη και ανάπτυξη ενός δικού του συστήματος καταγραφής, ταξινόμησης και αναγνώρισης δακτυλικών αποτυπωμάτων, ύστερα από έρευνα και μελέτη που πραγματοποίησε πάνω στα αρχεία του Galton. Αξιοσημείωτο είναι το γεγονός πως το σύστημά του, το οποίο ο ίδιος ονόμασε δακτυλοσκοπία, χρησιμοποιείται ακόμα και σήμερα σε πολλές χώρες, κυρίως ισπανόφωνες. Η ανακάλυψη του υπολογιστή ωστόσο, επέτρεψε τους εγκληματολόγους να έχουν πρόσβαση σε τεράστιες βάσεις δεδομένων με δακτυλικά αποτυπώματα, κάτι το οποίο ήταν αδύνατον να πραγματοποιηθεί από ένα ανθρώπινο χέρι σε σύντομο χρονικό διάστημα. Οι Βρετανοί εγκληματολόγοι εκμεταλλεύτηκαν αυτή τη τεχνογνωσία, την δεκαετία του 70’, καταγράφοντας 120,000 ζεύγη δακτυλικών αποτυπωμάτων ετησίως. Την ίδια περίπου περίοδο, το FBI προχώρησε στην ανακάλυψη του συστήματος AFIS (Automated Fingerprint Identification System) το οποίο έδινε την δυνατότητα στους υπολογιστές να αποθηκεύουν μεγάλο όγκο δεδομένων δακτυλικών αποτυπωμάτων και να πραγματοποιούν αυτόματο έλεγχο σύγκρισης και ταυτοποίησης δύο ή περισσοτέρων δακτυλικών αποτυπωμάτων.

3.3. Χαρακτηριστικά δακτύλων

Τα χαρακτηριστικά των δακτυλικών αποτυπωμάτων μαρτυρούν την ταυτότητα κάθε ατόμου, καθώς όπως έχουμε αναφέρει και παραπάνω, είναι μοναδικά, διαφέρουν από άνθρωπο σε άνθρωπο και διαμορφώνονται πριν από την γέννηση του ανθρώπου, όταν αυτός βρίσκεται ακόμη στην μήτρα της μητέρας του. Αξιοθαύμαστο είναι το γεγονός ότι δύο άνθρωποι μπορεί να μοιράζονται το ίδιο DNA, όμως ποτέ το ίδιο δακτυλικό αποτύπωμα. Τα χαρακτηριστικά που συναντάμε στα δακτυλικά αποτυπώματα αρχίζουν να σχηματίζονται την 10η βδομάδα της κυοφορίας του εμβρύου, εξαιτίας της πίεσης που ασκείται στα δάκτυλα των χεριών και των ποδιών του βρέφους και των γονιδίων, ωστόσο οι μικρολεπτομέρειες που προκύπτουν είναι εντελώς τυχαίες [21]. Τα κυριότερα από αυτά χαρακτηριστικά, είναι οι κοιλάδες (valleys) και οι παρυφές (ridges) που βρίσκονται στην επιδερμίδα των δακτύλων. Εάν παρατηρήσουμε την απεικόνιση ενός δακτυλικού αποτυπώματος διακρίνουμε, ότι οι κοιλάδες είναι λευκές γραμμές ενώ οι παρυφές, μαύρες γραμμές. Ακόμη μπορούμε να παρατηρήσουμε, ότι ανά σημεία οι κοιλάδες και οι παρυφές σχηματίζουν κάποια σχήματα, τα οποία ονομάζουμε ιδιαίτερες περιοχές (singular regions). Τα σχήματα που διαμορφώνονται στα σημεία αυτά, διαχωρίζονται συνήθως σε δύο περιοχές. Αυτές όπου σχηματίζονται σε τριγωνικές περιοχές τις οποίες ονομάζουμε σημεία δέλτα (delta points) και εκείνες που σχηματίζονται σε κυκλικές περιοχές, τις οποίες ονομάζουμε σημεία πυρήνα (core points). Συνήθως, η διαμόρφωση ενός ή παραπάνω σημείου δέλτα εξαρτάται από το πως διαμορφώνεται το σημείο του πυρήνα στο δάκτυλο. Οι πιο συνηθισμένες κατηγορίες των σημείων πυρήνα που μπορούν να σχηματιστούν σε ένα ανθρώπινο δάκτυλο είναι οι εξής:

- Τόξο (Arch)
- Δακτύλιος (Whorl)
- Απλός Βρόγχος (Simple Loop)
- Διπλός Βρόγχος (Double Loop)



Εικόνα 3.1: Κατηγορίες σημείων πυρήνα ανθρώπινου δακτύλου.

Επίσης στις κατηγορίες αυτές εντάσσονται οι κατηγορίες του Τεντωμένου Τόξου (Tented Arch) και αυτές όπου ο Απλός Βρόγχος διακρίνεται, σε Δεξί (Right Loop) και Αριστερό Βρόγχο (Left Loop). Αυτό που είναι σημαντικό να αναφέρουμε, είναι πως οι ιδιαίτερες περιοχές στο δάκτυλο και οι κατηγορίες αυτών, υπάρχουν για την ευκολότερη αναγνώριση, οργάνωση και ταξινόμηση ενός δακτυλικού αποτυπώματος σε μια ογκώδη βάση δεδομένων από δακτυλικά αποτυπώματα. Εφόσον, οι ειδικοί κατηγοριοποίησαν τα είδη των δακτυλικών αποτυπωμάτων, σειρά είχε η σύγκριση αυτών, βάση της δυνατότητά τους να ταυτοποιούν άτομα. Ως σημείο αναφοράς, όπως και μέτρο σύγκρισης για τα δακτυλικά αποτυπώματα καθιερώθηκαν οι μικρολεπτομέρειες (miniutiae), που δεν ήταν τίποτε άλλο από τον τερματισμό μιας παρυφής ή κοιλάδας (termination or ridge ending) και την διακλάδωση αυτών (bifurcations).

3.4. Είδη αισθητήρων

Όπως κάθε άλλο βιομετρικό σύστημα, έτσι και το σύστημα της δακτυλοσκοπίας ακολουθεί συγκεκριμένα βήματα για την επίτευξη του στόχου της (καταγραφή- αποθήκευση-σύγκριση), και για να πραγματοποιηθούν τα βήματα χρειάζονται και τα απαραίτητα υλικά. Τα υλικά αυτά είναι, ένας αισθητήρας (καταγραφή του επιθυμητού υλικού), ένας ηλεκτρονικός υπολογιστής (αποθήκευση του καταγραφομένου από τον αισθητήρα υλικού) και ένα λογισμικό (πραγματοποίηση σύγκρισης των αποθηκευμένων δεδομένων στον ηλεκτρονικό υπολογιστή). Οι αισθητήρες δακτυλικών αποτυπωμάτων είναι τα αναγκαία τεχνολογικά μέσα με τα οποία μπορούμε να απεικονίσουμε ένα δακτυλικό αποτύπωμα και να το αποηθηκεύσουμε σε μια επιθυμητή βάση δεδομένων σε έναν Η/Υ. Στην εποχή που διανύουμε, υπάρχουν πολυάριθμα είδη αισθητήρων, όμως οι πιο ευρέως διαδεδομένοι και χρησιμοποιημένοι είναι, οι οπτικοί και οι αισθητήρες χωρητικής ανίχνευσης [22].

- Οπτικοί αισθητήρες.

Πρόκειται για ένα από τους πλέον παλιούς τρόπους ανάγνωσης δακτυλικών αποτυπωμάτων. Ο χρήστης πιέζει το δάχτυλό του, στην επιφάνεια ενός πρίσματος που φωτίζεται από μια πηγή, η οποία συνήθως είναι LED λαμπάκι. Στο σημείο επαφής, το φως δεν ανακλάται, αλλά απορροφάται. Από την άλλη πλευρά του πρίσματος, το φως που εξέρχεται (η εικόνα) μεταδίδεται μέσω ενός φακού σε ένα αισθητήρα CCD/CMOS και τα δεδομένα που προκύπτουν μεταφέρονται σε ένα κύκλωμα ψηφιοποίησης (frame grabber). Τυπικά το σύστημα CCD/CMOS και το σύστημα ψηφιοποίησης μπορεί να βρίσκονται σε ένα μοναδικό ολοκληρωμένο κύκλωμα.

- Αισθητήρες χωρητικής ανίχνευσης.

Ο πιο δημοφιλής τρόπος ανίχνευσης μετά την οπτική ανίχνευση, βασίζεται στη μέτρηση της χωρητικότητας (capacitance). Το δάκτυλο δρα σαν οπλισμός ενός πυκνωτή καθώς πιέζεται στην επιφάνεια του ανιχνευτή. Η χωρητικότητα μεταβάλλεται ανάλογα με το αν ακουμπάει παρυφή ή κοιλάδα. Η μέτρηση αυτή της μεταβλητής χωρητικότητας μας δίνει τελικά την εικόνα του αποτυπώματος. Καθώς οι χωρητικότητες είναι πολύ μικρές και απαιτείται μεγάλη ευαισθησία για να γίνει σωστή ανάγνωση της τιμής τους (με βάση το ηλεκτρικό πεδίο), το πάχος της επίστρωσης του αισθητήρα πρέπει να είναι πολύ μικρό (σε επίπεδο λίγων microns), καθώς η χωρητικότητα μειώνεται με το τετράγωνο της απόστασης μεταξύ των οπλισμών του πυκνωτή (Στη χωρητική ανίχνευση ο ένας οπλισμός είναι το δάχτυλο και το άλλο ο αισθητήρας, έτσι είναι σημαντικό το ενδιάμεσό τους κενό να είναι όσο το δυνατό μικρότερο) [23]. Ένα άλλο μειονέκτημα της χωρητικής ανίχνευσης είναι η σχετικά εύκολη παρεμβολή της, από ηλεκτρικά πεδία που μπορεί να παράγονται από άλλες συσκευές. Τυχόν ηλεκτροστατική εκφόρτιση από το δέρμα του χρήστη μπορεί επίσης να καταστρέψει αισθητήρες αυτού του τύπου.

3.5. Ο αισθητήρας της εργασίας

Ο αισθητήρας λήψης δακτυλικών αποτυπωμάτων που επιλέξαμε να χρησιμοποιήσουμε για την υλοποίηση της εργασίας μας, είναι ένας adafruit αισθητήρας οπτικής ανίχνευσης της σειράς ZFM-20 της εταιρίας Zhiantec Technologies. Παρακάτω παρατηρούμε τα χαρακτηριστικά του αισθητήρα [18]:

- Τύπος αισθητήρα: ID Τυπική Τάση Εισόδου: 3.3VDC, 5VDC, 6V DC
- Ρεύμα Λειτουργίας: 120mA
- Διασύνδεση: Ψηφιακή
- Πρωτόκολλο Επικοινωνίας: TTL Serial
- Μέγιστο ρεύμα: 150mA
- Χρόνος απεικόνισης δακτυλικών αποτυπωμάτων: <1,0 δευτερόλεπτα
- Χωρητικότητα αποθήκευσης: 162 πρότυπα
- Αξιολογήσεις ασφαλείας (1-5 χαμηλή έως υψηλή ασφάλεια)
- Ποσοστό ψευδούς αποδοχής: <0,001% (επίπεδο ασφάλειας 3)
- Ποσοστό ψευδούς απόρριψης: <1,0% (επίπεδο ασφάλειας 3)
- Baud rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)
- Θερμοκρασία λειτουργίας: -20C to +50C
- Υγρασία εργασίας: 40% -85% RH

Εικόνα 3.2: Adafruit αισθητήρας οπτικής ανίχνευσης της σειράς ZFM-20.

Οι κυριότερες αιτίες που συνέβαλαν στην επιλογή του συγκεκριμένου αισθητήρα αφορά αρχικά, το κόστος σε συνδυασμό με την ποιότητα λειτουργίας που προσφέρει ο αισθητήρας, καθώς και ο σκοπός της παρούσας πτυχιακής εργασίας ο οποίος είναι καθαρά ερευνητικός και εκπαιδευτικός.

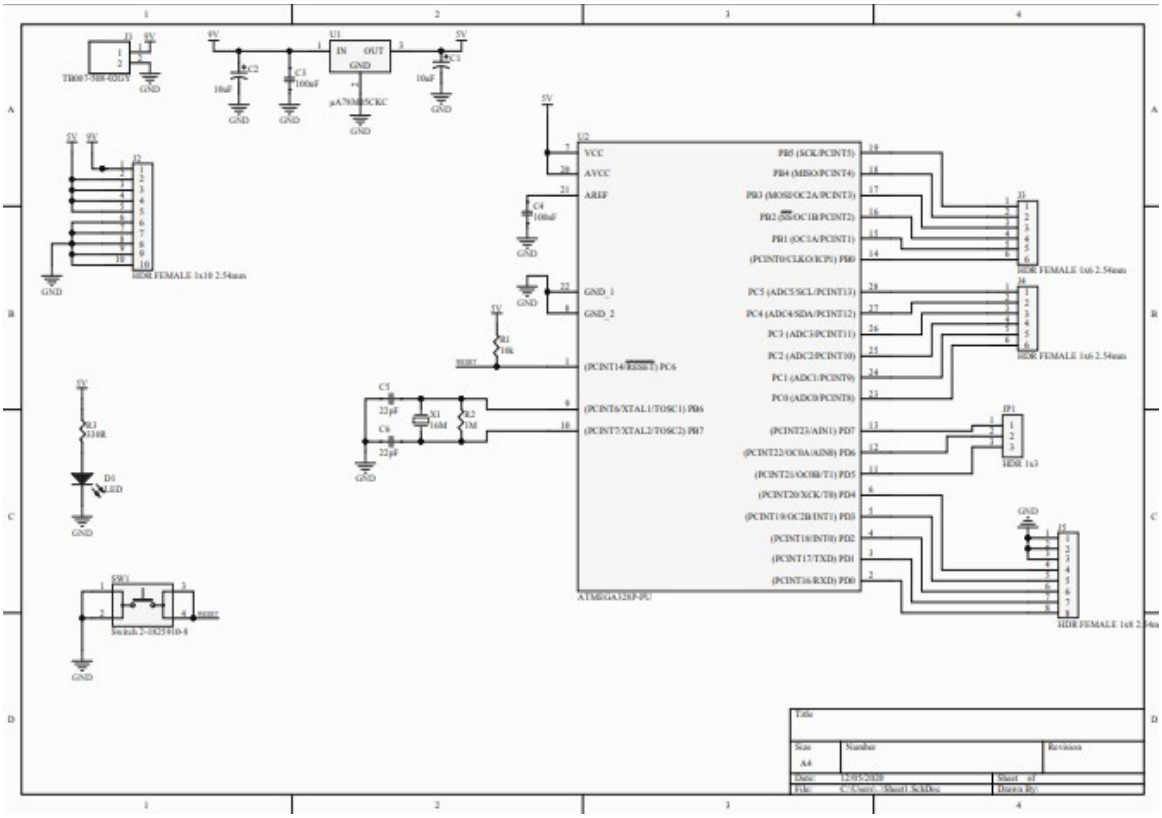
Κεφάλαιο 4ο:

Σχεδίαση και κατασκευή συστήματος Access Control

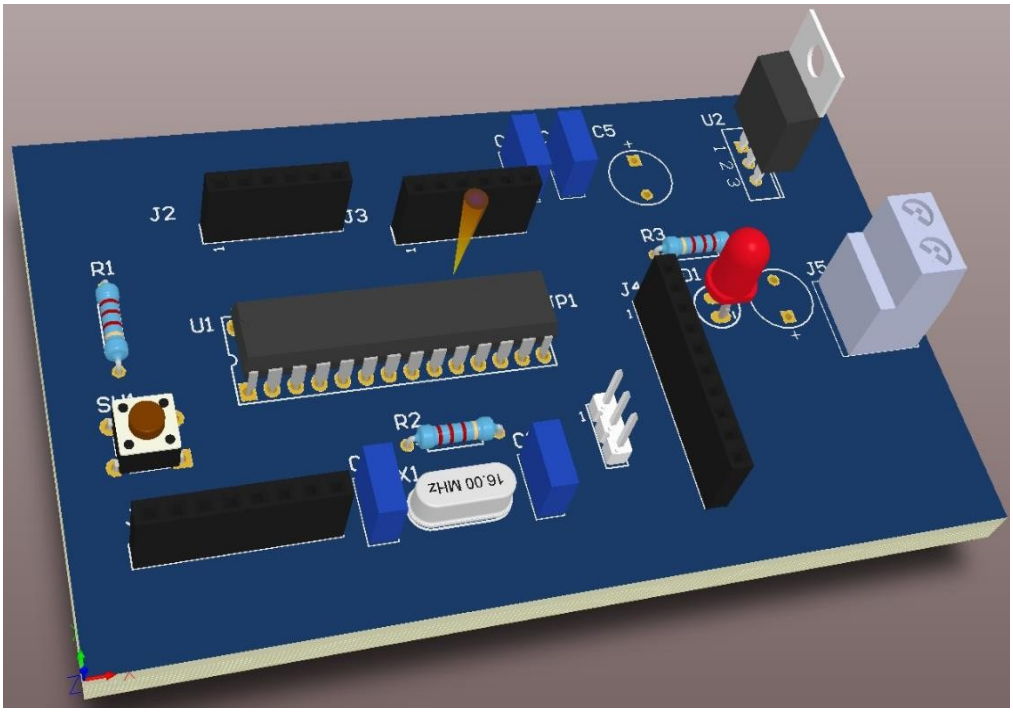
4.1. Altium Designer

Το λογισμικό Altium Designer αποτελεί ένα από τα πιο διαδεδομένα PCB προγράμματα σχεδίασης στην αγορά. Χρησιμοποιείται κυρίως, για την σχεδίαση του σχηματικού κυκλώματος, του PCB σχεδίου και της τρισδιάστατης μοντελοποίησης. Επίσης περιλαμβάνει εργαλεία για όλες τις εργασίες σχεδιασμού κυκλώματος όπως τη σχεδίαση σχηματικού, της γλώσσας προγραμματισμού HDL, η οποία χρησιμοποιείται για την αυτόματη ανάλυση και τη προσομοίωση ενός ηλεκτρονικού κυκλώματος, του σχεδιασμού PCB, τον σχεδιασμό και την ανάπτυξη ενσωματωμένου συστήματος και ολοκληρωμένου κυκλώματος FPGA (Field Programmable Gate Array).

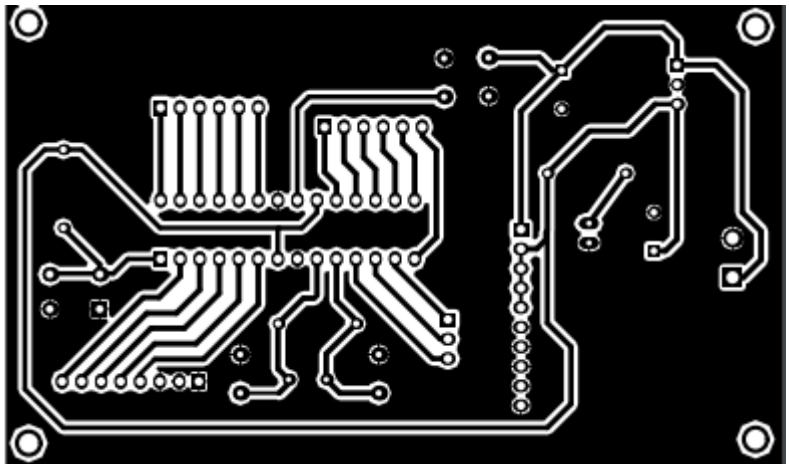
.



Εικόνα 4.1: Σχηματικό Κυκλώματος.



Εικόνα 4.2: Τρισδιάστατη απεικόνιση κυκλώματος.

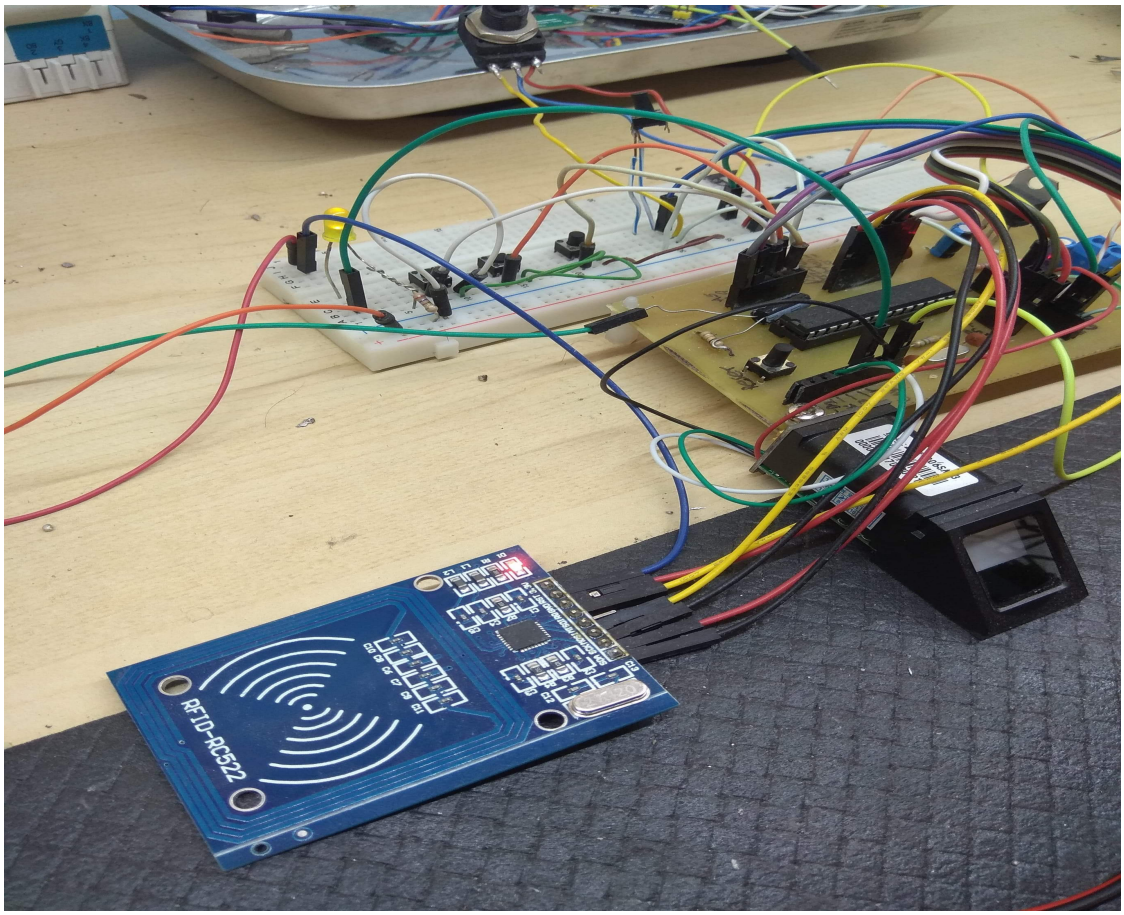


Εικόνα 4.3: Κάτοψη Πλακέτας

4.2 Διαδικασία κατασκευής της πλακέτας

Μετά τον σχεδιασμό του σχηματικού κυκλώματος στο πρόγραμμα Altium Designer, προχωρήσαμε στη μετατροπή του σχηματικού σε μία μονής όψεως πλακέτα. Η μέθοδος που επιλέξαμε για την μετατροπή του σχηματικού αναφέρεται ως φωτοχημική μέθοδος με χρήση φωτοευαίσθητης πλακέτας. Η διαδικασία που ακολουθήσαμε για το σχεδιασμό της πλακέτας έχει ως εξής:

1) Κόβουμε τη φωτοευαίσθητη πλακέτα στις διαστάσεις που χρειαζόμαστε.
2) Χαμηλώνουμε τα φώτα ή απομακρυνόμαστε από χώρους με έντονη φωτεινότητα, καθώς πρέπει να προστατέψουμε τη πλακέτα από κάθε είδους ακτινοβολία.
3) Τοποθετούμε τις ειδικές διαφάνειες ή φιλμ που έχουν αποτυπωμένο το σχηματικό κύκλωμα, κοντά στο λαμπτήρα που θα χρησιμοποιήσουμε για την έκθεση της πλακέτας.
4) Με σύντομες κινήσεις αφαιρούμε το προστατευτικό κάλυμμα της φωτοευαίσθητης πλακέτας και τη τοποθετούμε πάνω στις ειδικές διαφάνειες. Προσέχουμε η τυπωμένη επιφάνεια του φιλμ να εφαρμόζει στη πλακέτα. Επίσης, προσθέτουμε ένα βαρίδιο πάνω στη πλακέτα, ώστε όλη η επιφάνεια της πλακέτα να εφαρμόζει με το ειδικό φιλμ.
5) Ανάβουμε το λαμπτήρα και χρονομετρούμε περίπου 2.5 λεπτά.
6) Στη συνέχεια, τοποθετούμε τη φωτοευαίσθητη πλακέτα σε πλαστικό δοχείο το οποίο περιέχει διάλυμα με χλιαρό νερό και τη γνωστή σκόνη Tuboflo. Με το πέρασμα των δευτερολέπτων παρατηρούμε το σχηματικό να εμφανίζεται στη φωτοευαίσθητη πλακέτα.
7) Μετέπειτα, προχωράμε στην αποχάλκωση, δηλαδή τη τοποθέτηση της πλακέτας σε δοχείο όπου περιέχεται υγρό τριχλωριούχου σιδήρου για τριάντα λεπτά.
8) Το τελευταίο στάδιο της ολοκλήρωσης της πλακέτας, αφορά το τρύπημα της.



Εικόνα 4.4: Σύστημα Access Control, Όψη πλακέτας.

4.3. Λίστα υλικών κυκλώματος

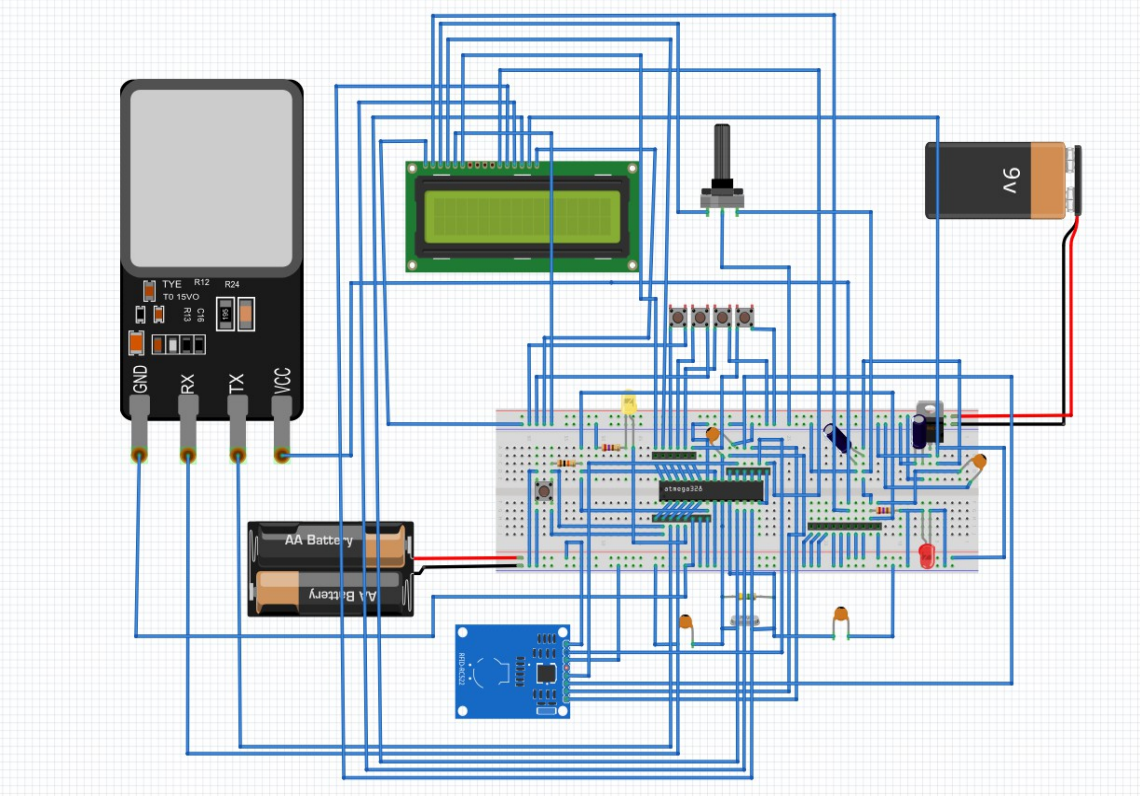
Υλικά πλακέτας:

1. Δύο ηλεκτρολυτικοί πυκνωτές 10μF.
2. Δύο κεραμικοί πυκνωτές τύπου (φακής) 100nF.
3. Δύο κεραμικοί πυκνωτές τύπου (φακής) 22pF.
4. Κρύσταλλος 16MHz..
5. Μία αντίσταση 10KΩ.
6. Μία αντίσταση 4.7MΩ.
7. Μία αντίσταση 270Ω.
8. Σταθεροποιητής τάσης τύπου CHN 522.
9. Ένα LED κόκκινου χρώματος.

Υλικά Breadboard:

1. Μία αντίσταση 270Ω.
2. Ένα ποτενσιόμετρο 10KΩ.
3. Τέσσερις διακόπτες (Buttons).
4. Διάταξη RFID τύπου (MFRC 522).
5. Αισθητήρας δακτυλικού αποτυπώματος (Adafruit σειράς ZFM-20).
6. Οθόνη LCD 16x2 Module.
7. Ένα LED κίτρινου χρώματος.

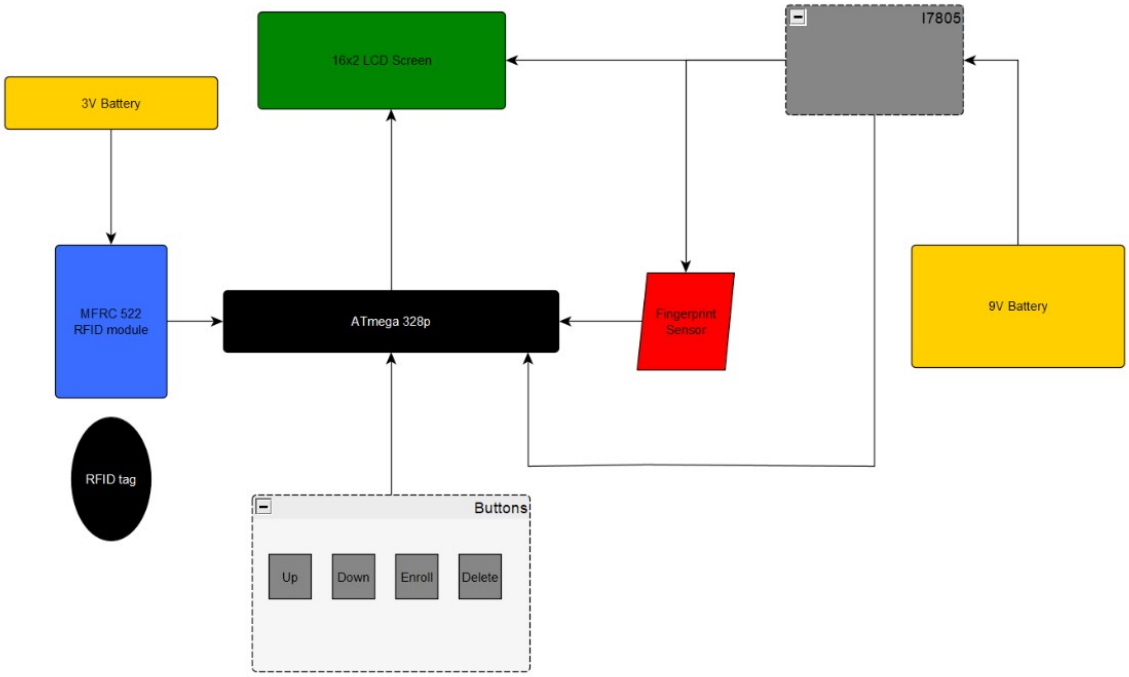
4.4. Ολοκληρωμένο σχέδιο κατασκευής



Εικόνα 4.5: Ολοκληρωμένο σχέδιο κατασκευής.

4.5. Ανάλυση μπλοκ διαγράμματος

Σε αυτό το σημείο της εργασίας, πραγματοποιείται εκτενής ανάλυση του κυκλώματος και του λειτουργικού συστήματος. Αρχικά, παρουσιάζουμε το Block διάγραμμα του συστήματος, διακρίνοντας τα φυσικά χαρακτηριστικά του. Μετέπειτα, γίνεται αναφορά στις υπομονάδες του συστήματος που συντέλεσαν στη κατασκευή και λειτουργία του κυκλώματος.



Εικόνα 4.6: Μπλοκ διάγραμμα συστήματος.

➔ ΠΕΡΙΦΕΡΕΙΑΚΕΣ ΜΟΝΑΔΕΣ

Για τη κατασκευή του συστήματος ασφαλείας της εργασίας, χρησιμοποιήθηκαν εκτός των βασικών μονάδων, όπως του μικροελεγκτή Atmega328P, του αισθητήρα δακτυλικού αποτυπώματος και της διάταξης RF-ID, επιπλέον υποσυστήματα ή αλλιώς περιφερειακές μονάδες. Τέτοιες περιφερειακές μονάδες αποτελούν οι διακόπτες (button), η οθόνη LCD και δύο μπαταρίες υπεύθυνες για την τροφοδοσία του κυκλώματος.

➔ ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΘΥΡΩΝ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ

Για την επικοινωνία του μικροελεγκτή με τα περιφερειακά του συστήματος, χρησιμοποιούνται κάποιες θύρες (πόρτες) εισόδου-εξόδου. Η χρήση αυτών περιγράφεται παρακάτω:

- PORT PD0, PD1 επικοινωνία μικροελεγκτή με τον αισθητήρα Fingerprint.
- PORT PB6, PB7 επικοινωνία μικροελεγκτή με RFID Module.
- PORT PD3, PD4, PB4, PB5 επικοινωνία μικροελεγκτή με οθόνη LCD.
- PORT PB0 – PB3 επικοινωνία μικροελεγκτή με διακόπτες (buttons) του συστήματος.

➔ ΤΡΟΦΟΔΟΣΙΑ ΣΥΣΤΗΜΑΤΟΣ

Για την ηλεκτρική παροχή ρεύματος της παρούσας εργασίας, έγινε χρήση μπαταριών 3 και 9Volt. Η μπαταρία των 3Volt χρησιμοποιήθηκε για την τροφοδοσία του RFID Module.Τα 9Volt της μπαταρίας, χρησιμοποιήθηκαν για την τροφοδοσία του υπόλοιπου κυκλώματος, ενώ παράλληλα προστέθηκε σταθεροποιητής τάσης στα 5Volt για την τροφοδοσία του αισθητήρα, των διακοπτών και της οθόνης LCD.

➔ ΟΘΟΝΗ LCD

Η οθόνη LCD είναι η μόνη περιφερειακή μονάδα του συστήματος, που λειτουργεί σαν έξοδος του μικροελεγκτή. Χρησιμοποιείται κυρίως, για την μεταφορά των μηνυμάτων του μικροελεγκτή προς τον χρήστη.

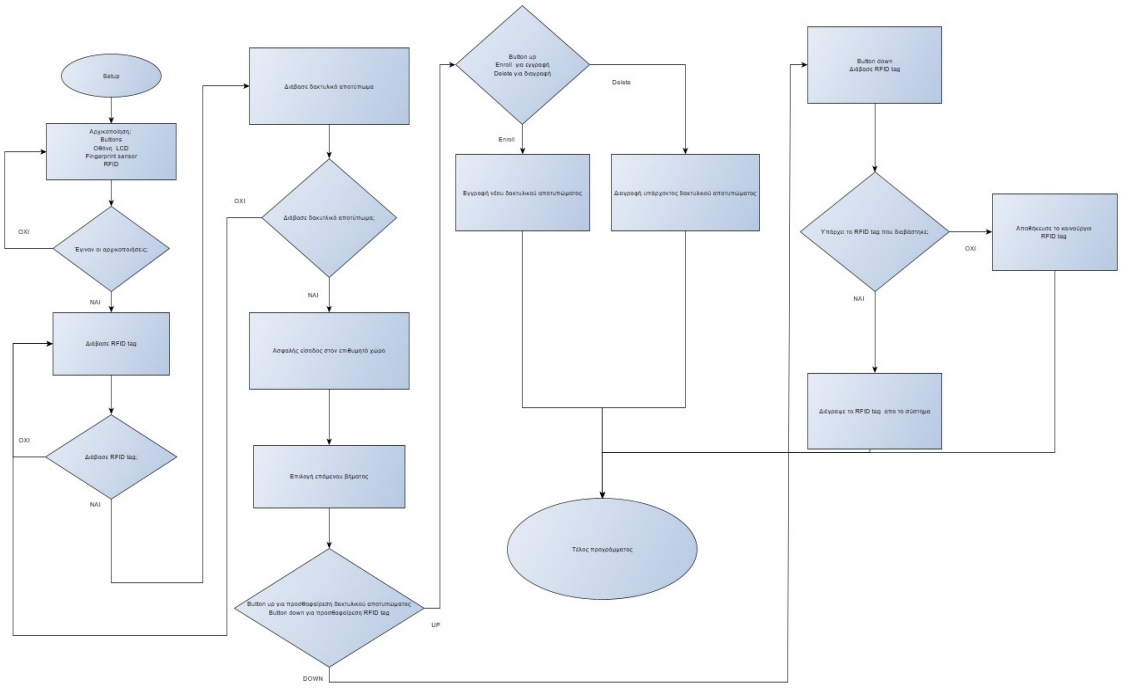
➔ MFRC 522 RFID Module

Το RFID Module επικοινωνεί με τον μικροελεγκτή, βάση του πρωτοκόλλου επικοινωνίας SPI, χρησιμοποιώντας τους ακροδέκτες (pin) 9 και 10.

➔ FINGERPRINT SENSOR

Ο αισθητήρας δακτυλικών αποτυπωμάτων επικοινωνεί με τον μικροελεγκτή, μέσω των ακροδεκτών 2 και 3. Το πρωτόκολλο επικοινωνίας που χρησιμοποιείται είναι το TTL Serial.

4.6. Διάγραμμα ροής συστήματος



Εικόνα 4.7: Διάγραμμα ροής συστήματος.

4.7. Δομή λογισμικού

Η λειτουργία και η ανάλυση του κυκλώματος, αποτελεί το πιο σημαντικό μέρος αυτής της εργασίας.

Ξεκινώντας, θα επικεντρωθούμε στη λειτουργία του κυκλώματος βήμα-βήμα.
Αρχικά, για να αποκτήσουμε πρόσβαση στο κύκλωμα, πρέπει να περάσουμε τους δύο ελέγχους πρόσβασης (access control) που μας εμφανίζονται. Ο πρώτος αφορά τον RFID αναγνώστη και ο δεύτερος τον αισθητήρα δακτυλικού αποτυπώματος.
Στην LCD οθόνη εμφανίζεται το μήνυμα (Place Rfid). Η επιτυχής πρόσβαση στο πρώτο στάδιο αναγνώρισης, οφείλεται στη χρήση της επιλεγμένης Hard ετικέτας, όπως και της επιλεγμένης κάρτας πρόσβασης. Εφόσον, οι ετικέτες που χρησιμοποιούνται είναι οι σωστές, τότε προχωρούμε στο επόμενο στάδιο ταυτοποίησης. Στη περίπτωση, που οι ετικέτες δεν είναι αποδεκτές, τότε στην οθόνη του κυκλώματος εμφανίζεται το μήνυμα (Rfid Not Found Try Later) και επιστρέφει στην αρχική του κατάσταση (Place Rfid).
Στο δεύτερο στάδιο ταυτοποίησης εμφανίζεται το μήνυμα (Place Finger). Εφόσον, το δακτυλικό αποτύπωμα είναι το αποδεκτό, τότε εμφανίζεται το μήνυμα (Access Granted Gate Closed) και πλέον έχουμε τη πλήρη πρόσβαση στο κύκλωμα. Στη περίπτωση που το δακτυλικό αποτύπωμα δεν είναι αποδεκτό, τότε εμφανίζεται το μήνυμα (Finger Not Found Try Later) και επιστρέφει στη κατάσταση (Place Finger).

```
87  void loop()
88  {
89
90   _Bool FlagRfid = true;
91   _Bool FlagFinger = true;
92
93    while(FlagRfid == true)
94    {
95      lcd.clear();
96      lcd.print("Place Rfid");
97
98      do {
99        successRead = getID();  // sets successRead to 1 when we get read from reader otherwise 0
100     }
101     while (!successRead);   //the program will not go further while you are not getting a successful read
102
103     if( isMaster(readCard) || findID(readCard) )
104     {
105       lcd.clear();
106       lcd.print("Rfid Pass");
107       delay(500);
108       if(isMaster(readCard))
109       {
110         FlagRfid = false;
111       }
112       break;
113     }
114     else{
115       lcd.clear();
116       lcd.print("Rfid not Pass");
117       delay(500);
118     }
119   }
```

Εικόνα 4.8: Διαδικασία προσπέλασης συστήματος ασφαλείας RFID.

```
119   }
120
121   SysTime = millis();
122   SysPrevTime = SysTime;
123
124   while(FlagFinger == true && (SysTime - SysPrevTime < Time_To_Place_Finger) )
125   {
126     lcd.clear();
127     lcd.print("Place Finger");
128     delay(1000);
129     int result=getFingerprintIDez();
130     if(result>=0)
131     {
132         digitalWrite(Led, HIGH);
133         lcd.clear();
134         lcd.print("Access Granted");
135         lcd.setCursor(0,1);
136         lcd.print("Gate Opened");
137         lcd.setCursor(0,1);
138         lcd.print("Gate Closed");
139         delay(2000);// kouts
140         digitalWrite(Led, LOW);// kouts
141         FlagFinger = false;
142         break;
143      }
144      SysTime = millis();
145    }
146
147   if(SysTime - SysPrevTime > Time_To_Place_Finger){
148     lcd.clear();
149     lcd.print("Timeout!!");
150     delay(1000);
151   }
```

Εικόνα 4.9: Διαδικασία προσπέλασης συστήματος ασφαλείας αισθητήρα Fingerprint.

Από τη στιγμή που αποκτούμε πρόσβαση στο κύκλωμα, εμφανίζεται το μήνυμα (Press Enroll or Delete).
Όταν το κύκλωμα βρίσκεται σε αυτή τη κατάσταση, μας δίνεται η δυνατότητα αποθήκευσης νέου RFID Tag ή δακτυλικού αποτυπώματος, ή διαγραφής της υπάρχουσας ετικέτας-αποτυπώματος.
Για λόγους ασφαλείας, υπάρχει χρονικό περιθώριο δέκα δευτερολέπτων, κατά τη διάρκεια του οποίου ο χρήστης πρέπει να αποκτήσει πρόσβαση στο σύστημα ή να επιλέξει ποια λειτουργία θα ακολουθήσει. Εφόσον ξεπεραστεί το χρονικό περιθώριο των δέκα δευτερολέπτων, επιστρέφει στην αρχική του κατάσταση.

Τα βήματα που ακολουθούμε για την αποθήκευση - διαγραφή RFID ετικετών είναι τα εξής:

Πατώντας το διακόπτη UP, εμφανίζεται το μήνυμα (Scan your tag).

➔Στη περίπτωση που το κύκλωμα ταυτοποιεί την ετικέτα, τότε την διαγράφει.
Μετέπειτα, επιστρέφει στην κατάσταση (Place Rfid). Για να έχουμε πρόσβαση στο σύστημα πρέπει να τοποθετήσουμε μια έγκυρη ετικέτα που είναι ήδη αποθηκευμένη στο κύκλωμα.

➔ Στη περίπτωση που το κύκλωμα δεν αναγνωρίζει την ετικέτα, τότε εμφανίζονται δύο μηνύματα (I don't know this tag) και (Add this tag), όπου και αποθηκεύει τη νέα ετικέτα. Στη συνέχεια επιστρέφει στην αρχική κατάσταση (Place Rfid) και ακολουθούμε την ίδια διαδικασία για την ταυτοποίηση του χρήστη.

```
if(FlagFinger == false)
{

  lcd.clear();
  lcd.print("Time to");
  lcd.setCursor(0,1);
  lcd.print("configuration");
  delay(2000);
  lcd.clear();
  lcd.print("Press Enroll or");
  lcd.setCursor(0,1);
  lcd.print("Delete");

  SysTime = millis();
  SysPrevTime = SysTime;

  while(SysTime - SysPrevTime < Time_For_Access_Configuration)
  {
    if(digitalRead(Button_up) == 0)
    {
     checkKeys();
    }
    else if (digitalRead(Button_down) == 0)
    {
      lcd.clear();
      lcd.print("Scan your");
      lcd.setCursor(0,1);
      lcd.print("tag");
      do
      {
        successRead = getID();  // sets successRead to 1 when we get read from reader otherwise 0
      }
      while (!successRead);   //the program will not go further while you are not getting a successful read
```

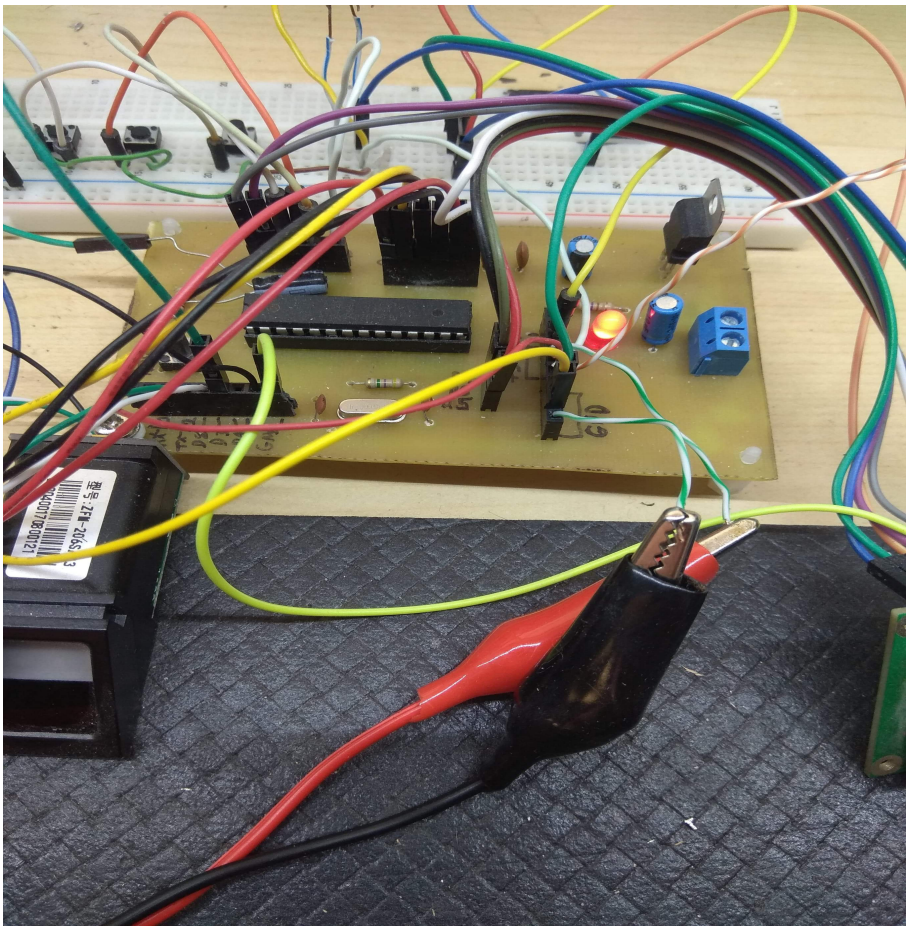
Εικόνα 4.10: Κώδικας παρεμβολής του χρήστη στην διαγραφή/καταγραφή στοιχείων ασφαλείας αισθητήρα δακτυλικού αποτυπώματος/RFID ετικέτας α).

```
      while (!successRead);   //the program will not go further while you are not getting a successful read

      if ( findID(readCard) ) { // If scanned card is known delete it
        lcd.clear();
        lcd.print("I know");
        lcd.setCursor(0,1);
        lcd.print("this tag");
        deleteID(readCard);
        lcd.clear();
        lcd.print("Detele");
        lcd.setCursor(0,1);
        lcd.print("this tag");
      }
      else {                    // If scanned card is not known add it
        lcd.clear();
        lcd.print("I dont know");
        lcd.setCursor(0,1);
        lcd.print("this tag");
        writeID(readCard);
        lcd.clear();
        lcd.print("Add");
        lcd.setCursor(0,1);
        lcd.print("this tag");
      }
    }
     SysTime = millis();
  }
}

}
```

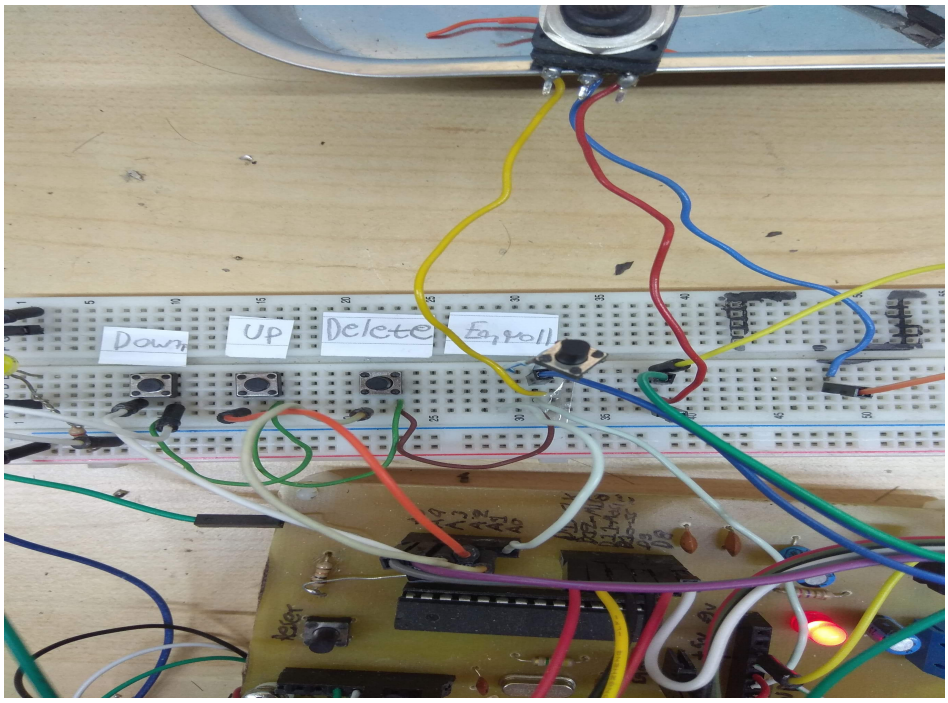
Εικόνα 4.11: Κώδικας παρεμβολής του χρήστη στην διαγραφή/καταγραφή στοιχείων ασφαλείας αισθητήρα δακτυλικού αποτυπώματος/RFID ετικέτας β).

Τα βήματα που ακολουθούμε για την αποθήκευση – διαγραφή δακτυλικού αποτυπώματος είναι τα εξής:

- Πατώντας ταυτόχρονα τους διακόπτες DOWN και Enroll, εμφανίζεται το μήνυμα (Enroll Finger Location:), όπου μας δίνεται η δυνατότητα να επιλέξουμε μία θέση από τις 162 που διαθέτει το κύκλωμα. Για την επιλογή θέσης χρησιμοποιούμε τους διακόπτες UP και DOWN. Εφόσον, επιλέξουμε τη θέση που θα αποθηκεύσουμε το αποτύπωμα, πατάμε το διακόπτη Enroll. Στη συνέχεια εμφανίζεται το μήνυμα (Finger ID: Place Finger), όπου τοποθετούμε το δακτυλικό αποτύπωμα. Περιμένουμε την εμφάνιση του μηνύματος (Remove Finger) για να απομακρύνουμε το δακτυλικό αποτύπωμα από τον αισθητήρα και στη συνέχεια με το μήνυμα (Stored!) μας ενημερώνει ότι το αποτύπωμα αποθηκεύτηκε. Τέλος επιστρέφει στην αρχική κατάσταση (Place Rfid), όπου συνεχίζεται η διαδικασία ταυτοπίησης.
- Πατώντας ταυτόχρονα τους διακόπτες DOWN και DELETE, εμφανίζεται το μήνυμα (Delete Finger Location:), όπου μας δίνεται η δυνατότητα να διαγράψουμε το αποτύπωμα σε μία από τις 162 θέσεις. Εφόσον επιλέξουμε ποιά θέση θα διαγράψουμε, πατάμε το διακόπτη Delete. Στη συνέχεια εμφανίζεται το μήνυμα (Finger Deleted Successfully), όπου το κύκλωμα μας ενημερώνει για την επιτυχή διαγραφή του δακτυλικού αποτυπώματος. Τέλος, επιστρέφει στην κατάσταση (Place Rfid), όπου συνεχίζεται η διαδικασία ταυτοποίησης.



Εικόνα 4.12: Τροφοδοσία Πλακέτας.



Εικόνα 4.13: Τοποθέτηση διακοπτών.

Μελλοντικές επεκτάσεις

Σε αυτό το σημείο, με την ολοκλήρωση της εργασίας, μπορούμε να προτείνουμε μελλοντικές τροποποιήσεις της εργασίας και του κυκλώματος, με σκοπό τη βελτίωση και εξέλιξή του. Μία από τις τροποποιήσεις που θα μπορούσαν να πραγματοποιηθούν, είναι η δυνατότητα του χρήστη να προσθέτει ή να διαγράφει RFID ετικέτες ή δακτυλικά αποτυπώματα, διαδοχικά, χωρίς να επιστρέφει στην αρχική του κατάσταση μετά την προσθήκη και διαγραφή μίας μόνο ετικέτας ή δακτυλικού αποτυπώματος.
Επόμενη βελτίωση, θα μπορούσε να θεωρηθεί η προσθήκη διακόπτη RESET.
Εφόσον ο χρήστης επιλέξει την επαναφορά του συστήματος στις εργοστασιακές του ρυθμίσεις, οι αποθηκευμένες RFID ετικέτες, όπως και τα δακτυλικά αποτυπώματα, να διαγράφονται αυτόματα, πλην της Master Card και του προκαθορισμένου από τις ρυθμίσεις, δακτυλικού αποτυπώματος.
Τέλος, σημαντική εξέλιξη του συστήματος αποτελεί, η ανάπτυξη λογισμικού στον ηλεκτρονικό υπολογιστή, όπου θα επικοινωνεί με τον μικροελεγκτή Atmega328P, καθώς και θα αποθηκεύει δεδομένα στις βάσεις δεδομένων που θα περιλαμβάνει.

Συμπεράσματα

Στη παρούσα πτυχιακή εργασία, προτάθηκε ο σχεδιασμός και η κατασκευή κυκλώματος Access Control, με τη χρήση δύο διαφορετικών τεχνολογιών ταυτοποίησης (RF-ID Card και αισθητήρα δακτυλικού αποτυπώματος). Αρχικά, αναφέραμε και αναλύσαμε διεξοδικά, τα τρία κύρια κομμάτια του συστήματος, που δεν είναι άλλα από το μικροελεγκτή, τη RF-ID τεχνολογία, όπως και τον αισθητήρα δακτυλικού αποτυπώματος. Σε αυτό το σημείο της εργασίας, μέσω της έρευνας που πραγματοποιήσαμε και των πληροφοριών που συλλέξαμε, προσπαθήσαμε να φανταστούμε την υλοποίηση του συστήματος, όπως και τη κατασκευή του κυκλώματος. Βάση των κριτηρίων της εργασίας αλλά και των δυνατοτήτων μας, προχωρήσαμε στην επιλογή του μικροελεγκτή (Atmega328P), του RF-ID συστήματος και του αισθητήρα δακτυλικού αποτυπώματος.
Μετέπειτα προχωρήσαμε στο κύριο μέρος της εργασίας, το οποίο αφορά τη σχεδίαση του κυκλώματος αλλά και τη σύνταξη του κώδικα. Στο σημείο αυτό, περιγράφουμε αναλυτικά τη λειτουργία του συστήματος, τη κατασκευή της πλακέτας και σχολιάζουμε τα κύρια μέρη του κώδικα.
Σαν συμπεράσματα της εργασίας, μπορούμε να θεωρήσουμε αρχικά,την επιτυχή κατασκευή και υλοποίηση του κυκλώματος σύμφωνα με τις απαιτήσεις της εργασίας ως ερευνητική. Στη συνέχεια συμπεράναμε κατά την διάρκεια των δοκιμών, τη δυνατότητα του μικροελεγκτή Atmega328P να εξυπηρετεί διάφορες εφαρμογές, ως ολοκληρωμένο κύκλωμα και τέλος τη δυνατότητα μελλοντικών επεκτάσεων της εργασίας εφόσον υπάρχουν οι κατάλληλοι πόροι.

Βιβλιογραφία

Data Sheet

[1] Atmel,"8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash",ATmega328P [DATASHEET] 7810D–AVR–01/15.

Internet Site

[2] Components Info,"ATmega328P Pinout Diagram, Pin Configuration, Brief Description & Datasheet",JANUARY 30, 2020, [Online]. Available: https://www.componentsinfo.com/atmega328p-pinout-configuration-datasheet/?fbclid=IwAR0EZjecreFYmehSlU89zK1WxEXV9sRCwcanFCRm84NweadQtMFYtjycMII.

[4] EMBEDDS, "Accessing AVR EEPROM memory in AVRGCC", [Online]. Available: https://embedds.com/accessing-avr-eeprom-memory-in-avrgcc/?fbclid=IwAR0EZjecreFYmehSlU89zK1WxEXV9sRCwcanFCRm84NweadQtMFYtjycMII.

[6] Wikipedia,"RFID",wikipedia, 9 April 2020, [Online]. Available:https://el.wikipedia.org/wiki/RFID.

[7] Suzanne Smiley,"17 Things You Might Not Know About Gen 2 RFID Tag Memory Banks",atlasRFIDstore, 1st Mar 2017,[Online]. Available: https://www.atlasrfidstore.com/rfid-insider/17-things-might-not-know-gen-2-rfid-tag-memory-banks.

[8] Suzanne Smiley,"Active RFID vs. Passive RFID: What's the Difference?",atlasRFIDstore,10th Dec 2019, [Online]. Available: https://www.atlasrfidstore.com/rfid-insider/active-rfid-vs-passive-rfid.

[9] Shain Armstrong,"Which RFID Frequency is Right for Your Application?",atlasRFIDstore, 29th Feb 2012, [Online]. Available: https://www.atlasrfidstore.com/rfid-insider/which-rfid-frequency-is-right-for-your-application.

[10] RFID4u,"how-to-select-a-correct-tag-frequency",rfid4u, [Online]. Available: https://rfid4u.com/rfid-basics-resources/how-to-select-a-correct-tag-frequency/?fbclid=IwAR2mgpsH7Wwler4sxj_sYJCW9ZdseZ-XVRddeAXrUpA5rajvvQ77XQnO2U0.

[11] NextPoints,"What are RFID tags and what are they used for?", NextPoints,València,Mirambell 35, [Online]. Available: https://nextpoints.com/en/rfid-blog/rfid-tags-types.

[12] smart-TEC,"rfid-technology",smart-tec, GmbH & Co. KG,Kolpingring 3,[Online]. Available: https://www.smart-tec.com/en/auto-id-world/rfid-technology.

[13] Kamran Ahsan,"RFID Components, Applications and System Integration with Healthcare Perspective",August 17th 2011, DOI: 10.5772/16968, [Online]. Available: https://www.intechopen.com/books/deploying-rfid-challenges-solutions-and-open-issues/rfid-components-applications-and-system-integration-with-healthcare-perspective?fbclid=IwAR2mgpsH7Wwler4sxj_sYJCW9ZdseZ-XVRddeAXrUpA5rajvvQ77XQnO2U0.

[14] Mehdia Ajana El Khaddar, Mohammed Boulmalf, Hamid Harroud, Mohammed Elkoutbi,"RFID Middleware Design and Architecture",Designing and Deploying RFID

Applications,June 2011, DOI: 10.3724/SP.J.1087.2008.001055, [Online]. Available: https://www.researchgate.net/publication/221912782_RFID_Middleware_Design_and_Architecture?fbclid=IwAR2mgpsH7Wwler4sxj_sYJCW9ZdseZ-XVRddeAXrUpA5rajvvQ77XQnO2U0.

[15] nordicid,"RFID VS BARCODE",28.08.19, [Online]. Available: https://www.nordicid.com.

[16] GROBOTRONICS,"Αναγνώστης NFC/RFID - MFRC-522 13.56MHz", [Online]. Available: https://grobotronics.com/mfrc-522-nfc-rfid-controller-breakout-board.html.

[18] GROBOTRONICS," Adafruit Αισθητήρας Δακτυλικών Αποτυπωμάτων", [Online]. Available: https://grobotronics.com/fingerprint-sensor.html.

[21] Alison Motluk," Fingerprints may illuminate life in the womb",NewScientist,30 November 2005.[Online] Available: https://www.newscientist.com/article/dn8396-fingerprints-may-illuminate-life-in-the-womb/?ignored=irrelevant&fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

[22] Miquel Gudino,"How Do Fingerprint Scanners Work? Optical vs Capacitive",arrow,1 February 2018.[Online] Available: https://www.arrow.com/en/research-and-events/articles/how-fingerprint-sensors-work?fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

E-BOOK

[19] Henry C.Lee, R. E. Gaensslen ," Advances in Fingerprint Technology", Taylor&FrancisGroup, Boca Raton FL,2001.[Online] Available: **https://books.google.dk/books?hl=da&lr=&id=xFnMBQAAQBAJ&oi=fnd&pg=PA1&dq=fingerprint%20technology%20history&ots=6C5XXH2-h3&sig=tc7vHWWpgyGxhPK2pLEi92wD_1Y&redir_esc=y&fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8#v=onepage&q=fingerprint%20technology%20history&f=false**.

[20] Hillary Moses Daluz," Fundamentals of FINGERPRINT ANALYSIS",Taylor&FrancisGroup, Boca Raton FL,2015. [Online] Available: **https://books.google.dk/books?id=p4xqBAAAQBAJ&pg=PA15&lpg=PA15&dq=fingerprints%20during%20qin%20dynasty&source=bl&ots=h6UL-sRdV3&sig=ACfU3U11ERkqc15SxkKo-d7YKu8glyqmvA&hl=da&sa=X&ved=2ahUKEwjo99y5pITqAhXBjqQKHT3SB7AQ6AEwC3oECAgQAQ&fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8#v=onepage&q=fingerprints%20during%20qin%20dynasty&f=false**.

[23] Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar," Handbook of Fingerprint Recognition",Springer, Verlag London Limited 2009.[Online] Available: **https://books.google.dk/books?id=1Wpx25D8qOwC&printsec=frontcover&dq=Handbook%20of%20Fingerprint%20Recognition&hl=da&sa=X&ved=0ahUKEwjchOjHj-3pAhVF3qQKHWz1DxYQ6AEIJzAA&fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8#v=onepage&q=Handbook%20of%20Fingerprint%20Recognition&f=false**.

Βιβλία

[3] William Stallings,"Computer Organization and Architecture, Designing for Performance, 8th Edition", ΤΖΙΟΛΑΣ, 2014.

[5] Σ.Μπουλταδάκης, Γ. Πατουλίδης, Ν. Ασημόπουλος, "ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ", ΘΕΣΣΑΛΟΝΙΚΗ, ΤΖΙΟΛΑΣ, 2011.

Internet Pictures

Εικόνα 1.1: "Μικροελεγκτής Atmega328P", Αναρτήθηκε από: https://en.wikipedia.org/wiki/ATmega328#/media/File:ATMEGA328P-PU.jpg.

Εικόνα 1.2: "Ακροδέκτες ATmega 328P", Αναρτήθηκε από: https://www.componentsinfo.com/atmega328p-pinout-configuration-datasheet/?fbclid=IwAR0EZjecreFYmehSlU89zK1WxEXV9sRCwcanFCRm84NweadQtMFYtjycMII.

Εικόνα 1.3: "Αρχιτεκτονική Atmega328P", Αναρτήθηκε από: https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html.

Εικόνα 1.4: "Μνήμη FLASH", Αναρτήθηκε από: https://ocw.aoc.ntua.gr/modules/document/file.php/ECE147/%CE%94%CE%B9%CE%B1%CF%86%CE%AC%CE%BD%CE%B5%CE%B9%CE%B5%CF%82/mP12-AVR.pdf?fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

Εικόνα 2.1: "Βασικά στοιχεία RFID συστήματος", Αναρτήθηκε από: https://www.elprocus.com/rfid-basic-introduction-simple-application/.

Εικόνα 2.2: "Hard Tags", Αναρτήθηκε από: https://www.inotecbsl.com/products/rfid-labels/rfid-hard-tags/.

Εικόνα 2.3: "Inlays Tags", Αναρτήθηκε από: https://www.amazon.com/YARONGTECH-860-960MHZ-Alien-73-5x21-2mm-Adhesive/dp/B01L97ULR4.

Εικόνα 2.4: "Ζώνη συχνοτήτων", Αναρτήθηκε από: http://www.seabreezerfid.com/rfid-frequencies-and-transmission-power.html?fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

Εικόνα 2.5: "α) Ενεργητική ετικέτα. β) Παθητική ετικέτα", Αναρτήθηκε από: https://medium.com/@verisium/how-nfc-tagging-technology-works-d730e56dbfde.

Εικόνα 2.6: "Handheld RFID Reader", Αναρτήθηκε από: http://www.rfidhandhelds.com/at870n-handheld-rfid-reader?fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

Εικόνα 2.7: "Fixed RFID Reader", Αναρτήθηκε από: https://www.zebra.com/us/en/products/rfid/rfid-readers.html?fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

Εικόνα 2.8: "Σύστημα RFID Middleware", Αναρτήθηκε από: https://www.researchgate.net/figure/Functions-of-the-RFID-middleware_fig6_278635444

Εικόνα 2.9: "Εκτυπωτής RFID", Αναρτήθηκε από: https://www.atlasrfidstore.com/rfid-insider/the-top-10-most-asked-questions-about-rfid-printers.

Εικόνα 2.10: "α) RFID Hard Ετικέτα. β) RFID Αναγνώστης. γ) RFID Κάρτα Ετικέτα", Αναρτήθηκε απόhttps: https://grobotronics.com/mfrc-522-nfc-rfid-controller-breakout-board.html.

Εικόνα 2.11: "Επαφές αναγνώστη (reader)", Αναρτήθηκε από: https://github.com/r00tGER/RFID-RC522.

Εικόνα 3.1: “Κατηγορίες σημείων πυρήνα ανθρώπινου δακτύλου”, Αναρτήθηκε από: https://lisa.ulb.ac.be/index.php/File:Class.jpg?fbclid=IwAR1BiQUjvrf9jQ-HbtvoP7Ex9j8BaOh1wQ2ohtI2FL99sKfcfZKo-LBkGM8.

Εικόνα 3.2: “Adafruit αισθητήρας οπτικής ανίχνευσης της σειράς ZFM-20”, Αναρτήθηκε από: https://grobotronics.com/fingerprint-sensor.html.

Παράρτημα

- **Κώδικας εφαρμογής**

codeprogramm_final

```
/*
 *Start LCD definitions
 */
#include<LiquidCrystal.h>
#define RS 18// kouts
#define EN 19// kouts
#define D4 8
#define D5 7
#define D6 6
#define D7 5
LiquidCrystal lcd(RS,EN,D4,D5,D6,D7);
/*
 * END
 */
 /*
  * Start FingerPrint definitions
  */
#include <SoftwareSerial.h>
#define RX 3
#define TX 2
SoftwareSerial fingerPrint(TX,RX);
#include <Adafruit_Fingerprint.h>
uint8_t id;
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerPrint);
void FingerInit();
/*
 * End
 */


 /* Start RFID definitions
  *
  */
#include <EEPROM.h>     // We are going to read and write PICC's UIDs from/to EEPROM
#include <SPI.h>        // RC522 Module uses SPI protocol
#include <MFRC522.h>  // Library for Mifare RC522 Devices

bool programMode = false;  // initialize programming mode to false

uint8_t successRead;    // Variable integer to keep if we have Successful Read from Reader

byte storedCard[4];   // Stores an ID read from EEPROM
byte readCard[4];   // Stores scanned ID read from RFID Module
byte masterCard[4];   // Stores master card's ID read from EEPROM
```

```
// Create MFRC522 instance.
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);

void Rfid_Init();
bool monitorWipeButton(uint32_t interval);
uint8_t getID();

 /*
  * END
  */

 /*
 * Start system definitions
 */
#define Button_enroll 14
#define Button_del 15
#define Button_up 16
#define Button_down 17
#define Led 4

#define Time_For_Access_Configuration 7000
#define Time_To_Place_Finger 10000

uint32_t SysTime = 0;
uint32_t SysPrevTime = 0;

void SysInit();

/*
 * End
 */

void setup()
{
    SysInit();
    FingerInit();
    Rfid_Init();
}

void loop()
{

 _Bool FlagRfid = true;
 _Bool FlagFinger = true;
```

```
while(FlagRfid == true)
{
  lcd.clear();
  lcd.print("Place Rfid");

  do {
    successRead = getID();  // sets successRead to 1 when we get read from reader otherwise 0
  }
  while (!successRead);   //the program will not go further while you are not getting a successful read

  if( isMaster(readCard) || findID(readCard) )
  {
    lcd.clear();
    lcd.print("Rfid Pass");
    delay(500);
    if(isMaster(readCard))
    {
      FlagRfid = false;
    }
    break;
  }
  else{
    lcd.clear();
    lcd.print("Rfid not Pass");
    delay(500);
  }
}

SysTime = millis();
SysPrevTime = SysTime;
```

```
while(FlagFinger == true && (SysTime - SysPrevTime < Time_To_Place_Finger) )
{
  lcd.clear();
  lcd.print("Place Finger");
  delay(1000);
  int result=getFingerprintIDez();
  if(result>=0)
  {
      digitalWrite(Led, HIGH);
      lcd.clear();
      lcd.print("Access Granted");
      lcd.setCursor(0,1);
      lcd.print("Gate Opened");
      lcd.setCursor(0,1);
      lcd.print("Gate Closed");
      delay(2000);// kouts
      digitalWrite(Led, LOW);// kouts
      FlagFinger = false;
      break;
   }
   SysTime = millis();
 }

if(SysTime - SysPrevTime > Time_To_Place_Finger){
  lcd.clear();
  lcd.print("Timeout!!");
  delay(1000);
}
```

```
if(FlagFinger == false)
{

  lcd.clear();
  lcd.print("Time to");
  lcd.setCursor(0,1);
  lcd.print("configuration");
  delay(2000);
  lcd.clear();
  lcd.print("Press Enroll or");
  lcd.setCursor(0,1);
  lcd.print("Delete");

  SysTime = millis();
  SysPrevTime = SysTime;

  while(SysTime - SysPrevTime < Time_For_Access_Configuration)
  {
    if(digitalRead(Button_up) == 0)
    {
     checkKeys();
    }
    else if (digitalRead(Button_down) == 0)
    {
      lcd.clear();
      lcd.print("Scan your");
      lcd.setCursor(0,1);
      lcd.print("tag");
      do
      {
        successRead = getID();  // sets successRead to 1 when we get read from reader otherwise 0
      }
      while (!successRead);   //the program will not go further while you are not getting a successful read

      if ( findID(readCard) ) { // If scanned card is known delete it
        lcd.clear();
        lcd.print("I know");
        lcd.setCursor(0,1);
        lcd.print("this tag");
        deleteID(readCard);
        lcd.clear();
        lcd.print("Detele");
        lcd.setCursor(0,1);
        lcd.print("this tag");
      }
```

```
        else {                    // If scanned card is not known add it
          lcd.clear();
          lcd.print("I dont know");
          lcd.setCursor(0,1);
          lcd.print("this tag");
          writeID(readCard);
          lcd.clear();
          lcd.print("Add");
          lcd.setCursor(0,1);
          lcd.print("this tag");
        }
      }
       SysTime = millis();
    }
  }

}

void SysInit()
{

    pinMode(Button_enroll, INPUT_PULLUP);
    pinMode(Button_up, INPUT_PULLUP);
    pinMode(Button_down, INPUT_PULLUP);
    pinMode(Button_del, INPUT_PULLUP);
    pinMode(Led, OUTPUT);
    delay(50);
    digitalWrite(Led, LOW);   // Turn on red LED

    lcd.begin(16,2);
    lcd.print("Fingerprint");
    lcd.setCursor(0,1);
    lcd.print("Access Control");
    delay(2000);
    lcd.clear();
    lcd.print("Koutsoupakis K.");
    lcd.setCursor(0,1);
    lcd.print("Lolas V.");
    delay(2000);

}
```

```
/* START FINGERPRINT FUNCTIONS */
void FingerInit()
{
    finger.begin(57600);
    //Serial.begin(9600);// kouts
    lcd.clear();
    lcd.print("Finding Module");
    lcd.setCursor(0,1);
    delay(1000);
    if (finger.verifyPassword())
    {
      lcd.clear();
      lcd.print("Sensor Detetected ");
      delay(1000);
    }
    else
    {
      lcd.clear();
      lcd.print("Sensor not Detetected");
      lcd.setCursor(0,1);
      lcd.print("Check Connections");
      while (1);
    }
}



void checkKeys()
{
   if(digitalRead(Button_enroll) == 0)
   {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    while(digitalRead(Button_enroll) == 0);
    Enroll();
   }
   else if(digitalRead(Button_del) == 0)
   {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    delet();
   }
}
```

```
void Enroll()
{
   int count=0;
   lcd.clear();
   lcd.print("Enroll Finger");
   lcd.setCursor(0,1);
   lcd.print("Location:");
   while(1)
   {
     lcd.setCursor(9,1);
     lcd.print(count);
     if(digitalRead(Button_up) == 0)
     {
       count++;
       if(count>25)
       count=0;
       delay(500);
     }
     else if(digitalRead(Button_down) == 0)
     {
       count--;
       if(count<0)
       count=25;
       delay(500);
     }
     else if(digitalRead(Button_del) == 0)
     {
          id=count;
          getFingerprintEnroll();
          return;
     }
     else if(digitalRead(Button_enroll) == 0)
     {
          return;
     }
 }
}
```

```
void delet()
{
   int count=0;
   lcd.clear();
   lcd.print("Delete Finger");
   lcd.setCursor(0,1);
   lcd.print("Location:");
   while(1)
   {
     lcd.setCursor(9,1);
     lcd.print(count);
     if(digitalRead(Button_up) == 0)
     {
       count++;
       if(count>25)
       count=0;
       delay(500);
     }

     else if(digitalRead(Button_down) == 0)
     {
       count--;
       if(count<0)
       count=25;
       delay(500);
     }
     else if(digitalRead(Button_del) == 0)
     {
          id=count;
          deleteFingerprint(id);
          return;
     }

       else if(digitalRead(Button_enroll) == 0)
     {
          return;
     }
 }
}
```

```
uint8_t getFingerprintEnroll()
{
  int p = -1;
  lcd.clear();
  lcd.print("Finger ID:");
  lcd.print(id);
  lcd.setCursor(0,1);
  lcd.print("Place Finger");
  delay(2000);
  while (p != FINGERPRINT_OK)
  {
    p = finger.getImage();
    switch (p)
    {
    case FINGERPRINT_OK:
      lcd.clear();
      lcd.print("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      lcd.clear();
      lcd.print("No Finger");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      lcd.clear();
      lcd.print("Communication");
      lcd.setCursor(0,1);
      lcd.print("Error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      lcd.clear();
      lcd.print("Imaging Error");
      break;
    default:
       lcd.clear();
      lcd.print("Unknown Error");
      break;
    }
  }
```

```
// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    lcd.clear();
    lcd.print("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
     lcd.clear();
     lcd.print("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    lcd.clear();
    lcd.print("Communication");
    lcd.setCursor(0,1);
    lcd.print("Error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    lcd.clear();
    lcd.print("Feature Not Found");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    lcd.clear();
    lcd.print("Feature Not Found");
    return p;
  default:
    lcd.clear();
    lcd.print("Unknown Error");
    return p;
}

lcd.clear();
lcd.print("Remove Finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
p = -1;
    lcd.clear();
    lcd.print("Place Finger");
    lcd.setCursor(0,1);
    lcd.print("Again");
```

```
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    lcd.print("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    lcd.print("...");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    lcd.print("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    lcd.print("Imaging error");
    break;
  default:
    lcd.print("Unknown error");
    return;
  }

 // OK success!

 p = finger.image2Tz(2);
 switch (p) {
   case FINGERPRINT_OK:
     lcd.print("Image converted");
     break;
   case FINGERPRINT_IMAGEMESS:
     lcd.print("Image too messy");
     return p;
   case FINGERPRINT_PACKETRECIEVEERR:
     lcd.print("Communication error");
     return p;
   case FINGERPRINT_FEATUREFAIL:
     lcd.print("Could not find");
     lcd.setCursor(0,1);
     lcd.print("fingerprint features");
     return p;
   case FINGERPRINT_INVALIDIMAGE:
    lcd.print("Could not find");
     lcd.setCursor(0,1);
     lcd.print("fingerprint features");
     return p;
   default:
     lcd.print("Unknown error");
     return p;
 }
```

```
  // OK converted!

  p = finger.createModel();
  if (p == FINGERPRINT_OK) {
    lcd.print("Prints matched!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    lcd.print("Communication");
    lcd.setCursor(0,1);
    lcd.print("Error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    lcd.print("Fingerprints");
    lcd.setCursor(0,1);
    lcd.print("did not match");
    return p;
  } else {
    lcd.print("Unknown error");
    return p;
  }

  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    lcd.clear();
    lcd.print("Stored!");
    delay(2000);
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    lcd.print("Communication");
    lcd.setCursor(0,1);
    lcd.print("Error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    lcd.print("Could not store");
    lcd.setCursor(0,1);
    lcd.print("in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    lcd.print("Error writing");
    lcd.setCursor(0,1);
    lcd.print("to flash memory");
    return p;
  }
  else {
    lcd.print("Unknown error");
    return p;
  }
}
```

```
int getFingerprintIDez()
{
  uint8_t p = finger.getImage();

  if (p != FINGERPRINT_OK)
  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)
  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)
  {
   lcd.clear();
   lcd.print("Finger Not Found");
   lcd.setCursor(0,1);
   lcd.print("Try Later");
   delay(2000);
  return -1;
  }
  // found a match!
  return finger.fingerID;
}


uint8_t deleteFingerprint(uint8_t id)
{
  uint8_t p = -1;
  lcd.clear();
  lcd.print("Please wait");
  p = finger.deleteModel(id);
  if (p == FINGERPRINT_OK)
  {
    lcd.clear();
    lcd.print("Finger Deleted");
    lcd.setCursor(0,1);
    lcd.print("Successfully");
    delay(1000);
  }

  else
  {
    lcd.clear();
    lcd.print("Something Wrong");
    lcd.setCursor(0,1);
    lcd.print("Try Again Later");
    delay(2000);
    return p;
  }
}

/* END FINGERPRINT FUNCTIONS */
```

```
/* START RFID FUNCTIONS */
void Rfid_Init()
{

    SPI.begin();           // MFRC522 Hardware uses SPI protocol
    mfrc522.PCD_Init();    // Initialize MFRC522 Hardware

    lcd.clear();
    byte vers = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
    if (vers == 0x91)
    {
      lcd.print("version =");
      lcd.print(vers);
    }
    else if (vers == 0x92)
    {
      lcd.print("version =");
      lcd.print(vers);
    }
    else if ( (vers == 0x00) || (vers == 0xFF))
    {
      lcd.print("Com error");
    }
    else
    {
      lcd.print("Unknown version");
    }
    delay(2500);
    lcd.clear();
    lcd.setCursor(0,0);
    if (digitalRead(Button_del) == LOW)
    {  // when button pressed pin should get low, button connected to ground
      digitalWrite(Led, HIGH); // Red Led stays on to inform user we are going to wipe
      lcd.print("Wipe Button");
      lcd.setCursor(0,1);
      lcd.print("Pressed");
      delay(2000);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Wipe EPPROM");
      lcd.setCursor(0,1);
      lcd.print("Hold it 10sec");
      delay(2000);
      lcd.clear();
      lcd.setCursor(0,0);
```

```
      bool buttonState = monitorWipeButton(10000); // Give user enough time to cancel operation
      if (buttonState == true && digitalRead(Button_del) == LOW) {    // If button still be pressed, wipe EEPROM
        lcd.print("Starting Wiping");
        lcd.setCursor(0,1);
        lcd.print("EEPROM");
        delay(2000);

        for (uint16_t x = 0; x < EEPROM.length(); x = x + 1) {    //Loop end of EEPROM address
          if (EEPROM.read(x) == 0) {              //If EEPROM address 0
            // do nothing, already clear, go to the next address in order to save time and reduce writes to EEPROM
          }
          else {
            EEPROM.write(x, 0);       // if not write 0 to clear, it takes 3.3mS
          }
        }

        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("EEPROM wiped");
        delay(2000);

        digitalWrite(Led, LOW);  // visualize a successful wipe
        delay(200);
        digitalWrite(Led, HIGH);
        delay(200);
        digitalWrite(Led, LOW);
        delay(200);
        digitalWrite(Led, HIGH);
        delay(200);
        digitalWrite(Led, LOW);
      }
      else {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Wiping");
        lcd.setCursor(0,1);
        lcd.print("Cancelled");
        delay(2000);
        digitalWrite(Led, LOW);
      }
    }
  }
```

```
bool monitorWipeButton(uint32_t interval) {
  uint32_t now = (uint32_t)millis();
  while ((uint32_t)millis() - now < interval)  {
    // check on every half a second
    if (((uint32_t)millis() % 500) == 0) {
      if (digitalRead(Button_del) != LOW)
        return false;
    }
  }
  return true;
}

uint8_t getID() {
  // Getting ready for Reading PICCs
  if ( ! mfrc522.PICC_IsNewCardPresent()) { //If a new PICC placed to RFID reader continue
    return 0;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) {   //Since a PICC placed get Serial and continue
    return 0;
  }
  // There are Mifare PICCs which have 4 byte or 7 byte UID care if you use 7 byte PICC
  // I think we should assume every PICC as they have 4 byte UID
  // Until we support 7 byte PICCs
  //Serial.println(F("Scanned PICC's UID:"));
  for ( uint8_t i = 0; i < 4; i++) {  //
    readCard[i] = mfrc522.uid.uidByte[i];
    //Serial.print(readCard[i], HEX);
  }
  mfrc522.PICC_HaltA(); // Stop reading
  return 1;
}
```

```
///////////////////////////////////////  Access Granted    ///////////////////////////////////
void granted ( uint16_t setDelay) {
  digitalWrite(Led, LOW);   // Turn off blue LED
  delay(1000);            // Hold green LED on for a second
}

///////////////////////////////////////// Access Denied  ///////////////////////////////////
void denied() {
  digitalWrite(Led, LOW);  // Make sure green LED is off
  delay(1000);
}

void cycleLeds() {
  digitalWrite(Led, LOW);  // Make sure red LED is off
  delay(200);
  digitalWrite(Led, HIGH);   // Make sure red LED is on
  delay(200);
}


//////////////////////////////////////// Read an ID from EEPROM //////////////////////////////
void readID( uint8_t number ) {
  uint8_t start = (number * 4 ) + 2;    // Figure out starting position
  for ( uint8_t i = 0; i < 4; i++ ) {     // Loop 4 times to get the 4 Bytes
    storedCard[i] = EEPROM.read(start + i);   // Assign values read from EEPROM to array
  }
}
```

```
//////////////////////////////////////// Add ID to EEPROM   ///////////////////////////////////
void writeID( byte a[] ) {
  if ( !findID( a ) ) {     // Before we write to the EEPROM, check to see if we have seen this card before!
    uint8_t num = EEPROM.read(0);     // Get the numer of used spaces, position 0 stores the number of ID cards
    uint8_t start = ( num * 4 ) + 6;  // Figure out where the next slot starts
    num++;                // Increment the counter by one
    EEPROM.write( 0, num );     // Write the new count to the counter
    for ( uint8_t j = 0; j < 4; j++ ) {   // Loop 4 times
      EEPROM.write( start + j, a[j] );  // Write the array values to EEPROM in the right position
    }
    successWrite();
  }
  else {
    failedWrite();
    lcd.clear();
    lcd.print("An error");
    lcd.setCursor(0,1);
    lcd.print("occurs");
    delay(2500);
  }
}


//////////////////////////////////////// Remove ID from EEPROM   ///////////////////////////////////
void deleteID( byte a[] ) {
  if ( !findID( a ) ) {     // Before we delete from the EEPROM, check to see if we have this card!
    failedWrite();      // If not
    lcd.clear();
    lcd.print("An error");
    lcd.setCursor(0,1);
    lcd.print("occurs");
    delay(2500);
  }
  else {
    uint8_t num = EEPROM.read(0);   // Get the numer of used spaces, position 0 stores the number of ID cards
    uint8_t slot;       // Figure out the slot number of the card
    uint8_t start;      // = ( num * 4 ) + 6; // Figure out where the next slot starts
    uint8_t looping;    // The number of times the loop repeats
    uint8_t j;
    uint8_t count = EEPROM.read(0); // Read the first Byte of EEPROM that stores number of cards
    slot = findIDSLOT( a );   // Figure out the slot number of the card to delete
    start = (slot * 4) + 2;
    looping = ((num - slot) * 4);
    num--;      // Decrement the counter by one
    EEPROM.write( 0, num );   // Write the new count to the counter
    for ( j = 0; j < looping; j++ ) {         // Loop the card shift times
      EEPROM.write( start + j, EEPROM.read(start + 4 + j));   // Shift the array values to 4 places earlier in the EEPROM
    }
    for ( uint8_t k = 0; k < 4; k++ ) {         // Shifting loop
      EEPROM.write( start + j + k, 0);
    }
    successDelete();
  }
}
```

```
///////////////////////////////////////// Check Bytes   ///////////////////////////////////
bool checkTwo ( byte a[], byte b[] ) {
  for ( uint8_t k = 0; k < 4; k++ ) {   // Loop 4 times
    if ( a[k] != b[k] ) {     // IF a != b then false, because: one fails, all fail
       return false;
    }
  }
  return true;
}

///////////////////////////////////////// Find Slot   ///////////////////////////////////
uint8_t findIDSLOT( byte find[] ) {
  uint8_t count = EEPROM.read(0);       // Read the first Byte of EEPROM that
  for ( uint8_t i = 1; i <= count; i++ ) {    // Loop once for each EEPROM entry
    readID(i);                // Read an ID from EEPROM, it is stored in storedCard[4]
    if ( checkTwo( find, storedCard ) ) {   // Check to see if the storedCard read from EEPROM
      // is the same as the find[] ID card passed
      return i;         // The slot number of the card
    }
  }
}

///////////////////////////////////////// Find ID From EEPROM   ///////////////////////////////////
bool findID( byte find[] ) {
  uint8_t count = EEPROM.read(0);     // Read the first Byte of EEPROM that
  for ( uint8_t i = 1; i < count; i++ ) {    // Loop once for each EEPROM entry
    readID(i);          // Read an ID from EEPROM, it is stored in storedCard[4]
    if ( checkTwo( find, storedCard ) ) {   // Check to see if the storedCard read from EEPROM
      return true;
    }
    else {    // If not, return false
    }
  }
  return false;
}
```

```
///////////////////////////////////////// Write Success to EEPROM   ///////////////////////////////////
// Flashes the green LED 3 times to indicate a successful write to EEPROM
void successWrite() {

  digitalWrite(Led, LOW);  // Make sure red LED is off

  delay(200);
  digitalWrite(Led, HIGH);   // Make sure green LED is on
  delay(200);
  digitalWrite(Led, LOW);  // Make sure green LED is off
  delay(200);
  digitalWrite(Led, HIGH);   // Make sure green LED is on
  delay(200);
  digitalWrite(Led, LOW);  // Make sure green LED is off
  delay(200);
  digitalWrite(Led, HIGH);   // Make sure green LED is on
  delay(200);
}

///////////////////////////////////////// Write Failed to EEPROM   ///////////////////////////////////
// Flashes the red LED 3 times to indicate a failed write to EEPROM
void failedWrite() {

  digitalWrite(Led, LOW);  // Make sure green LED is off
  delay(200);
  digitalWrite(Led, HIGH);   // Make sure red LED is on
  delay(200);
  digitalWrite(Led, LOW);  // Make sure red LED is off
  delay(200);
  digitalWrite(Led, HIGH);   // Make sure red LED is on
  delay(200);
  digitalWrite(Led, LOW);  // Make sure red LED is off
  delay(200);
  digitalWrite(Led, HIGH);   // Make sure red LED is on
  delay(200);
}


///////////////////////////////////////// Success Remove UID From EEPROM  ///////////////////////////////////
// Flashes the blue LED 3 times to indicate a success delete to EEPROM
void successDelete() {

  digitalWrite(Led, LOW);   // Make sure blue LED is off
  delay(200);
  digitalWrite(Led, HIGH);  // Make sure blue LED is on
  delay(200);
  digitalWrite(Led, LOW);   // Make sure blue LED is off
  delay(200);
  digitalWrite(Led, HIGH);  // Make sure blue LED is on
  delay(200);
  digitalWrite(Led, LOW);   // Make sure blue LED is off
  delay(200);
  digitalWrite(Led, HIGH);  // Make sure blue LED is on
  delay(200);
}

//////////////////////// Check readCard IF is masterCard   ///////////////////////////////////
// Check to see if the ID passed is the master programing card
bool isMaster( byte test[] ) {
  return checkTwo(test, masterCard);
}
/* END RFID FUNCTIONS */
```