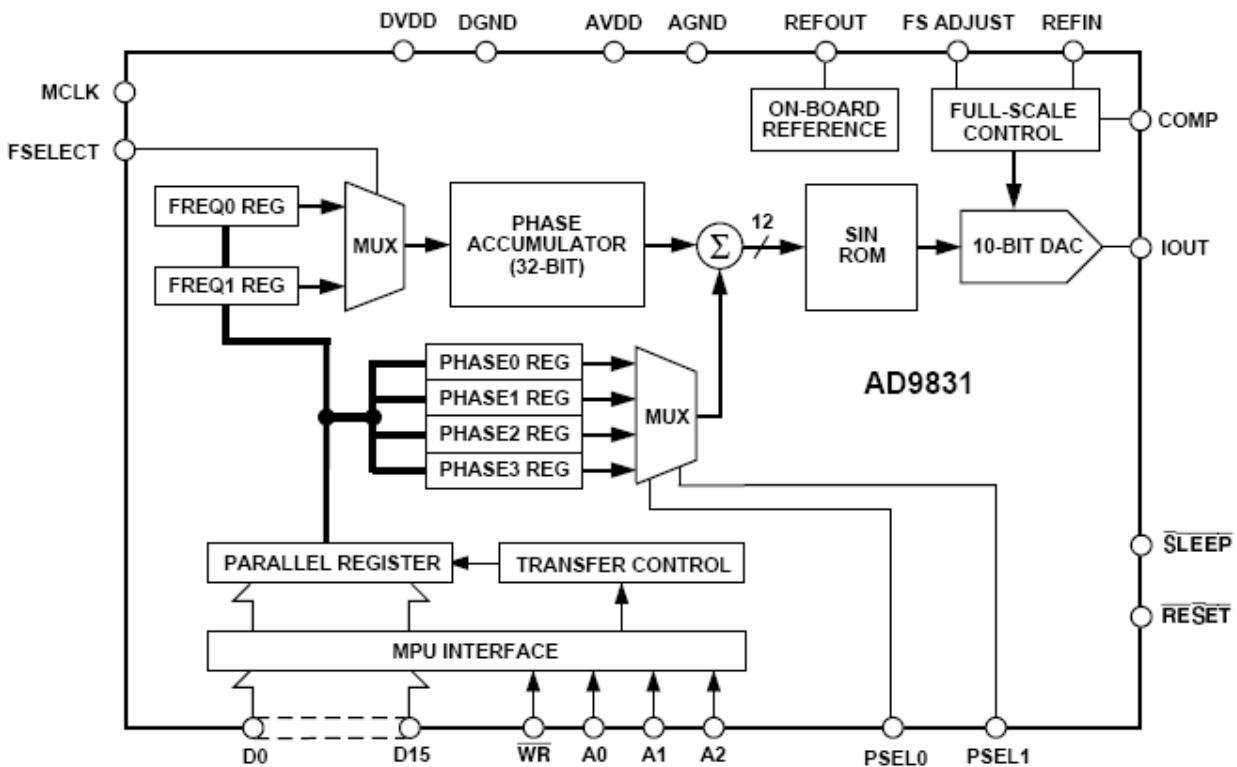


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ
ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

ΣΧΕΔΙΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ FSK ΠΟΜΠΟΔΕΚΤΗ
ΜΕ ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΑΣ “SOFTWARE RADIO”



εικόνα 5

ΕΙΣΗΓΗΤΗΣ: ΜΠΑΡΜΠΟΥΝΑΚΗΣ ΙΩΑΝΝΗΣ

ΣΠΟΥΔΑΣΤΗΣ: ΠΛΑΤΥΚΩΣΤΑΣ ΝΕΚΤΑΡΙΟΣ

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ.....	σελ 2
Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ DS89C420.....	σελ 3
Ο TRANSCEIVER AD9831.....	σελ 7
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ AD9831.....	σελ 11
AD9831 EVALUATION BOARD LAYOUT.....	σελ 12
ΑΝΑΠΤΥΞΙΑΚΟ ΚΥΚΛΩΜΑ ΓΙΑ ΤΟΝ DS89C420.....	σελ 14
ΑΝΑΠΤΥΞΙΑΚΟ ΚΥΚΛΩΜΑ ΓΙΑ ΤΟΝ AD9831.....	σελ 16
ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ ΤΗΣ ΑΠΟΚΡΙΣΗΣ ΣΥΧΝΟΤΗΤΑΣ ΤΟΥ ΦΙΛΤΡΟΥ.....	σελ 18
ΚΕΝΤΡΙΚΗ ΠΛΑΚΕΤΑ ΣΥΝΔΕΣΗΣ ΤΩΝ ΔΥΟ ΑΝΑΠΤΥΞΙΑΚΩΝ.....	σελ 19
ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΨΗΦΙΑΚΗΣ ΔΙΑΜΟΡΦΩΣΗΣ.....	σελ 24
ΚΩΔΙΚΑΣ ΓΙΑ FSK ΔΙΑΜΟΡΦΩΣΗ.....	σελ 25
ΛΙΣΤΑ ΕΞΑΡΤΗΜΑΤΩΝ ΚΑΤΑΣΚΕΥΗΣ.....	σελ 29

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια όλο και περισσότερες τεχνολογικές εφαρμογές υλοποιούνται πιο εύκολα και αξιόπιστα με τη βοήθεια των **μικροελεγκτών**. Από ηλεκτρολογικές εφαρμογές όπως ο έλεγχος των στροφών ενός κινητήρα, μέχρι συστήματα αυτομάτου ελέγχου, τηλεπικοινωνίες, τηλεφωνικά κέντρα κ.λπ. Ο γρήγορος και άμεσος προγραμματισμός τους, τους καθιστά αναγκαίους ακόμα και για τις πιο συνηθισμένες υλοποιήσεις ψηφιακών και αναλογικών κυκλωμάτων (κωδικοποιητές, πολυπλέκτες, ταλαντωτές κ.α.).

Επίσης είναι δυνατή η χρήση τους σε καθαρά τηλεπικοινωνιακές εφαρμογές, όπως η υλοποίηση πομποδεκτών με τη βοήθεια προγραμματιζόμενων ολοκληρωμένων, όπως για παράδειγμα τα **DDS (Direct Digital synthesis)** της εταιρείας Analog Device. Αυτή ακριβώς η σειρά ολοκληρωμένων και ο προγραμματισμός τους μέσω συμβατού 8051 μικροελεγκτή, αποτελεί τη γενικότερη ιδέα πάνω στην οποία στηρίχτηκε αυτή η πτυχιακή εργασία.

Πιο συγκεκριμένα η παρούσα εργασία ασχολείται με τον προγραμματισμό του πομποδέκτη AD9831 μέσω του μικροελεγκτή DS89C420 της Dallas semiconductor με στόχο την υλοποίηση διάφορων ψηφιακών διαμορφώσεων και απώτερο σκοπό την δημιουργία ενός επαναπρογραμματιζόμενου ψηφιακού πομπού καθώς και του αντίστοιχου δέκτη.

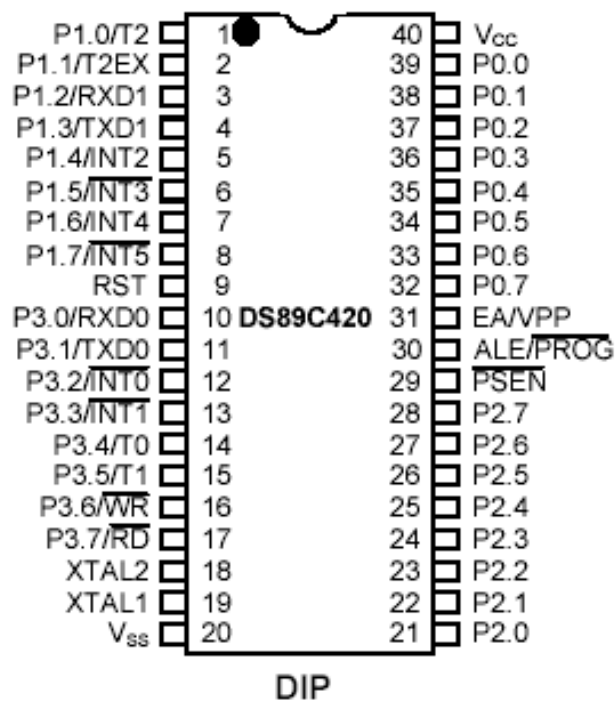
Το ολοκληρωμένο **AD9831** λοιπόν και ο προγραμματιστής του, **DS89C420**, τα οποία αποτελούν τον πυρήνα αυτής της κατασκευής, αναλύονται στις επόμενες σελίδες από πλευράς αρχιτεκτονικής, ώστε να γίνει κατανοητή στη συνέχεια και η συνεργασία τους σε μια κοινή κατασκευή, με σκοπό την υλοποίηση του προαναφερθέντος ψηφιακού πομποδέκτη.

Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ DS89C420

Το ολοκληρωμένο DS89C420 της Dallas semiconductor, αποτελεί έναν υψηλής ταχύτητας, συμβατό με **8051**, μικροελεγκτή ικανό για πάρα πολλές ηλεκτρονικές και ηλεκτρολογικές εφαρμογές. Παρακάτω θα δούμε επιγραμματικά ορισμένες πληροφορίες όσον αφορά την αρχιτεκτονική και την λειτουργία του ολοκληρωμένου, ώστε να κατανοήσουμε το ρόλο του στο σύνολο της πτυχιακής εργασίας.

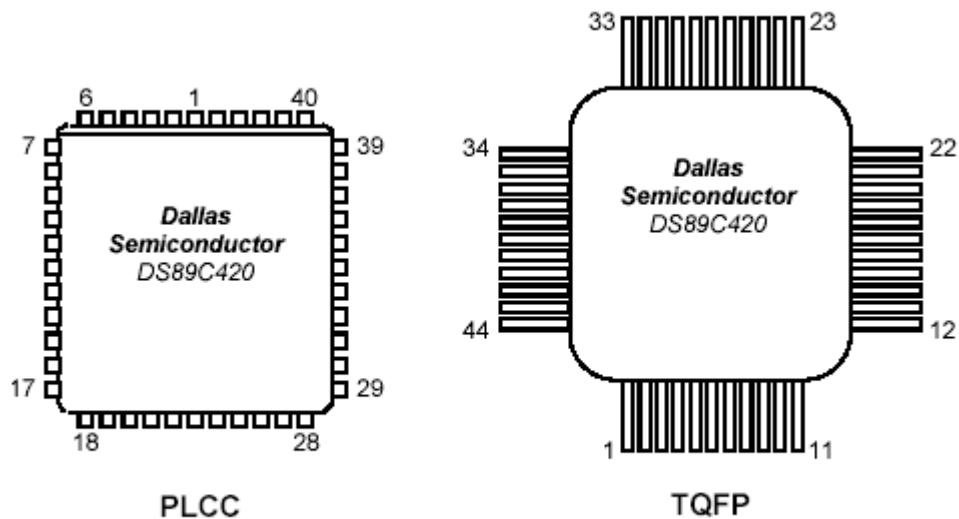
Καταρχήν ο συγκεκριμένος μικροελεγκτής, όπως και οι περισσότεροι αυτού του είδους, χρησιμοποιεί συνεχή τάση τροφοδοσίας από **4.5** έως **5.5 Volt**. Το ρολόι του επεξεργαστή του τρέχει σε μέγιστη συχνότητα στα **33MHz**, στη συγκεκριμένη κατασκευή ωστόσο θα χρησιμοποιήσουμε για το χρονοισμό του κρύσταλλο με συχνότητα ταλάντωσης στα **11.0592MHz**.

Οι εντολές που χρησιμοποιεί είναι οι ίδιες με αυτές του 8051 καθώς η αρχιτεκτονική του είναι βασισμένη σε αυτή του 8051. Στην παρακάτω εικόνα φαίνεται το ολοκληρωμένο σε συσκευασία **PDIP**.



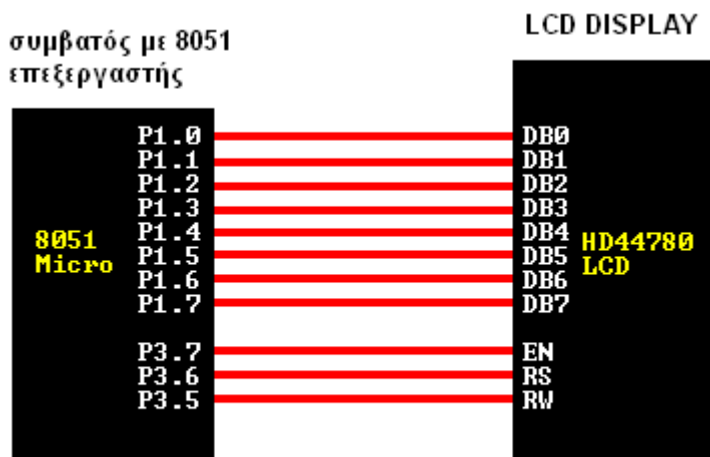
εικόνα 1

Ο DS89C420 κατασκευάζεται και σε συσκευασίες SMD όπως TQFP και PLCC αλλά στη συγκεκριμένη κατασκευή μας εξυπηρέτησε καλύτερα η κλασική PDIP συσκευασία.



εικόνα 2

Όπως παρατηρούμε στην εικόνα 1 ο DS89C420 περιλαμβάνει 4 ports των 8 bits (**P0,P1,P2,P3**) τα οποία μπορούν να συνδέσουν τον μικροελεγκτή με εξωτερικές συσκευές όπως μια LCD οθόνη, μια σειρά από leds, ή οποιαδήποτε άλλη ηλεκτρονική διάταξη στην οποία μπορούμε να περάσουμε δεδομένα σε ψηφιακή μορφή. Ένα απλό παράδειγμα φαίνεται στην εικόνα 3 στην οποία ο DS89C420 είναι συνδεδεμένος με μια LCD οθόνη.



εικόνα 3

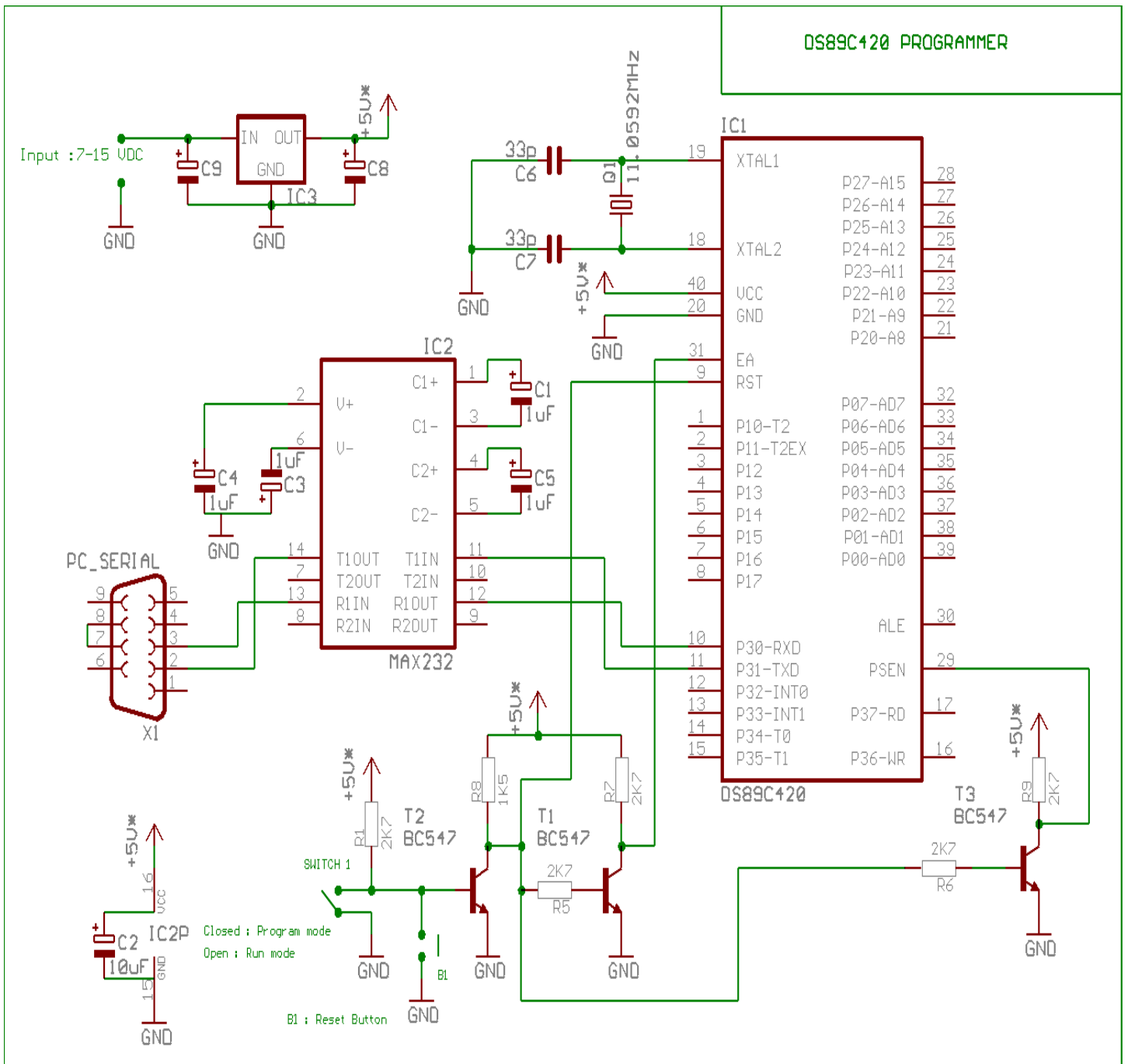
Παρατηρούμε ότι ένα port του μικροελεγκτή και επιπλέον 3 bit ελέγχου αρκούν για να απεικονίσουμε στο lcd display ένα αριθμό λέξεων τις οποίες είχαμε πρώτα αποθηκεύσει στον επεξεργαστή.

Εκτός βέβαια από τα 32 bit εισόδου/εξόδου (4 ports), ο DS89C420 διαθέτει δύο external interrupts (**INT0** και **INT1**) καθώς και δύο εισόδους για σύνδεση εξωτερικών timers (**T0** και **T1**). Επίσης δύο πολύ βασικά pins είναι τα **TXD** (transmit) και **RXD** (receive), τα οποία χρησιμεύουν στην επικοινωνία του μικροελεγκτή με τη σειριακή θύρα του υπολογιστή (**COM**), από την οποία προγραμματίζεται σε κώδικα μηχανής (assembly), με τη βοήθεια κατάλληλου λογισμικού (**ED95 8x5x**).

Τέλος διαθέτει το pin **EA** το οποίο επιτρέπει την επιλογή εξωτερικής ή εσωτερικής προγραμματιζόμενης μνήμης του επεξεργαστή.

Στην επόμενη σελίδα, στην εικόνα 4, ακολουθεί ένα σχηματικό διάγραμμα το οποίο αποτελεί την κυκλωματική διάταξη με την οποία προγραμματίζουμε τον DS89C420. Το ολοκληρωμένο **MAX232** μεσολαβεί για την ένωση του μικροελεγκτή με την COM του υπολογιστή. Επίσης όπως θα δούμε το σχέδιο περιλαμβάνει ένα κρύσταλλο των 11.0592MHz συνδεδεμένο στα pins **XTAL1** και **XTAL2** για το χρονισμό του επεξεργαστή. Επίσης μια ακόμη κυκλωματική διάταξη συνδέεται στα control bits **EA** και **PSEN** για την εναλλαγή από **program mode** (προγραμματισμός από τον υπολογιστή) σε **run mode** (εκτέλεση του κώδικα), καθώς και ένα button για το **reset του clock** του επεξεργαστή.

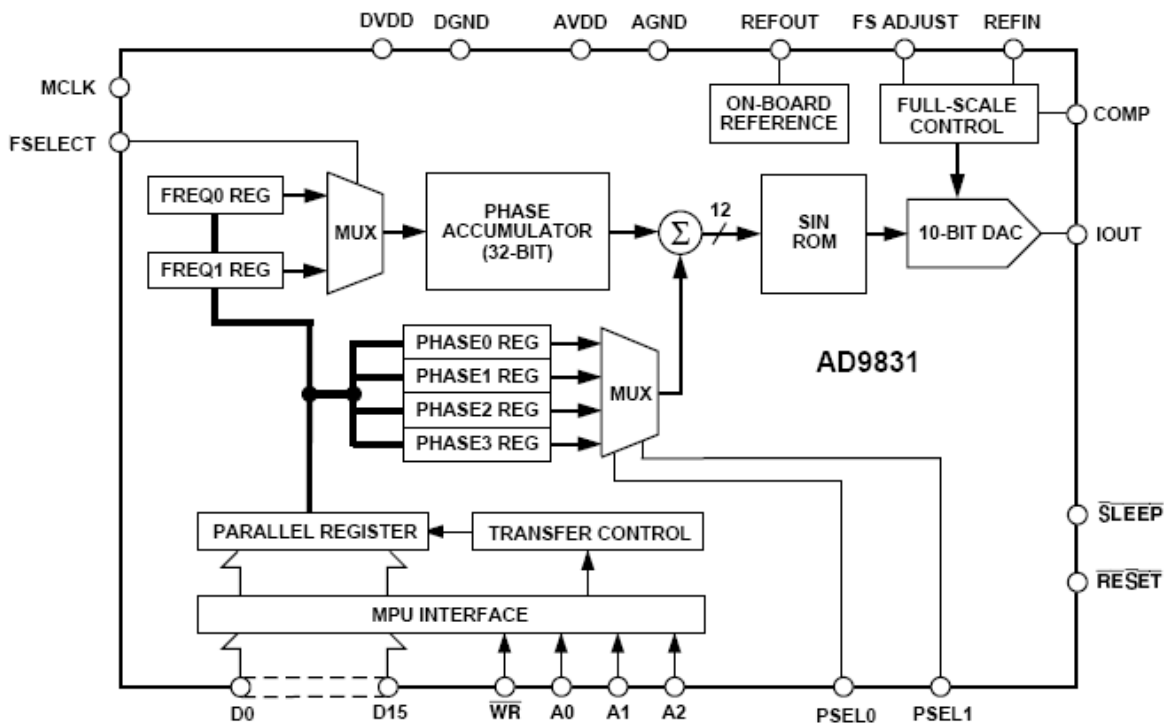
Επίσης το σχηματικό του αναπτυξιακού περιλαμβάνει και ένα σταθεροποιητή τάσης με σκοπό να επιτευχθεί σταθερή τροφοδοσία του μικροελεγκτή στα 5 Volts.



Οι κώδικες που θα χρησιμοποιήσουμε για τον προγραμματισμό του μικροελεγκτή γράφτηκαν σε δεκαεξαδική μορφή και επαληθεύτηκαν με τη βοήθεια του λογισμικού ED95 8x5x και του simulation για συμβατούς 8051 επεξεργαστές που διαθέτει. Η παρουσίαση και η επεξήγησή τους θα γίνει αναλυτικά στις επόμενες σελίδες, αφού όμως πρώτα μελετήσουμε το ολοκληρωμένο στο οποίο θα υλοποιηθούν οι κώδικες. Δηλαδή τον transceiver AD9831 της analog device.

Ο TRANSCEIVER AD9831

Το ολοκληρωμένο AD9831 κατασκευάζεται από την analog device και ανήκει στην κατηγορία **DDS (direct digital synthesis)**. Πρόκειται στην ουσία για έναν **numerical controlled oscillator** ικανό να πραγματοποιήσει διαμόρφωση ψηφιακής πληροφορίας κατά συχνότητα και κατά φάση (**FSK και PSK**). Με βάση την εικόνα 5 που ακολουθεί θα εξηγήσουμε όσο το δυνατόν πιο αναλυτικά την αρχιτεκτονική και της λειτουργίες του ολοκληρωμένου.



εικόνα 5

Όπως παρατηρούμε στην εικόνα 5 το ολοκληρωμένο, όσον αφορά τον τομέα της διαμόρφωσης, αποτελείται από τρεις βασικούς τομείς.

1) Numerical Controlled Oscillator (NCO) και Phase modulator

2) SINE Look-Up Table

3) Digital to Analog Converter

Το **NCO** αποτελείται αρχικά από δύο 32 bit καταχωρητές τους **FREQ0** και **FREQ1**. Σε αυτούς τους δύο καταχωρητές αποθηκεύουμε, μέσω του **MPU INTERFACE** δύο 32 bit λέξεις από τις οποίες κάθε μία αντιπροσωπεύει και μία ξεχωριστή συχνότητα όσον αφορά την διαμόρφωση FSK. Επίσης περιλαμβάνει και τέσσερις 12 bit registers (**PHASE REGS**) υπεύθυνους για την PSK διαμόρφωση, καθώς και έναν 32 bit **phase accumulator**, ο οποίος συνθέτει τη φάση του σήματος εξόδου. Πριν όμως μελετήσουμε τον τρόπο με τον οποίο οι ψηφιακές διαμορφώσεις υλοποιούνται μέσα στο ολοκληρωμένο, θα πρέπει να δούμε πως γίνεται

η μετατροπή των επιθυμητών συχνοτήτων, αρχικά σε δεκαεξαδική μορφή και στη συνέχεια σε δυαδική, ώστε να είναι δυνατή η αποθήκευσή της τιμής τους (MHz) στις μονάδες μνήμης του ολοκληρωμένου (καταχωρητές).

Δεδομένου ότι η συχνότητα ενός σήματος δίνεται από τη σχέση $\omega = 2\pi f$ και ότι η διαφορά φάσης ενός ημιτονοειδούς σήματος μπορεί να οριστεί από τη σχέση $\Delta\text{Phase} = \omega \delta t$ τότε έχουμε $\omega = \Delta\text{Phase} / \delta t = 2\pi f$

Εάν στην παραπάνω σχέση αντικαταστήσουμε τη συχνότητα αναφοράς με την περίοδο $1/f_{\text{MCLK}} = \delta t$ τότε οδηγούμαστε στην εξίσωση:

$$f = \Delta\text{Phase} \times f_{\text{MCLK}} / 2\pi \text{ (σχέση 1)}$$

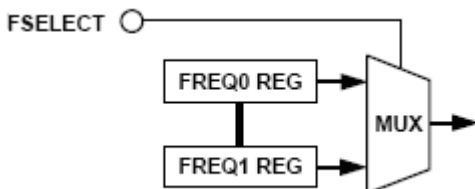
Γνωρίζουμε ότι το φασικό εύρος συνεχών σημάτων είναι από 0 έως 2π . Ο Phase accumulator απλώς κλιμακώνει το εύρος των φάσεων σε μια πολλαπλών bits ψηφιακή λέξη. Δεδομένου ότι ο accumulator είναι 32 bit, $2\pi = 2^{32}$. Επομένως η σχέση 1 γίνεται:

$$f = \Delta\text{Phase} \times f_{\text{MCLK}} / 2^{32}$$

με $0 < \Delta\text{Phase} < 2\pi$ και επειδή αναφερόμαστε στον AD9831 $0 < \Delta\text{Phase} < 2^{32} - 1$

Έτσι για παράδειγμα με 25MHz συχνότητα ρολογιού (f_{MCLK}) και 32 bit phase word 51EB852 έχουμε $f = 51EB852 \times 25\text{MHz} / 2^{32} = 0.500000000465 \text{ MHz}$

Με τον τρόπο αυτό βρίσκουμε τις επιθυμητές συχνότητες για τις ψηφιακές διαμορφώσεις και τις “μεταφέρουμε” στους κατάλληλους καταχωρητές μέσω των pins D0 έως D15 του AD9831.



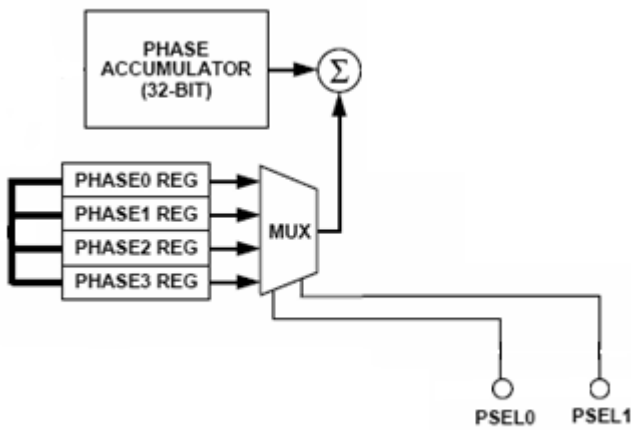
Με τις συχνότητες λοιπόν αποθηκευμένες στους καταχωρητές FREQ0 και FREQ1, όπως παρατηρούμε στο παραπάνω σχήμα, και με τη βοήθεια του bit ελέγχου **FSELECT**, που στην ουσία είναι η πληροφορία μας, παίρνουμε στην έξοδο του πολυπλέκτη (MUX) την κατάλληλη δεκαεξαδική λέξη για 0 ή 1. Αυτή είναι και η βασική αρχή με την οποία λειτουργεί η **FSK διαμόρφωση** στον AD9831. Οι 32 bit words φτάνουν στον προορισμό τους μέσω των control bits **A0, A1, A2** με βάση τον παρακάτω πίνακα.

A2	A1	A0	Destination Register
0	0	0	FREQ0 REG 16 LSBs
0	0	1	FREQ0 REG 16 MSBs
0	1	0	FREQ1 REG 16 LSBs
0	1	1	FREQ1 REG 16 MSBs

Φυσικά με παρόμοια διαδικασία υλοποιείται και η **διαμόρφωση κατά φάση (PSK)**. Το phase offset μπορεί να προστεθεί με τη βοήθεια των **phase registers** στους οποίους οι επιθυμητές τιμές τοποθετούνται όπως και στους προηγούμενους καταχωρητές (control bits).

A2	A1	A0	Destination Register
1	0	0	PHASE0 REG
1	0	1	PHASE1 REG
1	1	0	PHASE2 REG
1	1	1	PHASE3 REG

Η τιμή του κάθε καταχωρητή οδηγείται επιλεκτικά στην έξοδο του phase accumulator με τη βοήθεια των control bits **PSEL0** και **PSEL1**



Η αντιστοίχιση φαίνεται στον παρακάτω πίνακα.

output of the phase accumulator	control bit
PHASE0 REG 12 Bits	PSEL0 = PSEL1 = 0
PHASE1 REG 12 Bits	PSEL0 = 1 and PSEL1 = 0
PHASE2 REG 12 Bits	PSEL0 = 0 and PSEL1 = 1
PHASE3 REG 12 Bits	PSEL0 = PSEL1 = 1

Ας περάσουμε τώρα να δούμε τη χρησιμότητα της δεύτερης κατά σειρά βαθμίδας του **Sine Look-up Table**. Η αυτή βαθμίδα αποσκοπεί στο να μετατρέψει την πληροφορία της φάσης σε ένα ημιτονοειδές σήμα ώστε η έξοδος του κυκλώματος να είναι αξιοποιήσιμη. Από τη στιγμή που η πληροφορία της φάσης ταιριάζει απόλυτα σε μία τιμή πλάτους, η βαθμίδα **Look-up Table** μετατρέπει την πληροφορία της φάσης σε πλάτος. Για να είναι εφικτό αυτό, η βαθμίδα sine LUT κατευθύνεται χρησιμοποιώντας την ψηφιακή τιμή της φάσης.

Πρέπει επίσης να αναφέρουμε ότι όπως παρατηρήσαμε ο NCO περιλαμβάνει έναν 32 bits phase accumulator του οποίου η έξοδος μετατρέπεται σε 12 bits. Θα ήταν ανούσιο να κρατήσουμε την πλήρη ανάλυσή του διότι θα χρειαζόμαστε ένα look-up table 2^{32} εισόδων.

Τέλος για την ολοκλήρωση της διαδικασίας της διαμόρφωσης μένει η βαθμίδα του **Digital to Analog converter**. Η τελευταία βαθμίδα πριν την έξοδο του κυκλώματος αποτελείται από 10 bits και είναι ικανή να οδηγήσει ένα μεγάλο εύρος φορτίων σε διαφορετικές ταχύτητες. Η ουσιαστική χρησιμότητά της αφορά την μετατροπή του διαμορφωμένου σήματος από ψηφιακή σε αναλογική μορφή ώστε να είναι δυνατή η μετάδοση του σήματος εξόδου και κατά συνέπεια της πληροφορίας μέσω εκπομπής.

Στη συνέχεια θα μελετήσουμε τον προγραμματισμό του AD9831, αρχικά μέσω της παράλληλης θύρας του υπολογιστή και στη συνέχεια θα δούμε ποιες αλλαγές έγιναν ώστε ο προγραμματισμός να γίνει με τον μικροελεγκτή DS89C420.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ AD9831

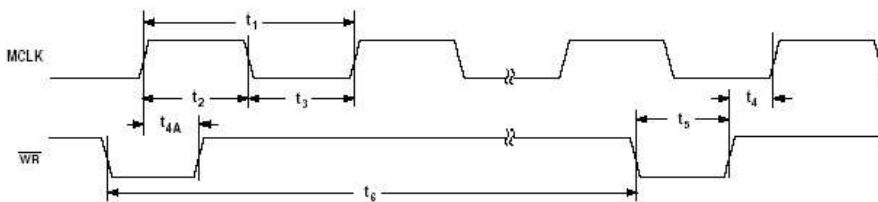
Όπως προαναφέραμε ο AD9831 είναι στην ουσία ένας επαναπρογραμματιζόμενος πομπός ο οποίος προγραμματίζεται από τη **παράλληλη θύρα** του υπολογιστή. Για να είμαστε πιο σαφείς, ο προγραμματισμός αφορά τον καθορισμό των συχνοτήτων ή του phase offset για την λειτουργία των ψηφιακών διαμορφώσεων μέσα στον επεξεργαστή. Επίσης μέσω του υπολογιστή ελέγχουμε το **reset** του phase accumulator, καθώς και την περίοδο του παλμού **WR**, ο οποίος είναι απαραίτητος για την αποθήκευση και την μανδάλωση των δεδομένων στον AD9831.

Με τις κατάλληλες τιμές στα control bits (A0, A1, A2) τα δεδομένα “φορτώνονται” στους καταχωρητές για κάθε $WR=0$ και αποθηκεύονται σε κάθε θετική ακμή του παλμού, όπως επίσης και οι τιμές στα control bits. Η εκάστοτε τιμή των καταχωρητών αναβαθμίζεται στη θετική ακμή του MCLK, δηλαδή του clock του επεξεργαστή. Εδώ ακριβώς απαιτείται προσοχή διότι δεν πρέπει να *συγχρονίζεται* ο WR με τον MCLK διότι *υπάρχει μια αβεβαιότητα του παλμού ρολογιού όσον αφορά το “φόρτωμα” του καταχωρητή προορισμού -ο εκάστοτε καταχωρητής πρέπει να “φορτώνεται” αμέσως ή πρέπει να αναβαθμίζεται στον επόμενο παλμό ρολογιού.*

Στο σχήμα που ακολουθεί φαίνεται ο συγχρονισμός του παλμού MCLK και WR

Parameter	Limit at T_{MIN} to T_{MAX} (A Version)	Units	Test Conditions/Comments
t_1	40	ns min	MCLK Period
t_2	16	ns min	MCLK High Duration
t_3	16	ns min	MCLK Low Duration
t_4^*	8	ns min	\overline{WR} Rising Edge to MCLK Rising Edge
t_{4A}^*	8	ns min	\overline{WR} Rising Edge After MCLK Rising Edge
t_5	8	ns min	\overline{WR} Pulse Width
t_6	t_1	ns min	Duration between Consecutive \overline{WR} Pulses
t_7	5	ns min	Data/Address Setup Time
t_8	3	ns min	Data/Address Hold Time
t_{9}^*	8	ns min	FSELECT, PSEL0, PSEL1 Setup Time Before MCLK Rising Edge
t_{9A}^*	8	ns min	FSELECT, PSEL0, PSEL1 Setup Time After MCLK Rising Edge
t_{10}	t_1	ns min	RESET Pulse Duration

* See Pin Description section.
Guaranteed by design but not production tested.

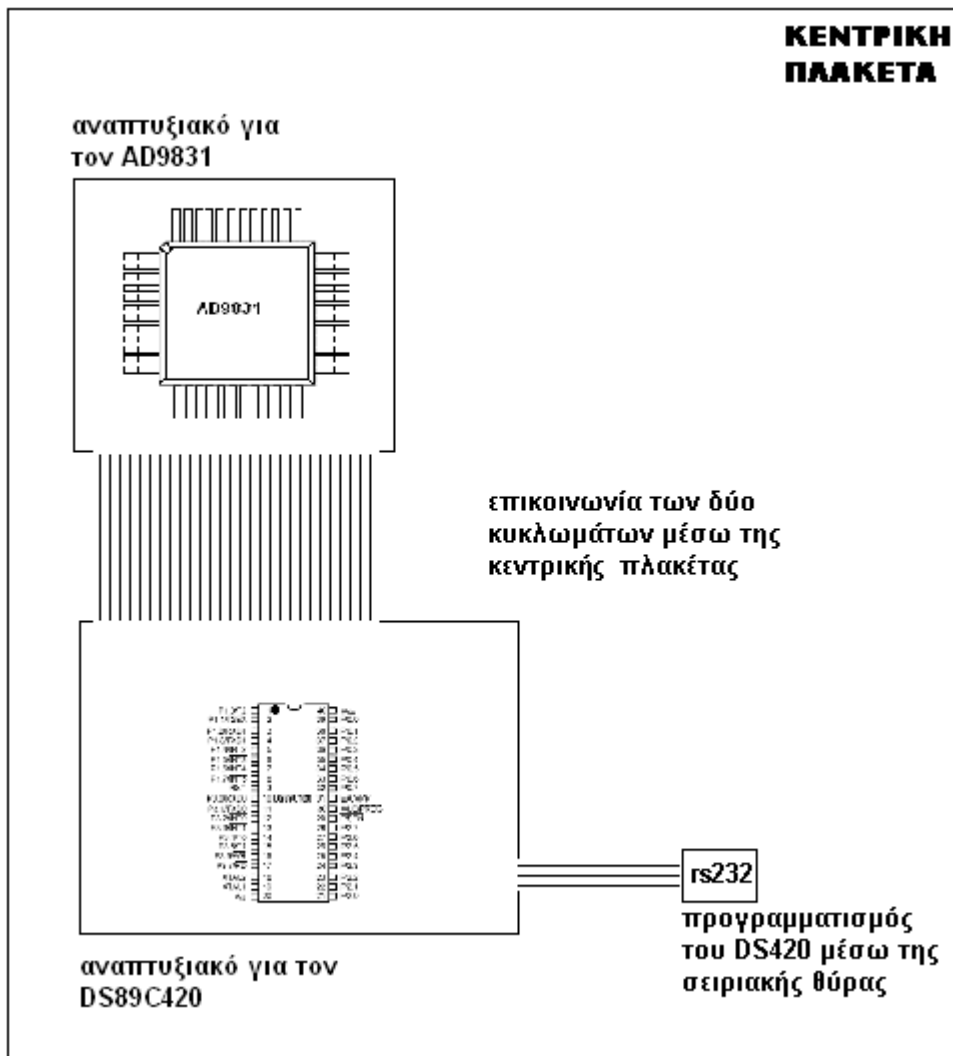


Στη επόμενη σελίδα φαίνεται το αναπτυξιακό του AD9831 το οποίο δίνει η εταιρία καθώς και η σύνδεσή του με τον υπολογιστή.

σύνδεση των παραπάνω ακροδεκτών και ορισμένων ακόμα με τις θύρες του DS89C420. Ο προγραμματισμός πλέον θα γίνει μέσω του μικροελεγκτή και όχι από την παράλληλη θύρα του υπολογιστή.

Η ιδέα υλοποιείται ως εξής:

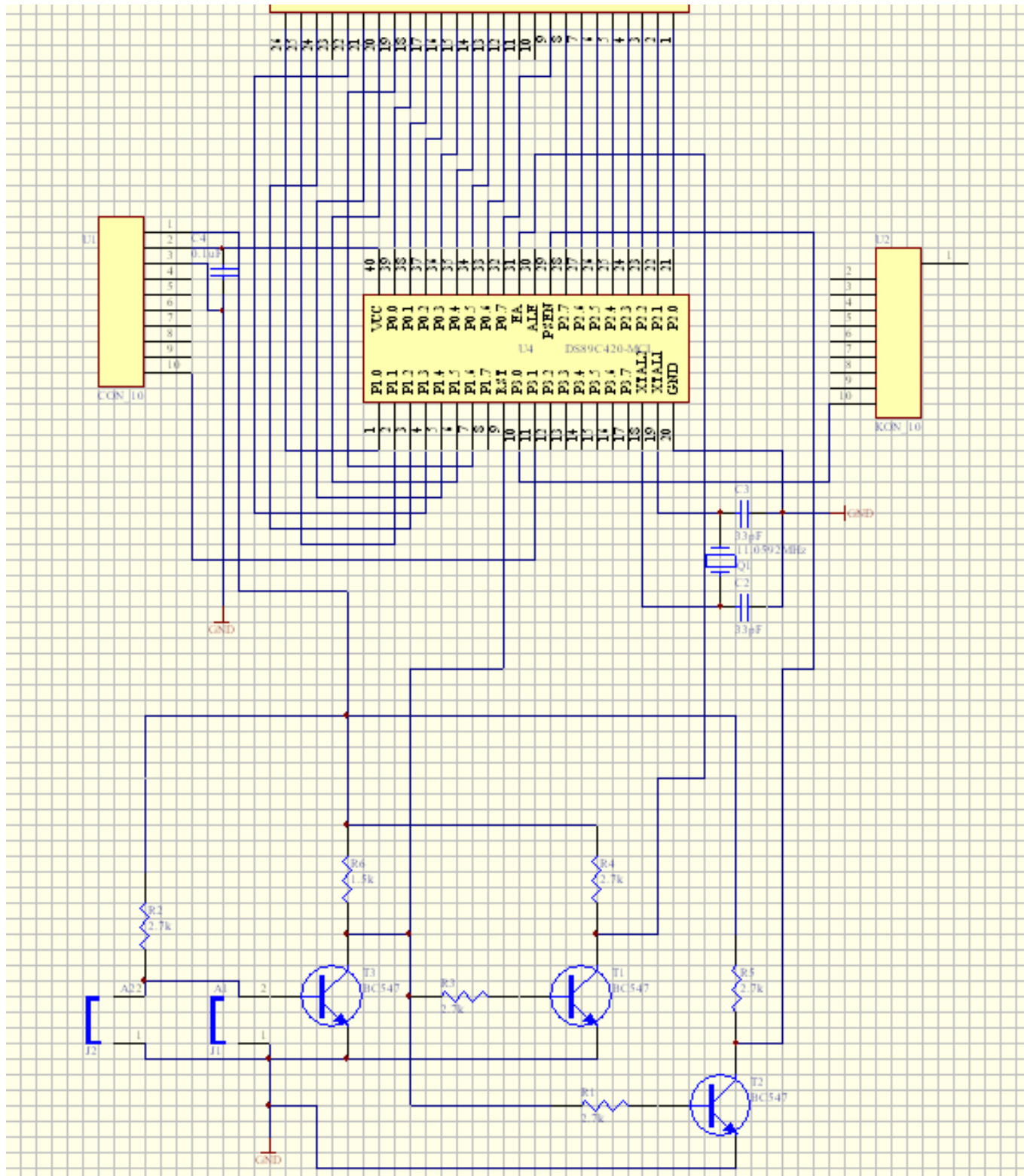
Η κατασκευή αποτελείται από τρεις πλακέτες, ένα στοιχειώδες αναπτυξιακό για τον μικροελεγκτή, ένα για τον AD9831 και μια κεντρική πλακέτα με υποδοχές για τις προηγούμενες δυο που αποσκοπεί στη σύνθεση ενός μεγάλου αναπτυξιακού στο οποίο θα επικοινωνούν οι δυο επεξεργαστές. Στο παρακάτω υπεραπλουστευμένο σχήμα φαίνεται η εικόνα των δυο ολοκληρωμένων συνδεδεμένα στη κεντρική πλακέτα.



Η επιλογή γι'αυτού του είδους την κατασκευή δεν έγινε στην τύχη. Ο απώτερος σκοπός ήταν να κατασκευαστεί ένα κεντρικό αναπτυξιακό το οποίο να διαθέτει δύο υποδοχές, ώστε να μπορούν να "εισέλθουν" οιοδήποτε transceivers DDS και μικροελεγκτές 8051. Η κεντρική αυτή πλακέτα θα πρέπει να παρέχει τροφοδοσία, υποστήριξη για σύνδεση με τον υπολογιστή, και γενικά ότι χρειάζεται για την υποστήριξη της σύνδεσής δυο επεξεργαστών. Αυτά όμως θα τα δούμε αναλυτικά στην συνέχεια αφού μελετήσουμε αναλυτικά τα σχέδια της κατασκευής ένα προς ένα.

ΑΝΑΠΤΥΞΙΑΚΟ ΚΥΚΛΩΜΑ ΓΙΑ ΤΟΝ DS89C420

Το αναπτυξιακό κύκλωμα για τον μικροελεγκτή, όπως και οι υπόλοιπες πλακέτες, σχεδιάστηκε με τη βοήθεια του προγράμματος **protel 99 SE** και βασίστηκε στο σχηματικό που αναφέραμε σε προηγούμενη σελίδα στην **εικόνα 4**. Το σχέδιο όμως που υλοποιήθηκε φαίνεται στο παρακάτω σχήμα.



Οι τρεις connectors που φαίνονται στο προηγούμενο σχήμα αποσκοπούν στην ένωση αυτής της πλακέτας με την κεντρική σύμφωνα με το αρχικό σχέδιο (εικόνα χ).

Οι ακροδέκτες που θα συνδεθούν με κεντρική πλακέτα θα είναι τα ports P0, P1, P2 του μικροελεγκτή καθώς και η τροφοδοσία η οποία θα παρέχεται από αυτήν με τη βοήθεια σταθεροποιητών τάσης. Τα τρία ports θα χρησιμεύσουν για τη μεταφορά δεδομένων από τον ένα μικροεπεξεργαστή στον άλλο.

Εκτός από τα προαναφερθέντα, η πλακέτα περιλαμβάνει ένα κρύσταλλο για το χρονισμό του μικροελεγκτή, ένα πυκνωτή μεταξύ τροφοδοσίας και γείωσης καθώς και ένα κύκλωμα για εναλλαγή από program mode σε run mode και έλεγχο του reset.

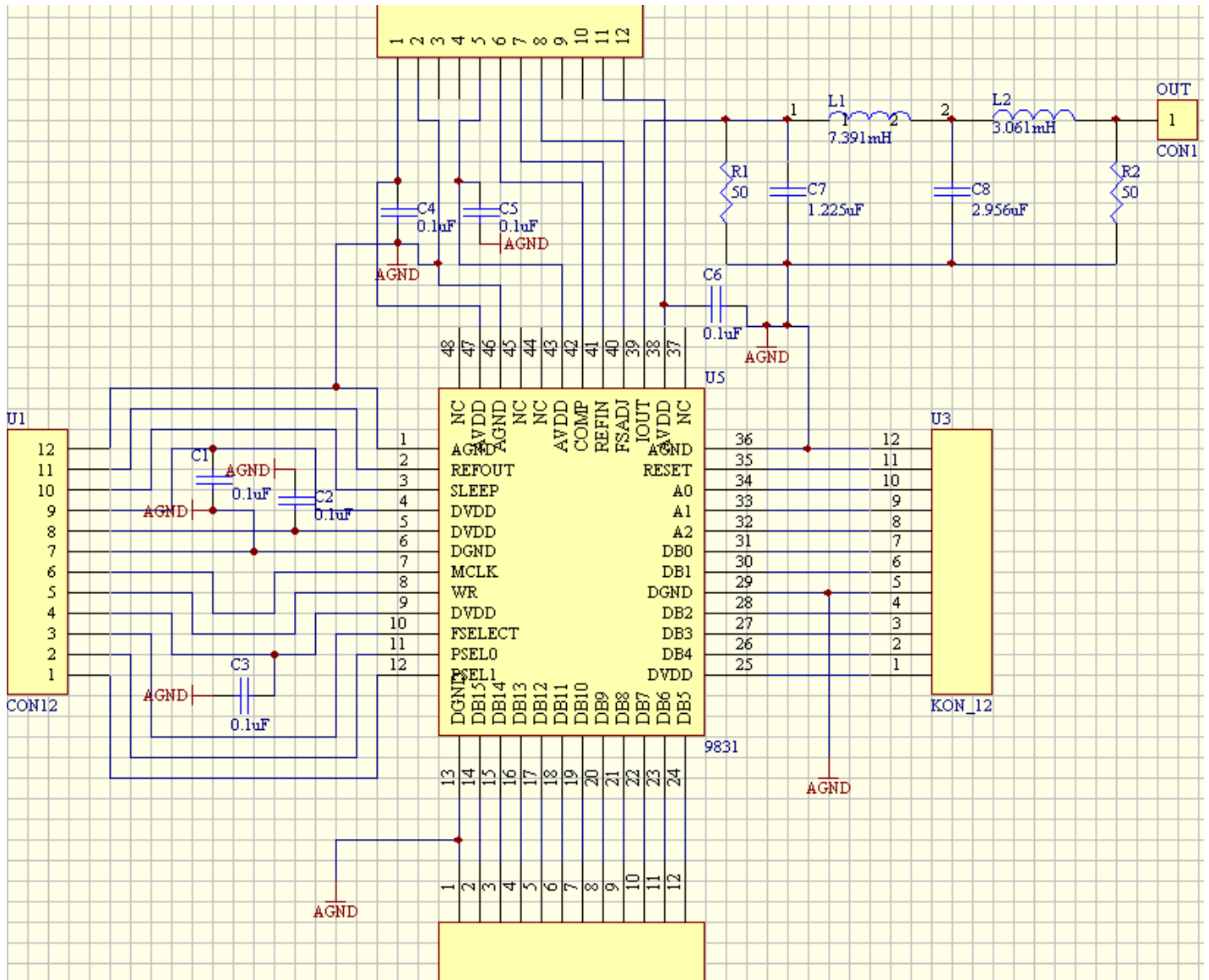
Η σύνδεση με τη σειριακή θύρα του PC για τον προγραμματισμό γίνεται με τη βοήθεια της κεντρικής πλακέτας στην οποία όπως θα δούμε στη συνέχεια υπάρχει RS232 connector. Τα pins RXD και TXD του DS89C420 συνδέονται με την κεντρική πλακέτα μέσω των connectors που αναφέραμε στην αρχή της σελίδας.

Το κύκλωμα στο κάτω μέρος της πλακέτας εξυπηρετεί με τη βοήθεια των jumpers τον έλεγχο του reset καθώς και την εναλλαγή της κατάστασης του μικροελεγκτή από program σε run mode. Η κατάσταση προγραμματισμού εξυπηρετεί την “φόρτωση” του εκάστοτε κώδικα από τον υπολογιστή στην μνήμη του DS89C420, και η κατάσταση εκτέλεσης την εφαρμογή του κώδικα.

Τέλος τα σημεία που η πλακέτα γειώνεται συνδέονται με τη γείωση του κοινού αναπτυξιακού. Αυτό συμβαίνει και με τη γείωση της πλακέτας του AD9831.

ΑΝΑΠΤΥΞΙΑΚΟ ΚΥΚΛΩΜΑ ΓΙΑ ΤΟΝ AD9831

Το σχηματικό υλοποιήθηκε και αυτό στο protel και σχεδιάστηκε σύμφωνα με το κύκλωμα που δίνει η εταιρεία κατασκευής του.



Η υλοποίηση του σχηματικού αυτού θα αποτελέσει την δεύτερη πλακέτα η οποία με τη σειρά της θα συνδεθεί στην κεντρική για την ολοκλήρωση της κατασκευής. Όπως αναφέραμε σε προηγούμενη σελίδα, η κεντρική πλακέτα είναι πλήρης όσον αφορά την παροχή τροφοδοσίας, παλμού ρολογιού, σύνδεσης με το PC και γενικά ότι χρειάζονται τα ολοκληρωμένα για να λειτουργήσουν. Έτσι λοιπόν και η πλακέτα του AD9831 περιλαμβάνει τα απολύτως απαραίτητα.

Αρχικά επτά πυκνωτές μεταξύ τροφοδοσίας και γείωσης (το ολοκληρωμένο διαθέτει 7 pins για τροφοδοσία). Ακόμα διαθέτει 4 connectors οι οποίοι θα συνδέσουν το κύκλωμα του πομπού με την κεντρική πλακέτα.

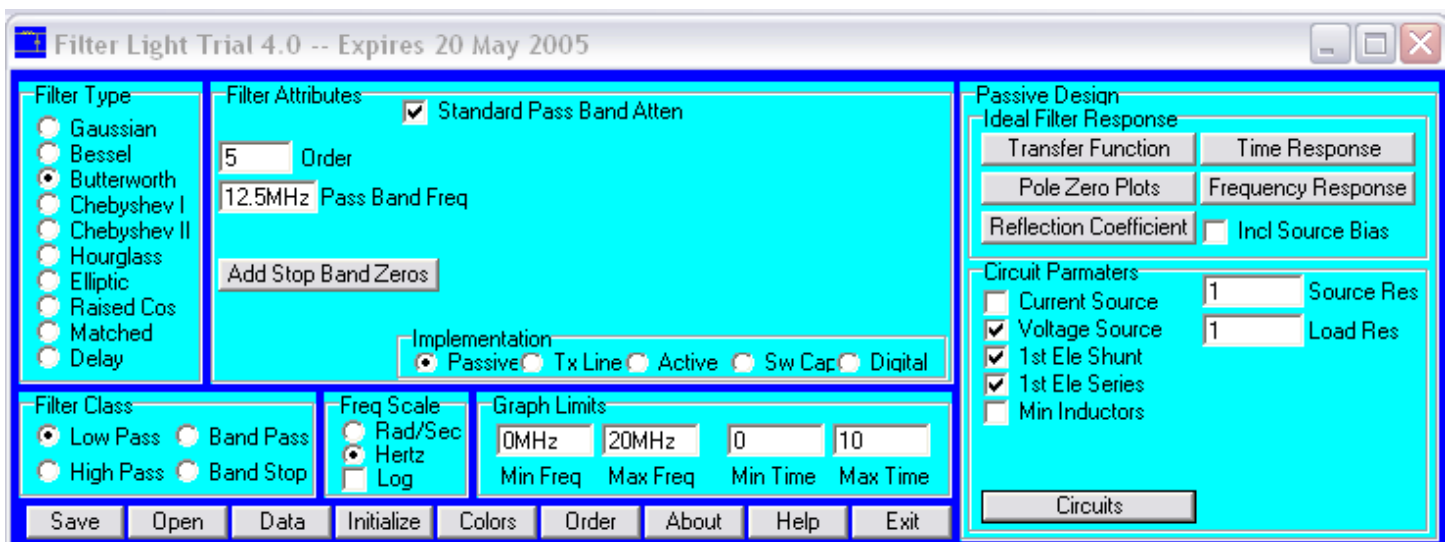
Τα pins του AD9831 που μας ενδιαφέρουν είναι καταρχήν τα **D0** έως **D15** από τα οποία θα διαβιάσουμε στους frequency και phase registers τις επιθυμητές τιμές. Επίσης συνδεδεμένοι πρέπει να είναι και οι ακροδέκτες **A0, A1, A2, PSEL0, PSEL1**, έτσι ώστε με τον κατάλληλο συνδυασμό 0 και 1 στην έξοδό τους να επιλέγουμε τον επιθυμητό καταχωρητή του AD9831.

Επίσης με τους connectors συνδέουμε και τα pins τροφοδοσίας και γείωσης ώστε να ενώνονται με την τροφοδοσία και τη γείωση της κεντρικής πλακέτας.

Ακόμη συνδέουμε τους ακροδέκτες **MCLK** και **WR** ώστε μέσω της κεντρικής πλακέτας να παρέχουμε στο ολοκληρωμένο τους παλμούς ρολογιού και WR.

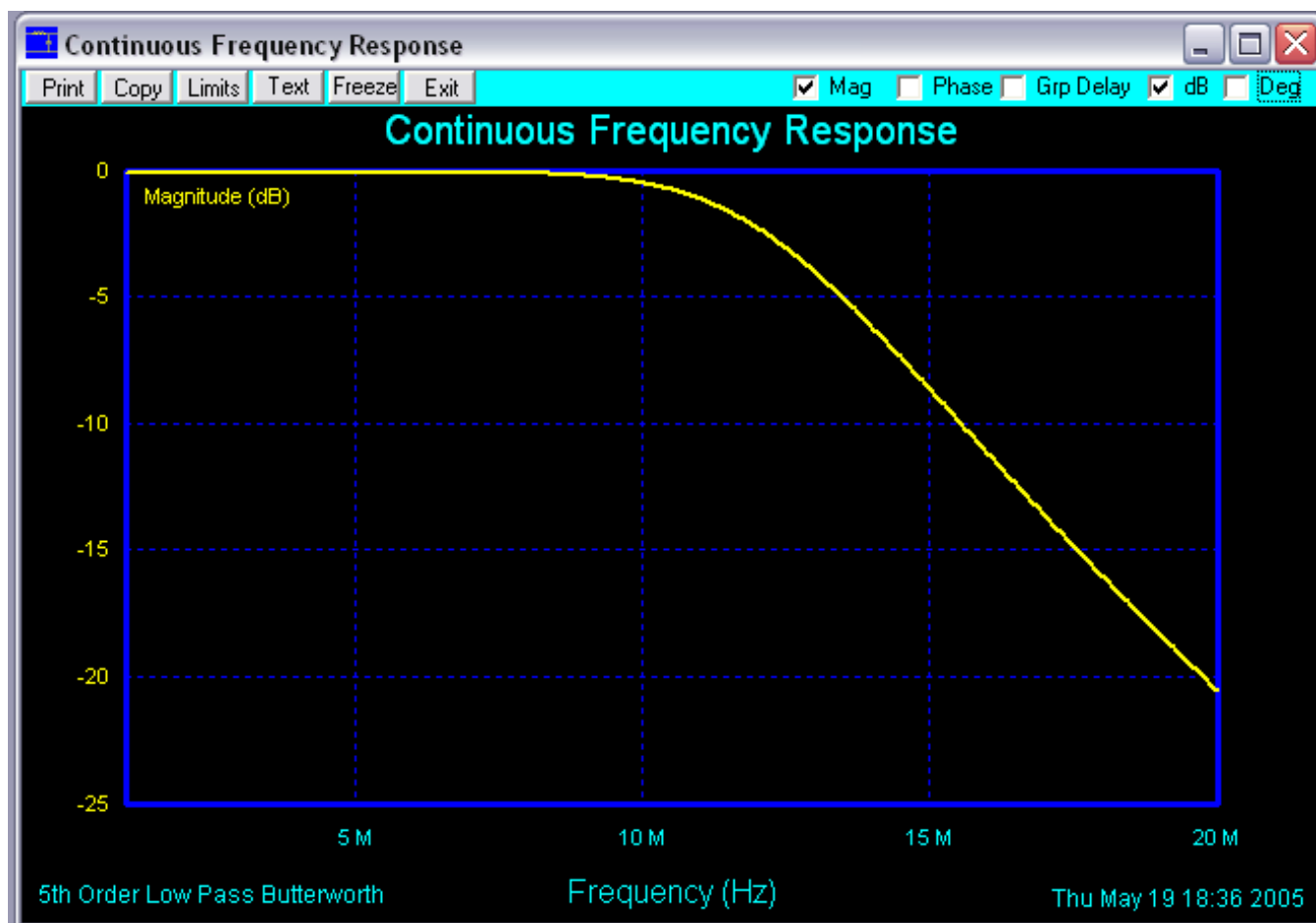
Τέλος τα pins **FSELECT** για την παροχή της, προς διαμόρφωση, πληροφορίας, καθώς και το **RESET** για τον phase accumulator.

Όπως παρατηρούμε στο σχηματικό στον ακροδέκτη **IOUT** από τον οποίο παίρνουμε την έξοδο του κυκλώματος, δηλαδή την διαμορφωμένη πληροφορία, είναι συνδεδεμένο ένα φίλτρο. Το φίλτρο αυτό σχεδιάστηκε με τη βοήθεια του λογισμικού **Filter Light** (trial version).



Το χαμηλοπερατό φίλτρο είναι τύπου Butterworth και 5^ο βαθμού και επιτρέπει συχνότητες έως 12.5MHz, δεδομένου ότι το ρολόι του AD9831 τρέχει στα 25MHz. Η προσθήκη του στο κύκλωμά μας είναι απαραίτητη για την αποφυγή αλλοίωσης της πληροφορίας στην έξοδο λόγω θορύβου. Παρακάτω φαίνεται η απόκρισή του όπως τη σχεδίασε το Filter Light.

ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ ΤΗΣ ΑΠΟΚΡΙΣΗΣ ΣΥΧΝΟΤΗΤΑΣ ΤΟΥ ΦΙΛΤΡΟΥ

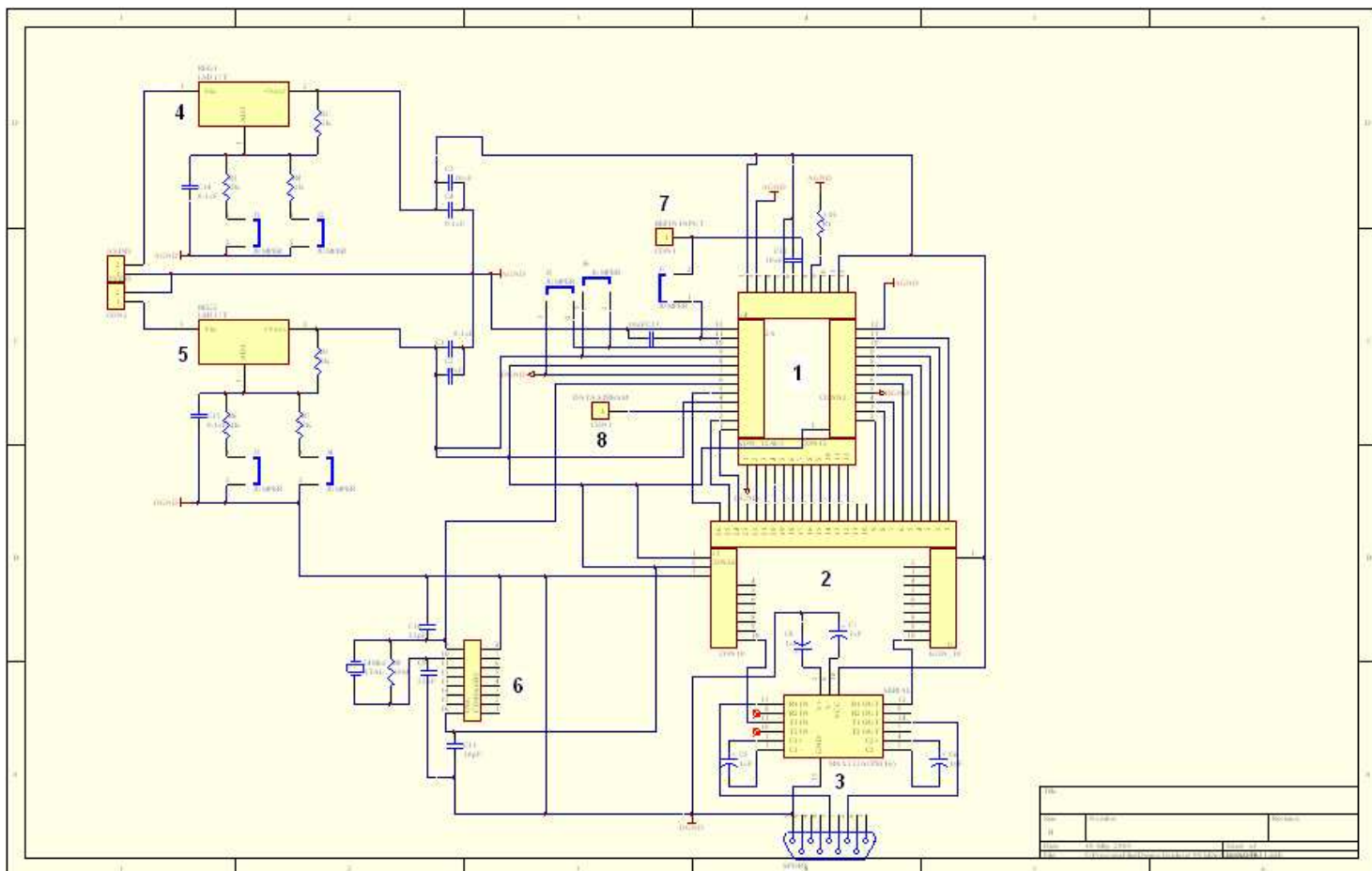


Τις τιμές των εξαρτημάτων για το σχεδιασμό του φίλτρου μας τις παρείχε επίσης το λογισμικό αν και οι τιμές των εξαρτημάτων που τελικά χρησιμοποιήθηκαν απείχαν λίγο από τις θεωρητικές τιμές.

Αυτό έγινε διότι δεν σταθηκε δυνατό να βρεθούν ακριβώς οι τιμές αυτές στο εμπόριο.

ΚΕΝΤΡΙΚΗ ΠΛΑΚΕΤΑ ΣΥΝΔΕΣΗΣ ΤΩΝ ΔΥΟ ΑΝΑΠΤΥΞΙΑΚΩΝ

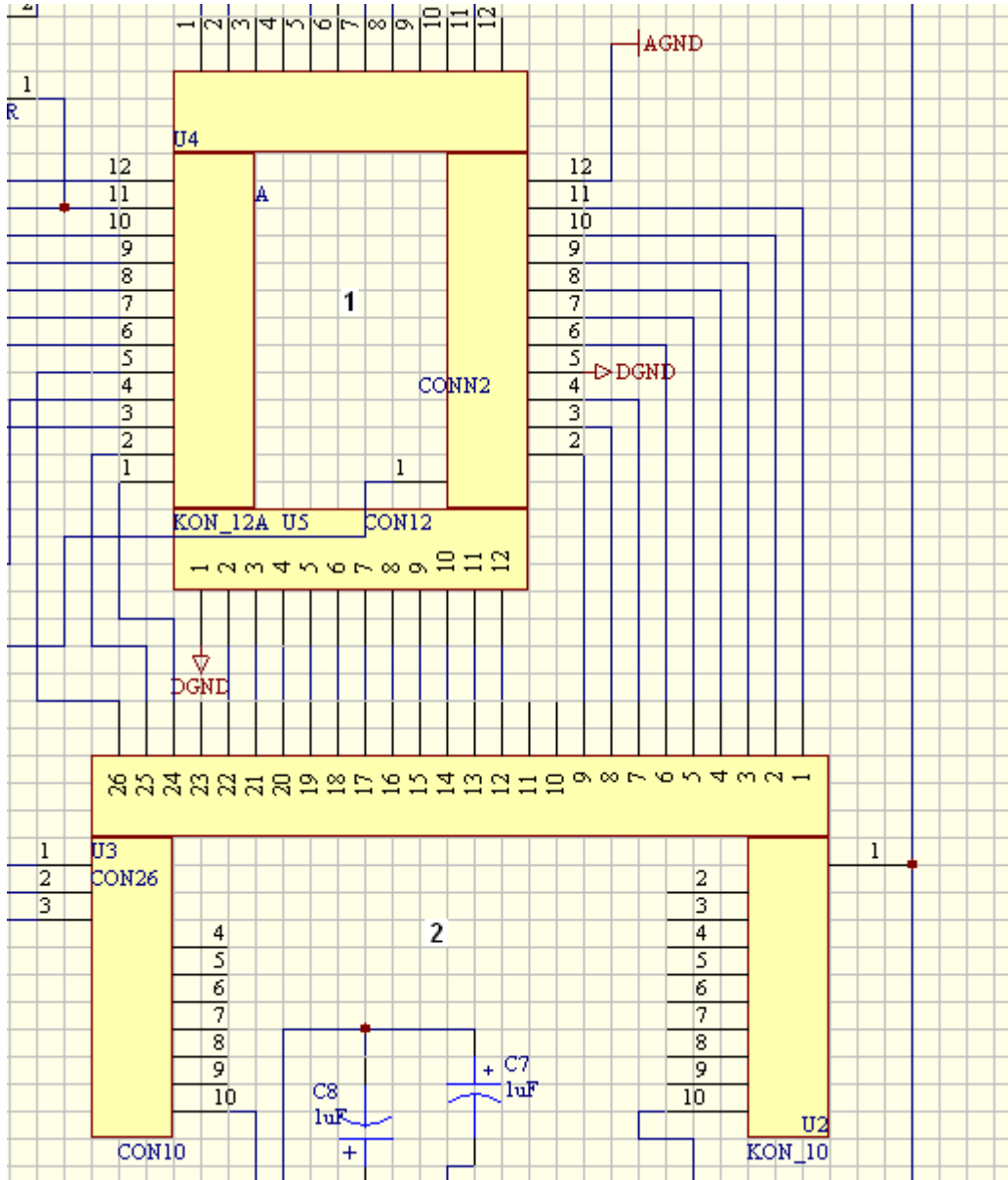
Ας περάσουμε τώρα να δούμε αναλυτικά την κεντρική πλακέτα πάνω στην οποία θα συνδεθούν τα δυο αναπτυξιακά με σκοπό τη συνεργασία τους για την υλοποίηση των ψηφιακών διαμορφώσεων. Όπως τα προηγούμενα κυκλώματα έτσι και αυτό υλοποιήθηκε στο protel 99 SE.



Για να κατανοήσουμε τη λειτουργία του αναπτυξιακού θα εξηγήσουμε μια προς μια τις βαθμίδες που το αποτελούν. Πριν προχωρήσουμε στην επόμενη σελίδα όμως πρέπει να τονίσουμε ότι αυτό είναι μόνο ένα σχηματικό. Οι θέσεις των εξαρτημάτων και οι αποστάσεις μεταξύ τους δεν έχουν καμία σχέση με την πραγματική κατασκευή δηλαδή το τυπωμένο κύκλωμα. Αλλά αυτά θα τα δούμε στη συνέχεια.

ΒΑΘΜΙΔΑ 1 & 2

Οι βαθμίδες 1 & 2 αποτελούν στην ουσία τις βάσεις στις οποίες θα μπου οι AD9831 και DS89C420 αντίστοιχα. Όπως παρατηρούμε οι ακροδέκτες συνδέονται μεταξύ τους έτσι ώστε να επιτυγχάνεται η επικοινωνία των δυο ολοκληρωμένων μέσω της κεντρικής πλακέτας.



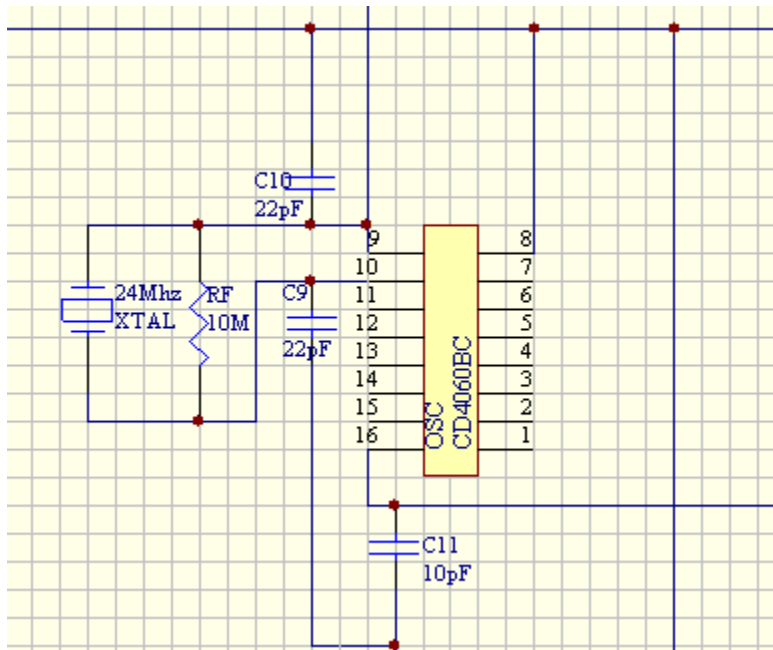
Από τα σχηματικά των δύο ολοκληρωμένων παρατηρούμε ότι τα ports P0, P1, και P2 του DS89C420 συνδέονται με τους ακροδέκτες D0-D15 και τα bits ελέγχου A0, A1, A2, PSEL0, PSEL1 του AD9831. Αυτή η σύνδεση μεταξύ των δύο επεξεργαστών είναι αρκετή ώστε να περάσουμε από τον μικροελεγκτή τα κατάλληλα δεδομένα στον πομπό για να εκτελέσει την ψηφιακή διαμόρφωση.

Έχοντας προγραμματίσει κατάλληλα τον μικροελεγκτή μέσω του υπολογιστή, αυτός με τη σειρά του δίνει τα απαραίτητα δεδομένα στον πομπό (συχνότητες, phase offset) ώστε να υλοποιήσει την επιθυμητή διαμόρφωση (FSK ή PSK). Στην ουσία δηλαδή ο DS89C420 αντικαθιστά την παράλληλη θύρα του υπολογιστή.

ΒΑΘΜΙΔΑ 6

Το κύκλωμα αυτό είναι στην ουσία ένας ταλαντωτής του οποίου η έξοδος αποτελεί τον παλμό ρολογιού για τον χρονισμό του AD9831. Η έξοδος του ταλαντωτή, από το pin 9 του ολοκληρωμένου, συνδέεται με τον ακροδέκτη MCLK του AD9831 για την παροχή του παλμού ρολογιού.

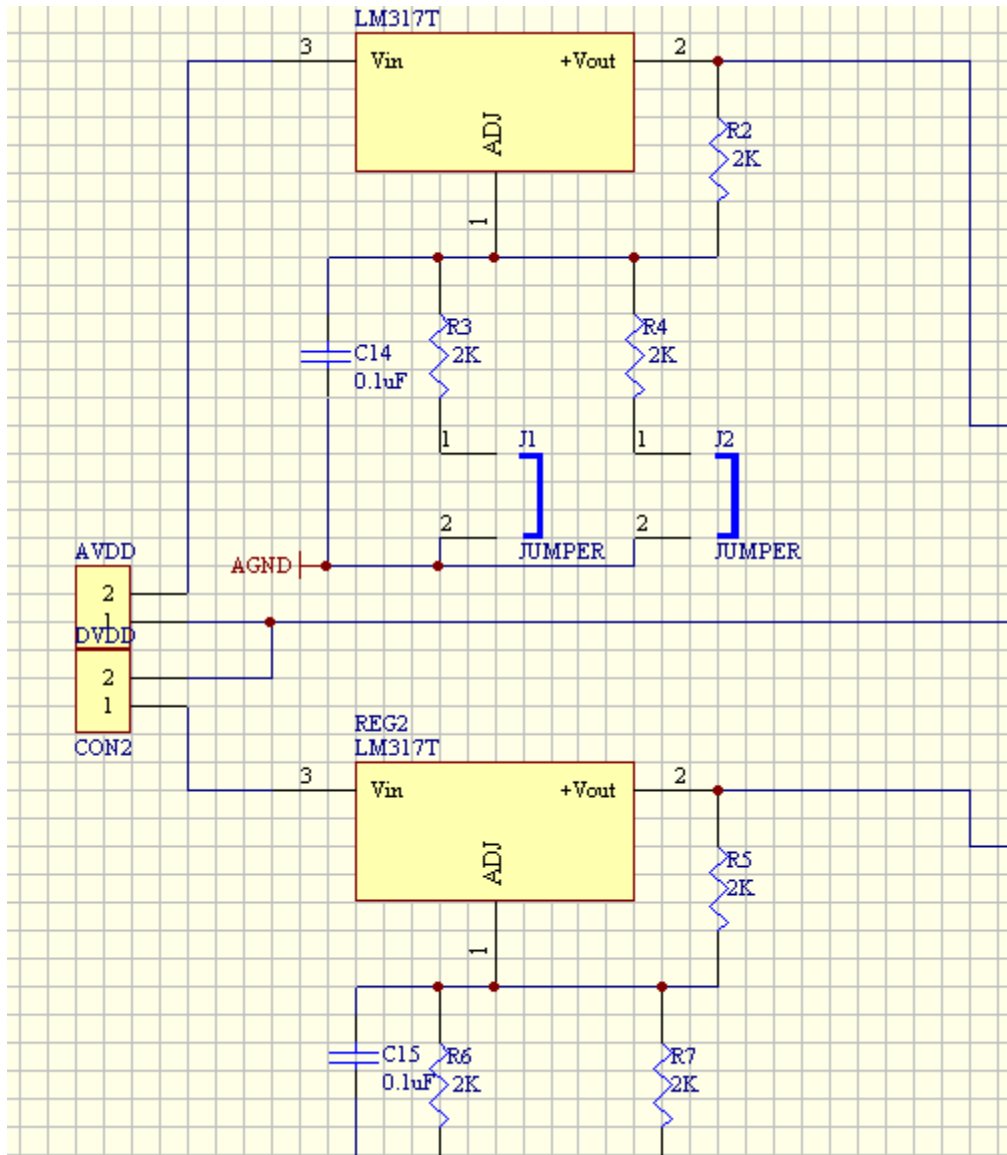
Ο κρύσταλλος που χρησιμοποιήθηκε στο κύκλωμα έχει συχνότητα ταλάντωσης 24MHz.



ΒΑΘΜΙΔΕΣ 4 & 5

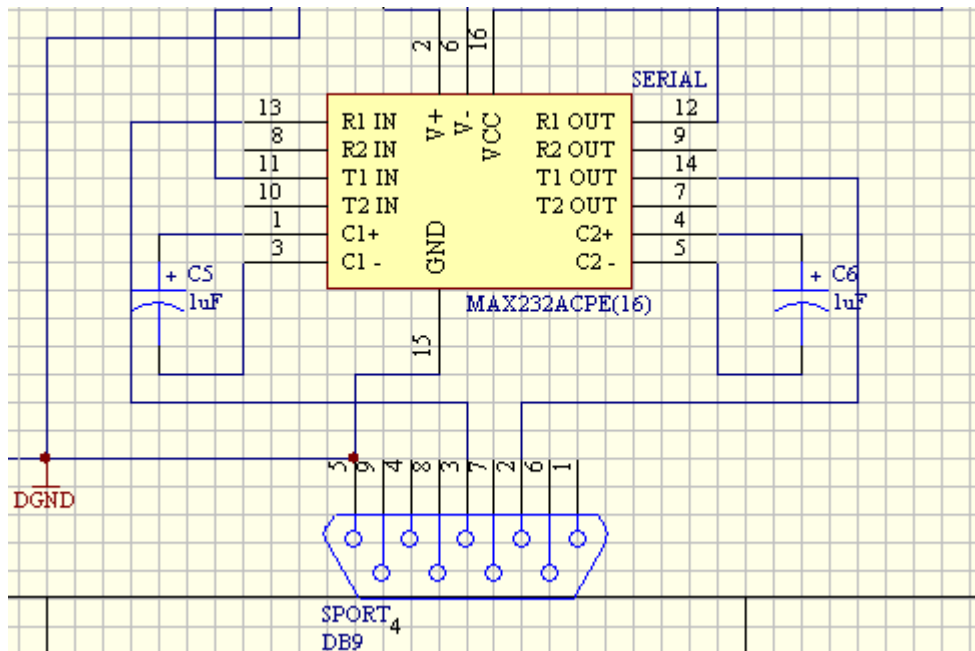
Οι βαθμίδες 4 & 5 είναι κυκλώματα που παρέχουν σταθερή τροφοδοσία 5 Volts στο αναπτυξιακό. Όπως παρατηρούμε στο σχηματικό υπάρχουν δύο ξεχωριστές παροχές τροφοδοσίας μια AVDD (analog) και μια DVDD (digital). Ο σχεδιασμός έγινε κατά αυτόν τον τρόπο διότι ο AD9831 χρησιμοποιεί δύο τάσεις τροφοδοσίας, μια αναλογική και μια ψηφιακή. Στην ουσία είναι η ίδια τάση, αλλά το για το σχεδιασμό του αναπτυξιακού ακολουθήσαμε πιστά την αρχιτεκτονική του AD9831.

Τα jumpers εξυπηρετούν την επιλογή της τάσης ανάμεσα σε 3.5V και 4.5V, τιμές στις οποίες λειτουργούν και οι δύο επεξεργαστές του αναπτυξιακού.



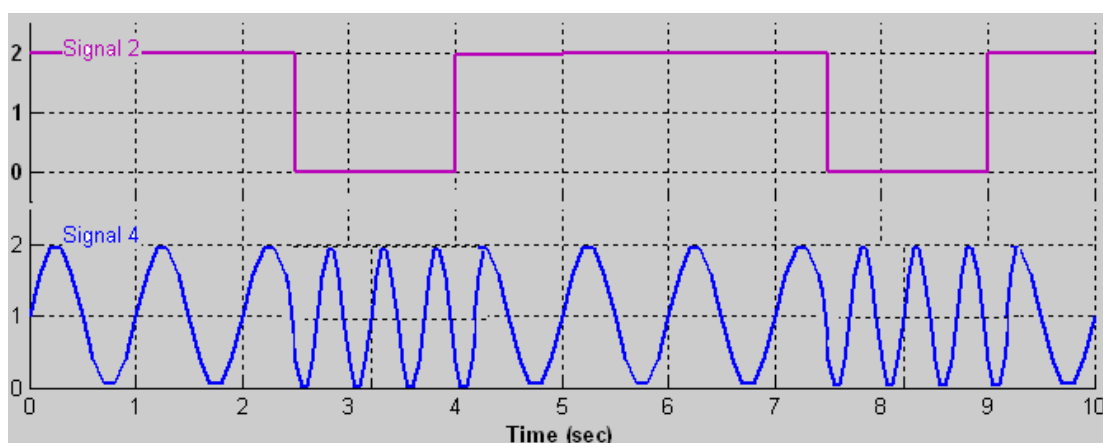
ΒΑΘΜΙΔΑ 3

Η βαθμίδα αυτή περιλαμβάνει το ολοκληρωμένο MAX 232 με τον οποίο ο DS89C420 συνδέεται με τη σειριακή θύρα του υπολογιστή για τον προγραμματισμό του. Στο σχηματικό φαίνεται και ο connector RS232 ο οποίος συνδέει την πλακέτα με τη σειριακή θύρα του ηλεκτρονικού υπολογιστή.



ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΨΗΦΙΑΚΗΣ ΔΙΑΜΟΡΦΩΣΗΣ

Αρχικά θα επιχειρήσουμε να υλοποιήσουμε μετάδοση ψηφιακής πληροφορίας με διαμόρφωση κατά συχνότητα. Για την πραγματοποίηση αυτού του είδους διαμόρφωσης απαιτείται η δημιουργία ενός αναλογικού σήματος του οποίου η συχνότητα θα μεταβάλλεται σύμφωνα με την αλλαγή της πληροφορίας από 0 σε 1. Ένα παράδειγμα φαίνεται στο παρακάτω σχήμα, όπου το **signal 2** είναι η πληροφορία και το **signal 4** το αναλογικό FSK σήμα.



Σκοπός μας λοιπόν είναι να φορτώσουμε στον AD9831 δύο 32bit δεκαεξαδικές λέξεις οι οποίες θα αντιστοιχούν στις 2 συχνότητες που θα αποτελέσουν το αναλογικό FSK σήμα. Οι λέξεις αυτές θα οδηγηθούν στον πομπό μέσω του μικροελεγκτή DS89C420, τον οποίο θα προγραμματίσουμε με το ED95 από τη σειριακή θύρα του υπολογιστή.

Για την ψηφιακή διαμόρφωση κατά συχνότητα μας αρκούν δύο ports του μικροελεγκτή από τα οποία θα περνάμε τις δεκαεξαδικές λέξεις, ένα port το οποίο θα ενημερώνει τα control bits του AD9831(A0, A1, A2) και τέλος ένα port για τη δημιουργία του παλμού WR, απαραίτητου για τη φόρτωση των δεδομένων από τον μικροελεγκτή στον πομπό.

Στη σελίδα που ακολουθεί φαίνεται το αρχικό στάδιο του προγράμματος για την υλοποίηση της ψηφιακής διαμόρφωσης κατά φάση. Μοναδικός σκοπός του όπως προαναφέραμε είναι να περάσει στον DS89C420 τις δύο συχνότητες και στη συνέχεια ο επεξεργαστής εκτελώντας το πρόγραμμα να οδηγήσει τις συχνότητες αυτές στον AD9831. Όλες οι υπόλοιπες διεργασίες που αφορούν την διαμόρφωση και εκπομπή της πληροφορίας είναι αποκλειστική δουλειά του transceiver AD9831.

ΚΩΔΙΚΑΣ ΓΙΑ FSK ΔΙΑΜΟΡΦΩΣΗ

```
defseg start,absolute
seg start
org 00h
```

main:

```
mov p2,#00000011b      ; control bits(A0, A1, A2)

mov p1,#00011000b      ; Freg1 16 MSBs
mov p0,#10110100b

acall write             ; κλήση του παλμού WR για την μεταφορά των data
acall delay

mov p2,#00000010b      ; control bits

mov p1,#10001001b      ; Freg1 16 LSBs
mov p0,#01011000b

acall write             ; κλήση του παλμού WR για την μεταφορά των data
acall delay

mov p2,#00000001b      ; control bits

mov p1,#01011100b      ; Freg0 16 MSBs
mov p0,#1001111b

acall write             ; κλήση του παλμού WR για την μεταφορά των data
acall delay

mov p2,#00000000b      ; control bits

mov p1,# 01101111b     ; Freg0 16 LSBs
```

mov p0, #01110110b

acall write
acall delay

; κλήση του παλμού WR για την μεταφορά των data

sjmp \$

delay:

; ρουτίνα καθυστέρησης για τον
; καθορισμό της περιόδου του παλμού

mov r2, #14h
djnz r2, \$

ret

write:

; δημιουργία ρουτίνας παλμού για το control
; bit WR

clr p3.7
acall delay

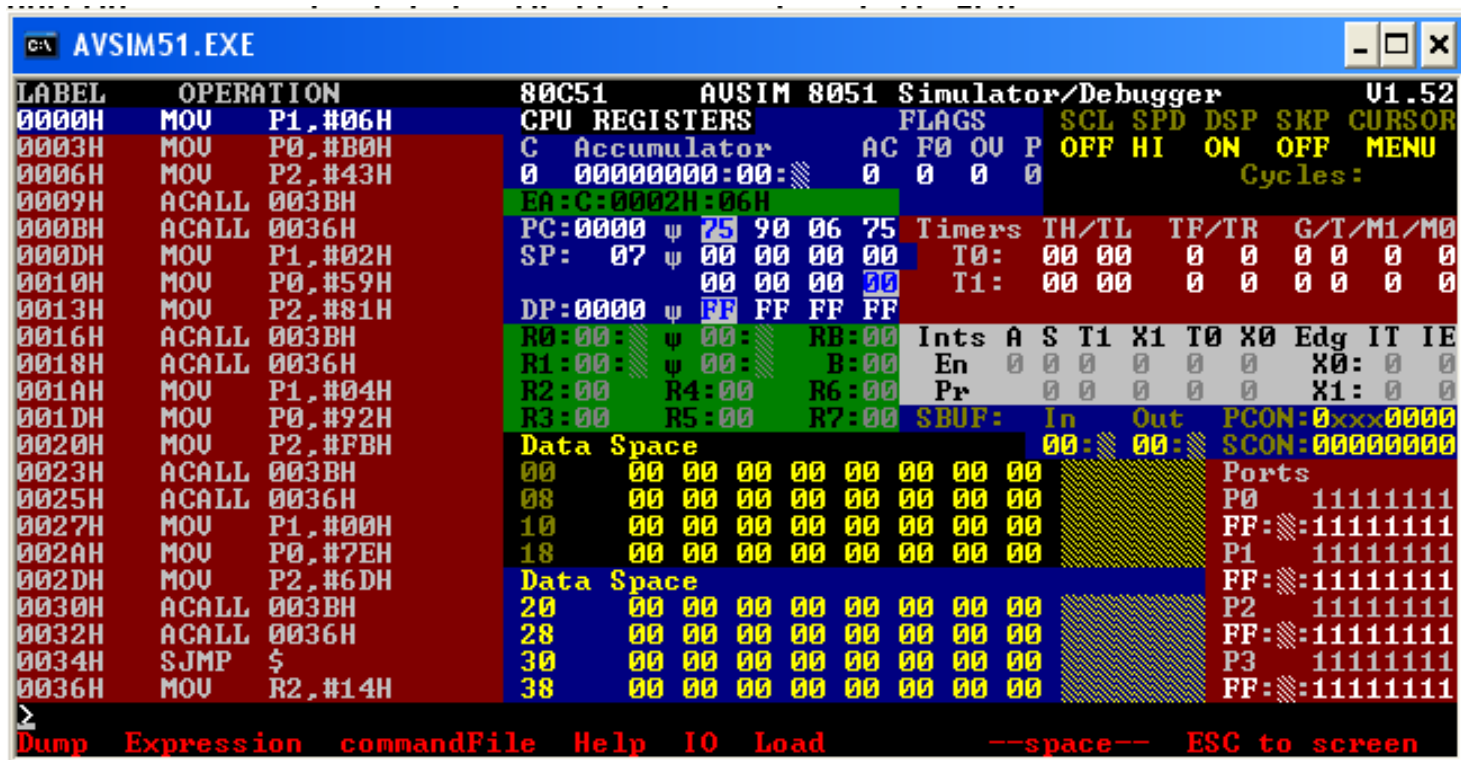
; θετική ημιπερίοδος του παλμού WR

setb p3.7
acall delay

; αρνητική ημιπερίοδος του παλμού WR

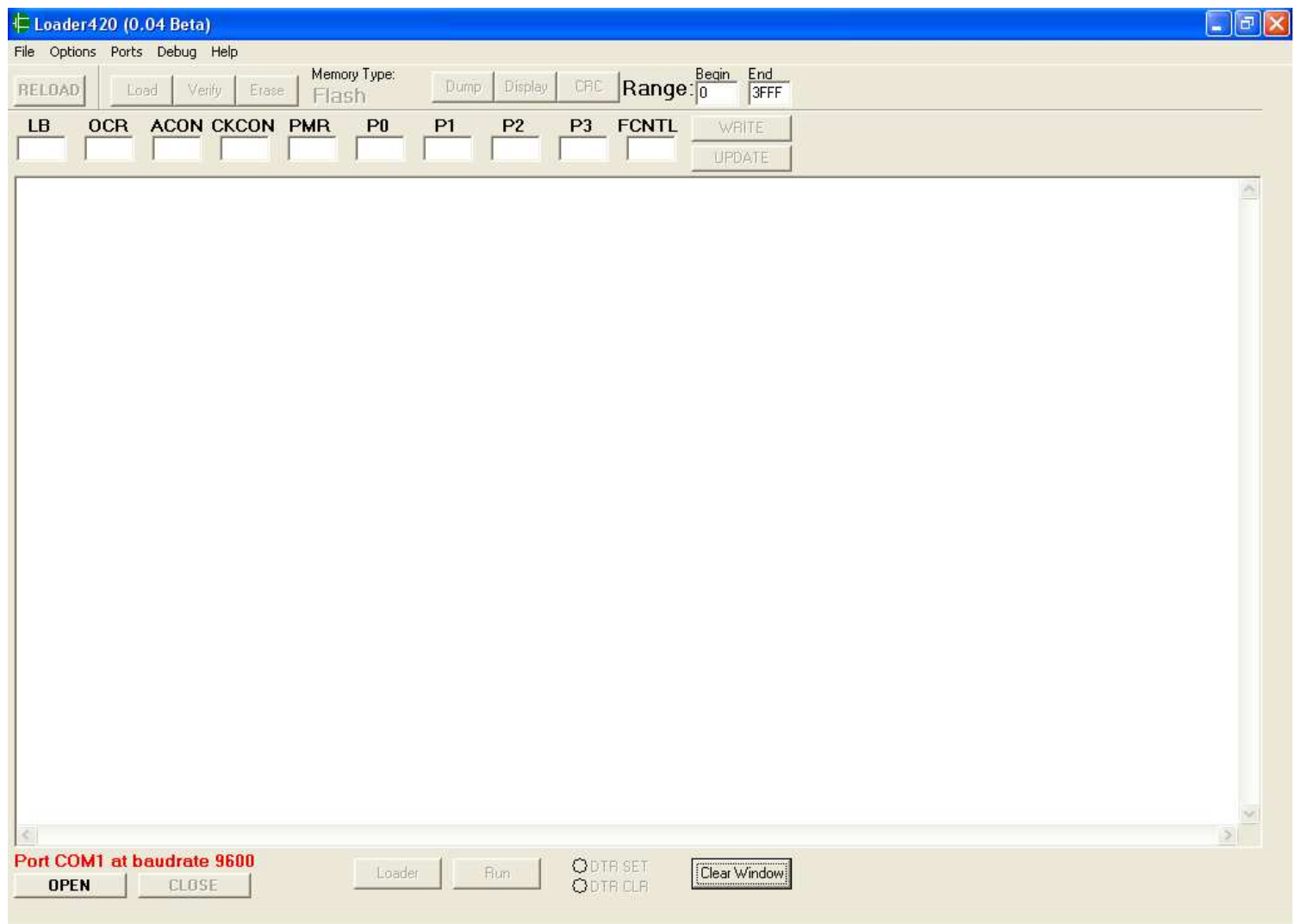
ret
end

Η δημιουργία του κώδικα έγινε με τη βοήθεια του λογισμικού ED98 το οποίο περιέχει όλες τις εντολές της αρχιτεκτονικής 8051. Παρέχει επίσης τη δυνατότητα εκτέλεσης του εκάστοτε κώδικα μέσω εξομοιωτή, ο οποίος απεικονίζει στην οθόνη όλες τις εισόδους και εξόδους του ολοκληρωμένου καθώς και τις μονάδες μνήμης του. Με αυτόν τον τρόπο η είναι εφικτός ο έλεγχος του κώδικα βήμα προς βήμα όσον αφορά τον τρόπο με τον οποίο επιδρά μέσα στο στις ακολουθιακές μονάδες του μικροελεγκτή. Ένα παράδειγμα του simulation φαίνεται στο παρακάτω σχήμα.



Όπως μπορούμε να παρατηρήσουμε φαίνονται όλες οι εντολές που αποτελούν τον κώδικα για την FSK διαμόρφωση. Επίσης διακρίνονται τα τέσσερα εξωτερικά ports, οι εσωτερικοί καταχωρητές, οι περιοχές μνήμης οι timers κ.α.

Ακόμη πρέπει να πούμε ότι με τη βοήθεια του ED98 μετατρέψαμε τον κώδικα σε αρχείο hex (δεκαεξαδικό). Το αρχείο αυτό αποθηκεύτηκε και εκτελέστηκε στον DS89C420 με το απαραίτητο λογισμικό loader 420 (επόμενη σελίδα). Το πρόγραμμα αυτό μας επιτρέπει να επικοινωνούμε άμεσα με τον μικροελεγκτή από την σειριακή θύρα του υπολογιστή μας.



Παρακάτω δίνεται η λίστα με τα εξαρτήματα που χρησιμοποιήθηκαν.

ΛΙΣΤΑ ΕΞΑΡΤΗΜΑΤΩΝ ΚΑΤΑΣΚΕΥΗΣ (ΣΥΝΟΛΟ 80)

ΑΝΑΠΤΥΞΙΑΚΟ ΚΥΚΛΩΜΑ ΓΙΑ ΤΟΝ DS89C420 -- 18 εξαρτήματα

1 μικροελεγκτής DS89C420
3 connectors(ακιδοδοσειρές)
2 πυκνωτές 33pF
1 κρύσταλλος 11,0592MHz
2 jumpers
3 transistors BC547
5 αντιστάσεις 2,7KΩ
1 αντίσταση 1,5KΩ

ΑΝΑΠΤΥΞΙΑΚΟ ΚΥΚΛΩΜΑ ΓΙΑ ΤΟΝ AD9831 -- 17 εξαρτήματα

1 ολοκληρωμένο AD9831 της analog
1 πηνίο 3,061mH
1 πηνίο 7,391mH
1 πυκνωτή 1,225μF
1 πυκνωτή 2,956μF
5 πυκνωτές 0,1μF
2 αντιστάσεις 50Ω
5 connectors(ακιδοδοσειρές)

ΚΕΝΤΡΙΚΗ ΠΛΑΚΕΤΑ -- 45 εξαρτήματα

2 σταθεροποιητές τάσης LM317T
6 αντιστάσεις 2KΩ
4 πυκνωτές 0,1μF
2 πυκνωτές 10μF
7 jumpers
1 κρύσταλλο 24MHz
1 αντίσταση 10MΩ
2 πυκνωτές 22pF
1 πυκνωτή 10pF
1 ολοκληρωμένο CD4060BC
1 ολοκληρωμένο MAX232A
4 πυκνωτές 1μF
1 connector RS232
2 πυκνωτές 10nF
1 αντίσταση 3,9KΩ
11 connectors(ακιδοδοσειρές)

ΤΕΛΟΣ