



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«Ηλεκτρονική κλειδαριά ασφαλείας  
ελεγχόμενη από πληκτρολόγιο και i-button»**

*Τσαρτσαράκης Χρήστος*

*A.M. 3831*

**Επιβλέπων: Ρηγάκης Ηρακλής**  
*Εργαστηριακός συνεργάτης*

**Χανιά 2014**

## ΠΕΡΙΛΗΨΗ

Από την αρχαιότητα ο άνθρωπος είχε την επιθυμία να κλειδώνει πολύτιμα, για αυτόν, πράγματα όπως θησαυρούς, εμπορεύματα ή ακόμη και αντικείμενα συναισθηματικής αξίας. Με τον καιρό έβρισκε νέους και πιο ασφαλείς τρόπους για να το καταφέρει. Σ' αυτό το ζήτημα τον βοήθησε και η τεχνολογία. Αναπτύσσοντας τρόπους για την διευκόλυνση της ζωής του μέσω των ηλεκτρονικών ανέπτυξε και τεχνολογικά τεχνάσματα για την ασφάλεια του, και την ασφάλεια πολύτιμων αντικειμένων, είτε αυτά είναι εμπορεύματα, είτε προσωπικά του αντικείμενα, είτε αντικείμενα της δουλειάς του.

Σε αυτή την εργασία παρουσιάζεται ένα είδος ασφάλειας το οποίο προσφέρει επιλεγμένη πρόσβαση σε μέρη που φυλάσσονται πολύτιμα, για τον κάτοχο, αντικείμενα. Η συγκεκριμένη κατασκευή προσφέρει πρόσβαση σε εξουσιοδοτημένα και μόνο άτομα τα οποία πρέπει να περάσουν από διπλό έλεγχο για να περάσουν από μία πόρτα. Το πρώτο στάδιο ασφάλειας είναι μία συσκευή που ονομάζεται i-button το οποίο είναι ένα micro chip που περιέχει έναν κωδικό αναγνώρισης και όταν τοποθετηθεί στη συσκευή, στέλνει αυτό το κωδικό για να επαληθευτεί η εγκυρότητα του. Από την στιγμή που εγκριθεί η συσκευή τότε έρχεται το δεύτερο επίπεδο ασφάλειας που είναι ένας προσωπικός κωδικός pin, ξεχωριστός για το κάθε άτομο, για τον οποίο υπάρχει το πληκτρολόγιο, πάνω στην κατασκευή, για την εισαγωγή του. Για να περάσει κάποιος από την πόρτα πρέπει να εγκριθούν και τα δύο επίπεδα ασφάλειας. Η όλη διαδικασία επιτυγχάνεται με την βοήθεια μικροελεγκτή. Ο μικροελεγκτής λαμβάνει τα δεδομένα από τις εισόδους, που στην περίπτωση μας είναι το i-button, το πληκτρολόγιο και ο Η/Υ, τα επεξεργάζεται και στη συνέχεια τα στέλνει στις εξόδους, που είναι η lcd οθόνη, ο Η/Υ και η κλειδαριά. Ο Η/Υ θεωρείται και είσοδος και έξοδος για τον μικροελεγκτή. Είσοδος θεωρείται διότι στέλνει αποτελέσματα για την εγκυρότητα των κωδικών ασφαλείας και έξοδος θεωρείται διότι ο μικροελεγκτής του στέλνει τους κωδικούς για να πραγματοποιήσει τον έλεγχο. Η επικοινωνία του μικροελεγκτή με τον Η/Υ επιτυγχάνεται μέσω μιας θύρας USB.

# RESUME

From the ancient time the human had the desire to lock valuable items, such as treasures, merchandise or even items with sentimental value. Over time he found new and more secure ways to succeed it. In this issue the tech helped him. Developing ways of making his life easier through electronics tech, he developed tricks for his safety and for the security of his valuable items, whether they are goods, or personal items, or objects of his work.

This paper presents a kind of security that offers selected access to places where there are valuable objects. This construction provides access only to authorized persons who have to go double check to go through a door. The first stage of a safety is a devise called “ i-button “ which is a microchip that contain a unique code. When this i-button places in the apparatus, it sends this code to verify the validity of the i-button. Once the devise is approved then the second level of security that is a personal pin code comes, separate for each person, for whom there is a keyboard ( on the construction ) for the input of the pin code. Both security levels must be approved, if someone want to pass the door. The whole process is accomplished with the aid of a microcontroller. The microcontroller receives the data from the inputs, ( which in our case are the i-button, the keyboard and the pc), process and sends them to the outputs ( the lcd display, the pc and to lock). The pc is considered input and output for the microcontroller. Input is considered because it sends the results of the validity of the safety password and output is considered because the microcontroller sends the codes to perform the audit. The communication of the microcontroller to the pc is achieved via a USB port.

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΕΙΣΑΓΩΓΗ

### ΚΕΦΑΛΑΙΟ 1

- 1.1 Περιγραφή του 8051
- 1.2 Χρονισμός
- 1.3 Οργάνωση μνήμης
- 1.4 Οι εντολές του 8051
  - 1.4.1 Διευθυνσιοδότηση
  - 1.4.2 Αριθμητικές εντολές
  - 1.4.3 Λογικές πράξεις
  - 1.4.4 Εντολές μεταφοράς δεδομένων
  - 1.4.5 Εντολές αλμάτων
- 1.5 Εξυπηρέτηση διακοπών
- 1.6 Περιγραφή του i-button
- 1.7 Περιγραφή της database
- 1.8 Ηλεκτρονικές κλειδαριές που κυκλοφορούν στην αγορά

### ΚΕΦΑΛΑΙΟ 2

- 2.1 Λειτουργία προγράμματος
  - 2.1.1 Γενική λειτουργία του προγράμματος
  - 2.1.2 Διαδικασίες που εκτελεί κατά την εισαγωγή του i-button
  - 2.1.3 Διαδικασίες που εκτελεί μετά την επιβεβαίωση της εγκυρότητας της συσκευής
  - 2.1.4 Διαδικασίες που εκτελεί εάν ο κωδικός pin δεν είναι έγκυρος
- 2.2 επικοινωνία πλακέτας με υπολογιστή

## ΚΕΦΑΛΑΙΟ 3

3.1 Σχηματικό

3.2 Part list

3.3 Περιγραφή του σχηματικού

3.4 πλακέτα

ΣΧΟΛΙΑ

ΒΙΒΛΙΟΓΡΑΦΙΑ

## ΕΙΣΑΓΩΓΗ

Σε αυτή την εργασία παρουσιάζεται μία ηλεκτρονική κλειδαριά ασφαλείας.

Σε διάφορα κτίρια όπως σε υπηρεσίες, ξενοδοχεία, εργοστάσια κ.α. υπάρχουν δωμάτια στα οποία η πρόσβαση περιορίζεται σε λίγα άτομα οπότε σε αυτά πρέπει να υπάρχει κάποια αυξημένη ασφάλεια ώστε να ελέγχεται ποιος θα εισέρχεται και θα εξέρχεται από αυτά και ποιος επιτρέπεται να χρησιμοποιήσει αυτά τα δωμάτια. Επιπλέον σε πολλές περιπτώσεις είναι αναγκαίο να υπάρχει και καταγραφή των ατόμων για το πότε ακριβώς χρησιμοποίησαν την συγκεκριμένη πόρτα και πόση ώρα μείνανε μέσα. Αυτό που επιτυγχάνουμε με την κλειδαριά είναι να περιορίζεται η χρήση κάποιας πόρτας μόνο σε άτομα που τους επιτρέπεται η πρόσβαση. Η κλειδαριά παρέχει δύο επίπεδα ασφαλείας. Πρώτα χρειάζεται να γίνει αναγνώριση μιας συσκευής που ονομάζεται i-button και εφόσον η συγκεκριμένη συσκευή είναι έγκυρη από το σύστημα έρχεται το δεύτερο επίπεδο ασφαλείας το οποίο είναι ένας προσωπικός κωδικός pin και εάν είναι και αυτός έγκυρος τότε δίνεται εντολή να ανοίξει η πόρτα. Όταν δοθεί η εντολή για το άνοιγμα της πόρτας αυτόματα γίνεται και αποθήκευση των στοιχείων του ιδιοκτήτη του ibutton καθώς και η ώρα και η ημερομηνία που άνοιξε την πόρτα.

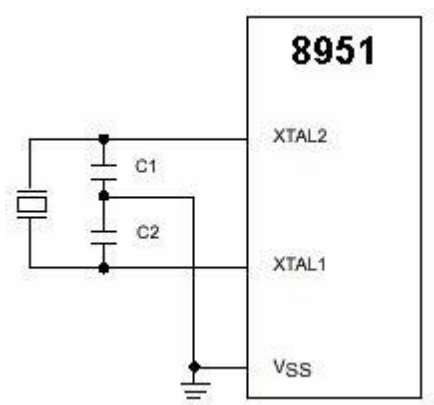
# ΚΕΦΑΛΑΙΟ 1

## 1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ 8051

Ο 8051 είναι ένας οκτάμπιτος μικροελεγκτής που πρωτοκατασκευάστηκε την δεκαετία του 80 από την INTEL. Κυκλοφορεί σε πολλές παραλλαγές από διάφορους κατασκευαστές και πολλές απ' αυτές περιλαμβάνουν ενσωματωμένα στον μικροελεγκτή αρκετά βοηθητικά περιφερειακά όπως timers, counters και σειριακή επικοινωνία. Επίσης ο μικροελεγκτής έχει ενσωματωμένη μνήμη ROM για την εγγραφή του κώδικα του προγράμματος και μνήμη RAM για την αποθήκευση δεδομένων κατά την διάρκεια εκτέλεσης του προγράμματος. Με όλες αυτές τις ευκολίες που παρέχουν οι κατασκευαστές ο μικροελεγκτής χρειάζεται ελάχιστα εξαρτήματα για την λειτουργία του. Φυσικά απαραίτητο είναι το πρόγραμμα που θα εκτελέσει ο μικροελεγκτής καθώς αυτό καθορίζει την όλη λειτουργία του. Το πρόγραμμα φορτώνεται στον μικροελεγκτή με κάποιο προγραμματιστή EPROM ή FLASH ROM ή ακόμα και σειριακά ανάλογα με τον τύπο του μικροελεγκτή. Τα αρχεία που φορτώνονται στη μνήμη του μικροελεγκτή είναι αρχεία HEX τα οποία μας δίνει ο compiler της γλώσσας προγραμματισμού που χρησιμοποιούμε. Για την κατασκευή μου χρησιμοποίησα την γλώσσα assembly καθώς αυτή χρησιμοποιείται στις περισσότερες εφαρμογές αυτού του είδους. Επίσης αρκετά χρησιμοποιείται και η γλώσσα C για τον προγραμματισμό μικροελεγκτή.

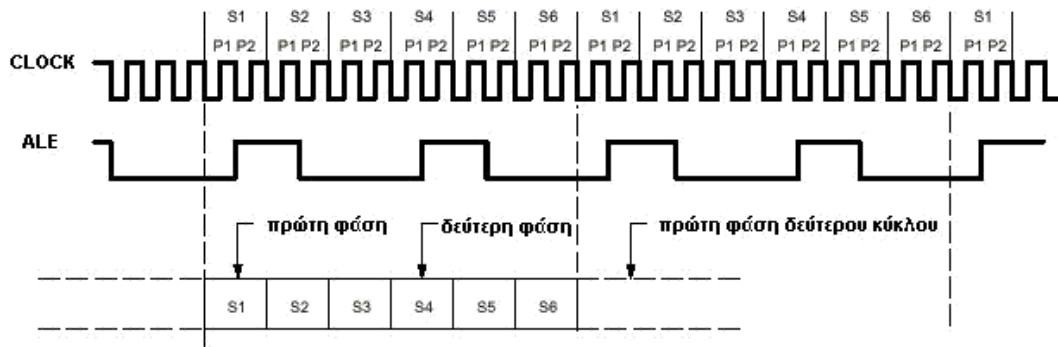
## 1.2 ΧΡΟΝΙΣΜΟΣ

Για να λειτουργήσει ο μικροελεγκτής χρειάζεται χρονισμό. Ο 8051 έχει ενσωματωμένο ταλαντωτή, για την πηγή των παλμών χρονισμού, ο οποίος για να λειτουργήσει πρέπει να συνδεθεί εξωτερικά του μικροελεγκτή ένας κρύσταλλος και δύο κεραμικοί πυκνωτές όπως φαίνεται στο σχήμα.



Η εσωτερική γεννήτρια παλμών καθορίζει το χρόνο που θα διαρκεί κάθε 'κύκλος μηχανής' κατά την εκτέλεση των εντολών. Από τον χρόνο αυτό εξαρτάται και η ταχύτητα με την οποία ο μικροελεγκτής θα εκτελεί τις εντολές του προγράμματος. Ένας κύκλος μηχανής αποτελείται από έξι καταστάσεις, S1 ως S6 στο σχήμα, και η κάθε κατάσταση από δύο περιόδους του ταλαντωτή. Οπότε ένας κύκλος μηχανής διαρκεί συνολικά 12 περιόδους του ταλαντωτή. Πιο συνηθισμένη είναι η χρήση ταλαντωτή στα 12 MHz, όπως και στην κατασκευή μας, οπότε η διάρκεια του κύκλου είναι 1  $\mu$ Sec.

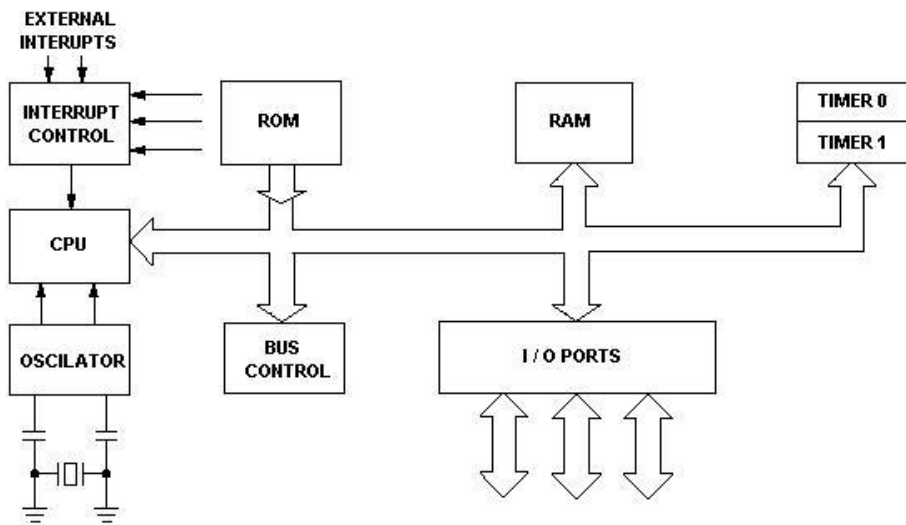




Κάποιες εντολές εκτελούνται σε ένα κύκλο όπως η “INC A” για παράδειγμα ενώ κάποιες άλλες απαιτούν περισσότερο χρόνο όπως η “DJNZ A” η οποία εκτελείται σε δύο κύκλους. Κάθε κύκλος έχει δύο φάσεις εκτέλεσης όπως φαίνεται στο σχήμα. Η πρώτη φάση αποτελείται από τις καταστάσεις S1, S2, S3 και η δεύτερη από τις καταστάσεις ξεκινάει από την κατάσταση S4 και διαρκεί ως την S6.

### 1.3 ΟΡΓΑΝΩΣΗ ΜΝΗΜΗΣ

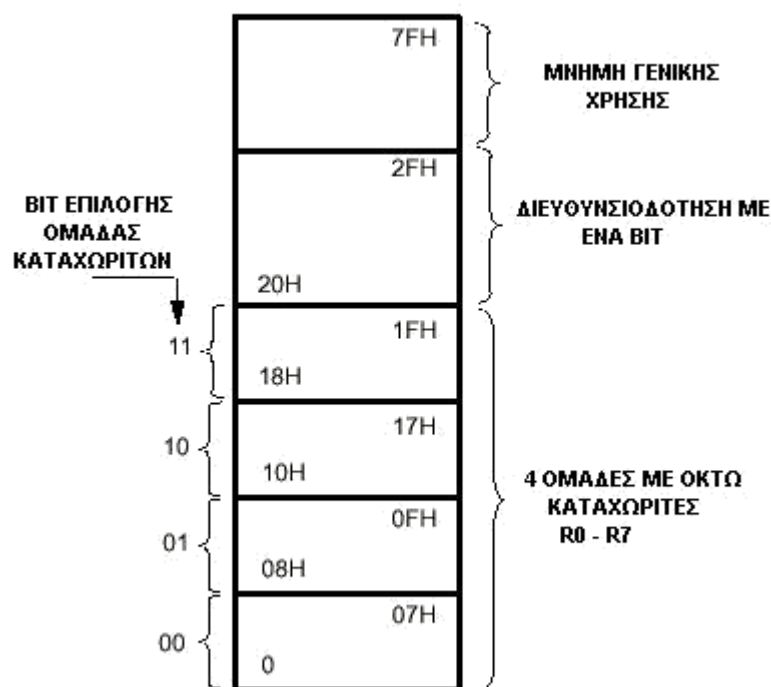
Ο 8051 κυκλοφορεί σε πολλές διαφορετικές εκδόσεις και οι περισσότερες από αυτές έχουν ενσωματωμένη μνήμη συνήθως 4 Kbs ROM και 256 bytes RAM. Ο μικροελεγκτής δέχεται και εξωτερική μνήμη και μπορεί να διαχειριστεί μέχρι 64 Kbs μνήμης η οποία διευθυνσιοδοτείται με διευθύνσεις μήκους 16 bit ( $2^{16} = 64$  Kbs). Για την προσπέλαση της εξωτερικής μνήμης χρησιμοποιούνται δύο ports και κάποια ακόμα βοηθητικά σήματα όπως το PSEN το RD και το WR. Επειδή γίνεται σαφής διαχωρισμός της εσωτερικής μνήμης ROM από την RAM η μνήμη RAM είναι ανεξάρτητη και μπορεί να διευθυνσιοδοτηθεί με οκτάμπιτες διευθύνσεις για περισσότερη ευκολία στον προγραμματισμό.



Το πρόγραμμα που θα εκτελεστεί είναι αποθηκευμένο στην μνήμη ROM. Όταν ο επεξεργαστής ξεκινάει την εκτέλεση διαβάζει από την θέση 0000H και εκεί πρέπει να βρίσκεται η πρώτη εντολή του προγράμματος. Επίσης κάθε αίτηση διακοπής (Interrupt) έχει μια προκαθορισμένη θέση στην μνήμη. Όταν γίνει αίτηση διακοπής στον επεξεργαστή αυτός θα ψάξει στην συγκεκριμένη θέση μνήμης για να βρει την πρώτη εντολή της ρουτίνας που θα εξυπηρετήσει την συγκεκριμένη διακοπή.

Το πρόγραμμα βρίσκεται αποθηκευμένο στη μνήμη ROM που είναι μόνο για ανάγνωση από τον μικροελεγκτή. Φυσικά αυτή δεν σβήνεται όταν ο μικροελεγκτής δεν τροφοδοτείται με ρεύμα και σ' αυτή την μνήμη μπορούμε να γράψουμε το πρόγραμμα όταν ο μικροελεγκτής βρίσκεται σε κατάσταση προγραμματισμού. Στη διάθεση μας έχουμε και κάποια μνήμη στην οποία μπορούμε να αποθηκεύουμε δεδομένα κατά την εκτέλεση του προγράμματος τα οποία θα χαθούν σε περίπτωση που διακοπεί η τροφοδοσία του μικροελεγκτή. Αυτή είναι η μνήμη RAM και είναι 256 bytes στην έκδοση του μικροελεγκτή που χρησιμοποιούμε. Τα 32 πρώτα bytes αυτής της μνήμης είναι χωρισμένα σε τέσσερις ομάδες των οκτώ καταχωρητών η κάθε μια. Οι καταχωρητές αυτοί είναι γενικής χρήσης και ονομάζονται R0 ως R7. Η επιλογή της ομάδας των καταχωρητών που θα χρησιμοποιηθεί γίνεται με δύο bit του καταχωρητή PSW (Program Status Word). Κατά την διάρκεια

εκτέλεσης του προγράμματος μπορούμε να αλλάζουμε ομάδα καταχωρητών και έτσι να έχουμε στη διάθεση μας 32 καταχωρητές αντί για 8. Με τον διαχωρισμό αυτό έχουμε αποτελεσματικότερη διαχείριση της μνήμης και ταχύτερη εκτέλεση του προγράμματος αφού οι εντολές με καταχωρητές καταλαμβάνουν λιγότερο χώρο καθώς έχουν μικρότερο μήκος από την απ' ευθείας διευθυνσιοδότηση στη μνήμη. Αμέσως μετά ακολουθούν 16 θέσεις μνήμης οι οποίες μπορούν να διευθυνσιοδοτηθούν με ένα μόνο bit και οι υπόλοιπες θέσεις είναι για γενική χρήση. Αυτά τα 128 bytes μνήμης μπορούν να προσπελαστούν είτε απ' ευθείας είτε έμμεσα και να χρησιμοποιηθούν όπως επιθυμεί ο προγραμματιστής για αποθήκευση δεδομένων.



Ο 8051 έχει άλλα 128 bytes μνήμης RAM τα οποία όμως δεν διευθυνσιοδοτούνται άμεσα και δεν μπορούμε να αποθηκεύουμε δεδομένα. Εκεί βρίσκονται οι διευθύνσεις των ports, οι καταχωρητές A και B, οι timers κ.τ.λ. Σε αυτές τις διευθύνσεις μπορούμε να χρησιμοποιήσουμε μόνο την έμμεση διευθυνσιοδότηση π.χ. SETB P1.0

Τέλος στην μνήμη υπάρχει και ο καταχωρητής PSW στον οποίο τα bits αντιπροσωπεύουν κάποια flags και τα δύο bits για την επιλογή της ομάδας καταχωρητών (Register Bank).

Συμβολισμός BIT	Διεύθυνση		Περιγραφή
CY	PSW.7	D7h	Carry flag
AC	PSW.6	D6h	Auxiliary carry flag
F0	PSW.5	D5h	Flag 0
RS1	PSW.4	D4h	Register bank select 1
RS0	PSW.3	D3h	Register bank select 0
OV	PSW.2	D2h	Overflow flag
	PSW.1	D1h	Reserved
P	PSW.0	D0h	Even Parity flag

## 1.4 ΟΙ ΕΝΤΟΛΕΣ 8051

### 1.4.1 Διευθυνσιοδότηση

Μπορούμε να προσπελάσουμε την εσωτερική μνήμη του μικροελεγκτή με διάφορους τρόπους. Ένας απ' αυτούς είναι η απ' ευθείας εγγραφή της διεύθυνσης που θέλουμε να γράψουμε ή να διαβάσουμε. Η διεύθυνση αποτελείται από 8 bit εφόσον ανήκει στην εσωτερική μνήμη RAM του μικροελεγκτή. Δεν μπορούμε μ' αυτόν τον τρόπο να προσπελάσουμε εξωτερική μνήμη RAM αν υπάρχει.

Ένας άλλος τρόπος είναι η έμμεση διευθυνσιοδότηση στην οποία η διεύθυνση που θέλουμε να προσπελάσουμε βρίσκεται σε κάποιο καταχωρητή. Για διευθύνσεις των 8 bits μπορούμε να χρησιμοποιήσουμε κάποιο καταχωρητή γενικής χρήσης όπως είναι οι καταχωρητές R0 ως R7 ή τον stack pointer. Για διευθύνσεις μήκους 16 bits μπορούμε να χρησιμοποιήσουμε μόνο τον DPTR. Με αυτόν τον τρόπο μπορούμε να προσπελάσουμε τόσο την εσωτερική όσο και την εξωτερική μνήμη RAM του μικροελεγκτή αν υπάρχει.

Μπορούμε εκτός από την μνήμη RAM να προσπελάσουμε και την μνήμη ROM του μικροελεγκτή χωρίς βέβαια να έχουμε την δυνατότητα να γράψουμε σ' αυτή. Αυτό είναι χρήσιμο για να διαβάσουμε από τον κώδικα του προγράμματος δεδομένα που είναι αποθηκευμένα σε μορφή πίνακα. Έτσι διαβάζουμε και στο πρόγραμμα μας τα μηνύματα που εμφανίζονται στην οθόνη υγρών κρυστάλλων αφού αυτά είναι αποθηκευμένα σαν πίνακες χαρακτήρων. Επειδή χρησιμοποιούμε διευθύνσεις των 16 bits για να προσπελάσουμε τη μνήμη ROM, η διεύθυνση “φορτώνεται” σε δεκαεξάμπιτο καταχωρητή όπως είναι ο DPTR.

## 1.4.2 Αριθμητικές εντολές

Οι αριθμητικές εντολές του 8051 φαίνονται στον παρακάτω πίνακα.

ΜΝΗΜΟΝΙΚΟ	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ $\mu\text{sec}$ ΜΕ 12 MHz clock
ADD A,<byte>	$A = A + \langle\text{byte}\rangle$	1
ADDC A,<byte>	$A = A + \langle\text{byte}\rangle + C$	1
SUBB A,<byte>	$A = A - \langle\text{byte}\rangle - C$	1
INC A	$A = A + 1$	1
INC <byte>	$\langle\text{byte}\rangle = \langle\text{byte}\rangle + 1$	1
INC DPTR	$\text{DPTR} = \text{DPTR} + 1$	2
DEC A	$A = A - 1$	1
DEC <byte>	$\langle\text{byte}\rangle = \langle\text{byte}\rangle - 1$	1
MUL AB	$B:A = B \times A$	4
DIV AB	$A = \text{Int}[A/B]$ $B = \text{Mod}[A/B]$	4
DA A	Decimal Adjust	1

Στον πίνακα φαίνεται και ο χρόνος που χρειάζεται για να εκτελεστεί η κάθε εντολή με τον χρονισμό των 12 MHz που χρησιμοποιούμε. Πρέπει να προσθέσουμε ότι κάθε byte της εσωτερικής RAM μπορεί να αυξηθεί ή να μειωθεί χωρίς να περάσει από τον accumulator. Το ίδιο μπορεί να γίνει και με τον καταχωρητή DPTR, το οποίο είναι χρήσιμο όταν αυτός χρησιμοποιείται για την προσπέλαση της εξωτερικής μνήμης,

καθώς γίνεται πιο γρήγορη η όλη διαδικασία.

Η εντολή 'MUL A,B' πολλαπλασιάζει το περιεχόμενο του καταχωρητή A με αυτό του B και βάζει το γινόμενο της πράξης και στους δύο καταχωρητές, κρατώντας στον A το λιγότερο σημαντικό μέρος και στον B το περισσότερο σημαντικό μέρος αν υπάρχει. Η εντολή 'DIV A,B' διαιρεί το περιεχόμενο του καταχωρητή A αυτό του B και μετά την πράξη ο A κρατάει το ακέραιο πηλίκο και ο B το υπόλοιπο.

### ***1.4.3 Λογικές πράξεις***

Στον παρακάτω πίνακα βλέπουμε τις εντολές των λογικών πράξεων που μπορεί να εκτελέσει ο 8051.

MNHMONIKO	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ΜΕ 12 MHz clock
ANL A,<byte>	A = A.AND. <byte>	1
ANL <byte>,A	<byte> = <byte> .AND.A	1
ANL <byte>,#data	<byte> = <byte> .AND.#data	2
ORL A,<byte>	A = A.OR.<byte>	1
ORL <byte>,A	<byte> = <byte> .OR.A	1
ORL <byte>,#data	<byte> = <byte> .OR.#data	2
XRL A,<byte>	A = A.XOR. <byte>	1
XRL <byte>,A	<byte> = <byte> .XOR.A	1
XRL <byte>,#data	<byte> = <byte> .XOR.#data	2
CRL A	A = 00H	1
CPL A	A = .NOT.A	1
RL A	Rotate ACC Left 1 bit	1
RLC A	Rotate Left through Carry	1
RR A	Rotate ACC Right 1 bit	1
RRC A	Rotate Right through Carry	1
SWAP A	Swap Nibbles in A	1



Στις λογικές πράξεις AND, OR, XOR και NOT μεταξύ 2 bytes η πράξη γίνεται bit προς bit. Για παράδειγμα η εντολή ‘ANL A, 0FH’ θα κάνει AND το περιεχόμενο του accumulator με τον δεκαεξαδικό αριθμό 0F που αντιστοιχεί στο δυαδικό 00001111. Δηλαδή αν ο accumulator έχει τον δυαδικό αριθμό 01010101 θα μας δώσει αποτέλεσμα 00000101 το οποίο θα μείνει στον accumulator. Λογικές πράξεις μπορούν να εκτελεστούν στην εσωτερική μνήμη RAM χωρίς τη μεσολάβηση του accumulator.

Οι εντολές περιστροφής μετατοπίζουν το περιεχόμενο του accumulator προς τα δεξιά ή προς τα αριστερά.

#### ***1.4.4 Εντολές μεταφοράς δεδομένων***

ΜΝΗΜΟΝΙΚΟ	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ΜΕ 12 MHz clock
MOV A,<src>	A = <src>	1
MOV <dest>,A	<dest> = A	1
MOV <dest>,<src>	<dest> = <src>	2
MOV DPTR,#data16	DPTR = 16-bit immed. constant	2
PUSH <src>	INC SP:MOV“@SP”,<src >	2
POP <dest>	MOV <dest>，“@SP”:DEC SP	2
XCH A,<byte>	ACC and <byte> exchange data	1
XCHD A,@Ri	ACC and @Ri exch. low nibbles	1

Στον παρακάτω πίνακα βλέπουμε τις εντολές που χρησιμοποιούμε στον 8051 για την μεταφορά δεδομένων από μια θέση μνήμης σε μια άλλη.

Με την εντολή MOV μπορούμε να μεταφέρουμε δεδομένα από μια θέση μνήμης σε μια άλλη χωρίς τη μεσολάβηση του accumulator. Οι εντολές μετακίνησης αφορούν δεδομένα μήκους 8 bit εκτός από την εντολή ‘MOV DPTR, #data’ με την οποία μπορούμε να μεταφέρουμε στον καταχωρητή DPTR 16 bit με μια μόνο εντολή. Αυτό είναι χρήσιμο για προσπέλαση δεδομένων σε πίνακες. Στον παρακάτω πίνακα φαίνονται δύο εντολές που χρησιμοποιούνται συχνά για το σκοπό αυτό.

ΜΝΗΜΟΝΙΚΟ	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ME 12 MHz clock
MOVC A,@A+DPTR	Read program memory at (A + DPTR)	2
MOVC A,@A+PC	Read program memory at (A + PC)	2

### 1.4.5 Εντολές αλμάτων

Στον παρακάτω πίνακα φαίνονται οι εντολές αλμάτων του 8051.

ΜΝΗΜΟΝΙΚΟ	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ µsec ΜΕ 12 MHz clock
JMP addr	Jump to address	2
JMP @A+DPTR	Jump to A + DPTR	2
CALL addr	Call subroutine at address	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1

Με τις εντολές αλμάτων μπορούμε να μετακινούμαστε μέσα στο πρόγραμμα. Οι εντολές του προγράμματος βρίσκονται αποθηκευμένες στην μνήμη με τη σειρά που θα εκτελεστούν και ο program counter κάθε φορά που εκτελείται μια εντολή αυξάνει κατά ένα έτσι ώστε να δείχνει την θέση της επόμενης εντολής που θα εκτελεστεί. Στην ουσία με τις εντολές αλμάτων δίνουμε στον program counter μια διαφορετική διεύθυνση από την ακριβώς επόμενη του και έτσι μπορούμε να μετακινηθούμε σε μια μακρινή θέση μνήμης και να εκτελέσουμε την εντολή που βρίσκεται εκεί. Αυτό είναι χρήσιμο όταν ελέγχουμε μια συνθήκη και πρέπει να εκτελεστούν άλλες εντολές αν αυτή ισχύει και άλλες αν δεν ισχύει. Επίσης εντολή άλματος είναι και η CALL με την οποία μετακινούμαστε στη θέση μνήμης που βρίσκεται μια ρουτίνα που θέλουμε να εκτελεστεί. Η ρουτίνα τελειώνει με την εντολή

RET με την οποία το πρόγραμμα κάνει και πάλι άλμα μία θέση μνήμης μετά από αυτή που κάλεσε την ρουτίνα για να συνεχιστεί η εκτέλεση του προγράμματος. Το ίδιο κάνει και η εντολή RETI μετά την εκτέλεση ρουτίνας εξυπηρέτησης μιας διακοπής.

Η εντολή JMP είναι μια γενική εντολή άλματος η οποία όμως έχει τρεις μορφές. Την SJMP την LJMP και την AJMP. Πιο συγκεκριμένα η SJMP δέχεται σαν διεύθυνση προορισμού 8 bit οπότε η μέγιστη μετακίνηση μπορεί να είναι  $-128$  ως  $+128$  bytes από την διεύθυνση της εντολής. Η εντολή LJMP δέχεται σαν διεύθυνση προορισμού 16 bit οπότε μπορούμε να έχουμε μετακίνηση 64 Kbytes δηλαδή οπουδήποτε μέσα στη μνήμη. Τέλος με την AJMP μπορούμε να κάνουμε άλματα εντός 2 Kbytes μέσα στο πρόγραμμα. Η διαφορά των τριών αυτών εντολών είναι στο μήκος της εντολής αφού όλη η εντολή αποτελείται από τον κωδικό της εντολής που είναι 8 bit και από την διεύθυνση προορισμού η οποία και διαφέρει σε καθεμία από τις εντολές. Κατά την δημιουργία του προγράμματος αν χρησιμοποιήσουμε για όλους τους τύπους άλματος την γενική εντολή JMP ο assembler κατά την μεταγλώττιση του προγράμματος θα την αντικαταστήσει με την πιο κατάλληλη από τις τρεις.

Τα ίδια ακριβώς ισχύουν και για την εντολή CALL. Η εντολή έχει δύο μορφές την LCALL και την ACALL για τις οποίες ισχύουν τα ίδια με την LJMP και την AJMP αντίστοιχα.

Οι παραπάνω εντολές αλμάτων δεν ελέγχουν κάποια συνθήκη αλλά κάνουν άλμα κατά την επιλογή του προγραμματιστή. Στον παρακάτω πίνακα βλέπουμε κάποιες ακόμα εντολές άλματος οι οποίες ελέγχουν και συνθήκη.

MNHMONIKO	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ΜΕ 12 MHz clock
JZ rel	Jump if A = 0	2
JNZ rel	Jump if A $\neq$ 0	2
DJNZ <byte>,rel	Decrease and jump if not zero	2
CJNE A,<byte>,rel	Jump if A $\neq$ <byte>	2
CJNE <byte>,#data,rel	Jump if <byte> $\neq$ #data	2

Με τις εντολές άλματος με συνθήκη μπορούμε να κάνουμε άλματα μέσα στο πρόγραμμα μέχρι 256 bytes από τη θέση που βρισκόμαστε. Οι εντολές JZ και JNZ ελέγχουν το περιεχόμενο του accumulator για την συνθήκη, οπότε πρέπει πρώτα να μεταφερθεί το περιεχόμενο της μνήμης που θέλουμε να ελεγχθεί στον accumulator για να γίνει ο έλεγχος. Ακόμα υπάρχουν οι εντολές DJNZ η οποία είναι χρήσιμη για τη δημιουργία βρόγχων επανάληψης και η CJNE για την σύγκριση δύο αριθμών.

## 1.5 ΕΞΥΠΗΡΕΤΗΣΗ ΔΙΑΚΟΠΩΝ

Ο μικροελεγκτής 8051 έχει πέντε διακοπές (interrupts), δύο εξωτερικές, δύο διακοπές από τους timers και μία από την σειριακή επικοινωνία. Κάθε διακοπή μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί ανάλογα με τις απαιτήσεις του προγράμματος. Επίσης υπάρχει η δυνατότητα να θέσουμε τη διακοπή σε υψηλή ή χαμηλή προτεραιότητα. Όλα αυτά μπορούν να γίνουν αλλάζοντας την τιμή δύο καταχωρητών, του ΙΕ και του ΙΡ. Αναλυτικότερα οι λειτουργίες φαίνονται στους παρακάτω πίνακες.

Καταχωρητής ΙΕ (interrupt enable register)									
	ΙΕ.7 ΕΑ	ΙΕ.6	ΙΕ.5	ΙΕ.4 ΕS	ΙΕ.3 ΕΤ1	ΙΕ.2 ΕΧ1	ΙΕ.1 ΕΤ0	ΙΕ.0 ΕΧ0	
ΙΕ.7	Απενεργοποιεί όλες τις διακοπές αν είναι 0								
ΙΕ.4	Απενεργοποιεί την διακοπή της σειριακής αν είναι 0								
ΙΕ.3	Απενεργοποιεί την διακοπή του timer 1 αν είναι 0								
ΙΕ.2	Απενεργοποιεί την εξωτερική διακοπή 1 αν είναι 0								
ΙΕ.1	Απενεργοποιεί την διακοπή του timer 0 αν είναι 0								
ΙΕ.0	Απενεργοποιεί την εξωτερική διακοπή 0 αν είναι 0								

Καταχωρητής IP (interrupt priority register)							
IP.7	IP.6	IP.5	IP.4 PS	IP.3 PT1	IP.2	IP.1	IP.0
IP.4		Προτεραιότητα της σειριακής διακοπής. Αν είναι 0 την θέτει σε χαμηλή προτεραιότητα					
IP.3		Προτεραιότητα της διακοπής του timer 1. Αν είναι 0 την θέτει σε χαμηλή προτεραιότητα					
IP.2		Προτεραιότητα της εξωτερικής διακοπής 1. Αν είναι 0 την θέτει σε χαμηλή προτεραιότητα					
IP.1		Προτεραιότητα της διακοπής του timer 0. Αν είναι 0 την θέτει σε χαμηλή προτεραιότητα					
IP.0		Προτεραιότητα της εξωτερικής διακοπής 0. Αν είναι 0 την θέτει σε χαμηλή προτεραιότητα					



## 1.6 Περιγραφή του i-button

Το i-button είναι μια συσκευή που σχεδιάστηκε από την Dallas Semiconductor Corp. Η συσκευή i-button είναι ένα chip το οποίο βρίσκεται μέσα σε ένα ατσάλινο κουτί πάχους 16mm. Η συσκευή i-button μπορεί να τοποθετηθεί σχεδόν οπουδήποτε και αυτό επειδή είναι αρκετά ανθεκτική σε αντίξοες συνθήκες και σε εσωτερικούς και σε εξωτερικούς χώρους. Είναι αρκετά μικρό και μπορεί να τοποθετηθεί στην θήκη με τα κλειδιά ή στο πορτοφόλι και να χρησιμοποιείται καθημερινά για εφαρμογές όπως ο έλεγχος πρόσβασης σε κτίρια και υπολογιστές, στη διαχείριση περιουσιακών στοιχείων καθώς και διάφορες εργασίες καταγραφής δεδομένων.

Η συσκευή i-button χρησιμοποιεί την ατσάλινη θήκη της ως μια διεπαφή ηλεκτρονικών επικοινωνιών. Κάθε θήκη έχει μία επαφή δεδομένων που ονομάζεται 'lid' και μία επαφή γείωσης που ονομάζεται 'base'. Κάθε μία από αυτές τις επαφές συνδέονται στο chip που βρίσκεται μέσα στην θήκη. Το 'lid' είναι το πάνω μέρος της θήκης και το 'base' το πλαϊνό και κάτω μέρος της θήκης. Τις δύο αυτές επαφές τις χωρίζει μία ροδέλα πολυπροπυλενίου. Ακουμπώντας την συσκευή i-button στις δύο αυτές επαφές μπορείς να επικοινωνήσεις με αυτήν μέσω του πρωτοκόλλου 1-wire το οποίο χωρίζεται σε δύο ταχύτητες επικοινωνίας :

1. Standard mode στα 16 kbps
2. Overdrive mode στα 142 kbps

Κάθε συσκευή i-button έχει μια μοναδική και αναλλοίωτη διεύθυνση χαραγμένη με laser πάνω στο chip που βρίσκεται μέσα στη θήκη. Η διεύθυνση μπορεί να χρησιμοποιηθεί ως ένα αναγνωριστικό για κάθε συσκευή i-button.

Αρκετές συσκευές μοιράζονται το ίδιο bus. Κάθε συσκευή έχει ένα μοναδικό 64-bit serial number. Τα λιγότερο σημαντικά ψηφία του serial number είναι ένας αριθμός των 8-bit που περιγράφει τον τύπο της συσκευής. Τα σημαντικότερα ψηφία είναι ένα 8-bit CRC. Υπάρχουν αρκετές στάνταρ εντολές εκπομπής καθώς και εντολές που χρησιμοποιούνται για την διευθυνσιοδότηση μιας συγκεκριμένης συσκευής. Η συσκευή που έχει οριστεί ως master μπορεί να στείλει μια εντολή και μετά την διεύθυνση της συσκευής. Η επόμενη εντολή εκτελείται μόνο από την συγκεκριμένη συσκευή.

Το πρωτόκολλο απαρίθμησης του 1-wire bus είναι ένας αλγόριθμος που χρησιμοποιεί η master συσκευή για να διαβάσει τις διευθύνσεις της κάθε συσκευής που είναι συνδεδεμένες στο bus. Από τη στιγμή που η διεύθυνση περιλαμβάνει τον τύπο της συσκευής και το CRC, η ανάκτηση των δεδομένων παρέχει μια αξιόπιστη απογραφή των συσκευών στο bus. Το 1-wire network της Dallas έχει υλοποιηθεί ως open drain master συσκευή η οποία είναι συνδεδεμένη σε μία ή περισσότερες open drain slave συσκευές. Μία απλή pull-up αντίσταση είναι κοινή σε όλες τις συσκευές και δίνει στο bus 3 ή 5 volts, και μπορεί να τροφοδοτήσει τις slave συσκευές. Η επικοινωνία μπορεί να επιτευχθεί εάν η master ή slave συσκευή είναι low, π.χ. εάν γειωθεί η pull-up αντίσταση μέσω της εξόδου του mosfet η μεταφορά δεδομένων των 16,3Kbit/s επιτυγχάνεται. Επίσης υπάρχει και το overdrive mode το οποίο επιταχύνει την επικοινωνία με ένα συντελεστή 10.

Η master συσκευή ξεκινάει την εκπομπή με ένα reset παλμό ο οποίος μεταδίδει στο wire λογικό 0 για 480 μs. Αυτό κάνει reset όλες τις slave συσκευές στο bus. Μετά από αυτό κάθε slave συσκευή εμφανίζεται με έναν “presence” παλμό που κρατάει low το bus περίπου για 60 μs αφού η master συσκευή έχει αποδεσμεύσει το bus.

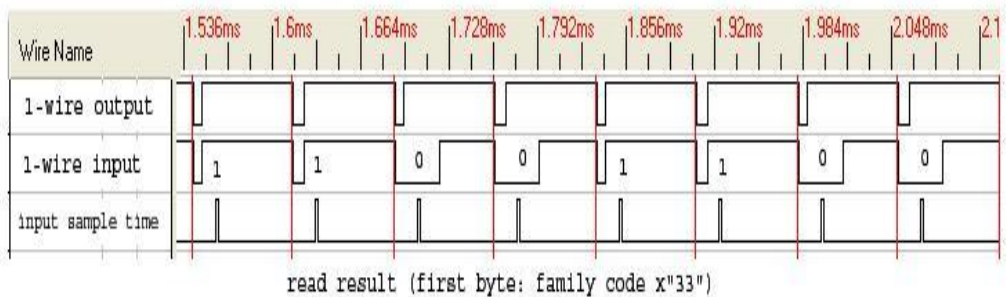
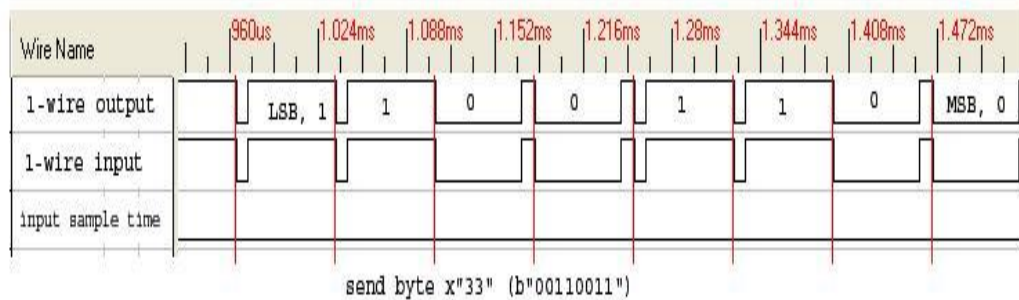
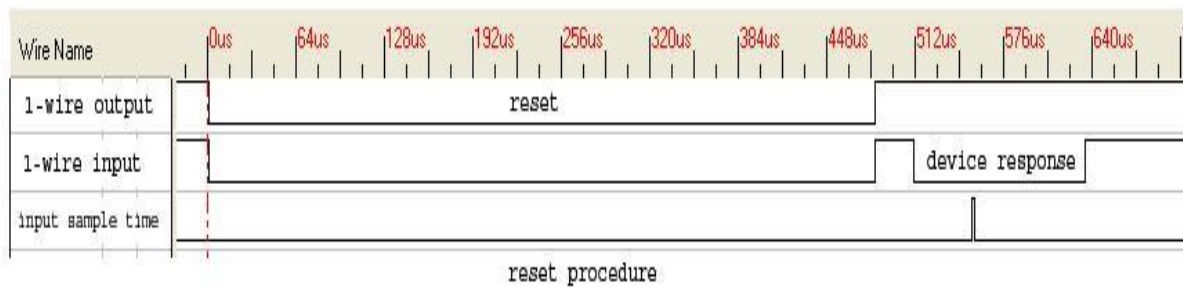
Για λογικό “1” το software της master συσκευής στέλνει ένα σύντομο low pulse (1-15 μs). Για λογικό “0” το software στέλνει ένα low pulse διάρκειας 60μs. Η αρνητική ημιπερίοδος χρησιμοποιείται για να ξεκινήσει ο μονοσταθής πολυδονητής της slave συσκευής. Ο πολυδονητής, αναπόφευκτα, έχει αναλογική ανοχή η οποία επηρεάζει την χρονική ακρίβεια και για αυτό το λόγο ο παλμός εξόδου είναι 60 μs και ο παλμός εκκίνησης δεν μπορεί να είναι μεγαλύτερος από 15 μs.

Όταν μια ειδική 1-wire περιφερειακή σύνδεση δεν είναι εφικτή, ένα UART μπορεί να χρησιμοποιηθεί για να εφαρμοστεί ένα 1-wire master bus. Διαθέσιμα επίσης είναι και τα σειριακά ή τα USB bridge chips τα οποία πληρούν τις προϋποθέσεις χρονισμού και κυματομορφών του 1-wire bus και είναι ιδιαιτέρως χρήσιμα σε μεγάλες αποστάσεις καλωδίων (μεγαλύτερο από 100m καλώδιο).

Όταν πάρει πληροφορία η master συσκευή στέλνει για 1-15μs ένα παλμό 0 volt. Εάν η slave μονάδα μετάδοσης στείλει λογικό “1” η κατάσταση του παλμού δεν αλλάζει και το bus μεταφέρεται στην Pull up ταση. Εάν η slave μονάδα μετάδοσης στείλει λογικό “0” η γραμμή πληροφορίας γειώνεται για 60 μs.

## Παράδειγμα επικοινωνίας με την συσκευή

1 Wire reset, write and read example with DS2432



## **1.7 Περιγραφή της database**

Με τον όρο βάση δεδομένων (database) εννοείται μία συλλογή από συστηματικά μορφοποιημένα σχετιζόμενα δεδομένα στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση. Ειδικότερα στην επιστήμη της πληροφορικής και στην καθημερινή χρήση των ηλεκτρονικών υπολογιστών, με τον όρο βάσεις δεδομένων αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων δεδομένων ηλεκτρονικά και ψηφιακά αποθηκευμένων, στο λογισμικό που χειρίζεται τέτοιες συλλογές και στο γνωστικό πεδίο που τις μελετά. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει μέσω του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

## 1.8 Ηλεκτρονικές κλειδαριές που κυκλοφορούν στην αγορά

### 1) RFID door locking system:

Πρόκειται για κλειδαριά ασφαλείας που έχει ως βασικό του χαρακτηριστικό τα ραδιοκύματα.

Τα συστήματα RFID απαρτίζονται από δύο κύρια μέρη. Το πρώτο είναι οι πομποδέκτες (transponders) που συχνά αναφέρονται και ως ετικέτες RFID (RFID tags). Οι ετικέτες RFID είναι μικρά chips που αποτελούνται από ένα ολοκληρωμένο κύκλωμα, το οποίο περιλαμβάνει μνήμη ώστε να αποθηκεύει δεδομένα- πληροφορίες, και μία κεραία. Το μέγεθός τους μπορεί να είναι τόσο μικρό όσο το μισό ενός κόκκου άμμου (1/3 του χιλιοστού), ανάλογα με το τύπο τις ετικέτας. Το δεύτερο μέρος είναι οι αναγνώστες ή αισθητήρες (readers), οι οποίοι ανακτούν τα δεδομένα από τις ετικέτες RFID. Οι αναγνώστες RFID έχουν ενσωματωμένα μια κεραία και μια μονάδα ελέγχου.

### 2) Smart card door locking system:

Η έξυπνη κάρτα ενσωματώνει ένα μικροεπεξεργαστή, ο οποίος βρίσκεται κάτω από μια επαφή από χρυσό, προσαρμοσμένο στη μια πλευρά της. Η τροφοδοσία της κάρτας με ενέργεια εξασφαλίζεται από τον αναγνώστη έξυπνης κάρτας (smart card reader), στον οποίο εισάγεται η κάρτα προκειμένου να χρησιμοποιηθεί. Αυτός μπορεί να επικοινωνήσει με κάποιο κεντρικό υπολογιστή, όπου υπάρχουν τα στοιχεία του χρήστη, προκειμένου να εξασφαλιστεί η πρόσβαση σε

δεδομένα. Η μνήμη RAM μιας έξυπνης κάρτας έχει μέγεθος μέχρι 8 Kbytes, η μνήμη ROM μέχρι 346 Kbytes, η μνήμη PROM (προγραμματιζόμενη ROM) μέχρι 256 kbytes. Ο μικροεπεξεργαστής είναι συνήθως 16 bytes, ενώ υποστηρίζει μικρή ομάδα εντολών (εξασφαλίζοντας μικρό μέγεθος), κυρίως αυτών που είναι απαραίτητες για την επικοινωνία με τον αναγνώστη καρτών / υπολογιστή και την κρυπτογράφηση των περιεχόμενων δεδομένων.

3) magnetic stripe card door locking system:

Αυτός ο τύπος κάρτας έχει την δυνατότητα αποθήκευσης δεδομένων με μαγνητικό τρόπο.

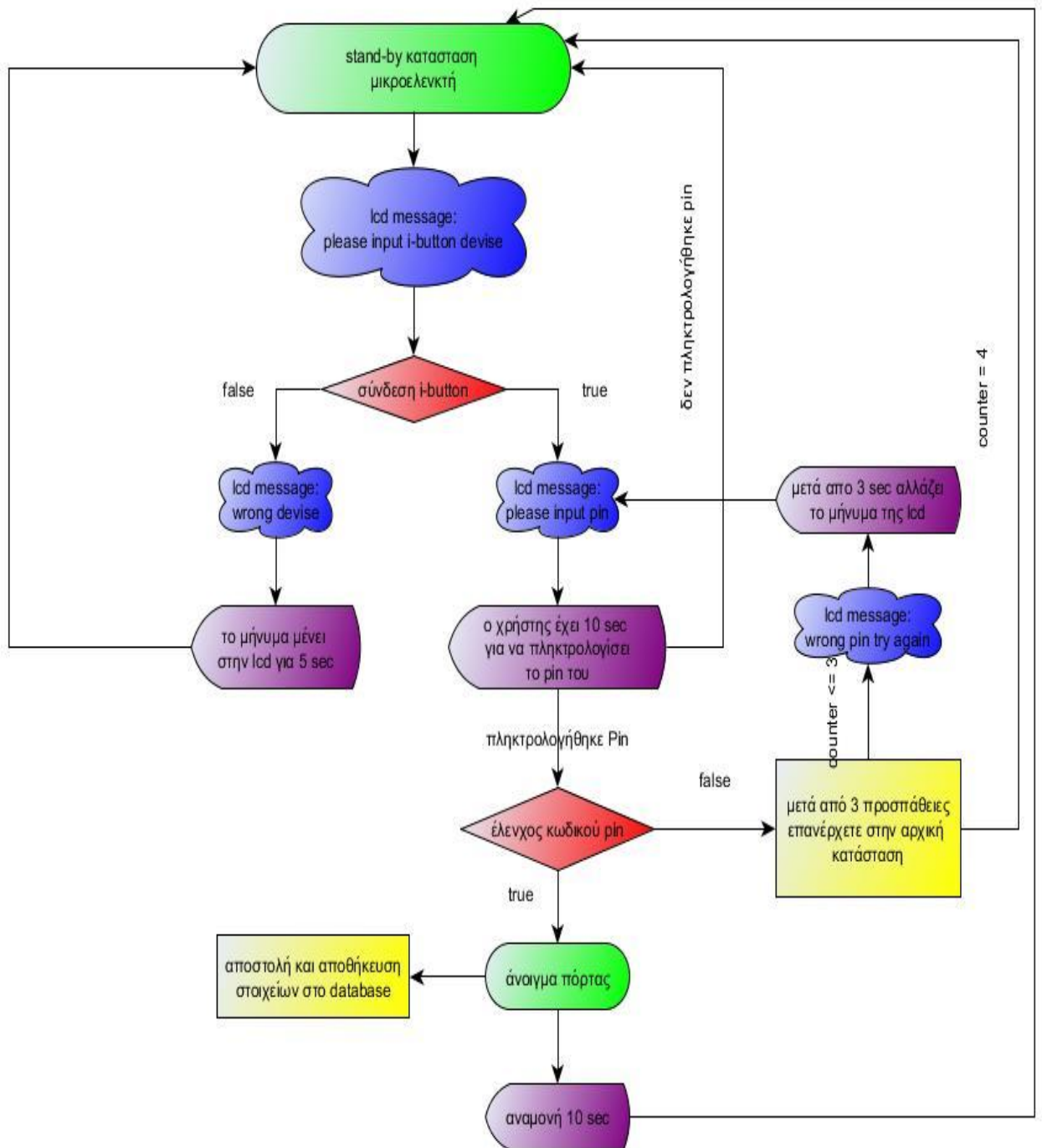
Μπορεί να αποθηκεύσει δεδομένα τροποποιώντας το μαγνητισμό των μικροσκοπικών, με βάση το σίδηρο, μαγνητικών σωματιδίων σε μια ταινία από μαγνητικό υλικό.

# ΚΕΦΑΛΑΙΟ 2

## 2.1 Λειτουργία προγράμματος

### 2.1.1 Γενική λειτουργία του προγράμματος

Ο μικροελεγκτής θα βρίσκεται σε μια κατάσταση stand-by περιμένοντας κάποιο interrupt από κάποια συσκευή 1-wire button (i-button) με ένα μήνυμα στην οθόνη που θα αναγράφει “please input i-button device”. Από τη στιγμή που θα δοθεί μήνυμα ότι έχει συνδεθεί κάποια συσκευή 1-wire button θα ξεκινήσει έλεγχο για το εάν η συσκευή είναι έγκυρη ή άκυρη. Εάν η συσκευή βρεθεί άκυρη τότε θα εμφανιστεί μήνυμα στην lcd το οποίο θα λέει “wrong device” και θα παραμένει για 5 sec εάν η συσκευή είναι έγκυρη θα εμφανιστεί μήνυμα στην lcd που θα λέει “please input your pin” και ο χρήστης θα έχει στην διάθεσή του 10 sec για να πληκτρολογήσει το pin του. Αν δεν πληκτρολογήσει το pin μέσα σε 10 sec θα γίνει reset στον μικροελεγκτή και θα επανέλθει στην αρχική του κατάσταση η οποία είναι το stand-by με το μήνυμα “please input i-button device” στην lcd του. Εφόσον πληκτρολογήσουμε το pin θα γίνει ένας ακόμα έλεγχος για το αν το pin είναι έγκυρο ή άκυρο. Εάν το pin είναι έγκυρο θα δώσει σήμα να στο relay ώστε να ανοίξει η πόρτα το οποίο θα διαρκεί 10 sec και στην συνέχεια θα κάνει reset στον μικροελεγκτή ώστε να επανέλθει στην stand-by λειτουργία του. Εάν το pin είναι άκυρο θα εμφανιστεί μήνυμα που θα λέει “false pin try again” και ο χρήστης θα έχει 3 προσπάθειες, μετά θα ακολουθεί reset του μικροελεγκτή.



δεν πληκτρολογήθηκε pin

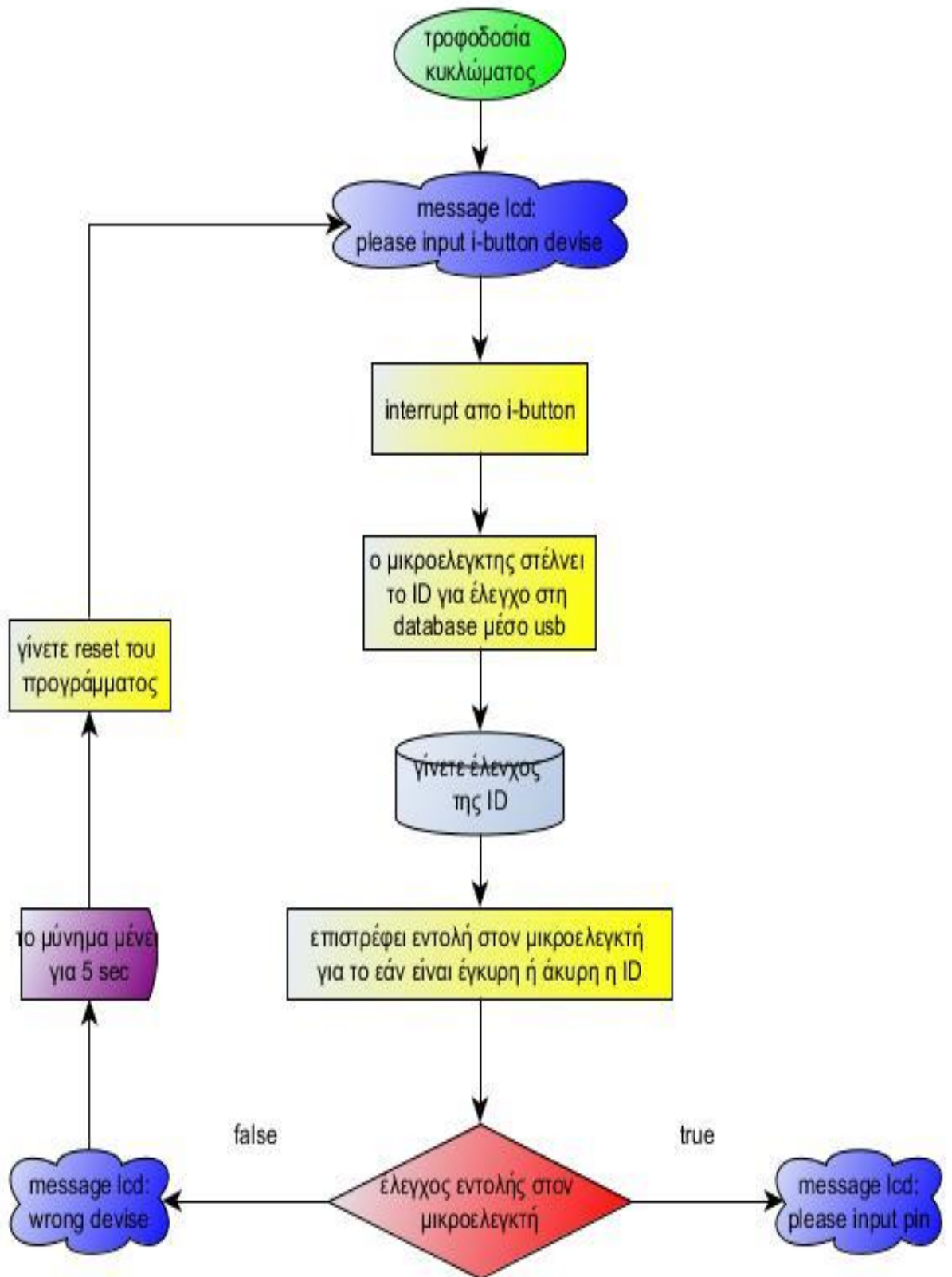
counter = 4

counter <= 3



### ***2.1.2 Διαδικασίες που εκτελεί κατά την εισαγωγή του i-button.***

Με το που δώσουμε τάση στην πλακέτα, στην οθόνη θα εμφανιστεί το μήνυμα “pleaseinputi-buttondevise” και θα περιμένει κάποιο interrupt από μια συσκευή i-button. Μόλις συνδεθεί η συσκευή i-button ο μικροελεγκτής θα στείλει σήμα στον υπολογιστή, που έχουμε συνδεδεμένο με την πλακέτα μέσω της θύρας usb, το οποίο θα ξεκινήσει αναζήτηση στην database για την εγκυρότητα της συσκευής που χρησιμοποιήθηκε . Στην database υπάρχουν αποθηκευμένα τα IDτων i-button καθώς και τα ονοματεπώνυμα και οι κωδικοίρίπτων χρηστών που τους επιτρέπετε η πρόσβαση. Μετά τον έλεγχο εάν η συσκευή είναι έγκυρη το πρόγραμμα θα προχωρήσει στο επόμενο βήμα, εάν η συσκευή i-button είναι άκυρη τότε θα στείλει μήνυμα στην οθόνη που θα αναγράφει για 5 sec“wrongdevise” και στη συνέχεια θα γίνετε reset του προγράμματος και θα επανέρχεται στην αρχική stand-by κατάσταση οποία είναι η lcd να αναγράφει το μήνυμα “pleaseinputi-buttondevise” και το πρόγραμμα να περιμένει να συνδεθεί νέα συσκευή.

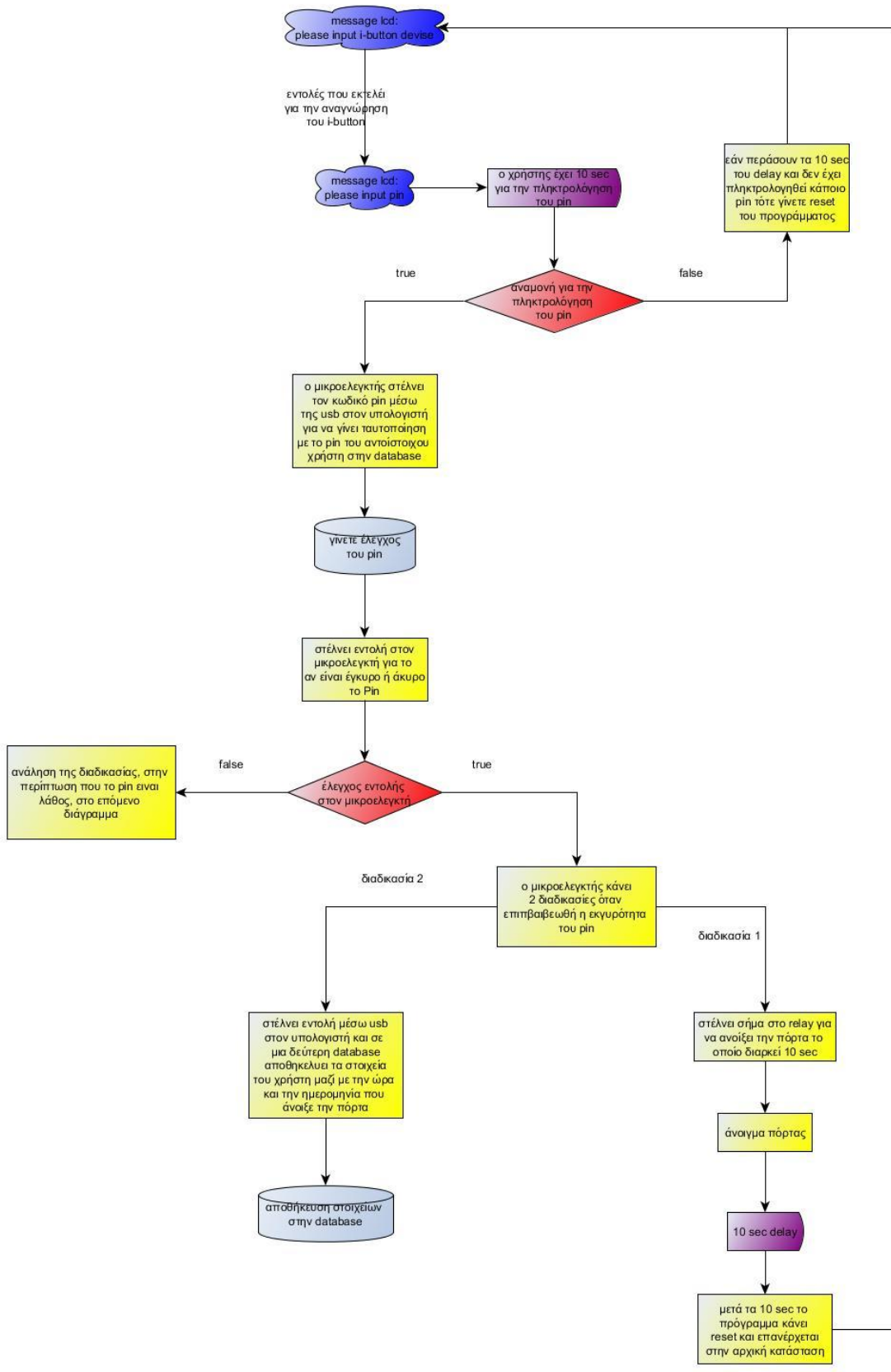


### ***2.1.3 Διαδικασίες που εκτελεί μετά την επιβεβαίωση της εγκυρότητας της συσκευής.***

Από την στιγμή που επιβεβαιωθεί η εγκυρότητα της συσκευής i-button το αμέσως επόμενο βήμα του προγράμματος είναι να στείλει μήνυμα στην οθόνη που θα αναγράφει “pleaseinputpin” το οποίο θα παραμένει για 10sec και το πρόγραμμα θα περιμένει να πληκτρολογηθεί ένας τετραψήφιος κωδικός pin. Εάν δεν πληκτρολογηθεί κάποιος κωδικός μέσα σε αυτά τα 10sec τότε θα γίνεται reset του προγράμματος και θα επανέρχεται στην stand-by κατάσταση. Μόλις ο κωδικός πληκτρολογηθεί τότε ο μικροελεγκτής θα στείλει σήμα μέσω της usb για να ελέγξει την εγκυρότητα του pin στην database όπου είναι αποθηκευμένα τα στοιχεία του χρήστη της συσκευής. Από την στιγμή που και ο κωδικός pin είναι έγκυρος τότε το πρόγραμμα θα δώσει τάση στο relay, για 10sec, ώστε να ανοίξει η πόρτα και θα στείλει μέσω usb εντολή να αποθηκευτούν σε μία άλλη database τα στοιχεία του χρήστη τα οποία θα είναι

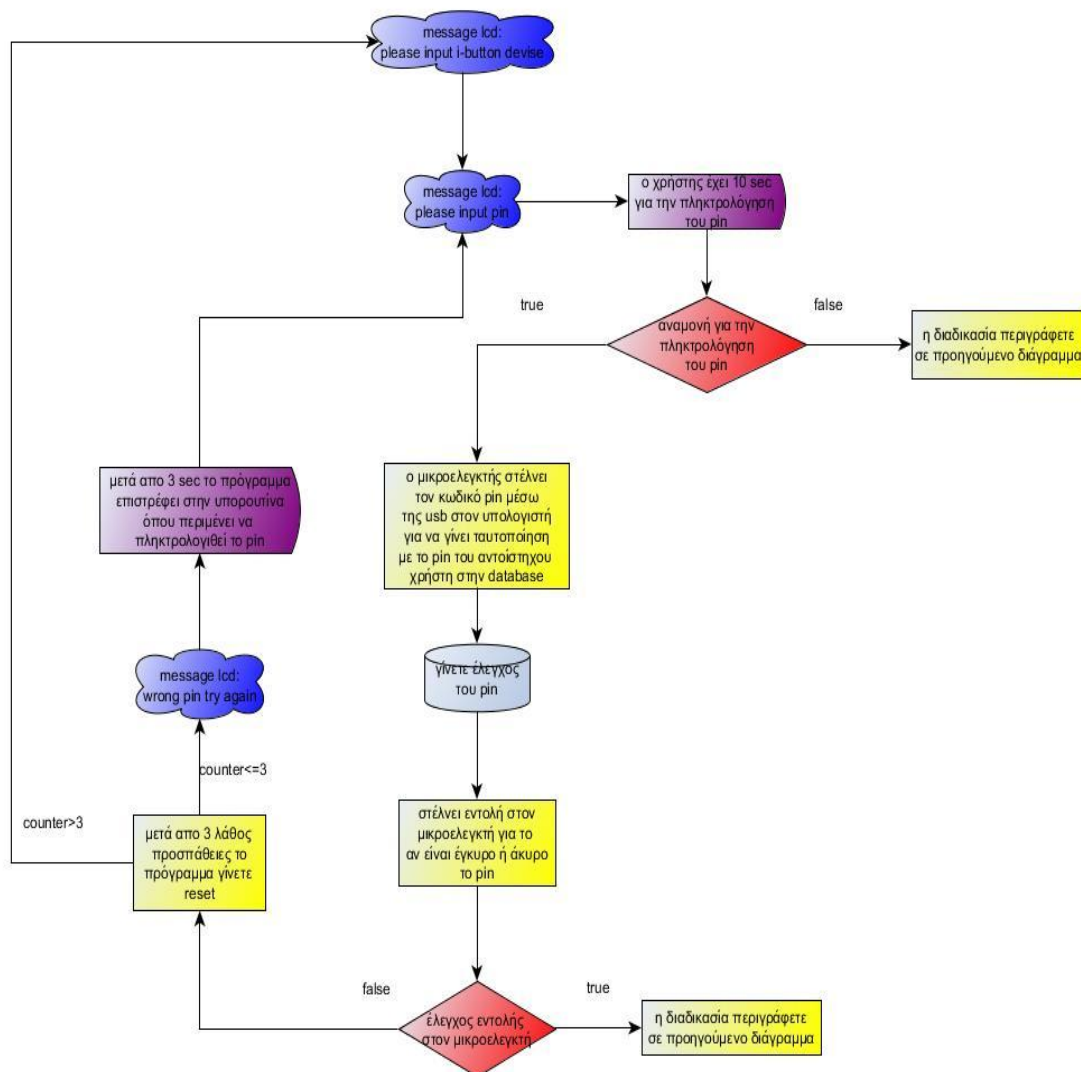
- 1. το ID του i-button,**
- 2. ο προσωπικός του κωδικός pin,**
- 3. το ονοματεπώνυμο του χρήστη**
- 4. η ώρα και η ημερομηνία που χρησιμοποίησε την συσκευή για να ανοίξει την πόρτα.**

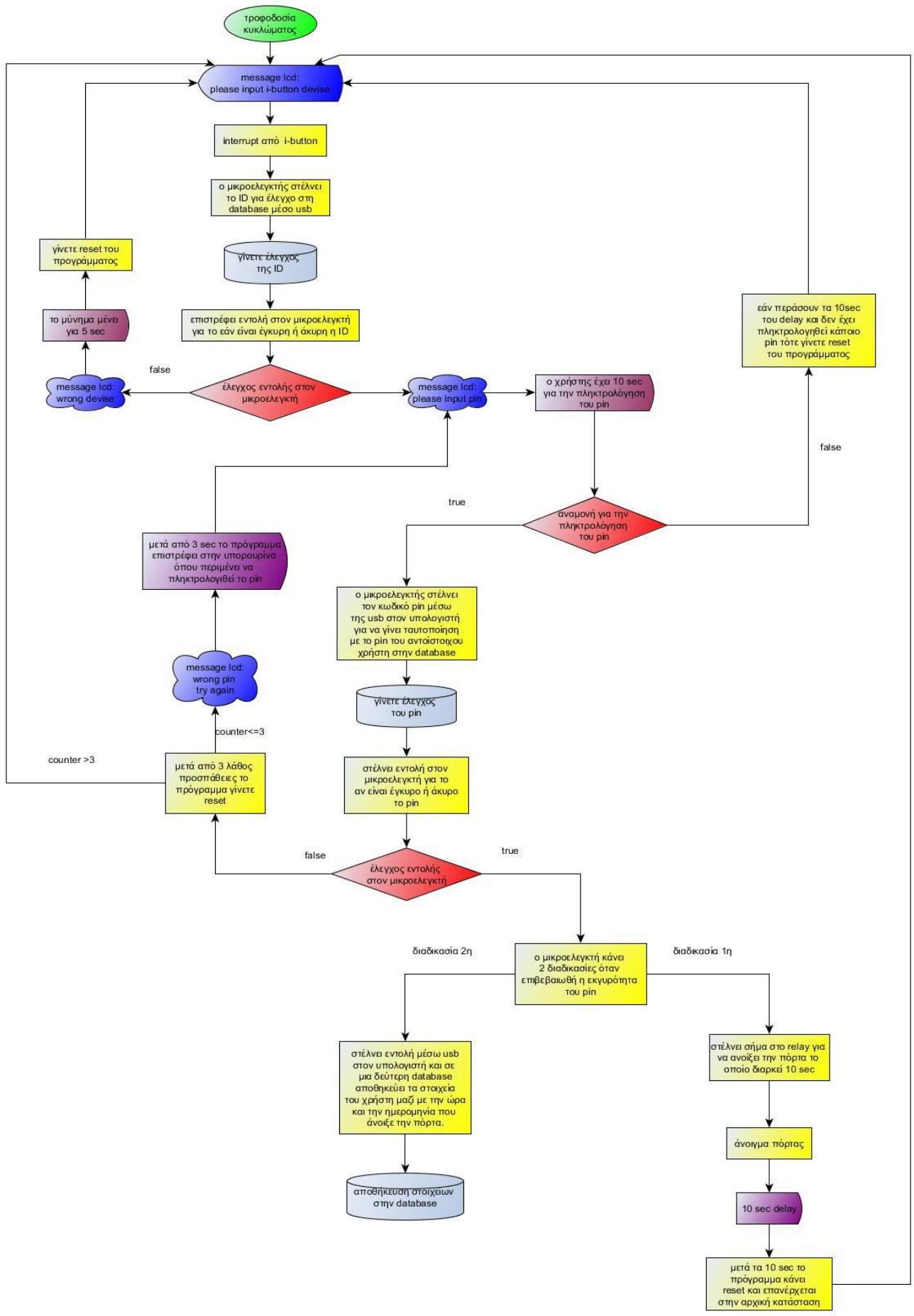
Μόλις περάσουν τα 10 sec το πρόγραμμα θα κόψει την παροχή ρεύματος στο relay και θα κάνει reset ώστε να επανέλθει στην αρχική stand-by κατάσταση με το μήνυμα στην lcd το οποίο είναι “pleaseinputi-buttondevice” και το πρόγραμμα να περιμένει σύνδεση κάποιου i-button.



## 2.1.4 Διαδικασίες που εκτελεί εάν ο κωδικός pin δεν είναι έγκυρος.

Εάν το Pin που θα πληκτρολογηθεί δεν είναι έγκυρο τότε το πρόγραμμα θα ελέγχει ένα counter ο οποίος αν είναι μικρότερος ή ίσος του 3 θα στέλνει στην οθόνη για 3sec μήνυμα που θα αναγράφει “wrong pin try again” και στη συνέχεια θα επανεμφανίζει το μήνυμα “please input pin”. Αν ο counter είναι μεγαλύτερος του 3 τότε το πρόγραμμα θα γίνει reset και θα επανέρχεται σε stand-by κατάσταση.





## 2.2 επικοινωνία πλακέτας με υπολογιστή

Η επικοινωνία της πλακέτας με τον υπολογιστή επιτυγχάνετε μέσω usb. Για να μπορέσει ο μικροελεγκτής να στείλει δεδομένα στον υπολογιστή μέσω της usb χρειάζεται ένα chip το οποίο είναι ένα usb to uart bridge. Στην κατασκευή μου χρησιμοποιώ το cp2102. Επίσης η usb τροφοδοτεί με 5V τα αντίστοιχα κομμάτια της πλακέτας που χρειάζονται 5V για να λειτουργήσουν. Αυτά είναι:

1. ο μικροελεγκτής
2. η lcd οθόνη
3. τα button
4. το i-button
5. το cp2102
6. η τροφοδοσία για να κολλήσει το relay

Με εξωτερική πηγή 24V που περνάει από έναν regulator για να μετατρέψει τα 24V σε 12V τροφοδοτώ την κλειδαριά η οποία χρειάζεται τροφοδοσία 12V.

Στον υπολογιστή υπάρχουν 2 database. Στο πρώτο database είναι αποθηκευμένα τα στοιχεία του κάθε χρήστη που του επιτρέπεται η πρόσβαση στην πόρτα. Τα στοιχεία αυτά είναι:

1. το ID του i-button
2. ο κωδικός pin
3. το όνομα του χρήστη
4. το επώνυμο του χρήστη

Από αυτό το database ο μικροελεγκτής ελέγχει εάν είναι έγκυρη η συσκευή i-button και αν είναι έγκυρος ο κωδικός pin που χρησιμοποιεί.

Στο δεύτερο database αποθηκεύονται τα στοιχεία του χρήστη που χρησιμοποίησε την πόρτα εφόσον έχει γίνει έλεγχος στα στοιχεία του χρήστη από τον μικροελεγκτή και βρέθηκαν έγκυρα και η συσκευή και το pin. Τα στοιχεία που αποθηκεύονται σε αυτό το database είναι:

- 1. το ID του i-button του χρήστη**
- 2. ο κωδικός pin του χρήστη**
- 3. το όνομα του χρήστη**
- 4. το επώνυμο του χρήστη**
- 5. η ημερομηνία που χρησιμοποίησε το pin και την συσκευή του**
- 6. η ώρα που χρησιμοποίησε το Pin και την συσκευή του.**

Στην περίπτωση που υπάρχουν παραπάνω από μία πόρτες οι οποίες θέλουμε να είναι ασφαλισμένες αλλά να ανοίγουν με τις ίδιες συσκευές του κάθε χρήστη και να μην έχει ο κάθε χρήστης παραπάνω από ένα i-button τότε όλες οι πόρτες θα είναι συνδεδεμένες σε έναν υπολογιστή με μία κοινή database η οποία θα περιέχει αποθηκευμένα τα στοιχεία του χρήστη που αναφέρθηκαν παραπάνω και επιπλέον άλλες τόσες database όσες και οι πόρτες που θα ελέγχονται και ανάλογα με το ποιος μικροελεγκτής θα στέλνει σήμα στον υπολογιστή θα αποθηκεύονται τα στοιχεία του χρήστη στο αντίστοιχο database.

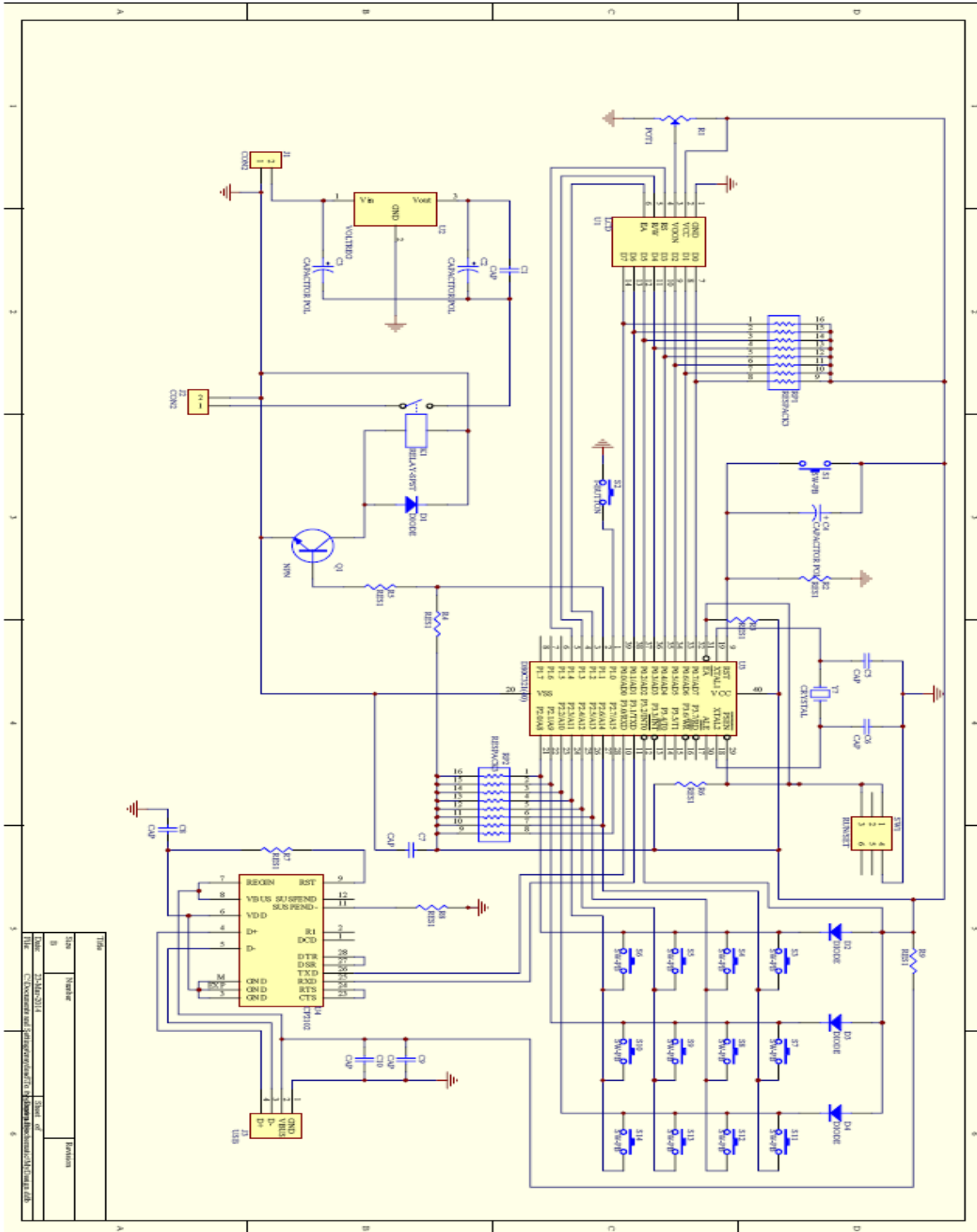
Οι database είναι γραμμένες σε κώδικα visual basic. Για την σχεδιάσή τους χρησιμοποιήθηκε η SQL Server Compact 3.5. Όπως αναφέρθηκε και πριν οι database που θα χρησιμοποιηθούν είναι τουλάχιστον δύο (εκτός και αν έχουμε παραπάνω από μία πόρτα οπότε σε αυτήν την περίπτωση θα έχουμε τόσες database όσες και οι πόρτες συν μία ακόμα). Στην μία database θα υπάρχουν καταγεγραμμένα τα στοιχεία των



ατόμων που έχουν εξουσιοδότηση για την χρήση της πόρτας ώστε να γίνεται έλεγχος για την εγκυρότητα του pin και του ID του i-button και στην δεύτερη database θα γίνεται αποθήκευση των στοιχείων του χρήστη καθώς και η ώρα και η ημερομηνία που χρησιμοποίησε το pin και το ID του για να υπάρχει ο έλεγχος για την χρήση της πόρτας. Η χρησιμότητα των database είναι για να γίνεται ευκολότερα η αναζήτηση του κάθε χρήστη στην περίπτωση που θέλουμε να διορθώσουμε κάτι στα είδη υπάρχοντα στοιχεία ή στην αναζήτηση πληροφοριών για την χρήση της πόρτας καθώς και για την ταξινόμηση στοιχείων εάν χρειαστεί. Για να γίνει αποθήκευση των στοιχείων του χρήστη το πρόγραμμα θα παίρνει από την πρώτη database, που υπάρχουν είδη τα στοιχεία αποθηκευμένα, το όνομα, το επώνυμο, το pin και το ID και την ώρα και την ημερομηνία θα τα παίρνει από τον υπολογιστή μέσω ενός κώδικα, σε visual basic, που θα ενημερώνεται για την τρέχουσα ώρα και ημερομηνία που υπάρχει στον υπολογιστή.

# ΚΕΦΑΛΑΙΟ 3

## 3.1 Σχηματικό



## 3.2 Part list:

1. R1	10K $\Omega$ Potentiometer
2. R2,R5,R9,R8	10K $\Omega$
3. R7	4,7K $\Omega$
4. R4	1K $\Omega$
5. R3,R6	100K $\Omega$
6. C2,C3	10nf
7. C4,C9	10uF
8. C5,C6	33pF
9. C1,C7,C8,C10	100nF
10. D1	1N4148
11. D2,D3,D4	LL103
12. RP1,RP2	100K $\Omega$ Resistor pack
13. K1	Relay 12volt SRD-S-112D
14. Q1	NPN BC547
15. U1	LCD ABC016002F69-YLY-R-01
16. U2	LM7805 Voltage regulator
17. U3	DS89C450
18. U4	CP2102
19. S1,S3-S14	Buttons
20. S2	i-button
21. J1	24Volt Input connector
22. J2	Output connector
23. J3	usb connector
24. SW1	run/set switch
25. Y1	11.05920MHz Crystal

### 3.3 Περιγραφή του σχηματικού

Στην παρακάτω εικόνα φαίνετε το πληκτρολόγιο του κυκλώματος το οποίο χρησιμεύει ώστε ο χρήστης να μπορεί να πληκτρολογήσει τον προσωπικό κωδικό pin. Η αντιστοίχιση των button με τους αριθμούς είναι:

S3=1

S7=2

S11=3

S4=4

S8=5

S12=6

S5=7

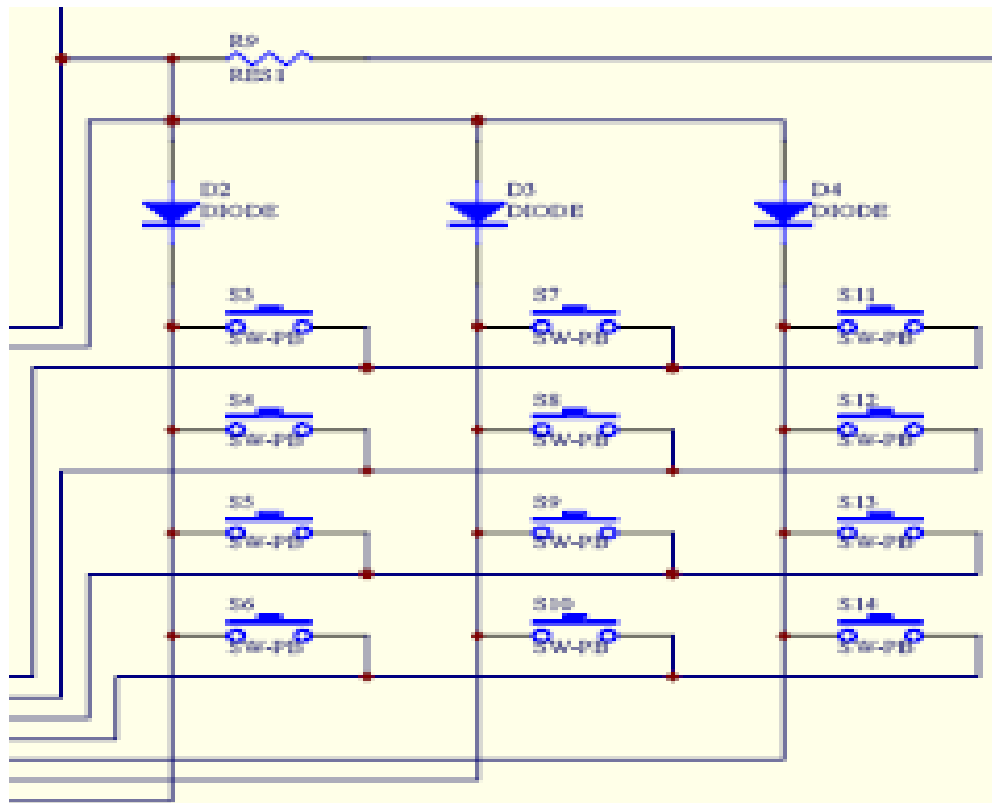
S9=8

S13=9

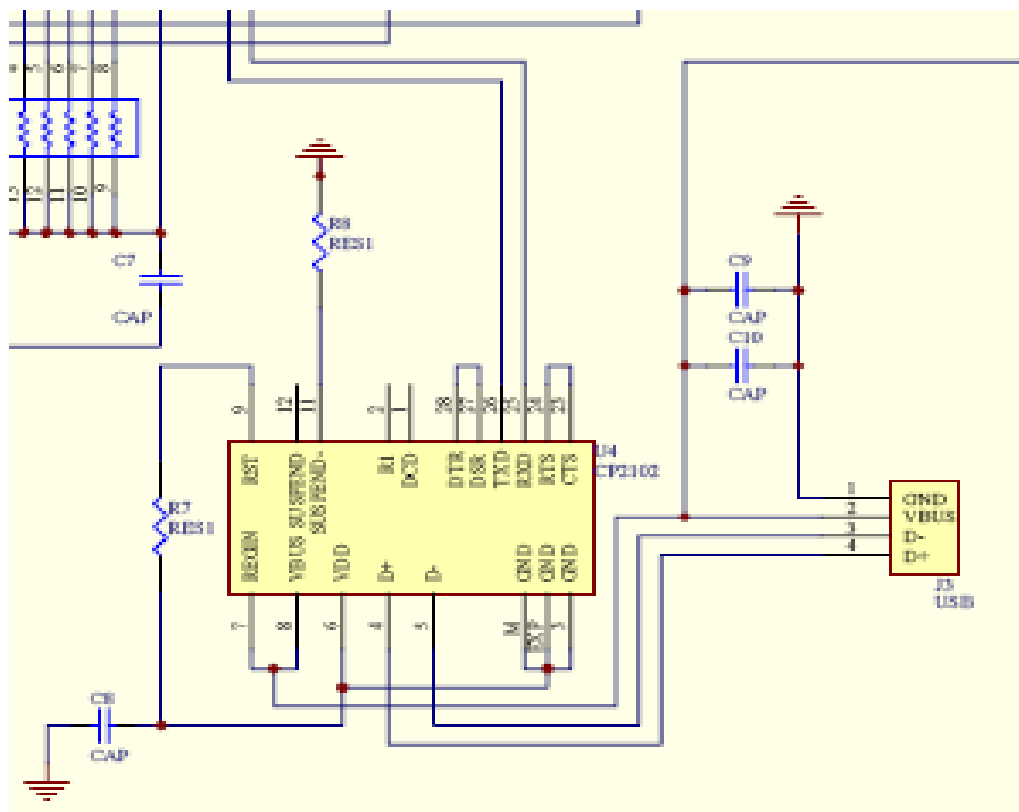
S6=\*

S10=0

S14=#



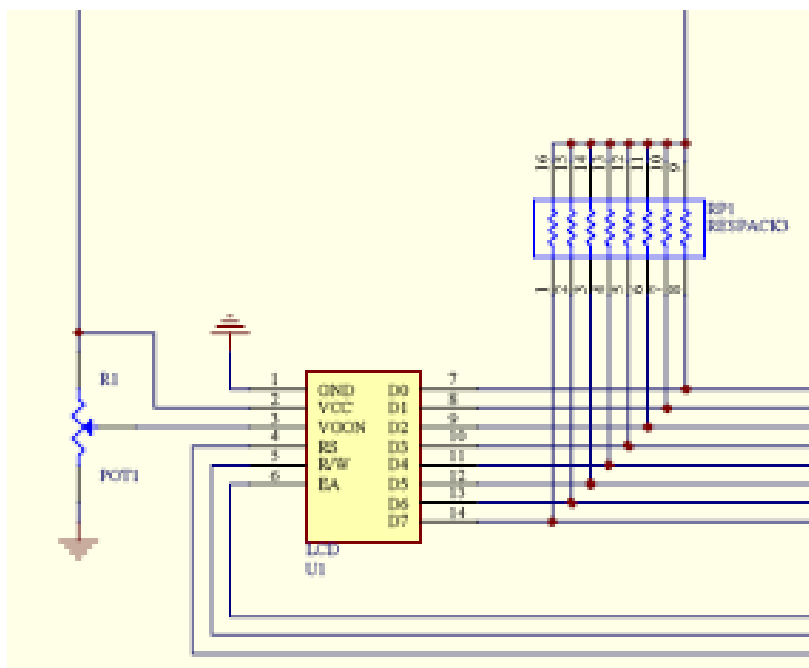
Στην παρακάτω εικόνα φαίνεται το cp2102 με την usb θύρα.



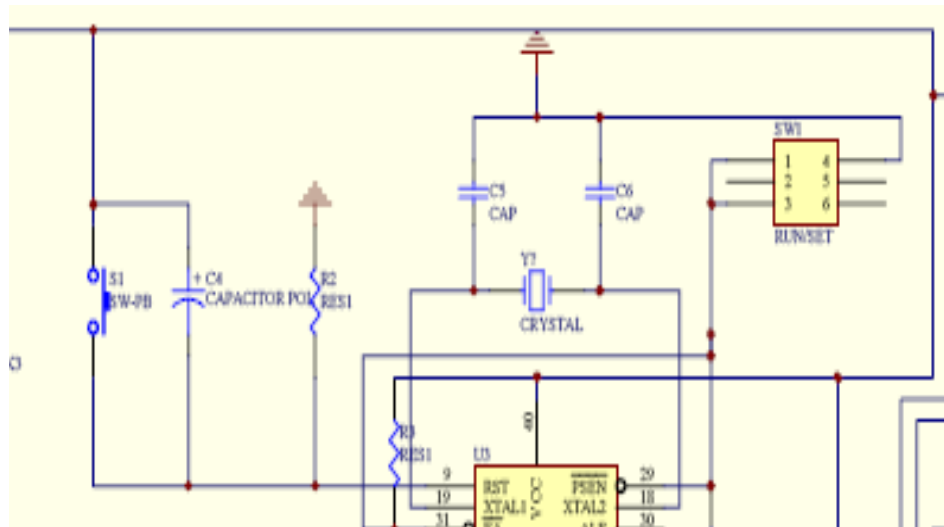
Παρακάτω φαίνεται η lcd του κυκλώματος στην οποία lcd εμφανίζονται τα αντίστοιχα μηνύματα για την κάθε περίπτωση.

Τα μηνύματα αυτά είναι:

1. Please input i-button devise αυτό το μήνυμα είναι όταν ο μικροελεγκτής περιμένει να πάρει interrupt από κάποια συσκευή i-button.
2. Wrong devise αυτό το μήνυμα εμφανίζεται εάν η συσκευή i-button δεν επιβεβαιωθεί με τα στοιχεία που υπάρχουν στην database.
3. Please input pin αυτό το μήνυμα εμφανίζεται αφού έχει επιβεβαιωθεί η εγκυρότητα της συσκευής i-button και περιμένει ο χρήστης να πληκτρολογήσει το pin του.
4. Wrong pin please try again αυτό το μήνυμα εμφανίζεται όταν ο κωδικός pin δεν ταιριάζει με αυτόν που αντιστοιχεί στην συσκευή i-button που χρησιμοποιήθηκε.



Παρακάτω φαίνεται το button του reset, ο διακόπτης run/set και ο κρύσταλλος του κυκλώματος. Το button του reset σε συνδυασμό με τον διακόπτη run/set χρησιμοποιούνται για τον προγραμματισμό του μικροελεγκτή.

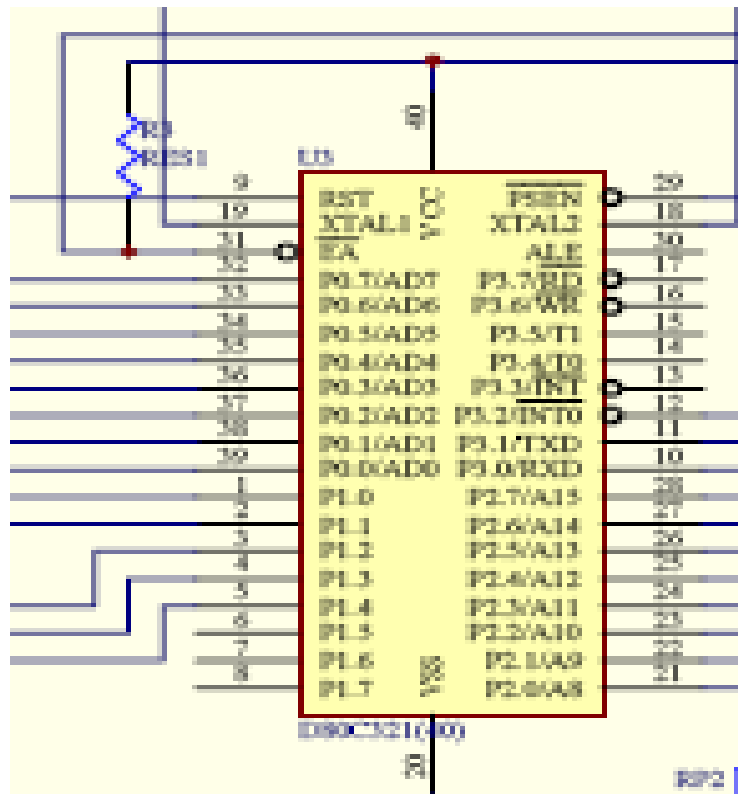


Παρακάτω φαίνεται η είσοδος και η έξοδος του κυκλώματος καθώς και η σύνδεση του i-button στον μικροελεγκτή. Στην είσοδο του κυκλώματος υπάρχει ένας voltage regulator που μετατρέπει τα 24Volt σε 12Volt για την τροφοδοσία της κλειδαριάς. Η έξοδος του κυκλώματος είναι ένας connector που συνδέεται στην κλειδαριά για να την τροφοδοτήσει με τα 12Volt. Στο ενδιάμεσο κύκλωμα βλέπουμε το relay το οποίο μόλις πάρει σήμα από τον μικροελεγκτή θα κολλήσει και θα επιτρέψει την είσοδό μας να τροφοδοτήσει την έξοδό μας με τα 12Volt.





Παρακάτω φαίνεται το κομμάτι του μικροελεγκτή του κυκλώματος το οποίο είναι και το κεντρικό κομμάτι της πλακέτας.



### 3.4 Πλακέτα



## **ΣΧΟΛΙΑ**

Ανάλογα με τις απαιτήσεις θα μπορούσαν να γίνουν διάφορες αλλαγές και στο hardware και στο software. Για παράδειγμα θα μπορούσαμε να προσθέσουμε επιπλέον επίπεδα ασφάλειας π.χ. επιβεβαίωση του κωδικού pin ή να αφαιρέσουμε το pin και να λειτουργεί μόνο με το i-button, οπότε θα μπορούσαμε να αφαιρέσουμε και το πληκτρολόγιο της κατασκευής. Η επικοινωνία θα μπορούσε να γίνει μέσω σειριακής θύρας ή θα μπορούσε να γίνει δικτυακά με UTP καλώδιο (με τις απαραίτητες αλλαγές σε software και hardware φυσικά). Επίσης θα μπορούσε να χρησιμοποιηθεί πλακέτα διπλής όψης ώστε να μικρύνει η κατασκευή.

## **Βιβλιογραφία**

Predko M., προγραμματίζοντας το μικροελεγκτή 8051, Τζιολα 1999

## **Σύνδεσμοι**

<http://www.wikipedia.org/>

<http://www.maximintegrated.com/en.html>

<http://msdn.microsoft.com/el-gr/default.aspx>