

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΚΡΗΤΗΣ  
ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ  
Σ.Τ.Ε.Φ. ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ηλεκτρονικός συγκολλητής πλαστικών σωλήνων**



**ΣΠΟΥΔΑΣΤΕΣ**  
ΒΑΛΤΖΑΚΗΣ ΝΙΚΟΛΑΟΣ  
ΠΑΠΑΓΡΗΓΟΡΙΟΥ ΓΕΩΡΓΙΟΣ

**ΕΙΣΗΓΗΤΗΣ**  
ΤΣΑΚΙΡΗΣ ΑΝΤΩΝΙΟΣ

ΦΕΒΡΟΥΑΡΙΟΣ 2005

This is a project of an electronic plastic pipes connector.  
Polyethylene plastic pipes are connected using electric current.  
To connect two pipes we use a plastic union that has a resistance inside it. Plastic pipes are placed at both sides of the union and an electric voltage is applied at the resistance. Electric current produces heat, melts the pipes and the union and connect the two pipes together.

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΚΕΦΑΛΑΙΟ 1

1.1 Εισαγωγή	4
--------------	---

## ΚΕΦΑΛΑΙΟ 2

2.1 Περιγραφή του 8051	5
2.2 Χρονισμός	6
2.3 Οργάνωση της μνήμης	7
2.4 Οι εντολές του 8051	11
2.4.1 Διευθυνσιοδότηση	11
2.4.2 Αριθμητικές εντολές	12
2.4.3 Λογικές πράξεις	13
2.4.5 Εντολές αλμάτων	15
2.5 Εξυπηρέτηση διακοπών	18

## ΚΕΦΑΛΑΙΟ 3

3.1 Το κύκλωμα	20
----------------	----

## ΚΕΦΑΛΑΙΟ 4

4.1 Το πρόγραμμα	25
------------------	----

## ΠΑΡΑΡΤΗΜΑ Α

Σχηματικό διάγραμμα κυκλώματος	33
Τυπωμένο κύκλωμα	36
Διάγραμμα ροής	38
Πρόγραμμα μικροελεγκτή	39

## ΠΑΡΑΡΤΗΜΑ Β

Data sheets	47
-------------	----

# ΚΕΦΑΛΑΙΟ 1

## 1.1 ΕΙΣΑΓΩΓΗ

Η κατασκευή που παρουσιάζεται παρακάτω είναι ένας ηλεκτρονικός συγκολλητής πλαστικών σωλήνων.

Στο εμπόριο κυκλοφορούν πλαστικοί σωλήνες από πολυαιθυλένιο. Για να συνδεθούν αυτοί οι σωλήνες μεταξύ τους χρησιμοποιείται ένας πλαστικός σύνδεσμος, διατομής λίγο μεγαλύτερης από τη διατομή των σωλήνων, ο οποίος στην εσωτερική του επιφάνεια έχει μια αντίσταση σύρματος. Η αντίσταση αυτή αποτελείται από λίγες σπείρες που καταλήγουν σε δύο ακροδέκτες. Οι σωλήνες προς κόλληση τοποθετούνται στις δύο πλευρές του συνδέσμου. Όταν εφαρμοστεί κατάλληλη τάση και για κατάλληλο χρόνο στους ακροδέκτες, η αντίσταση θερμαίνει τους πλαστικούς σωλήνες και το σύνδεσμο, το υλικό που είναι κατασκευασμένοι λιώνει και αφού κρυώσει οι σωλήνες έχουν κολληθεί μεταξύ τους.

Η τιμή της τάσης που χρειάζεται καθώς και ο χρόνος που πρέπει να εφαρμοστεί η τάση αυτή, εξαρτάται από τη διατομή του σωλήνα και από τον κατασκευαστή. Η κατασκευή με τη χρήση μικροελεγκτή ελέγχει και την τάση και το χρόνο της κόλλησης.

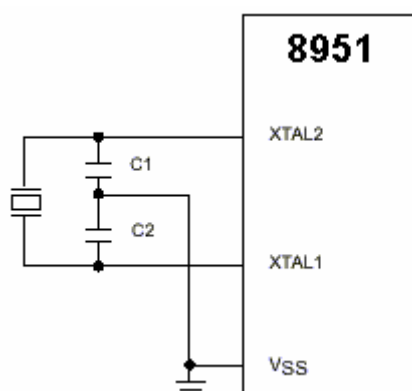
## ΚΕΦΑΛΑΙΟ 2

### 2.1 Περιγραφή του 8051

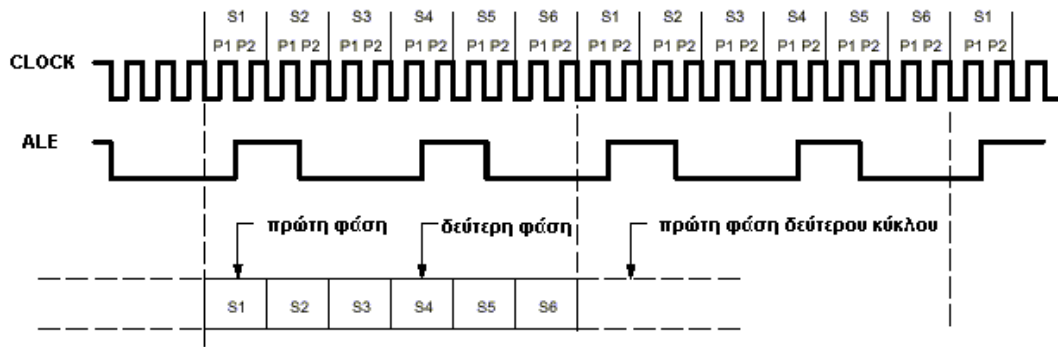
Ο 8051 είναι ένας οκτάμπιτος μικροελεγκτής που πρωτοκατασκευάστηκε την δεκαετία του 80 από την INTEL. Κυκλοφορεί σε πολλές παραλλαγές από διάφορους κατασκευαστές και πολλές απ' αυτές περιλαμβάνουν ενσωματωμένα στον μικροελεγκτή αρκετά βοηθητικά περιφερειακά όπως timers, counters και σειριακή επικοινωνία. 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. Εμείς για την κατασκευή μας χρησιμοποιήσαμε την γλώσσα assembly καθώς αυτή χρησιμοποιείται στις περισσότερες εφαρμογές αυτού του είδους. Επίσης αρκετά χρησιμοποιείται και η γλώσσα C για τον προγραμματισμό μικροελεγκτή.

## 2.2 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.

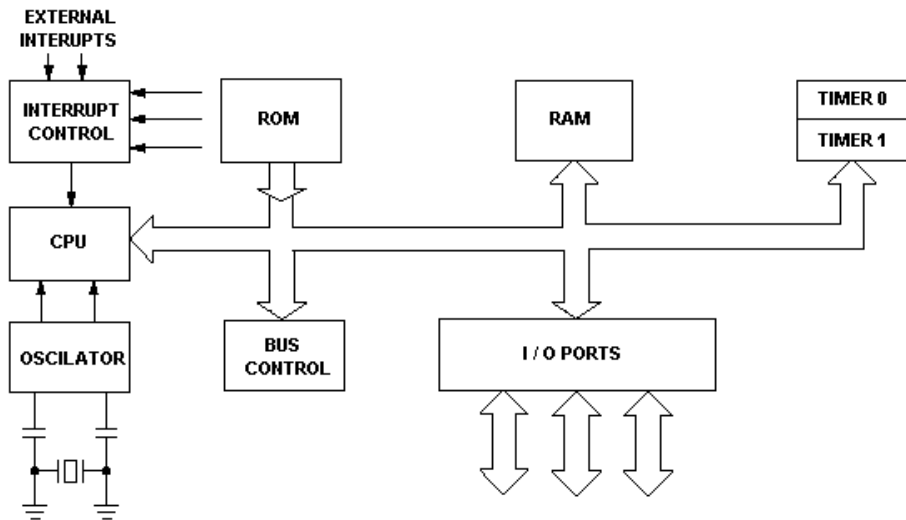


Κάποιες εντολές εκτελούνται σε ένα κύκλο όπως η “INC A” για παράδειγμα ενώ κάποιες άλλες απαιτούν περισσότερο χρόνο όπως η “DJNZ A” η οποία εκτελείται σε δύο κύκλους. Κάθε κύκλος έχει δύο φάσεις εκτέλεσης όπως φαίνεται στο σχήμα. Η πρώτη φάση αποτελείται από τις καταστάσεις S1, S2, S3 και η δεύτερη από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

## 2.3 ΟΡΓΑΝΩΣΗ ΤΗΣ 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

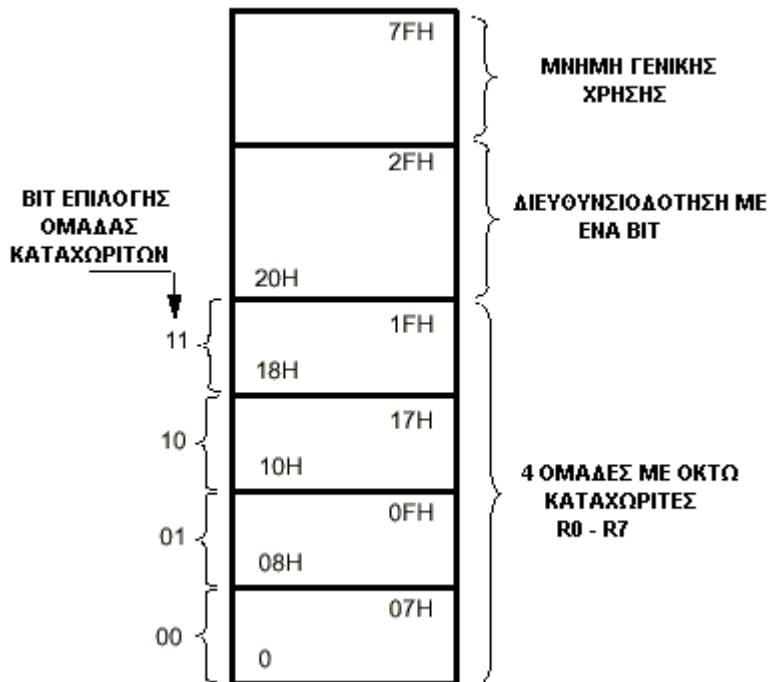
RAM είναι ανεξάρτητη και μπορεί να διευθυνσιοδοτηθεί με οκτάμπιτες διευθύνσεις για περισσότερη ευκολία στον προγραμματισμό.



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. Για παράδειγμα στο πρόγραμμα μας χρησιμοποιούμε την εξωτερική διακοπή (INT0) όταν πατηθεί κάποιο πλήκτρο στο πληκτρολόγιο. Τότε ο μικροελεγκτής πάει στη θέση μνήμης 0003H στην οποία βρίσκεται η ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής και διαβάζει το πλήκτρο που πατήθηκε μεταφέροντας το σε κάποια θέση στη μνήμη για να το επεξεργαστεί αργότερα.



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

## 2.4 ΟΙ ΕΝΤΟΛΕΣ ΤΟΥ 8051

### 2.4.1 ΔΙΕΥΘΥΝΣΙΟΔΟΤΗΣΗ

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.

## 2.4.2 ΑΡΙΘΜΗΤΙΚΕΣ ΕΝΤΟΛΕΣ

Οι αριθμητικές εντολές του 8051 φαίνονται στον παρακάτω πίνακα.

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

Στον πίνακα φαίνεται και ο χρόνος που χρειάζεται για να εκτελεστεί η κάθε εντολή με τον χρονοισμό 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 από το B και μετά την πράξη ο A κρατάει το ακέραιο πηλίκο και ο B το υπόλοιπο.

### 2.4.3 ΛΟΓΙΚΕΣ ΠΡΑΞΕΙΣ

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

<b>ΜΝΗΜΟΝΙΚΟ</b>	<b>ΛΕΙΤΟΥΡΓΙΑ</b>	<b>ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ME 12 MHz clock</b>
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 0000101 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.

## 2.4.4 ΕΝΤΟΛΕΣ ΜΕΤΑΦΟΡΑΣ ΔΕΔΟΜΕΝΩΝ

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

ΜΝΗΜΟΝΙΚΟ	ΛΕΙΤΟΥΡΓΙΑ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ME 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

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 ΜΕ 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

## 2.4.5 ΕΝΤΟΛΕΣ ΑΛΜΑΤΩΝ

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.

<b>ΜΝΗΜΟΝΙΚΟ</b>	<b>ΛΕΙΤΟΥΡΓΙΑ</b>	<b>ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ ΣΕ μsec ME 12 MHz clock</b>
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.

## 2.5 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 IE και του IP. Αναλυτικότερα οι λειτουργίες φαίνονται στους παρακάτω πίνακες.

Καταχωρητής IE (interrupt enable register)									
	IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0	
	EA			ES	ET1	EX1	ET0	EX0	
IE.7	Απενεργοποιεί όλες τις διακοπές αν είναι 0								
IE.4	Απενεργοποιεί την διακοπή της σειριακής αν είναι 0								
IE.3	Απενεργοποιεί την διακοπή του timer 1 αν είναι 0								
IE.2	Απενεργοποιεί την εξωτερική διακοπή 1 αν είναι 0								
IE.1	Απενεργοποιεί την διακοπή του timer 0 αν είναι 0								
IE.0	Απενεργοποιεί την εξωτερική διακοπή 0 αν είναι 0								

Καταχωρητής IP (interrupt priority register)									
	IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0	
				PS	PT1	PX1	PT0	PX0	
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								

## ΚΕΦΑΛΑΙΟ 3

### 3.1 ΤΟ ΚΥΚΛΩΜΑ

Η κατασκευή είναι μια γεννήτρια παλμών σταθερής συχνότητας και μεταβλητού κύκλου εργασίας (Duty Cycle). Οι παλμοί αυτοί οδηγούν ένα FET με το οποίο ελέγχουμε το ρεύμα που θα περάσει μέσα από το στοιχείο που θερμαίνει το σωλήνα. Η γεννήτρια υλοποιείται με μικροελεγκτή και χρειάζεται ελάχιστα εξαρτήματα για την κατασκευή της. Ο χρήστης εισάγει από το πληκτρολόγιο τα δεδομένα σε μορφή τάσης αφού έτσι αναφέρονται από τους κατασκευαστές των σωλήνων και ο μικροελεγκτή παράγει παλμούς που η RMS τιμή της τάσης τους είναι ίση με την τάση που έχει εισαχθεί. Επίσης ο χρήστης εισάγει και το χρόνο που θα διαρκέσει η κόλληση. Κατά την εκκίνηση η κόλληση ξεκινάει με μικρότερο ρεύμα από αυτό που χρειάζεται και αυξάνει σταδιακά στη μέγιστη τιμή του. Δηλαδή ξεκινάει με το 1/8 του απαιτούμενου ρεύματος για 0,8 δευτερόλεπτα, στη συνέχεια με το 1/4 του ρεύματος για άλλα 0,8 δευτερόλεπτα και μετά με το απαιτούμενο ρεύμα για όσο χρόνο έχει εισάγει ο χρήστης.

Το κύκλωμα της γεννήτριας των παλμών και αυτό του ελέγχου του ρεύματος είναι ανεξάρτητα και απομονωμένα μεταξύ τους ώστε να μην περάσει κάποια υψηλή τάση από το κύκλωμα ισχύος στα 5 volt του κυκλώματος ελέγχου. Η σύζευξη μεταξύ τους γίνεται με οπτοζεύκτη (IC2).

Η τροφοδοσία του κυκλώματος ελέγχου γίνεται από τον μετασχηματιστή και αφού ανορθωθεί από τις διόδους D1 ως D4 και φιλτραριστεί από τον πυκνωτή C1 σταθεροποιείται στα 5 volt από το 7805 (IC2). Μόλις το κύκλωμα τροφοδοτηθεί με τάση

ξεκινάει να τρέχει το πρόγραμμα στον μικροελεγκτή. Αναλυτικότερα η λειτουργία του προγράμματος φαίνεται στο επόμενο κεφάλαιο.

Οι πληροφορίες για την εισαγωγή των δεδομένων και για την εξέλιξη της όλης διαδικασίας εμφανίζονται σε μια οθόνη υγρών κρυστάλλων δύο γραμμών με δεκαέξι χαρακτήρες ανά γραμμή. Η οθόνη είναι συνδεδεμένη στον CON2 στην πλακέτα με μια καλωδιωτική δεκατεσσάρων αγωγών. Δύο από αυτούς είναι η τροφοδοσία της οθόνης με 5 volt. Οκτώ αγωγοί χρειάζονται για την μεταφορά των δεδομένων από τον μικροελεγκτή στην οθόνη και τρεις ακόμα για την επιλογή αποστολής εντολής ή χαρακτήρα (RS), την επιλογή αποστολής προς την οθόνη ή ανάγνωσης από αυτήν (R/W) και την ενεργοποίηση της οθόνης (EN). Τέλος ένας ακόμα αγωγός χρειάζεται για την ρύθμιση της αντίθεσης της οθόνης. Η μεταφορά των δεδομένων τα οποία είναι 8 bit γίνεται από το port 0 του μικροελεγκτή. Απαραίτητη είναι η σειρά με τις αντιστάσεις των 10 KΩ (Pull Up Resistor) γιατί το port 0 είναι ανοιχτού συλλέκτη. Οι εντολές RS και EN δίνονται από το P2.5 και P2.6 αντίστοιχα ενώ το R/W βρίσκεται πάντα σε λογικό 0 αφού μόνο γράφουμε στη μνήμη της οθόνης στην κατασκευή μας. Το ποτενσιόμετρο P1 ρυθμίζει την αντίθεση της οθόνης η οποία εξαρτάται από την τάση στο δρομέα του.

Η εισαγωγή του χρόνου και της τάσης γίνεται από το πληκτρολόγιο το οποίο είναι συνδεδεμένο στο CON1. Οι οκτώ ακροδέκτες του πληκτρολογίου οδηγούνται στις οκτώ αντίστοιχες εισόδους του αποκωδικοποιητή πληκτρολογίου MM74C922 (IC4). Όταν πατηθεί κάποιο πλήκτρο αυτός το αναγνωρίζει και βγάζει στην έξοδο του, η οποία αποτελείται από 4 bit, τον δεκαεξαδικό αριθμό που αντιστοιχεί στο πλήκτρο που πατήθηκε. Ταυτόχρονα

θέτει σε λογικό 1 το pin 12 και μέσω του τρανζίστορ T2 στέλνει αίτηση διακοπής στον μικροελεγκτή έτσι ώστε να διαβάσει το πλήκτρο από το port 2.

Πλήκτρο που πατήθηκε	Έξοδος MM74C922			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Αφού ολοκληρωθεί η εισαγωγή των δεδομένων ο μικροελεγκτής αρχίζει την παραγωγή των παλμών για όσο χρόνο έχει οριστεί. Ο παλμός από το P1.0 μέσω της R4 οδηγείται στη

βάση του τρανζίστορ T1. Από τον συλλέκτη του T4 και μέσω του δικτυώματος των αντιστάσεων R2 και R3 οδηγείται το led του οπτοζεύκτη. Το τρανζίστορ του οπτοζεύκτη μέσω του CON4 συνδέεται με το κύκλωμα ισχύος που ελέγχει το ρεύμα για την κόλληση του σωλήνα.

Το κύκλωμα ισχύος αποτελείται από τη γέφυρα ανόρθωσης, τον πυκνωτή εξομάλυνσης και το FET IRFP240. Ο συλλέκτης του τρανζίστορ του οπτοζεύκτη συνδέεται μέσω μιας αντίστασης με την πύλη του FET. Το φορτίο είναι συνδεδεμένο σε σειρά με τις επαφές πηγής και καταβόθρας και έτσι οι παλμοί μεταφέρονται τελικά στην αντίσταση του συνδέσμου.

Επίσης υπάρχει ένα κύκλωμα που αποτελείται από τον πυκνωτή C7, την αντίσταση R1 και το μπουτόν reset για την επανεκκίνηση του επεξεργαστή. Τέλος στα pin 18 και 19 του μικροελεγκτή είναι συνδεδεμένος ο κρύσταλλος των 12 MHz και δύο κεραμικοί πυκνωτές για το χρονισμό του μικροελεγκτή.

<b>Λίστα εξαρτημάτων</b>	
<b>Αντιστάσεις</b>	
R1	8,2 ΚΩ
R2 R3	120 Ω
R4 R6	4,7 ΚΩ
R5	10 ΚΩ
R7	10 ΚΩ pull up resistor
<b>Πυκνωτές</b>	
C1	1000 μF / 25V
C2 C3 C4	100 nF / 25V
C5 C6	30 pF
C7 C10	10 μF / 25V
C8 C9	1 μF / 25V
C11 C12	6800 μF/100V
<b>Ημιαγωγοί</b>	
D1 – D4	1N4001
D5	1N4148
T1 T2	BC548
T3	IRFP240
D6	KBPC50004 (Γέφυρα ανόρθωσης 200V / 50A)
<b>Ολοκληρωμένα</b>	
IC1	LM7805
IC2	4N25
IC3	AT89C51
IC4	MM74C922
<b>Διάφορα</b>	
2 Μετασχηματιστές 12V / 200mA, μετασχηματιστής 42V /12A	
Κρύσταλλος 12 MHz, Ποτενσιόμετρο 10ΚΩ, 2 ασφάλειες 1A	



## ΚΕΦΑΛΑΙΟ 4

### 4.1 ΤΟ ΠΡΟΓΡΑΜΜΑ

Τρεις είναι οι βασικές εργασίες που κάνει ο μικροελεγκτής μέσω του προγράμματος. Να διαβάζει από το πληκτρολόγιο τα εισαγόμενα δεδομένα, να στέλνει πληροφορίες στο χρήστη μέσω της οθόνης και να παράγει τον κατάλληλο παλμό για να οδηγήσει το στοιχείο που ελέγχει την κόλληση.

Στην αρχή του προγράμματος ορίζουμε ένα όνομα που αντιστοιχεί σε κάποιες θέσεις μνήμης ώστε να μας διευκολύνει στη συνέχεια στον προγραμματισμό, αφού όταν θέλουμε να κάνουμε χρήση αυτής της διεύθυνσης μνήμης θα την καλούμε με το όνομα που είναι πιο εύκολο στην απομνημόνευση. Αυτό φαίνεται στον παρακάτω πίνακα.

KEYB	EQU	P2
RAM	EQU	21H
KEYIN	EQU	20H.0
DF	EQU	20H.1
VOL1	EQU	27H
VOL2	EQU	28H
VOLT	EQU	29H
SFT1	EQU	30H
SFT2	EQU	31H
DF1	EQU	20H.2
R_S	EQU	P2.4
_E	EQU	P2.5

Αμέσως μετά βρίσκονται οι εντολές :

ORG 0000H

JMP MAIN

Ο μικροελεγκτής όταν ξεκινάει την εκτέλεση του προγράμματος η πρώτη εντολή που εκτελεί είναι αυτή που βρίσκεται στη θέση μνήμης 0000H. Με τις εντολές αυτές τοποθετούμε σ' αυτή τη θέση μνήμης την εντολή άλματος στο κυρίως πρόγραμμα το οποίο ξεκινάει από την θέση " MAIN: ". Μετά από αυτή την εντολή και πριν από την αρχή του κυρίως προγράμματος βρίσκεται η ρουτίνα εξυπηρέτησης της διακοπής EX0 που είναι η διακοπή σε περίπτωση που πατηθεί κάποιο πλήκτρο στο πληκτρολόγιο. Η διεύθυνση της είναι η 0003H. Όταν πατηθεί πλήκτρο καλείται η ρουτίνα με την οποία ο μικροελεγκτής διαβάζει το PORT2 που είναι συνδεδεμένος ο αποκωδικοποιητής πληκτρολογίου και το βάζει στη θέση μνήμης 21H . Στη συνέχεια θέτει 1 το bit 20H.0 που χρησιμοποιείται σαν σημαία και επιστρέφει στην εκτέλεση του προγράμματος με την εντολή RETI.

Στο κυρίως πρόγραμμα το πρώτο πράγμα που κάνουμε είναι η αρχικοποίηση της οθόνης. Η οθόνη για να λειτουργήσει έχει ενσωματωμένη μια πολύ μικρή μνήμη και ένα ελεγκτή (HD 44780). Σ' αυτό τον ελεγκτή στέλνουμε τις εντολές οι οποίες είναι 8 bit. Κατά την αρχικοποίηση κάνουμε μηδενισμό της οθόνης και τοποθετούμε τον κέρσορα στην πρώτη θέση της πρώτης γραμμής. Στη συνέχεια ενεργοποιούμε την εξωτερική διακοπή EX0 μέσω του καταχωρητή IE και αρχικοποιούμε κάποιες σημαίες (flags) που θα χρησιμοποιήσουμε στο πρόγραμμα.

Αμέσως μετά εμφανίζεται στην οθόνη το πρώτο μήνυμα. Τα μηνύματα βρίσκονται στο τέλος του προγράμματος στον πίνακα μηνυμάτων και κάθε ένα έχει μια αρχική διεύθυνση η οποία είναι 16 bit και “φορτώνεται στον καταχωρητή DPTR. Ο καταχωρητής R0 μηδενίζεται αρχικά και στην συνέχεια κάθε φορά που διαβάζεται και τυπώνεται ένας χαρακτήρας αυτός αυξάνεται κατά ένα και μας δείχνει την επόμενη θέση που βρίσκεται χαρακτήρας για διάβασμα. Ο χαρακτήρας μεταφέρεται στον accumulator και καλώντας τη ρουτίνα APPEAR εμφανίζεται στην οθόνη. Αυτό εκτελείται σε βρόγχο μέχρι το τέλος του μηνύματος. Έτσι γίνεται διάβασμα και εμφάνιση των χαρακτήρων ένας ένας από τα αριστερά προς τα δεξιά και σχηματίζεται το μήνυμα. Από το βρόγχο βγαίνουμε όταν φτάσουμε στο χαρακτήρα 0 με τον οποίο τελειώνουν τα μηνύματα. Στον παρακάτω πίνακα βλέπουμε όλα τα μηνύματα που εμφανίζονται στο πρόγραμμα.

PRESS "E"
INSERT VOLTAGE
INSERT TIME
PRESS "E" TO BEGIN
ERROR
INS. NEW VALUES
CANCEL, PRESS KEY
COMPLETED

Το πρώτο μήνυμα είναι το “PRESS E”. Το πλήκτρο E αντιστοιχεί στην εντολή “ENTER”. Μετά την εμφάνιση του μηνύματος το πρόγραμμα περιμένει μέχρι να πατηθεί κάποιο πλήκτρο και να γίνει διακοπή στον επεξεργαστή. Όταν πατηθεί

πλήκτρο καλείται η ρουτίνα εξυπηρέτησης της διακοπής EX0 και το πλήκτρο που πατήθηκε μεταφέρεται από το PORT2 στην μνήμη και στη συνέχεια στον accumulator για να γίνει έλεγχος αν είναι το "ENTER". Αν είναι το σωστό πλήκτρο συνεχίζεται η διαδικασία εκτέλεσης του προγράμματος ενώ αν δεν είναι, το πρόγραμμα επιστρέφει πάλι στην αναμονή για να πατηθεί νέο πλήκτρο. Πρέπει να σημειώσουμε ότι ενώ το PORT2 είναι 8 bit η έξοδος του αποκωδικοποιητή του πληκτρολογίου είναι 4 bit. Γι' αυτό υπάρχει η εντολή "ANL A, #0FH" με την οποία κάνουμε λογικό AND σε αυτό που διαβάσαμε από το PORT2 με αποτέλεσμα να μηδενίζουμε τα τέσσερα περισσότερο σημαντικά bits και να κρατάμε αναλλοίωτα τα υπόλοιπα τέσσερα τα οποία έχουν και την πληροφορία που χρειαζόμαστε.

Στη συνέχεια ξεκινάει η διαδικασία της εισαγωγής δεδομένων τάσης και χρόνου από το χρήστη. Αφού εμφανιστεί το μήνυμα "INSERT VOLTAGE" το πρόγραμμα και πάλι περιμένει εισαγωγή από το πληκτρολόγιο. Όταν πατηθεί κάποιο πλήκτρο καλείται η ρουτίνα "KEY\_ROUT" η οποία αναλαμβάνει την αποθήκευση στη μνήμη RAM του πλήκτρου που πατήθηκε καθώς και τον διαχωρισμό της εισαγωγής τάσης από την εισαγωγή χρόνου. Μετά την εισαγωγή της τάσης, η οποία πρέπει να αποτελείται από δύο ψηφία και να είναι από 8 ως 48 volt, εμφανίζεται το μήνυμα "INSERT TIME" για την εισαγωγή του χρόνου. Ο χρόνος εισάγεται σε δευτερόλεπτα, αποτελείται από τρία ψηφία και μπορεί να είναι από 1 ως 999. Αφού ολοκληρωθεί η διαδικασία εισαγωγής, στην οθόνη εμφανίζεται ο χρόνος και η τάση και ζητείται επιβεβαίωση των στοιχείων για να ξεκινήσει η διαδικασία της κόλλησης. Η επιβεβαίωση γίνεται με το πλήκτρο "ENTER". Σε όλη την παραπάνω διαδικασία με το πλήκτρο C μπορούμε να ακυρώσουμε

ότι έχουμε εισάγει μέχρι αυτή τη στιγμή και να δώσουμε νέες τιμές σε περίπτωση λάθους.

Αφού πατηθεί το πλήκτρο ENTER ξεκινάει ο υπολογισμός του παλμού καλώντας την ρουτίνα "DATA\_CALC". Η "DATA\_CALC" ξεκινάει με έλεγχο των τιμών που έχουν δοθεί αν είναι μέσα στα επιτρεπτά όρια. Αρχικά ελέγχεται ο χρόνος αν είναι μηδέν. Αν είναι μηδέν η κόλληση δεν προχωράει και εμφανίζεται μήνυμα σφάλματος. Στην συνέχεια το ψηφίο που μας δείχνει τις δεκάδες πολλαπλασιάζετε με το δέκα για να γίνει δεκάδες και προσθέτονται σ' αυτό και οι μονάδες για να έχουμε την πραγματική τιμή της τάσης. Η τάση αυτή ελέγχεται να είναι μεταξύ του 8 και του 48. Ο έλεγχος αυτός δεν υπάρχει σαν εντολή γι' αυτό γίνεται με αφαίρεση και έλεγχο του κρατούμενου. Πιο αναλυτικά αν από την τάση αφαιρέσουμε το 48 και έχουμε κρατούμενο σημαίνει ότι αυτή είναι μικρότερη από το 48 άρα δεν υπάρχει πρόβλημα. Επίσης αν από την τάση αφαιρέσουμε το 8 και έχουμε κρατούμενο σημαίνει ότι αυτή είναι μικρότερη του 8 άρα εμφανίζεται μήνυμα λάθους. Αν οι τιμές είναι εκτός των επιτρεπτών ορίων εμφανίζονται τα μηνύματα "ERROR" και "INSERT NEW VALUES". Αυτά μένουν στην οθόνη για 5 δευτερόλεπτα και μετά το πρόγραμμα ξεκινάει και πάλι από την αρχή με την αρχικοποίηση των καταχωρητών. Η αρχικοποίηση της οθόνης γίνεται μόνο την πρώτη φορά που θα δοθεί τάση στο κύκλωμα και μετά δεν χρειάζεται να ξαναγίνει όσο υπάρχει τροφοδοσία.

Αν οι τιμές που έχουν δοθεί είναι μέσα στα επιτρεπτά όρια η εκτέλεση του προγράμματος συνεχίζεται με τον υπολογισμό του κύκλου εργασίας (Duty Cycle) του παλμού. Αν με το 100% του κύκλου εργασίας έχουμε 50 volts τότε ο υπολογισμός γίνεται με την εξίσωση  $VOL1 = \text{τάση} * 100 / 50$  όπου το VOL1 είναι ο χρόνος

που η έξοδος βρίσκεται σε λογικό 1 σε ποσοστό επί τις εκατό του όλου κύκλου. Η έξοδος θα βρίσκεται σε λογικό 0 σε ποσοστό επί τις εκατό του όλου κύκλου για χρόνο  $VOL2 = 100 - VOL1$ . Αφού γίνει και αυτός ο υπολογισμός το πρόγραμμα επιστρέφει από την ρουτίνα υπολογισμού και καλείται η ρουτίνα παραγωγής των παλμών.

Η ρουτίνα παραγωγής των παλμών ξεκινάει με την εμφάνιση στην οθόνη του μηνύματος "CANCEL,PRESS KEY". Η διαδικασία μπορεί να ακυρωθεί με το πάτημα οποιουδήποτε πλήκτρου. Στην οθόνη φαίνονται και η τάση και ο χρόνος που έχουν δοθεί. Ο κύκλος εργασίας του παλμού που βρέθηκε στην προηγούμενη ρουτίνα διαιρείται δια 8 ώστε κατά την εκκίνηση να έχουμε το 1/8 του τελικού κύκλου εργασίας δηλαδή το 1/8 της τελικής τάσης. Αυτό γίνεται για 0,8 δευτερόλεπτα και στη συνέχεια επαναλαμβάνεται η ίδια διαδικασία με το 1/4 της τάσης για άλλα 0,8 δευτερόλεπτα. Αυτή είναι η διαδικασία της εκκίνησης (soft start) και διαρκεί συνολικά 1,6 δευτερόλεπτα. Στη συνέχεια φορτώνονται οι καταχωρητές R5, R6 και R7 με τις μονάδες, τις δεκάδες και τις εκατοντάδες των δευτερολέπτων αντίστοιχα και ξεκινάνε οι παλμοί που δίνουν την τελική τάση στο κύκλωμα.

Πιο αναλυτικά για τους παλμούς εκτελείται η παρακάτω διαδικασία. Θέτουμε σε λογικό 1 την έξοδο (P1.0) και καλούμε την χρονοκαθυστέρηση DELAY1 η οποία γίνεται με βάση το VOL1, που όπως είπαμε είναι το ποσοστό επί τις εκατό του χρόνου που ο παλμός είναι σε υψηλή στάθμη. Μετά η έξοδος γίνεται 0 και καλείται η χρονοκαθυστέρηση DELAY2. Αυτή η διαδικασία αποτελεί ένα κύκλο, διαρκεί συνολικά 5 msec και εκτελείται 200 φορές το δευτερόλεπτο. Έτσι η συχνότητα του παλμού είναι 200 Hz. Όταν ολοκληρωθεί το ένα δευτερόλεπτο μειώνεται κατά ένα ο

καταχωρητής των δευτερολέπτων και γίνεται έλεγχος αν έχουν ολοκληρωθεί οι μονάδες των δευτερολέπτων. Αν έχει συμβεί αυτό μειώνεται κατά ένα ο καταχωρητής των δεκάδων και ξαναφορτώνονται δέκα μονάδες στον καταχωρητή των μονάδων. Η ίδια διαδικασία γίνεται και για τις εκατοντάδες και συνεχίζεται μέχρι να μηδενιστούν τα δευτερόλεπτα. Ταυτόχρονα από αυτό τον έλεγχο παίρνουμε πληροφορίες κάθε δευτερόλεπτο για την εμφάνιση στην οθόνη του χρόνου που απομένει μέχρι το τέλος της διαδικασίας. Όταν ολοκληρωθεί η διαδικασία εμφανίζεται το μήνυμα "COMPLETED" για 5 δευτερόλεπτα και το πρόγραμμα επιστρέφει και πάλι στην αρχή.

Στο τέλος του προγράμματος φαίνονται οι ρουτίνες χρονοκαθυστέρησης SFTDEL1, SFTDEL2, DELAY1 και DELAY2. Επίσης υπάρχει και μια χρονοκαθυστέρηση για την εμφάνιση των μηνυμάτων στην οθόνη για χρόνο 5 δευτερολέπτων. Η ρουτίνα APPEAR εκτελείται για την αποστολή εντολών στην οθόνη ενώ η APPEARΔ για την αποστολή χαρακτήρα για εμφάνιση. Τέλος έχουμε τον πίνακα μηνυμάτων. Στην επόμενη σελίδα βλέπουμε ένα συνοπτικό διάγραμμα ροής.