



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Κρήτης – Παράρτημα Χανίων**



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

*"Γεννήτρια προτύπων σημάτων (patterns) ασπρόμαυρου & έγχρωμου
οπτικού σήματος (video) με χρήση μικροελεγκτή, για τον έλεγχο της ποιότητας
της εικόνας των τηλεοπτικών δεκτών (τηλεοράσεων)"*

Γιώργος Παπουτσάκης

Επιβλέπων καθηγητής

Δρ. Κουριδάκης Στέλιος

ΧΑΝΙΑ ΔΕΚΕΜΒΡΙΟΣ 2010

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά όσους με βοήθησαν για την ολοκλήρωση της παρούσας εργασίας, και ιδιαίτερα τον καθηγητή κύριο Κουριδάκη Στέλιο για την πολύτιμη βοήθειά του, και για τις γνώσεις που απέκτησα όλο αυτό το διάστημα.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ	3
ΚΕΦΑΛΑΙΟ 2: ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	6
ΚΕΦΑΛΑΙΟ 3: ΑΡΧΕΣ ΛΕΙΤΟΥΡΓΙΑΣ & ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	8
ΚΕΦΑΛΑΙΟ 4: ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ & ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ.....	16
ΚΕΦΑΛΑΙΟ 5: ΑΝΑΛΥΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ & PCB LAYOUT.....	21
ΚΕΦΑΛΑΙΟ 6: ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ.....	26
ΚΕΦΑΛΑΙΟ 7: ΑΝΑΛΥΣΗ ΒΟΗΘΗΤΗΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΓΙΑ ΤΟ PC ΣΕ ΓΛΩΣΣΑ Visual Basic.....	57
ΒΙΒΛΙΟΓΡΑΦΙΑ	64
ΠΑΡΑΡΤΗΜΑ 1: DATASHEET'S ΤΩΝ ΕΞΑΡΤΗΜΑΤΩΝ.....	65

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Η παρούσα πτυχιακή εργασία αφορά την μελέτη των σημάτων που χρησιμοποιούνται στην αναλογική έγχρωμη τηλεόραση, και στην μελέτη και κατασκευή μονάδας παραγωγής βασικών σημάτων, έγχρωμου και ασπρόμαυρου βίντεο (patterns) , που χρησιμοποιούνται για τον έλεγχο ή επισκευή συσκευών τηλεόρασης & οθονών CRT.

Τα σήματα που παράγει η παρούσα κατασκευή είναι: σήματα για τα τρία βασικά χρώματα (RGB) 0-1 Vp-p και σήματα συγχρονισμού, οριζόντιας απόκλισης, κάθετης απόκλισης, και σύνθετο σήμα συγχρονισμού: (Vertical – Horizontal & Composite synchro Pulses) σε επίπεδο TTL.

Στο σημείο αυτό αξίζει να σημειωθεί, για να έχει ο αναγνώστης μια καλύτερη και πιο σφαιρική εικόνα για την παρούσα μελέτη και υλοποίηση, ότι η πτυχιακή εργασία και κατασκευή αποτελεί επιμέρους τμήμα ενός μεγαλύτερου project. Το τελικό project αφορά ένα πλήρης εργαστηριακό όργανο, το οποίο θα είναι μια ολοκληρωμένη μονάδα παραγωγής σημάτων βίντεο και ήχου, καθώς και διαμόρφωση ήχου και εικόνας στα τηλεοπτικά κανάλια στην περιοχή των UHF με χρήση κυκλώματος PLL. Αποτελείται από τις εξής υπό μονάδες: Την κεντρική μονάδα (παρούσα πτυχιακή εργασία), που με χρήση μικροελεγκτή, παράγει τα σήματα RGB καθώς και τα σήματα συγχρονισμού, και επίσης πραγματοποιεί τον έλεγχο των άλλων υπό μονάδων μέσω δίαυλου I²C Bus. Την μονάδα μετατροπής των RGB σημάτων και σημάτων συγχρονισμού σε σύνθετο σήμα βίντεο στο σύστημα PAL (PAL composite video), την μονάδα του ήχου, την μονάδα διαμόρφωσης με χρήση PLL, και τέλος την μονάδα τροφοδοσίας.

Ο μικροελεγκτής που χρησιμοποιήθηκε για την υλοποίηση του κυκλώματος είναι ο DS89C430 της Dallas Semiconductors, ο οποίος είναι μια βελτιωμένη έκδοση του κλασικού 8051 της Intel. Χρησιμοποιήθηκε ένας γρήγορος μετατροπέας Digital->Analog, Ο ADV7125 της Analog Devices, ο οποίος περιέχει τρεις A/D και είναι σχεδιασμένος για χρήση σε εφαρμογές RGB παραγωγής video. Επίσης χρησιμοποιήθηκε το ολοκληρωμένο MAX7456 που είναι μία μονάδα OSD (On Screen Display), το οποίο κάνει εισαγωγή κειμένου, ή άλλων αποθηκευμένων στην μνήμη του χαρακτήρων σε ένα υπάρχον σήμα video. Το ολοκληρωμένο αυτό δίνει επίσης την δυνατότητα να φτιάξει ο χρήστης custom χαρακτήρες και να τους αποθηκεύσει στην εσωτερική EPROM που διαθέτει. Για τον σκοπό της αποθήκευσης custom χαρακτήρων στην μνήμη του MAX7456, γράφτηκε πρόγραμμα στην Visual Basic, το οποίο στην συνέχεια το εξελίξα ώστε να μπορεί ο χρήστης να γράψει και να εμφανίσει κείμενο στην οθόνη.

Τέτοιου είδους εργαστηριακά όργανα – συσκευές έχουν αναπτυχθεί και υπάρχουν στο εμπόριο και στους πάγκους των τεχνικών χρόνια τώρα, και η παρούσα μελέτη δεν αποτελεί κάποια καινοτομία. Ωστόσο από την μελέτη και την υλοποίηση ενός κυκλώματος για την παραγωγή σημάτων βίντεο, όπως αυτό που παρουσιάζω, με χρήση μικροελεγκτή, απέκτησα πολύτιμη εμπειρία και μου έδωσε πολλές γνώσεις και πάνω στο αντικείμενο της τηλεόρασης, στην επεξεργασία αναλογικών σημάτων, καθώς και στην ψηφιακή τεχνολογία και στην χρήση των μικροελεγκτών.

Abstract:

The following thesis is related to the study of signals that are used in analog color television and the study and construction of a production unit, basic signals, color and black and white patterns, which are used for the control or repair of television sets & CRT screens.

The signals produced from this construction are: signals for the three basic colors (RGB) 0-1 Vp-p and synchronization signals of horizontal deviation, vertical deviation and composite synchronization signal: (Vertical -Horizontal & Composite sync Pulses) in TTL level.

At this point it is worth mentioning that this thesis and construction is a part of a bigger project, so that the reader has a more round picture for the current study and implementation. The final project is about a complete laboratory instrument, that will be a complete video and sound production unit, as well as modulation of sound and picture in television channels in UHF area with the use of PLL circuit.

It consists of the following subunits: The central unit (Current thesis), which, with the use of microcontrollers, produces RGB signals as well as synchronization signals, and it also controls the other subunits by I²C Bus. The RGB and synchronization signals conversion unit in composite video signal in PAL system, the sound unit, the modulator unit with use of PLL and finally the power supply unit.

The microcontroller that was used for this circuit is the DS89C430 of Dallas Semiconductors, which is an improved version of the classic 8051 from Intel. There was used a fast Digital->Analog converter ADV7125 of Analog Devices, which includes three A/D's and is designed for use in RGB video production applications. There was also used the chip MAX7456 which is an OSD (On Screen Display) unit, which makes text insertion, or other saved in memory characters in an already existent video signal. This chip also gives the opportunity for the user to make custom characters, and save them to the internal EPROM that it has. For this purpose a PC program was created in Visual Basic so the user can download characters to the EPROM, the program was improved so the user can write text to the screen also.

Lab instrument like this, are developed and exist at the market, many years now, and the current thesis is not an innovation. Also from the study and construction of such a circuit, for the production of video signals, like the one I am presenting to you, with the use of microcontroller, I gained important experience and knowledge on the subject of television, on analog signals process, and the digital technology and use of microcontrollers.

ΚΕΦΑΛΑΙΟ 2: Σύντομη ιστορική αναδρομή

Καθότι οι πρώτες συσκευές τηλεοράσεως ήταν πλήρως αναλογικά κυκλώματα, χρειαζόνταν συχνά ρύθμιση ώστε η εικόνα να αναπαράγεται πιστά δίχως παραμόρφωση. Οι πρώτες συσκευές που χρησιμοποιήθηκαν για τον έλεγχο και την ρύθμιση των συσκευών τηλεοράσεως ήταν τα μονοσκόπια (monoscope) . Η λειτουργία τους ήταν ανάλογη με αυτήν μιας κάμερας λήψεως, μόνο που ήταν πολύ μικρότερες συσκευές, όπου ο καθοδικός σωλήνας που έκανε την λήψη ήταν αρκετά μικρός και είχε ενσωματωμένη την δοκιμαστική καρτέλα (test card) που τραβούσε στο εσωτερικό του. Στους τηλεοπτικούς σταθμούς, για τον ίδιο σκοπό χρησιμοποιήθηκαν επίσης κάμερες λήψεως όπου τραβούσαν συνεχόμενα την δοκιμαστική καρτέλα. Τα πλεονεκτήματα του μονοσκοπίου ήταν ότι πάντα ήταν εστιασμένο και ότι δεν υπήρχε ο κίνδυνος, να “καεί” η εικόνα της δοκιμαστικής καρτέλας στον φώσφορο της κάμερας μόνιμα. Τα μονοσκόπια έγιναν λιγότερο δημοφιλή στα τέλη του 1960 λόγω της αδυναμίας τους στην παραγωγή έγχρωμου βίντεο. Οι τηλεοπτικοί σταθμοί, για την μετάδοση έγχρωμων καρτελών χρησιμοποιούσαν έγχρωμες κάμερες λήψης όπου τραβούσαν τις έγχρωμες κάρτες. Με την εξέλιξη των ηλεκτρονικών και των ψηφιακών κυκλωμάτων, έκαναν την εμφάνισή τους συσκευές, που με ψηφιακά κυκλώματα έκαναν σύνθεση πλέον, του έγχρωμου σήματος βίντεο και των έγχρωμων καρτελών.

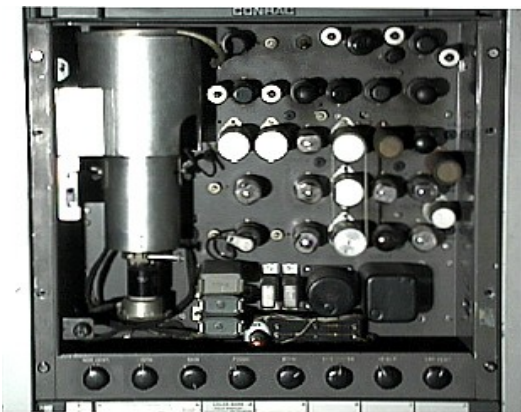
Συσκευές Monoscope:



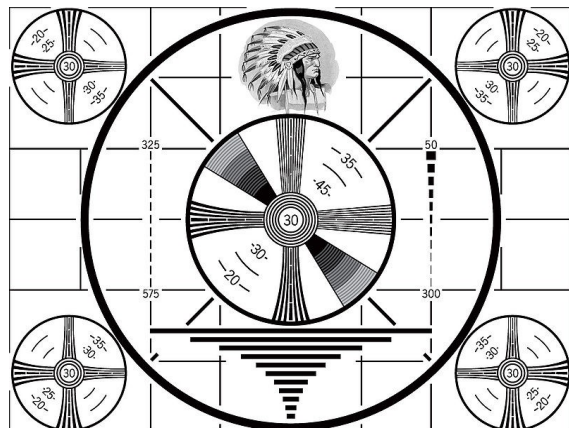
καθοδικός σωλήνας μονοσκοπίου



Κάρτα στο εσωτερικό του σωλήνα

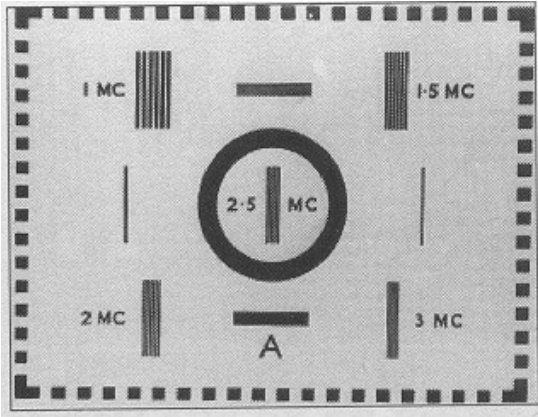


Συσκευή της RCA (RCA TK-1C Monoscope)



Η ασπρόμαυρη Κάρτα του RCA TK-1C Monoscope

Καρτέλες που χρησιμοποιήσαν τηλεοπτικοί σταθμοί:

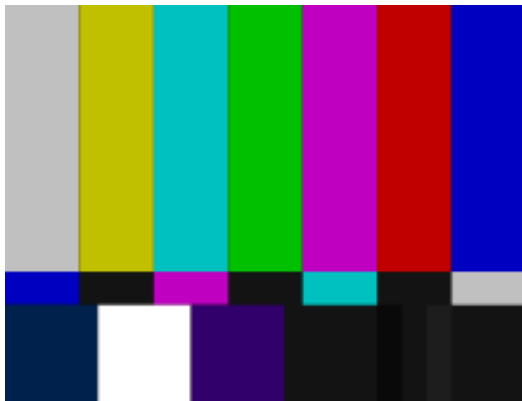


Η πρώτη ασπρόμαυρη καρτέλα που αναμεταδόθηκε απο τηλεοπτικό σταθμό (BBC test card A)



Πρώτη έγχρωμη καρτέλα (BBC test card F)

Ηλεκτρονικές Γεννήτριες:



SMPTE color bars (CBS Laboratories)



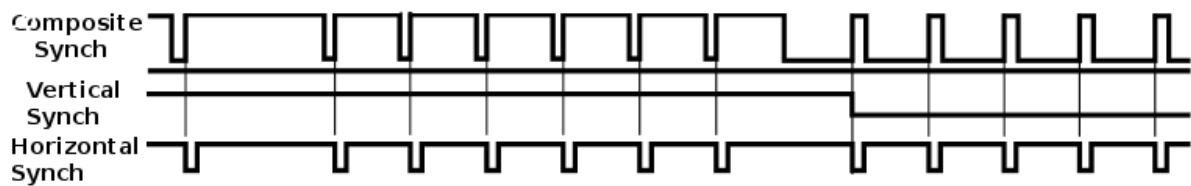
Απο τις πιο διαδεδομένες γεννήτριες, η Phillips PM5544 Test Pattern generator

ΚΕΦΑΛΑΙΟ 3: Αρχές λειτουργίας & Τεχνικά χαρακτηριστικά συστήματος

Έξοδος:

Το κύκλωμα που κατασκεύασα παρέχει εξόδους RGB & εξόδους για τα σήματα συγχρονισμού.

Οι εξόδοι RGB είναι 1 V_{p-p}, με το επίπεδο του μαύρου στο 0V, μπορούν να πάρουν 16 διακριτές τιμές από 0V μέχρι 1V. Τα σήματα συγχρονισμού που παρέχει η κατασκευή είναι αρνητικοί παλμοί κάθετου & οριζόντιου συγχρονισμού καθώς και σύνθετο σήμα συγχρονισμού σε επίπεδο TTL. Παρακάτω φαίνονται τα τρία αυτά σήματα συγχρονισμού:



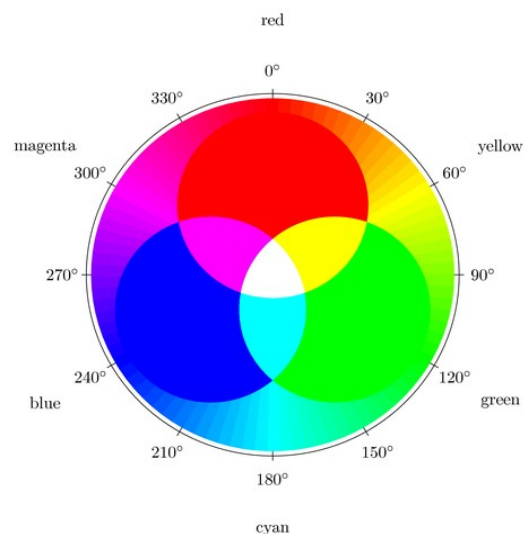
Οι χρόνοι των σημάτων είναι σύμφωνα με τα ευρωπαϊκά πρότυπα, και αναλυτικά: περίοδος οριζόντιας σάρωσης = 64μsec, περίοδος κάθετης σάρωσης 20msec, 625 οριζόντιες γραμμές σε 2 interlaced πεδία (άρτιο – περιττό) 312,5 γραμμές ανά πεδίο, συχνότητα πεδίου 50Hz, συχνότητα καρέ, που αποτελείται από 2 πεδία 25Hz, aspect ratio = 4:3.

Πρότυπο χρωματικής κωδικοποίησης RGB:

Το πρότυπο χρώματος RGB είναι ένα προσθετικό πρότυπο στο οποίο τα χρώματα κόκκινο, πράσινο και μπλε συνδυάζονται με διάφορους τρόπους για να αναπαραχθούν τα άλλα χρώματα. Το όνομα του προτύπου και η σύντηξη RGB προέρχονται από τα τρία βασικά χρώματα, το κόκκινο (Red), πράσινο (Green), και το μπλε (Blue).

Μια κοινή εφαρμογή του προτύπου χρώματος RGB είναι οι οθόνες καθοδικού σωλήνα, υγρών κρυστάλλων ή πλάσματος. Κάθε εικονοστοιχείο στην οθόνη μπορεί να αντιπροσωπευθεί ως τιμές έντασης για το κόκκινο, πράσινο και μπλε.

Το πρότυπο RGB προδιαγράφει ξεχωριστή μεταφορά κάθε μίας χρωματικής συνιστώσας (κόκκινη, πράσινη και μπλε) σε ξεχωριστό καλώδιο και προσφέρει την καλύτερη δυνατή ποιότητα. Στο διπλανό σχήμα (χρωματικός κύκλος) φαίνεται πώς συνδυάζοντας τα τρία βασικά χρώματα δημιουργείται όλο το οπτικό φάσμα.



Τροφοδοσία κυκλώματος:

Η τροφοδοσία του κυκλώματος είναι 5Vdc, και η κατανάλωση 250mA, για πλήρη έξοδο (100% white). Για τις ανάγκες του κυκλώματος κατασκευάστηκε τροφοδοτικό 5V – 1A.

Διασύνδεση Χρήστη (User Interface):

Η διασύνδεση με τον χρήστη γίνεται μέσω ενός πληκτρολογίου τεσσάρων πλήκτρων, και της οθόνης υγρών κρυστάλλων LCD (4 x 20) που διαθέτει. Επίσης διαθέτει θύρα RS-232, που χρησιμοποιείται για τον προγραμματισμό του μικροελεγκτή DS89C430, για την αποθήκευση custom χαρακτήρων στην EPROM του ολοκληρωμένου MAX7456, και για να γράψει ο χρήστης κείμενο στην οθόνη της τηλεόρασης. Τα 2 τελευταία μέσω του προγράμματος που γράφτηκε στην Visual Basic. Το κύκλωμα για τον προγραμματισμό του DS89C430 έχει ενσωματωθεί στην πλακέτα της κατασκευής έτσι ώστε να μπορεί να προγραμματίζεται χωρίς να απαιτείται να αφαιρέσουμε τον μικροελεγκτή και να χρησιμοποιήσουμε ξεχωριστό αναπτυξιακό.

Έχει υλοποιηθεί διάυλος I²C Bus για την σύνδεση με τις άλλες μονάδες στο τελικό project, και διάυλος SPI για τον προγραμματισμό του MAX7456.

Πρωτόκολλο RS-232:

Το πλέον γνωστό πρότυπο για τη σειριακή μεταφορά δεδομένων μεταξύ συστημάτων είναι το RS-232 (EIA232). Το αρχικό πρότυπο καθορίζει το φυσικό (ηλεκτρικό) επίπεδο, τα αντίστοιχα σήματα και τις υποδοχές σύνδεσης (connectors) για την επικοινωνία μεταξύ ενός “Data Terminal Equipment” (DTE, συνήθως ένας υπολογιστής) και ενός “Data Communications Equipment” (DCE, συνήθως ένα modem). Στην πράξη όμως το RS-232 χρησιμοποιείται σε μία ευρεία κλίμακα εφαρμογών. Τα σήματα του RS232 περιλαμβάνουν γραμμές μετάδοσης και λήψης δεδομένων (TxD, RxD), σήματα ελέγχου modem (request to send - RTS, clear to send - CTS, data terminal ready - DTR, data set ready - DSR κ.ά.), καθώς και την κοινή τάση αναφοράς (γείωση). Οι λογικές στάθμες είναι οι ακόλουθες:

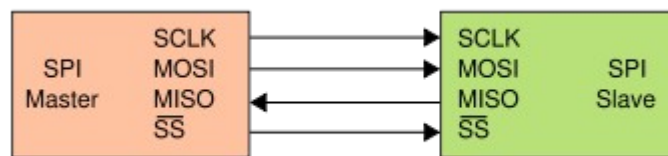
- Το ‘1’ αντιπροσωπεύεται από τάση -3 έως -25V (mark).
- Το ‘0’ αντιπροσωπεύεται από τάση 3 έως 25V (space).

Η μέγιστη ταχύτητα μεταφοράς δεδομένων είναι περίπου 20Kbps, η σύνδεση είναι πάντοτε μεταξύ δύο σημείων (point-to-point) και το μήκος του καλωδίου μπορεί να φτάσει τα 10 έως 15m.

Για τη μετάφραση των λογικών επιπέδων TTL/CMOS από και προς τα επίπεδα του RS-232 χρησιμοποιούνται ειδικά κυκλώματα οδήγησης. Στην παρούσα κατασκευή χρησιμοποιήθηκε το MAX232. Τα κυκλώματα αυτά χρησιμοποιούν διατάξεις με πυκνωτές (charge pumps) για να παράγουν από την απλή τροφοδοσία +5V τις απαιτούμενες από το πρότυπο RS-232 τάσεις.

Πρωτόκολλο SPI:

Το Πρωτόκολλο **SPI** ή **Serial Peripheral Interface Bus** επιτρέπει την σειριακή σύγχρονη επικοινωνία μεταξύ ολοκληρωμένων σε πλήρης αμφίδρομη επικοινωνία. Ο διάλογος υλοποιήθηκε για πρώτη φορά από την εταιρία Motorola. Οι συσκευές επικοινωνούν μεταξύ τους σε mode Master/Slave. Ο Master του διαύλου είναι το ολοκληρωμένο που παράγει το frame των δεδομένων και το μεταδίδει προς τα ολοκληρωμένα slave. Μπορούν σε έναν SPI διάλογο να διασυνδεθούν περισσότερες από μία συσκευές slave χρησιμοποιώντας της γραμμές Chip Select. Πολλές φορές το SPI το αποκαλούν "σειριακό διάλογο 4 καλωδίων".

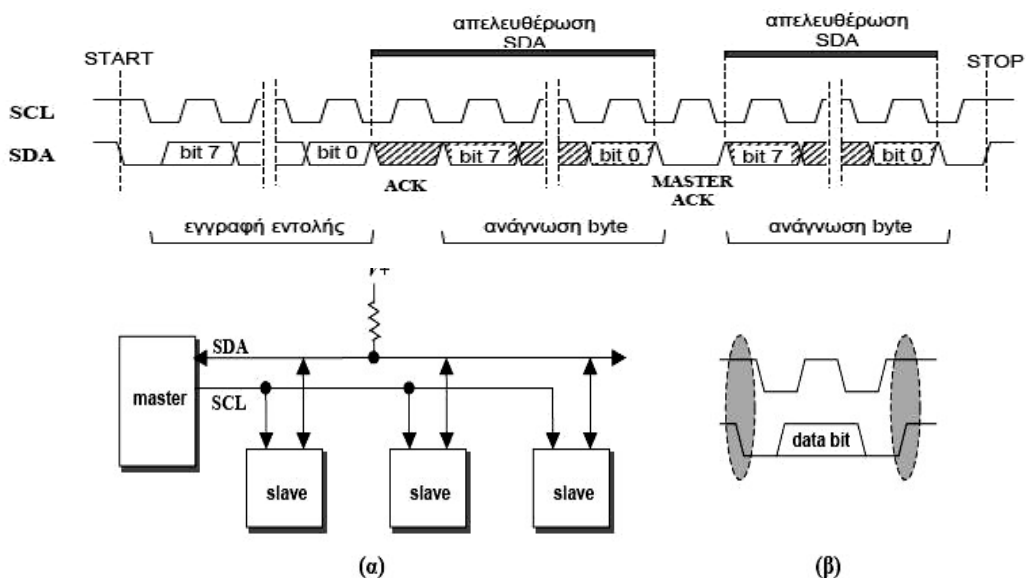


Χαρακτηριστικά του SPI:

1. Επιτρέπει την σύγχρονη επικοινωνία.
2. Είναι σειριακό.
3. Είναι πλήρως αμφίδρομο (full-duplex).
4. Δεν είναι plug-and-play.
5. Υπάρχει ένας και μόνο ένας Master στον διάλογο, ενώ μπορεί να υπάρξουν ένας ή περισσότεροι Slaves.

Πρωτόκολλο I²C Bus:

Ένα ευρέως χρησιμοποιούμενο σειριακό interface για διασύνδεση ολοκληρωμένων κυκλωμάτων είναι το I²C της Philips. Ο διάυλος I2C επινοήθηκε από τη Philips στις αρχές της δεκαετίας του 80. Ο διάυλος αυτός υποστηρίζει σειριακή επικοινωνία μεταξύ ολοκληρωμένων κυκλωμάτων κάθε τεχνολογίας (CMOS, NMOS, Bipolar). Στηρίζεται στην αρχιτεκτονική των δύο συρμάτων, με τη βοήθεια των οποίων τα επιμέρους ολοκληρωμένα κυκλώματα μιας συσκευής ανταλλάσσουν σειριακά δεδομένα και σήματα συγχρονισμού μεταξύ τους. Οι δύο γραμμές του είναι η SCL και η SDA και είναι ανυψωμένες σε υψηλή τάση με αντιστάσεις ανύψωσης σε τάση (Pull Up Resistors), έχοντας τάση συνήθως από 3.3 V έως 5, αλλά αυτό ποικίλει από διάταξη σε διάταξη. Κάθε συσκευή που συνδέεται σε αυτές τις γραμμές, χαρακτηρίζεται από μια μοναδική διεύθυνση και μπορεί να λειτουργήσει ως πομπός ή ως δέκτης δεδομένων. Το εύρος διευθύνσεων του διαύλου είναι 7 bit, με 16 διευθύνσεις δεσμευμένες, έτσι ο συνολικός αριθμός ολοκληρωμένων(IC) που μπορούν να συνδεθούν στον διάυλο είναι 112. Η πιο συνηθισμένη ταχύτητα που λαμβάνει χώρα σε έναν I2C διάυλο είναι αυτή των 100 kbit/sec και ονομάζεται standard mode. Υπάρχει ωστόσο και το low speed mode όπου οι ταχύτητες σε αυτό δεν ξεπερνούν τα 10 kbit/s. Οι συσκευές που συνδέονται στον I2C διακρίνονται σε Master και Slave. Ο Master είναι αυτός που έχει την πρωτοβουλία στο κύκλωμα και στέλνει τους παλμούς χρονισμού στις συσκευές που είναι εξαρτώμενες (Slave). Η αρχή αποστολής δεδομένων γίνεται με τη συνθήκη εκκίνησης (Start). Αντίστοιχα η παύση αποστολής δεδομένων γίνεται με τη συνθήκη τερματισμού (Stop). Τα δεδομένα μεταφέρονται στον διάυλο κατά bytes. Ανάμεσα στα bytes που αποστέλλονται από τον πομπό προς τον δέκτη, ο δέκτης παράγει παλμούς επιβεβαίωσης ACK (Acknowledge).



Χρησιμότητα της συσκευής:

Η συσκευή αυτή προορίζεται για τον έλεγχο της ποιότητας της εικόνας των τηλεοπτικών δεκτών & οθονών CRT ώστε να πάρουμε μια πλήρως γραμμική απεικόνιση, με σωστό χρωματικό περιεχόμενο. Πιο συγκεκριμένα, μπορούμε να ελέγξουμε και να ρυθμίσουμε την παραμόρφωση στην γεωμετρία της εικόνας, την γραμμικότητα, την καθαρότητα των χρωμάτων, την σύγκλιση των χρωμάτων.

Τα patterns που έχουν σχεδιαστεί:

Έχουν υλοποιηθεί 14 συνολικά patterns, με τα οποία ο τεχνικός μπορεί να ελέγξει την ποιότητα της εικόνας, την γραμμικότητα, τα χρώματα, και να πραγματοποιήσει όλες τις απαραίτητες ρυθμίσεις σε έναν τηλεοπτικό δέκτη. Στις παρακάτω φωτογραφίες φαίνονται τα patterns όπως εμφανίζονται σε μία συσκευή τηλεόρασης.

Σκακιέρα (ChessBoard):



Με αυτό το pattern μπορούμε να ελεγχουμε την παραμόρφωση στην γεωμετρία της εικόνας, καθώς και την απόκριση στις χαμηλές συχνότητες.

Ασπρόμαυρες λωρίδες (B&W Bars):



Επίσης για τον έλεγχο του συστήματος στις χαμηλές συχνότητες.

Διαβαθμίσεις γκριζου (ράμπα) 7 σημείων (GrayScale Ramp):



Ο σκοπός της διαβάθμισης του γκρι είναι να εξασφαλίσει την ασπρόμαυρη απεικόνιση με σκιές μόνο του γκρι χωρίς ίχνος χρώματος σε όλα τα επίπεδα λαμπρότητας. Τα σημεία χαμηλής λαμπρότητας φέρνουν τα σημεία αποκοπής των τριών πυροβόλων σε σύμπτωση επιβεβαιώνοντας έτσι ότι οι σκιασμένες περιοχές δεν έχουν κανένα ίχνος έγχρωμης απόχρωσης.

Έγχρωμες λωρίδες (Color Bars):



Έγχρωμο pattern για τον έλεγχο όλων των χρωμάτων.

Διαβαθμίσεις κόκκινου, μπλε, πράσινου (Color Ramp):



Ράστερ μπλε, πράσινο, κόκκινο, λευκό (Blue, Green, Red, White Raster):

Η καθαρότητα της εικόνας εξασφαλίζεται όταν η κάθε δέσμη της λυχνίας εικόνας προσκρούει στον αντίστοιχο φώσφορο σε ολόκληρη την επιφάνεια της οθόνης. Όταν η καθαρότητα των χρωμάτων είναι κανονική θα πρέπει το λευκό ράστερ να έχει λευκό ομοιόμορφο χρώμα σε ολόκληρη την επιφάνεια της οθόνης. Ωστόσο δεν είναι δυνατόν να ελέγξουμε σωστά την καθαρότητα των χρωμάτων μόνο από το λευκό. Θα πρέπει να έχουμε ομοιόμορφο και έντονο χρώμα και για τα τρία χρώματα.



Σταυρός (Cross):



Με αυτό το pattern μπορούμε να ελέγξουμε εάν η εικόνα είναι εστιασμένη στο κέντρο της οθόνης και να την ρυθμίσουμε.

Πλέγμα (Net):

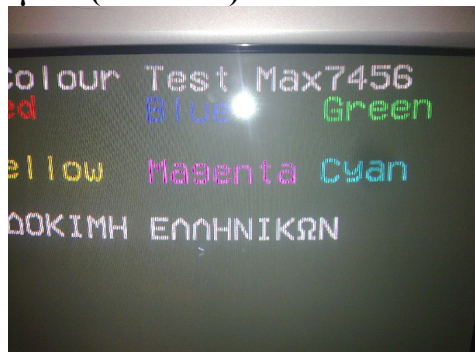


Τελείες (Dots) :

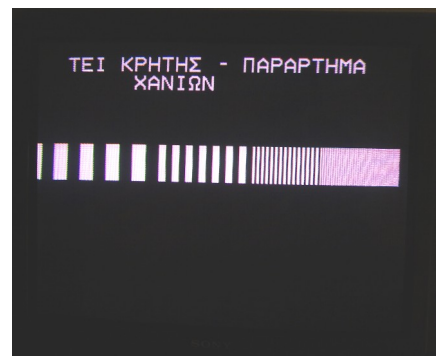


Με το πλέγμα και τις τελείες μπορούμε να ελέγξουμε την σύγκλιση των χρωμάτων. Τα δύο αυτά pattern είναι πολύ χρήσιμα για την ρύθμιση της σύγκλισης των τριών πυροβόλων, όταν δεν συγλίνουν στο ίδιο σημείο και τα τρία, αντί για μία λευκή τελεία βλέπουμε δύο ή τρεις έγχρωμες ανάλογα με το αν βρίσκονται και τα τρία εκτός σύγκλισης ή μόνο το ένα σε σχέση με τα άλλα δύο. Με το pattern του πλέγματος μπορούμε να τα ρυθμίσουμε ώστε να συγκλίνουν μεταξύ τους. Το pattern αυτό είναι χρήσιμο επίσης για τον έλεγχο της παραμόρφωσης στην γεωμετρία της εικόνας.

Κείμενο (Text Test):

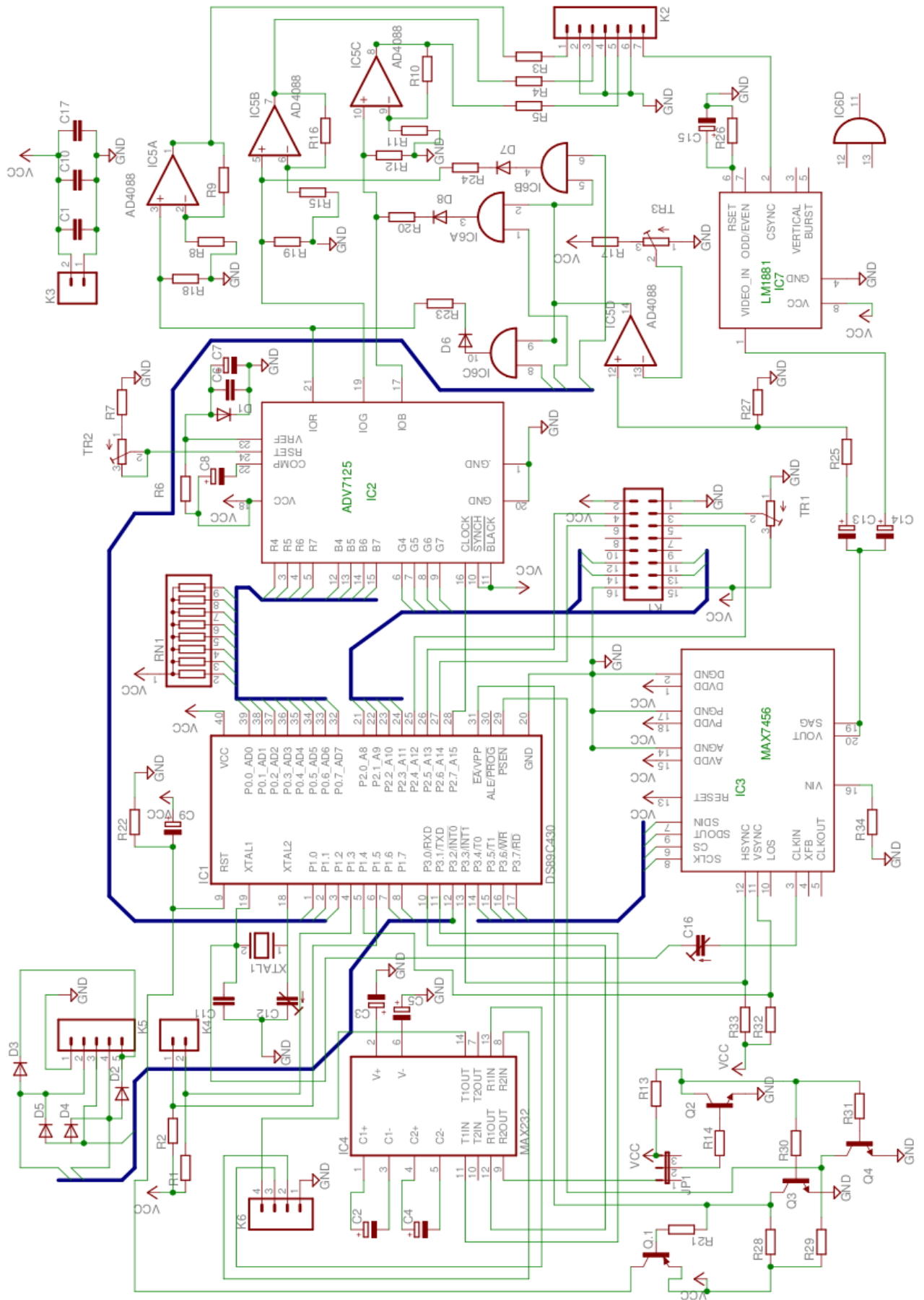


Απόκριση συχνότητας(Freq Test):

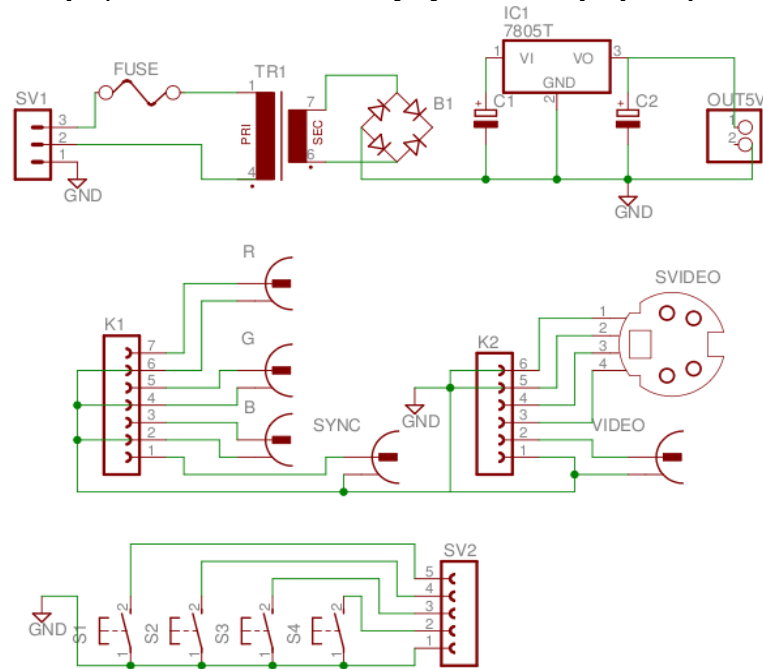


Με το pattern της απόκρισης συχνότητας μπορούμε να ελέγξουμε την απόκριση στις υψηλές συχνότητες.

ΚΕΦΑΛΑΙΟ 4: Περιγραφή Λειτουργίας & Σχηματικό διάγραμμα
Σχηματικό κεντρικής πλακέτας:



Σχηματικό τροφοδοτικού – πλακέτας εξόδων - ηλεκτρολόγιο:



Ο μικροελεγκτής DS89C430:

Ο DS89C430, είναι ένας μικροελεγκτής συμβατός με τον κλασικό INTEL 8051, είναι αρκετά πιο γρήγορος από τον κλασικό, αφού μπορεί και εκτελεί τις εντολές 12 φορές πιο γρήγορα από τον κλασικό 8051, και μπορεί να λειτουργήσει με συχνότητα ρολογιού έως και 33 Mhz. Έχει 16KB εσωτερικής μνήμης flash, με δυνατότητα προγραμματισμού μέσω της σειριακής θύρας από ένα ενσωματωμένο λειτουργικό φόρτωσης του προγράμματος στην μνήμη. Διαθέτει τέσσερις 8Bit I/O Ports. Έχει τρεις 16Bit Timers, 6 εξωτερικά interrupts, Δύο καταχωρητές DPTR (Dual DPTR). Τα τεχνικά χαρακτηριστικά του αναλυτικά και ο χάρτης μνήμης του μικροελεγκτή υπάρχουν στο Datasheet του κατασκευαστή που υπάρχει στις τελευταίες σελίδες.

Ο τριπλός μετατροπέας Digital to Analog ADV7125:

Ο ADV7125 είναι ένας CMOS τριπλός (Triple) 8-Bit High Speed Video DAC, και μπορεί να δουλέψει με ταχύτητες clock έως και 330 Mhz. Διαθέτει τρεις παράλληλες πόρτες εισόδου 8bit για την αποστολή των δεδομένων, μια είσοδο clock, μια είσοδο sync που χρησιμοποιείται για να κάνει εισαγωγή του σήματος συγχρονισμού στο σήμα green εάν θέλουμε, μια είσοδο black που στέλνει και τις τρεις εξόδους στο επίπεδο του μαύρου. Επίσης διαθέτει είσοδο Power Save Mode. Έχει 3 εξόδους ρεύματος, υψηλής εσωτερικής αντίστασης (high impedance, analog output current source) και τρεις συμπληρωματικές εξόδους (complementary outputs). Διατίθεται σε θήκη 48-lead LQFP. Περισσότερα τεχνικά χαρακτηριστικά υπάρχουν στο datasheet του κατασκευαστή στο παράρτημα στις τελευταίες σελίδες.

To ολοκληρωμένο OSD (On Screen Display) MAX7456:

To MAX 7456 είναι ένα On Screen Display ολοκληρωμένο, με εσωτερική EPROM (Single-Channel Monochrome On-Screen Display with Integrated EEPROM), εισάγει αλφαριθμητικούς χαρακτήρες σε ένα υπάρχον σήμα video, έχει δυνατότητα αποθήκευσης 256 χαρακτήρων στην εσωτερική EPROM, τους οποίους ο χρήστης μπορεί να τροποποιήσει ή να φτιάξει δικούς του, (256 User-Defined Characters or Pictographs in Integrated EEPROM). Οι χαρακτήρες είναι DotMatrix με μέγεθος κάθε χαρακτήρα 12 x 18 Pixel, Η οθόνη χωρίζεται σε 16 γραμμές και 30 στήλες (16 Rows x 30 Characters), για να λειτουργήσει στο σύστημα PAL χρειάζεται έναν κρύσταλλο 27 Mhz. Εκτός απο εισαγωγή κειμένου σε υπάρχον σήμα video, μπορεί να ρυθμιστεί σε internal sync mode, όπου δεν χρειάζεται είσοδο video για να λειτουργήσει, και παράγει τους παλμούς συγχρονισμού εσωτερικά, χρησιμοποιώντας την συχνότητα του κρυστάλλου 27Mhz. Επίσης εκτός απο έξοδο comosite video διαθέτει δυο εξόδους συγχρονισμού Vertical Sync και Horizontal Sync σε επίπεδο TTL. Η επικοινωνία με το ολοκληρωμένο γίνεται μέσω του σειριακού πρωτοκόλλου SPI.

LCD Display:

To LCD Display που έχει χρησιμοποιηθεί είναι ένα 4X20 αλφαριθμητικών χαρακτήρων Display, διαθέτει τον επεξεργαστή HITACHI HD44780 Controller, μπορεί να δουλέψει σε 8 ή 4 Bit Mode. Διαθέτει BackLight Led's μπλε χρώματος. Το Datasheet και το Set εντολών του HITACHI HD44780 Controller υπάρχει στο παράρτημα1.

Περιγραφή λειτουργίας:

To κύκλωμα δουλεύει ως εξής: Το ολοκληρωμένο MAX7456 (IC3) έχει την δυνατότητα να παράγει έξοδο video χωρίς σήμα εισόδου (video in), αυτή την λειτουργία την ονομάζει internal synch mode. Επίσης το ολοκληρωμένο αυτό δίνει έξοδο Vertical & Horizontal Synch. Την λειτουργία αυτήν εκμεταλλεύτηκα, και χρησιμοποίησα το MAX7456, εκτός από γεννήτρια χαρακτήρων, και σαν γεννήτρια συγχρονισμού. Τα δύο αυτά σήματα συγχρονισμού τα οδηγώ σε δύο interrupt του μικροελεγκτή, τον οριζόντιο στο INT_0 (P3.0), και τον κάθετο στο INT_2 (P1.4) . Για να δουλέψει το σύστημα σωστά χρειάστηκε τα δύο ολοκληρωμένα να έχουν ίδιο clock, και γιαυτό έχω χρησιμοποιήσει κοινό κρύσταλλο. Σε κάθε οριζόντιο παλμό συγχρονισμού ο μικροελεγκτής αυξάνει τον dptr_0, ο οποίος χρησιμοποιείται για να μετράει σε ποια γραμμή βρίσκεται η σάρωση. Έπειτα καλείται η

αντίστοιχη ρουτίνα που παράγει την έξοδο video για το συγκεκριμένο pattern και την συγκεκριμένη γραμμή. Σε κάθε παλμό κάθετου συγχρονισμού ο DPTR_0 μηδενίζεται.

Στην αρχική μου προσέγγιση ο μικροελεγκτής έφτιαχνε τους παλμούς συγχρονισμού και την έξοδο video μέσω του D/A, και οι πρώτες μου δοκιμές έγιναν με αυτόν τον τρόπο. Με τον τρόπο που τελικά υλοποιήθηκε, απλουστεύθηκε ο κώδικας του μικροελεγκτή. Η σταθερότητα της εξόδου εξαρτάται από το πόσο σταθερό είναι το MAX7456, και όχι από τον μικροελεγκτή.

Την έξοδο RGB την παράγει το ολοκληρωμένο ADV7125 (IC2), όπου είναι ένας τριπλός Analog to Digital Converter, σχεδιασμένος για εφαρμογές video, και μπορεί να δουλέψει σε ταχύτητες μέχρι και 330 Mhz. Αν και είναι 8-bit, έχουν χρησιμοποιηθεί μόνο τα 4 πιο σημαντικά bit, τα οποία έχουν συνδεθεί ως εξής: RED : P0.0 – P0.3 , GREEN: P0.4 – P0.7 BLUE: P2.4 – P2.7 , παράλληλα με το BLUE έχει συνδεθεί και η είσοδος DATA του LCD Display για οικονομία διότι διαφορετικά δεν αρκούσαν οι πόρτες του μικροελεγκτή. Το Clock του ADV7125 έχει συνδεθεί στο P2.0. Τα τρία σήματα ελέγχου του LCD έχουν συνδεθεί στα P2.1 – P2.3. Το ADV7125 έχει δύο επιπλέον εισόδους Blank & Synch η πρώτη οδηγεί και τα τρία σήματα στο επίπεδο του μαύρου και η δεύτερη κάνει εισαγωγή του σήματος συγχρονισμού στην έξοδο GREEN, τις εισόδους αυτές δεν τις έχω χρησιμοποιήσει. Για να λειτουργήσει το ADV7125 χρειάζεται μια τάση αναφοράς, και μια αντίσταση Rset που ρυθμίζει το ρεύμα εξόδου. Την τάση αναφοράς την φτιάχνω με το δικτύωμα D1, R6, C6, C7 η D1 είναι μια απλή δίοδος πυριτίου 1N 4148 πολωμένη ορθά, οπότε η τάση αναφοράς είναι 0.6 V. Σε σειρά με την Rset (R7) έχω βάλει ένα trimmer (TR2) ώστε να μπορώ να ρυθμίσω την τάση εξόδου στο 1 V p-p ακριβώς, η Rset ρυθμίζει το ρεύμα εξόδου, αλλά αφού οι αντιστάσεις εξόδου στις εξόδους είναι σταθερές ρυθμίζει έμμεσα την μέγιστη τάση, η έξοδος του ADV7125 είναι ρεύμα, όχι τάση, η έξοδοι οδηγούνται στους απομονωτές εξόδου (IC5 a, b, c) οι οποίοι παίζουν και τον ρόλο αθροιστή για τα σήματα από τον D/A και της γεννήτριας κειμένου .

Η composite video έξοδος του IC3 – MAX7456, οδηγείται στο ολοκληρωμένο LM1881 (IC7) και σε έναν συγκριτή, το IC7 είναι ένας video synch separator, και το χρησιμοποιώ για να εξάγω το σύνθετο σήμα συγχρονισμού. Ο συγκριτής (IC5-d) ανιχνεύει την στάθμη του λευκού και δίνει έξοδο 5V. Τον συγκριτή τον έχω υλοποιήσει με τον τέταρτο τελεστικό που υπάρχει στο IC5, στην αρνητική του είσοδο έχω φτιάξει έναν διεραϊτή τάσης με την αντίσταση R17 και το trimmer TR3, για να ρυθμίσω το κατώφλι του στο επίπεδο λευκού της εξόδου του MAX7456. Την έξοδο του συγκριτή την οδηγώ σε 3 πύλες AND

(IC6), οι οποίες χρησιμεύουν για να μπορώ να διαλέξω χρώμα στο κείμενο, μέσω του μικροελεγκτή, η δεύτερη είσοδος κάθε AND οδηγείται αντίστοιχα στα P1.0, P1.1, P1.2 του μικροελεγκτή. Οι έξοδοι των AND οδηγούνται μέσω μιας διόδου (D6, D7, D8) και μιας αντίστασης (R20, R23, R24) σε σειρά στους απομονωτές εξόδου (output buffer) (IC5 a-b-c), η διάδος χρειάστηκε γιατί χωρίς αυτήν, όταν η έξοδος της AND ήταν 0V δημιουργούσε βύθιση στο σήμα εξόδου, η αντίσταση μαζί με την αντίσταση εισόδου του απομονωτή (R12, R18, R19), δημιουργεί έναν διαιρέτη τάσης ώστε να είναι και το video από την γεννήτρια κείμενου

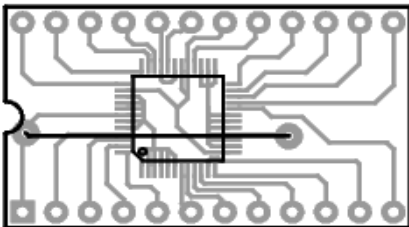
1 Vp-p.

Το ολοκληρωμένο MAX 232 μετατρέπει τις RS232 στάθμες σε στάθμες TTL, τα τρανζίστορ Q1, Q2, Q3, Q4 αποτελούν το κύκλωμα για τον προγραμματισμό του μικροεπεξεργαστή, μέσω της συριακής θύρας. Με το jumper1 επιλέγουμε κανονική λειτουργία της σειριακής, ή λειτουργία προγραμματισμού. Στον connector K1 συνδέεται το LCD, το οποίο έχει συνδεθεί σε 4 bit mode, με το TR1 ρυθμίζεται το contrast του LCD. Ο διάυλος I2C Bus υλοποιείται από τα pins P1.3 (SDA) και P1.5 (SCL) και συνδέεται στον connector K4, οι αντιστάσεις R1 R2 είναι pull-up resistors. Στον κονέκτορ K5 συνδέεται το πληκτρολόγιο, στον K3 η τάση τροφοδοσίας 5V, στον K2 η έξοδος, και στον K6 η συριακή θύρα.

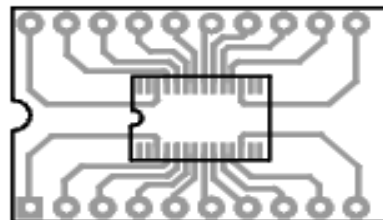
ΚΕΦΑΛΑΙΟ 5: PCB Layout & Κατασκευή

Για την τελική κατασκευή φτιάχτηκαν συνολικά 5 PCB. Η κεντρική πλακέτα, δύο μικρά που λειτουργούν σαν Adaptors για τα ολοκληρωμένα, MAX7456, και ADV7125, μια πλακέτα για τα βύσματα της εξόδου, και μία για το τροφοδοτικό. Τα ολοκληρωμένα MAX7456 και ADV7125 μπορούσα να τα βρω μόνο σε smd θήκες, και έτσι έφτιαξα 2 adaptors που τα μετατρέπουν σε DIP για να μπορέσω να τα βάλω πάνω σε ένα breadboard, και να μελετήσω την λειτουργία τους πριν υλοποιήσω την τελική κατασκευή και πλακέτα. Η τελική πλακέτα είναι μια αρκετά προσεγμένη σχεδίαση, αφού υπάρχουν σήματα video, που δέν θέλουμε να έχουν θόρυβο ή παραμόρφωση. Έχει σχεδιαστεί έτσι ώστε οι αγωγοί που μεταφέρουν σήματα video να είναι μικρού μήκους, και μακριά από αγωγούς που μεταφέρουν ψηφιακά σήματα. Επίσης έχουν τοποθετηθεί κεραμικοί πυκνωτές στην τροφοδοσία των ολοκληρωμένων, πολύ κοντά στα ολοκληρωμένα, για μην υπάρχει θόρυβος απο την τροφοδοσία. Η πλακέτα που σχεδιάστηκε είναι διπλής όψης, και στα σημεία που δεν υπάρχουν αγωγοί είναι καλυμμένη με γείωση και απο τις δύο πλευρές. Όλα τα PCB Layouts έχουν σχεδιαστεί με το PCB Designer της gEDA, το οποίο είναι ένα open source λογισμικό γραμμένο για Linux.

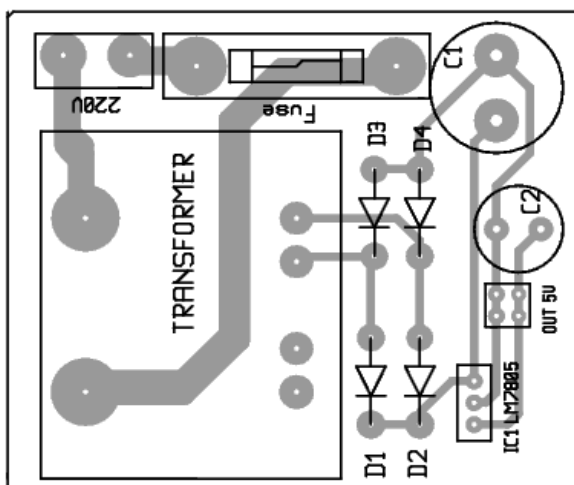
PCB Layouts:



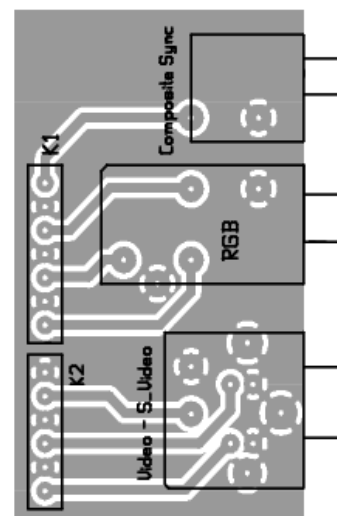
ADV7125 Adaptor.



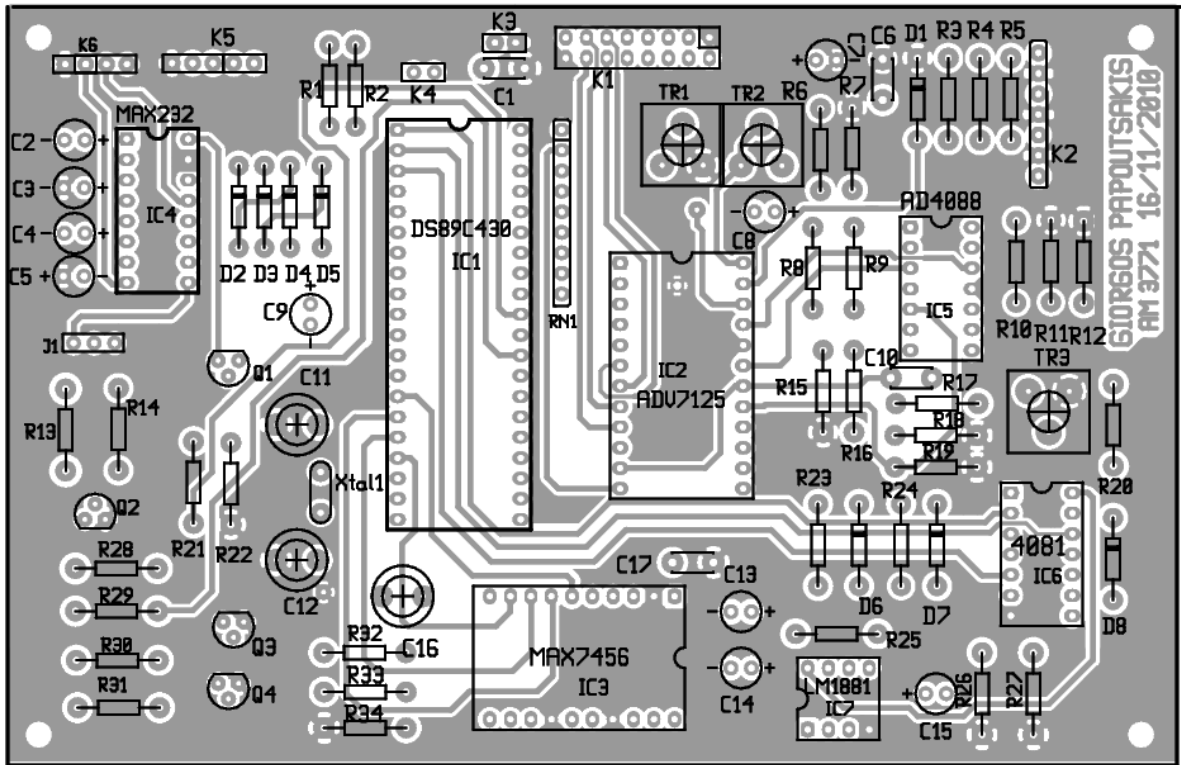
MAX 7456 Adaptor.



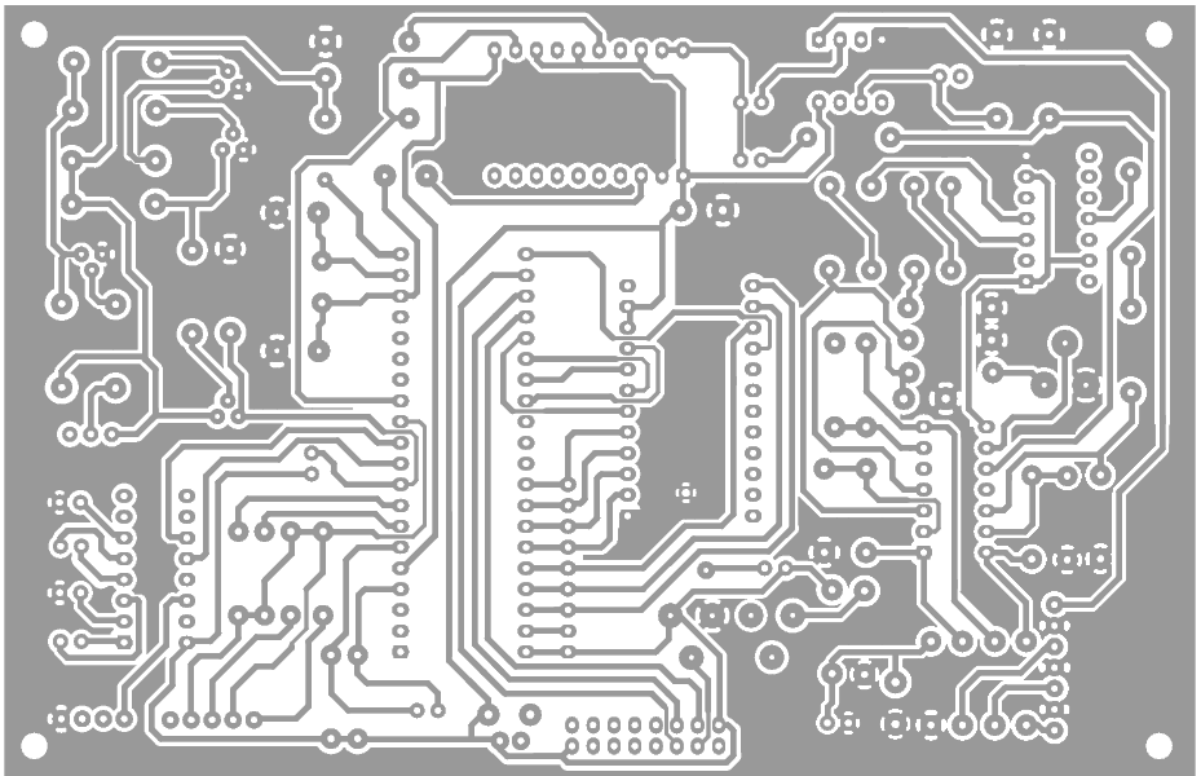
Τροφοδοτικό



Πλακέτα Εξόδων

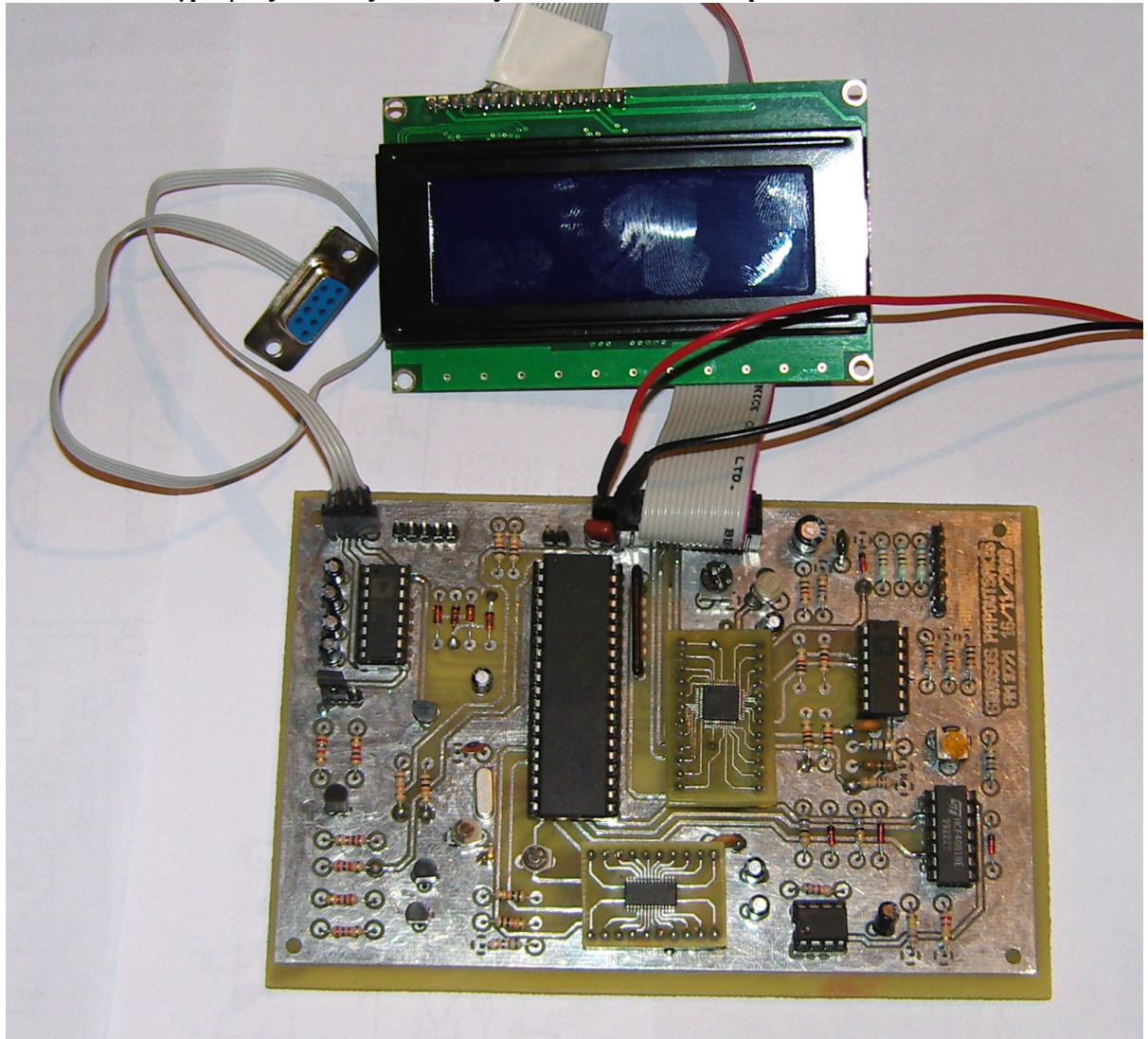


Κεντρική πλακέτα – Πάνω όψη (Πλευρά εξαρτημάτων).

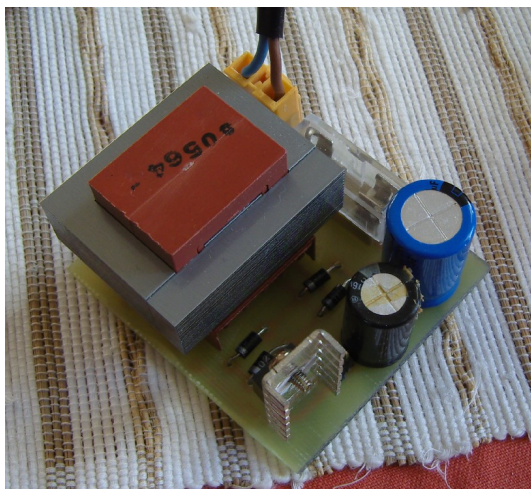


Κεντρική πλακέτα – Κάτω όψη.

Φωτογραφίες απο τις πλακέτες που κατασκευάστηκαν:



Κεντρική πλακέτα



Πλακέτα τροφοδοτικού

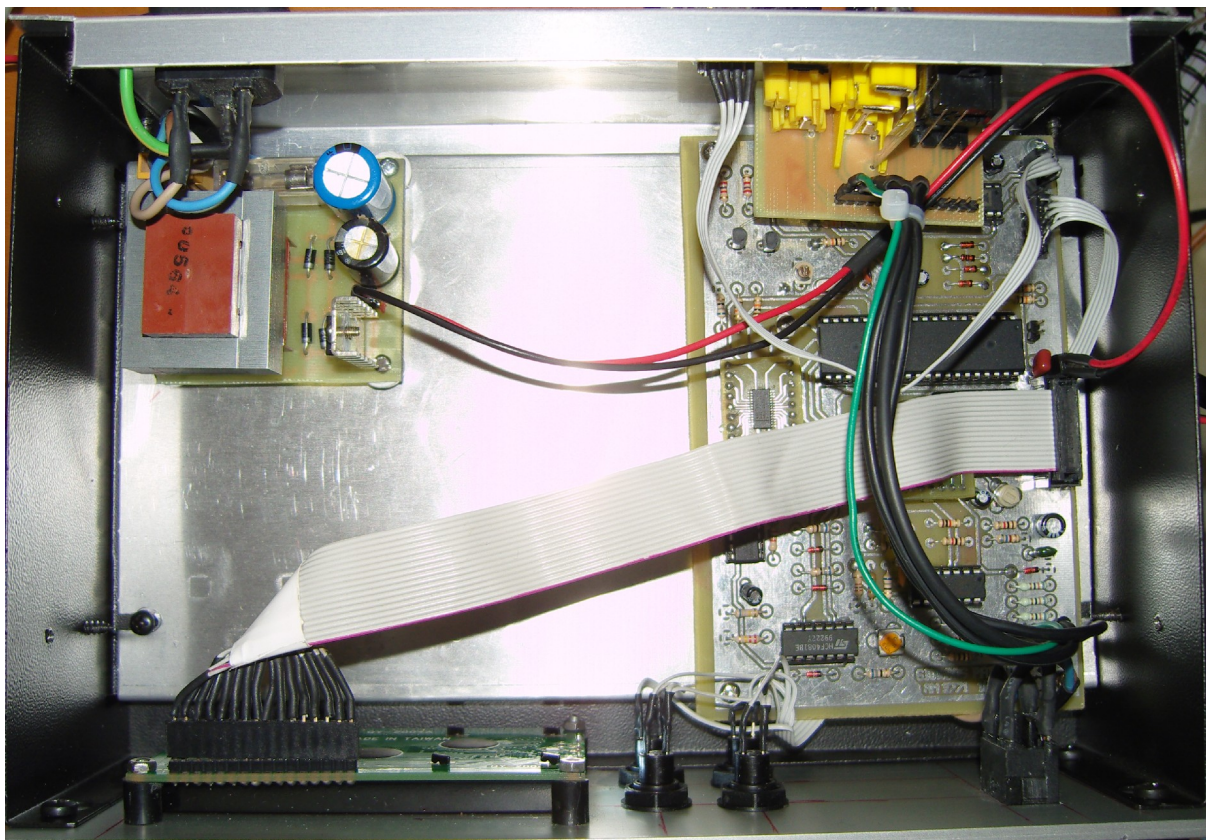


Πλακέτα εξόδων

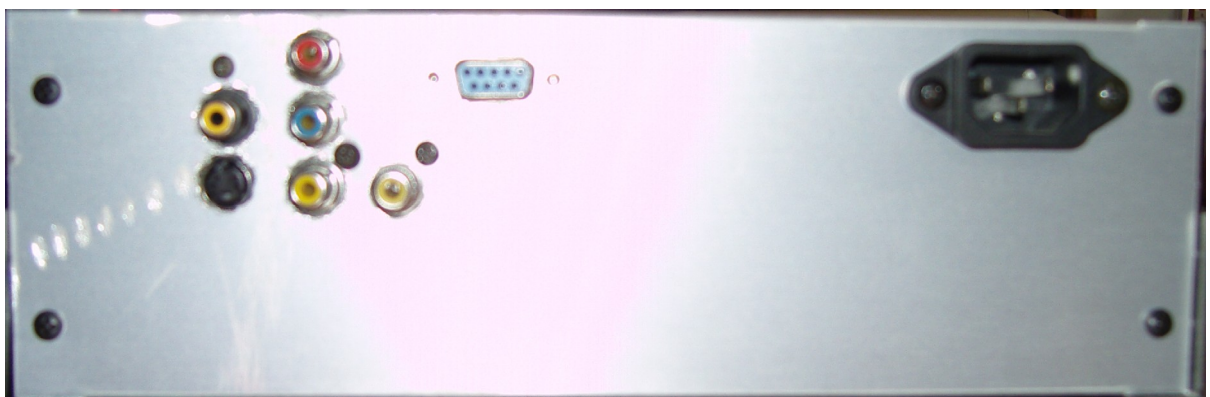
Φωτογραφίες από την κατασκευή στο κουτί της:



Πρόσοψη



Εσωτερικό



Πίσω όψη

Κατάλογος Υλικών:

Κεντρική πλακέτα:

R1: 10KΩ D1: 1N4148
R2: 10KΩ D2: 1N4148
R3: 75Ω D3: 1N4148
R4: 75Ω D4: 1N4148
R5: 75Ω D5: 1N4148
R6: 1KΩ D6: 1N41151
R7: 560Ω D7: 1N41151
R8: 1KΩ D8: 1N41151
R9: 1KΩ
R10: 1KΩ IC1: DS89C430
R11: 1KΩ IC2: ADV7125
R12: 100Ω IC3: MAX7456
R13: 2.7KΩ IC4: MAX232
R14: 2.7KΩ IC5: AD8044
R15: 1KΩ IC6: 4081 Quad AND
R16: 1KΩ IC7: LM1881
R17: 68Ω
R18: 100Ω RN1: 10KΩ Network
R19: 100Ω
R20: 68Ω TR1: 10KΩ Trimmer
R21: 19KΩ TR2: 1KΩ Trimmer
R22: 10KΩ TR3: 47KΩ Trimmer
R23: 68Ω
R24: 68Ω J1: JUMPER
R25: 1.9KΩ
R26: 680KΩ K1: 16 PIN HEADER (LCD)
R27: 220KΩ K2: 6 PIN (RGS OUT)
R28: 2.7KΩ K3: 2 PIN (5V IN)
R29: 2.7KΩ K4: 2 PIN (I²C BUS)
R30: 2.7KΩ K5: 5 PIN (KEYBOARD)
R31: 2.7KΩ K6: 4 PIN (RS_232)
R32: 10KΩ
R33: 10KΩ XTAL1: 27Mhz Crystal
R34: 68Ω

C1: 100nF
C2: 1μF 50V
C3: 1μF 50V
C4: 1μF 50V
C5: 1μF 50V
C6: 10nF
C7: 0.47μF 16V
C8: 0.1μF 16V
C9: 10μF 16V
C10: 10nF
C11: 39pF
C12: trimmer
C13: 0.1μF 16V
C14: 0.1μF 16V
C15: 0.1μF 16V
C16: trimmer
C17: 10nF

Τροφοδοτικό:

Transformer : 12 VAC - 1A

Fuse: 100mA fast fuse

D1: 1N4001

D2: 1N4001

D3: 1N4001

D4: 1N4001

C1: 4700μF 16V

C2: 3300μF 16V

IC1: LM7805

Πλακέτα εξόδων:

Video – Svideo: RCA – Svideo Socket

RGB: 3 RCA Socket

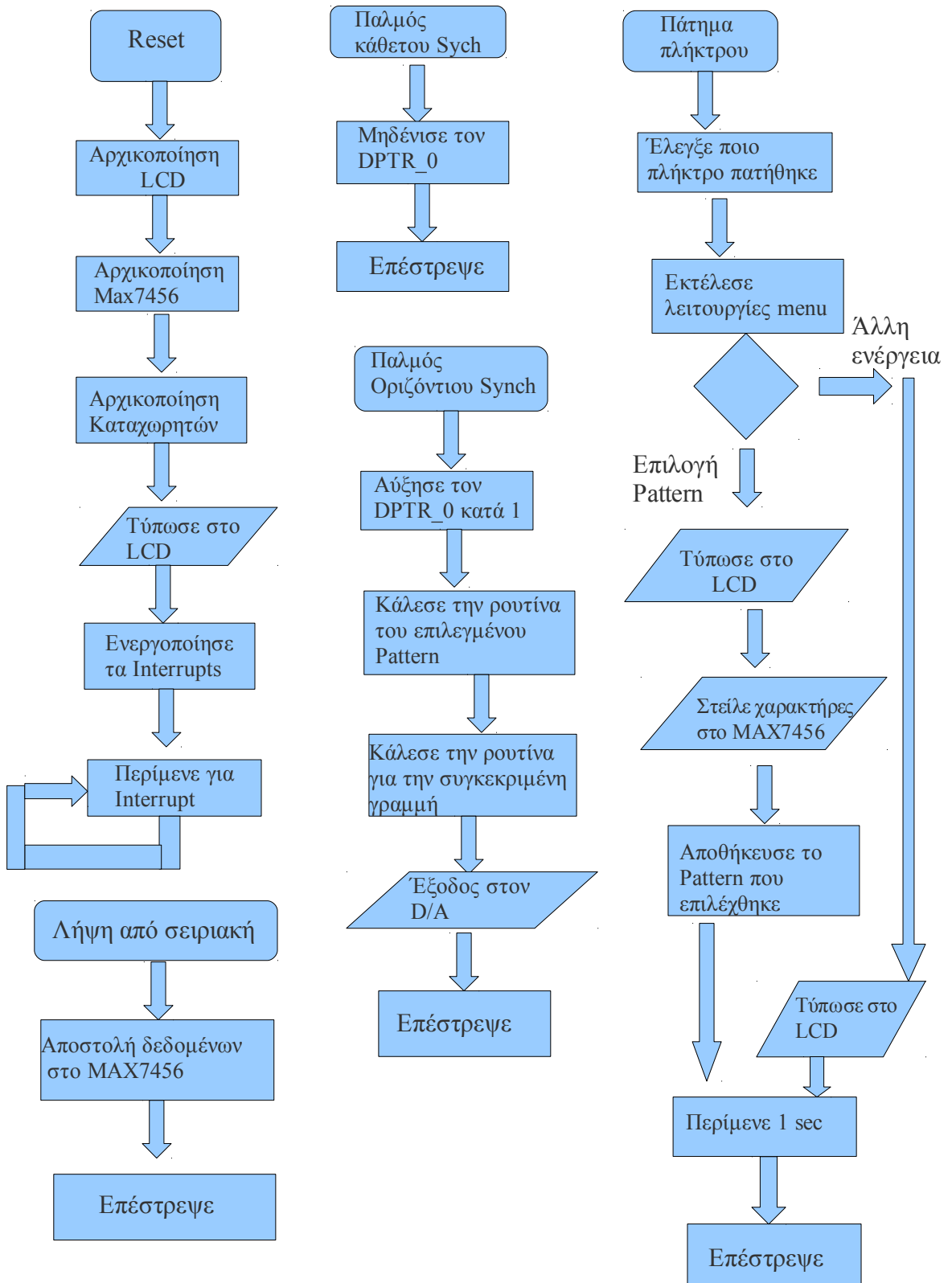
Synch: 1 RCA Socket

K1: 7 Pin Header (RGB-Sync In)

K2: 6 Pin Header (Video – Svideo In)

ΚΕΦΑΛΑΙΟ 6: Ανάλυση του κώδικα του μικροελεγκτή

Διάγραμμα Ροής:



Κώδικας του μικροελεγκτή σε Assembly:

Ο κώδικας του μικροελεγκτή έχει γραφτεί με την βοήθεια του λογισμικού Keil μVision 4, σε γλώσσα Assembly.

```
org 0000h
ljmp main

;***** external interupt INT0 (keyboard)*****
org 0003h
clr ex0
ljmp keyboard

;***** external interrupt INT1 (vertical) *****

org 13h
jnb p3.3,$
ljmp externint1
;timer 0
org 0bh
clr tr0
sjmp timer00

;***** timer 1 *****

org 1bh
reti
;***** serial port 0 interrupt *****
org 23h
acall serial
reti
;***** timer 1 interrupt *****

;***** external interrupt 2 (horizontal) *****
org 0043h
```

```
ljmp externint0

org 0080h
;/** timer 0 routine **/
timer00:
    djnz r5,timemore
    setb ex0
    reti
timemore:
    setb tr0
    reti
;/** keyboard routine **/
keyboard:
    clr ea
    setb p1.6
    setb p1.7
    jb p1.6,T2
    jb p1.7,T3
        sjmp C00
T3:        sjmp C01
T2:    jb p1.7, T4
        sjmp C10
T4:        sjmp C11

c00:
;//menu button
setb 21h.0
mov A,#0c0h
lcall next_line
mov 86h,#0000001b
MOV DPTR,#clear
lcall lcd_string
mov A,#0c0h
lcall next_line
mov 86h,#0000001b
MOV DPTR,#MESSAGE9
lcall lcd_string
mov A,#094h
lcall next_line
```

```
    acall lcdpatt

    lcall lcd_string

    sjmp t6
c01:  jnb 21h.0,t6
    mov r7,23h
    cjne r7,#1,deccc
    mov 23h,#15
    sjmp t7

deccc: dec r7
    mov 23h,r7
t7:
    sjmp t8

c10:  ;ok button
    jnb 21h.0,t6
    clr 21h.0
    mov A,#0c0h
    lcall next_line
    mov 86h,#00000001b
    MOV DPTR,#clear
    lcall lcd_string
    mov A,#0c0h
    lcall next_line

    mov 86h,#00000001b
    MOV DPTR,#MESSAGE2
    lcall lcd_string
    acall lcdpatt
    mov A,#0c9h
    lcall next_line
    lcall lcd_string
    mov A,#94h
    lcall next_line
    mov 86h,#00000001b
    MOV DPTR,#clear
    lcall lcd_string
```

```
        sjmp t6

c11:   jnb 21h.0,t6
        mov r7,23h
        cjne r7,#15,inccc
        mov 23h,#1
        sjmp t8

inccc: inc r7
        mov 23h,r7

t8:    mov A,#94h
        lcall next_line

        mov 86h,#00000001b
        MOV DPTR,#clear
        lcall lcd_string
        mov A,#94h
        lcall next_line

        acall lcdpatt
        mov A,#94h
        lcall next_line
        lcall lcd_string

t6:

t5:    mov r5,#10
        setb tr0
        setb ea
        reti

; ***** pattern selection *****
lcdpatt:
        lcall maxclean
        mov 86h,#0
        mov r7,23h
        clr tr1
        cjne r7,#1,lcd2
        mov 86h,#00000001b
        MOV DPTR,#patt1
        ljmp lcdexit

lcd2:  cjne r7,#2,lcd3
```

```
    mov 86h,#00000001b
    MOV DPTR,#patt2
    ljmp lcdexit
lcd3:  cjne r7,#3,lcd4
    mov 86h,#00000001b
    MOV DPTR,#patt3
    lcall lcd_string
    mov 86h,#0h
    mov dptr,#22H
    setb p1.0
    setb p1.1
    setb p1.2
    mov 86h,#00000001b
    mov dptr,#INTRO
    acall max7456_print
    mov dptr,#45H
    mov 86h,#00000001b
    mov dptr,#INTRO2
    acall max7456_print
    mov dptr,#0B5H
    mov 86h,#00000001b
    mov dptr,#INTRO3
    acall max7456_print
    mov dptr,#0D3H
    mov 86h,#00000001b
    mov dptr,#INTRO3
    acall max7456_print
    ljmp lcdexit
lcd4:  cjne r7,#4,lcd5
    mov 86h,#00000001b
    MOV DPTR,#patt4
    ljmp lcdexit
lcd5:  cjne r7,#5,lcd6
    mov 86h,#00000001b
    MOV DPTR,#patt5
    lcall lcd_string
    mov 86h,#0h
    mov dptr,#35
    mov 86h,#00000001b
    mov dptr,#mesg1
    acall max7456_print
```



```
    mov 86h,#0h
    mov dptr,#64
    mov 86h,#00000001b
    mov dptr,#mesg3
    acall max7456_print
    mov 86h,#0h
    mov dptr,#124
    mov 86h,#00000001b
    mov dptr,#msg4
    acall max7456_print
    mov 86h,#0h
    mov dptr,#185
    mov 86h,#00000001b
    mov dptr,#msg5
    acall max7456_print
    ljmp lcdexit
lcd6:  cjne r7,#6,lcd7
        mov 86h,#00000001b
        MOV DPTR,#patt6
        ljmp lcdexit
lcd7:  cjne r7,#7,lcd8
        mov 86h,#00000001b
        MOV DPTR,#patt7
        mov 86h,#0
        ljmp lcdexit
lcd8:  cjne r7,#8,lcd9
        mov 86h,#00000001b
        MOV DPTR,#patt8
        mov 86h,#0
        sjmp lcdexit
lcd9:  cjne r7,#9,lcd10
        mov 86h,#00000001b
        MOV DPTR,#patt9
        mov 86h,#0
        sjmp lcdexit
lcd10: cjne r7,#10,lcd11
        mov 86h,#00000001b
        MOV DPTR,#patt10
        mov 86h,#0
        sjmp lcdexit
lcd11: cjne r7,#11,lcd12
        mov 86h,#00000001b
```

```
MOV DPTR,#patt11
mov 86h,#0
sjmp lcdexit
lcd12:  cjne r7,#12, lcd13
        mov 86h,#00000001b
        MOV DPTR,#patt12
        mov 86h,#0
        sjmp lcdexit
lcd13:cjne r7,#13, lcd14
        mov 86h,#00000001b
        MOV DPTR,#patt13
        mov 86h,#0
        sjmp lcdexit
lcd14: cjne r7,#14, lcd15
        mov 86h,#00000001b
        MOV DPTR,#patt14
        mov 86h,#0
        sjmp lcdexit
lcd15:  cjne r7,#15,lcdexit
        mov 86h,#00000001b
        MOV DPTR,#patt15
        lcall lcd_string
        mov 86h,#0h
        setb tr1
        setb p1.0
        setb p1.1
        setb p1.2
        mov 20h,#0
        mov 40h,#01111000b
        lcall spiwritereg1
lcdexit:
ret
;***** MAX7456 initialize *****

max7456_initialize:
mov 40h,#0
mov 20h,#01111000b
lcall spiWriteReg1
        mov 40h,#6ch
mov 20h,#00001111b
lcall spiWriteReg1
//mov 40h,#01
```

```
//mov 20h,#1000111b
//lcall spiWriteReg1
//mov 40h,#2
//mov 20h,#20h
//lcall spiWriteReg1
//mov 40h,#3
//mov 20h,#10h
//lcall spiWriteReg1
//mov 40h,#4
//mov 20h,#11000100b
//lcall spiWriteReg1
ret
;/** max7456 printing message routine**/
```

Max7456_print:

```
;/setb 86h.0
```

NC0:

```
mov 40h,#5
mov 20h,dph
lcall spiwritereg1
mov 40h,#6
mov 20h,dpl
lcall spiwritereg1
mov 86h,#00000001b
mov A,#00
MOVC A,@A+DPTR
CJNE A,#2FH,NC11
mov 86h,#00
mov 40h,#0
mov 20h,#01111000b
lcall spiWriteReg1
RET
```

NC11:

```
mov 40h,#7
mov 20h,A
lcall spiwritereg1
inc dptr
mov 86h,#0
inc dptr
AJMP NC0
ret
```

```
;/***** serial port routine *****/
serial:
    mov r7,24h
    mov a,sbuf
    cjne r7,#1,sec
    mov 40h,A
    mov 24h,#0
    mov scon,#50h
    ret

sec:
    mov 20h,A
    mov 24h,#1
    lcall spiwritereg1
    mov 24h,#1
    mov scon,#50h
    ret
;/***** MAIN routine *****/
main:
    mov 23h,#3
klj:   mov R0,#255
jkl:   lcall delay1
    djnz R0,jkl
    acall max7456_initialize
    ICALL LCD_INIT ;Initialize lcd
    mov 86h,#1
    MOV DPTR,#MESSAGE1
    CALL LCD_STRING ;Display message on LCD
    mov A,#0c0h
    CALL NEXT_LINE ;Place cursor to;second Line
    mov 86h,#1
    MOV DPTR,#MESSAGE2
    CALL LCD_STRING ;Display message on LCD
    mov A,#0c9h
    lcall next_line
    mov 86h,#00000001b
    acall lcdpatt
    lcall lcd_string
    mov 86h,#0
    setb p3.2
    setb p3.3
    setb IT1
    setb EX0
```

```
setb EX1
setb 0bah
clr 0b8h
MOV SCON,#50h
setb ie.4
mov TH1,#0c5h;          /* TH1: reload value for 1200 baud */
mov tmod,#21h
mov TH0,#255
mov TL0,#255
setb 0E8h.0
mov 0f1h, #00000100b
mov 0f8h,#00000101b
setb et0
setb et1
setb EA
mov 24h,#1
sjmp $
;***** INT_1 routine (Horizontal Pulse) *****
externint1:

inc dptr
mov r7,23h

cjne r7,#1,vid2
    acall VideoCheckMate
    sjmp videxit
vid2:  cjne r7,#2,vid3
        acall VideoBwBars
        sjmp videxit
vid3:  cjne r7,#3,vid4
        //ret
        sjmp videxit
vid4:  cjne r7,#4,vid5
        acall videoBW7step
        sjmp videxit
vid5:  cjne r7,#5,vid6
        acall videotext
        sjmp videxit
vid6:  cjne r7,#6,vid7
        acall videocolbars
        sjmp videxit
vid7:  cjne r7,#7,vid8
```

```
        lcall videocolorramp
        sjmp videxit
vid8:   cjne r7,#8,vid9
        acall videoredrast
        sjmp videxit
vid9:   cjne r7,#9,vid10
        acall videobluerast
        sjmp videxit
vid10:  cjne r7,#10,vid11
        acall videogreenrast
        sjmp videxit
vid11:  cjne r7,#11,vid12
        acall videowhiterast
        sjmp videxit
vid12:  cjne r7,#12,vid13
        acall videoNet
        sjmp videxit
vid13:  cjne r7,#13,vid14
        acall videoCross
        sjmp videxit
vid14:  cjne r7,#14,videxit
        acall videodots

videxit:
        reti
;***** INT_0 (Vertical Pulse) *****
externint0:
        mov 20h,91h
        clr 20h.4
        mov 91h,20h
        mov dptr,#0

        nop
        nop
        nop
        nop
        nop
        nop
        reti
;***** Routine for checkmate pattern *****
videocheckmate:

        acall startcheck
```

```
mov r7,dph
cjne r7,#1,check2
    mov A,dpl
    clr c
    subb A,#7
jc checkA2
    ;//first bar
    mov r7,#7
    djnz r7,$
acall queen
    ret
checkA2:
    mov r7,#8
    djnz r7,$
    nop
    acall king
    ret
    ;// sec bar
    ret
check2: mov A,dpl
    clr c
    Subb A,#211
    jc check3
    mov r7,#8
    nop
    djnz r7,$
    acall king
    ret
    ;// sec bar
    ret
check3: mov A,dpl
    clr c
    subb A,#159
    jc check4
    ;// third
    mov r7,#4
    nop
    djnz r7,$
    acall queen
    ret
check4: mov A,dpl
```

```
        clr c
        subb A,#107
        jc check5
        ;// fourth
                mov r7,#3
                nop
                nop
                djnz r7,$
        acall king
        ret
check5: mov A,dpl
        clr c
        subb A,#55
        jc check6
        ;// fifth
        acall queen
        ret
check6: ;//sixth
                mov r7,#1
                nop
                nop
                djnz r7,$
                acall king

        ret
king:   acall delaycheck
                mov p0,#0ffh
                mov p2,#0f0h
                setb p2.0
                acall delaycheck
                mov p0,#0h
                mov p2,#0
                setb p2.0
                acall delaycheck
                mov p0,#0ffh
                mov p2,#0f0h
                setb p2.0
                acall delaycheck
                mov p0,#0h
                mov p2,#0
                setb p2.0
                acall delaycheck
```



```
        mov p0,#0ffh
        mov p2,#0f0h
        setb p2.0
        acall delaycheck
        mov p0,#0h
        mov p2,#0
        setb p2.0
        ret
queen:  mov p0,#11111111b
        mov p2,#11110000b
        setb p2.0
        acall delaycheck

        mov p0,#0
        mov p2,#0
        setb p2.0
        acall delaycheck

        mov p0,#11111111b
        mov p2,#11110000b
        setb p2.0
        acall delaycheck

        mov p0,#0
        mov p2,#0
        setb p2.0
        acall delaycheck
        mov p0,#11111111b
        mov p2,#11110000b
        setb p2.0
        acall delaycheck

        mov p0,#0
        mov p2,#0
        setb p2.0
        acall delaycheck

        mov p0,#11111111b
        mov p2,#11110000b
        setb p2.0
        acall delaycheck
```

```

        mov p0,#0
        mov p2,#0
        setb p2.0

        ret

delaycheck:
        mov r0,#42
        djnz R0,$
        ret

startcheck:
        mov r0,#40
        djnz R0,$
        ret
; ***** routine for B&W Bars pattern *****

videobwbars:
        acall startcheck
        nop
        nop
        nop
        nop
        acall queen
        ret
; Routine for cross pattern *****

videocross:
        mov r7,dpl
        cjne r7,#163,cross
        nop
        nop
        acall linehorizontal
        ret

cross:
        mov r7,#195
        djnz r7,$
        mov p2,#0f0h
        mov p0,#0ffh

        setb p2.0
        mov p2,#0
        mov p0,#0
        setb p2.0
        ret
; ***** routine for Grayscale pattern *****

videoBW7step:
```

```
nop
acall startcheck
mov p2,#0
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#00100000b
mov p0,#00100010b
setb p2.0
acall delaycheck
mov p2,#01000000b
mov p0,#01000100b
setb p2.0
acall delaycheck
mov p2,#01100000b
mov p0,#01100110b
setb p2.0
acall delaycheck
mov p2,#10000000b
mov p0,#10001000b
setb p2.0
acall delaycheck
mov p2,#10100000b
mov p0,#10101010b
setb p2.0
acall delaycheck
mov p2,#0f0h
mov p0,#0ffh
setb p2.0
acall delaycheck
mov p2,#00h
mov p0,#0h
setb p2.0
ret
```

```
;***** routine for colored bars pattern *****
```

videocolbars:

```
acall startcheck
mov p2,#11110000b
mov p0,#11111111b
setb p2.0
acall delaycheck
```

```
mov p2,#11110000b
mov p0,#00001111b
setb p2.0
acall delaycheck
mov p2,#11110000b
mov p0,#11110000b
setb p2.0
acall delaycheck
mov p2,#11110000b
mov p0,#00000000b
setb p2.0
acall delaycheck
mov p2,#00000000b
mov p0,#11111111b
setb p2.0
acall delaycheck
mov p2,#00000000b
mov p0,#00001111b
setb p2.0
acall delaycheck
mov p2,#00000000b
mov p0,#11110000b
setb p2.0
acall delaycheck
mov p2,#00h
mov p0,#0h
setb p2.0
ret
;routine for text test pattern *****
```

videotext:

```
mov A,dpl
clr c
Subb A,#120
jc text1
setb p1.0
setb p1.1
setb p1.2
ret
```

```
text1:  mov A,dpl
        clr c
        Subb A,#80
```

```

        jc text2
        setb p1.0
        setb p1.2
        clr p1.1
        lcall delayp
        lcall delayp
        lcall delayp
        clr p1.2
        setb p1.1
        setb p1.0
        lcall delayp
        lcall delayp
        ;acall delayp
        clr p1.0
        setb p1.2
        setb p1.1
        ret
text2:   mov A,dpl
        clr c
        Subb A,#55
        jc text3
        setb p1.0
        clr p1.1
        clr p1.2
        lcall delayp
        lcall delayp
        clr p1.0
        setb p1.1
        lcall delayp
        lcall delayp
        lcall delayp
        clr p1.1
        setb p1.2
        ret
text3:   setb p1.0
        setb p1.1
                setb p1.2
ret
;***** routine for Net pattern *****
videonet:
        mov r7,DPH
        cjne r7,#1,net0
```

```
        mov r7,DPL
        cjne r7,#15,net11
        acall linehorizontal
        ret

net0:   mov r7,DPL
        cjne r7,#70,net1
        acall linehorizontal
        ret

net1:   cjne r7,#120,net2
        acall linehorizontal
        ret

net2:   cjne r7,#170,net3
        acall linehorizontal
        ret

net3:   cjne r7,#220,net10
        acall linehorizontal
        ret

net11:  mov r7,#2
        nop
        nop
        djnz r7,$

net10:  acall dots
        ret
        ; routine for dots pattern *****

videodots:
mov r7,DPH
        cjne r7,#1,dots0
        mov r7,DPL
        cjne r7,#15,dots11
        acall dots
        ret

dots0:  mov r7,DPL
        cjne r7,#70,dots1
        acall dots
        ret

dots1:  cjne r7,#120,dots2
        acall dots
        ret

dots2:  cjne r7,#170,dots3
        acall dots
```

```
        ret
dots3:  cjne r7,#220,dots10
        acall dots
        ret
dots11: mov r7,#2
        nop
        nop
        djnz r7,$

dots10:
ret
        ;*****routine for 1 white line *****
linehorizontal:
        mov r7,#34
        nop
        nop
        djnz r7,$
mov p0,#11111111b
mov p2,#0f0h
setb p2.0
mov r7,#7
wa:acall delaycheck
djmpz r7,wa
mov r7,#10
djmpz r7,$
mov p0,#0
mov p2,#0
setb p2.0
ret
        ;***** routine for 1 dots line *****
dots:
        mov r7,#69

        djnz r7,$
        mov r7,#6
dot:mov p0,#0ffh
        mov p2,#0f0h
        setb p2.0
        mov p0,#0h
        mov p2,#0h
        setb p2.0
        acall delaycheck
        djnz r7,dot
```

```
ret
;***** routine for blue raster pattern *****
videobluerast:

acall startcheck
mov p0,#11110000b
mov p2,#0
setb p2.0
mov r7,#7
sa:acall delaycheck
djnz r7,sa
mov r7,#10
djnz r7,$
mov p0,#0
mov p2,#0
setb p2.0
ret
;*****8 routine for white raster pattern *****
videoredrast:

acall startcheck
mov p0,#00001111b
mov p2,#0
setb p2.0
mov r7,#10
djnz r7,$
mov r7,#7
ra:acall delaycheck
djnz r7,ra
mov p0,#0
mov p2,#0
setb p2.0
ret
;routine for green raster pattern *****
videogreenrast:
acall startcheck

mov p0,#00000000b
mov p2,#11110000b
setb p2.0
mov r7,#7
ga:acall delaycheck
```



```
djnz r7,ga
mov r7,#10
djnz r7,$
mov p0,#0
mov p2,#0
setb p2.0
ret
; ***** routine for white raster pattern *****
videowhiterast:
```

```
acall linehorizontal
ret
```

```
; ***** one colorscale line *****
greenstep:
```

```
mov r7,#40
djnz r7,$
mov p2,#00h
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#00100000b
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#01000000b
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#01100000b
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#10000000b
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#10100000b
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#0f0h
```

```
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#00h
mov p0,#0
setb p2.0
ret
    bluestep:
mov r7,#35
djnz r7,$
mov p2,#00h
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#00100000b
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#01000000b
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#01100000b
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#10000000b
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#10100000b
setb p2.0
acall delaycheck
mov p2,#00h
mov p0,#0f0h
setb p2.0
acall delaycheck
mov p2,#00h
mov p0,#0
setb p2.0
ret
```

```
redstep:
mov r7,#31
djnz r7,$
mov p2,#00h
mov p0,#0
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#00000010b
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#00000100b
setb p2.0
acall delaycheck
mov p2,#0
mov p0,#00000110b
setb p2.0
lcall delaycheck
mov p2,#0
mov p0,#00001000b
setb p2.0
lcall delaycheck
mov p2,#0
mov p0,#00001010b
setb p2.0
lcall delaycheck
mov p2,#00h
mov p0,#0fh
setb p2.0
lcall delaycheck
mov p2,#00h
mov p0,#0
setb p2.0
ret
;routine for color ramps pattern *****8
```

videocolorramp:

```
mov r7,dph
cjne r7,#1,colorramp2
    mov A,dpl
    clr c
```

```
        subb A,#7
jc      colorramp1

        ret

colorramp1:

        lcall greenstep
        ret

colorramp2:

        mov A,dpl
        clr c
        subb A,#212
jc      colorramp3

        lcall greenstep
        ret

colorramp3:

        mov A,dpl
        clr c
        subb A,#185
jc      colorramp4
        ret

colorramp4:

        mov A,dpl
        clr c
        subb A,#135
jc      colorramp5
        lcall bluestep
        ret

colorramp5:

        mov A,dpl
        clr c
        subb A,#105
jc      colorramp6
        ret

colorramp6:

        mov A,dpl
        clr c
        subb A,#55
jc      colorramp7
        lcall redstep

colorramp7:        ret
```

```
//***** SPI ROUTINES *****
```

```
spiReadReg1:
```

```
    setb p2.0
    clr p2.7
    clr p2.0
    mov A,40h
    lcall write
    lcall read
    setb p2.0
    ret
```

```
read:
```

```
    mov r0,#8
```

```
ll:
```

```
    setb p2.7
    mov c , p2.6
    mov A,40h
    rlc A
    mov 40h,A
    clr p2.7
    djnz r0,ll
    ret
```

```
; ***** clean the max7456 routine *****
```

```
MAXclean:
```

```
    mov 40h,#0
    mov 20h,#01110000b
    lcall spiWriteReg1
    mov 40h,#4
    mov 20h,#01111111b
    acall spiwritereg1
    ret
```

```
spiWriteReg1:
```

```
    setb p3.4
    clr p3.6
    clr p3.4
    mov A,40h
    acall write
    setb p3.5
    mov A,20h
    acall write
    setb p3.5
```

```
        setb p3.4
        ret
write:
        mov r0,#8
loopa:  rlc A
        jc one
        clr p3.5
        sjmp go
one:    setb p3.5
        ;//acall delayp
go:     setb p3.6
        ;//acall delayp
        clr p3.6
        ;//acall delayp
        djnz r0,loopa
        ret

// ***** LCD ROUTINES *****
LCD_INIT:
        clr p2.1
        ANL P2,#0FH
        CALL LOOP
        MOV DPTR,#LCDCODE
        MOV A,#0H
        MOV R6,#0H
        MOV R7,#0H
        CLR P2.3
NEXT:
        INC R6
        MOVC A,@A+DPTR
        MOV R7,A
        ANL A,#0F0H

        ANL P2,#0FH
        ORL P2,A
        ICALL ENABLE
        MOV A,R7
        ANL A,#0FH
        swap a
        ANL P2,#0FH
        ORL P2,A
        ICALL ENABLE
        MOV A,R6
```

```
CJNE R6,#09H,NEXT
RET
LCD_STRING:
MOV P2,#00H
SETB P2.3
MOV A,#00H
MOV R6,#00H

NC:
mov 86h,#1
MOVC A,@A+DPTR
CJNE A,#2FH,NC1
RET

NC1:
LCALL LCD_WRITE
INC R6
MOV A,R6
AJMP NC

LCD_WRITE:
SETB P2.3
CALL LO
RET

NEXT_LINE:
MOV P2,#00H
CLR P2.3
CALL LO
RET

LO:
MOV R7,A
ANL A,#0F0H
ANL P2,#0FH
ORL P2,A
CALL ENABLE
MOV A,R7
ANL A,#0FH
swap a
ANL P2,#0FH
ORL P2,A
CALL ENABLE
RET

ENABLE:
SETB P2.2
```

```
CALL DELAYL
CLR P2.2
CALL DELAYL
RET
DELAYL:
SETB PSW.4
MOV R7,#80
HDH:
MOV R6,#250
DJNZ R6,$
DJNZ R7,HDH
CLR PSW.4
RET
LOOP:
MOV R7,#250
LOOP1:
CALL DELAYL
CALL DELAYL
DJNZ R7,LOOP1
RET
delayh:
mov r0,#200
djnz R0,$
ret
delayp:
mov r0,#50

djnz R0,$
ret
;***** DATA *****
;LCD INITIALIZING CODE

LCDCODE:
DB 02H
DB 02H
DB 02H
DB 28H
DB 28H
DB 28H
DB 0CH
DB 06H
DB 01H
```



```
clear: db "          /"
MESSAGE1: DB "TV SIGNAL GENERATOR/" ;Change Message1
MESSAGE9: db "-PATTERN SELECTION-/"
patt1: db "ChessBoard/"
patt2: db "B&W Bars/"
patt3: db "Freq. Test/"
patt4: db "Gray Ramp 7/"
patt5: db "Text Test/"
patt6: db "Color Bars/"
patt7: db "Color Ramp/"
patt8: db "Red Raster/"
patt9: db "Blue Raster/"
patt10: db "GreenRaster/"
patt11: db "WhiteRaster/"
patt12: db "Net/"
patt13: db "Cross/"
patt14: db "Dots/"
patt15: db "PC MODE/"
MESSAGE2: DB "PATTERN:/" ;Change Message2
mesg1: DB "Colour Test Max7456/"
mesg3: DB "Red  Blue  Green/"
msg4:  DB "Yellow Magenta Cyan/"
msg5: DB 27h,"OKIMH E",29h,29h,"HNIK",3bh,"N/"
INTRO: DB "TEI KPHTH", 2CH, " - " ,2BH,"APAPTHMA/"
INTRO2: DB "XANI",3BH ,"N/"
INTRO3: DB 0FFH,0,0FFH,0,0FFH,0,0FFH,0,0FFH,0,1,1,1,1,1,1,1,3,3,3,3,2,2,2,2,2,2, "/"
msg2: DB 216, 215 ,214,213,212,211,210,209,208," "
end
```

ΚΕΦΑΛΑΙΟ 7: ανάλυση βοηθητικού προγράμματος για το PC σε γλώσσα Visual Basic

Πρόγραμμα για τον έλεγχο του OSD MAX7456:

Για να λειτουργήσει το πρόγραμμα στην Visual Basic πρέπει να επιλέξουμε απο το μενού της συσκευής PATTERN SELECTION -> PC MODE.

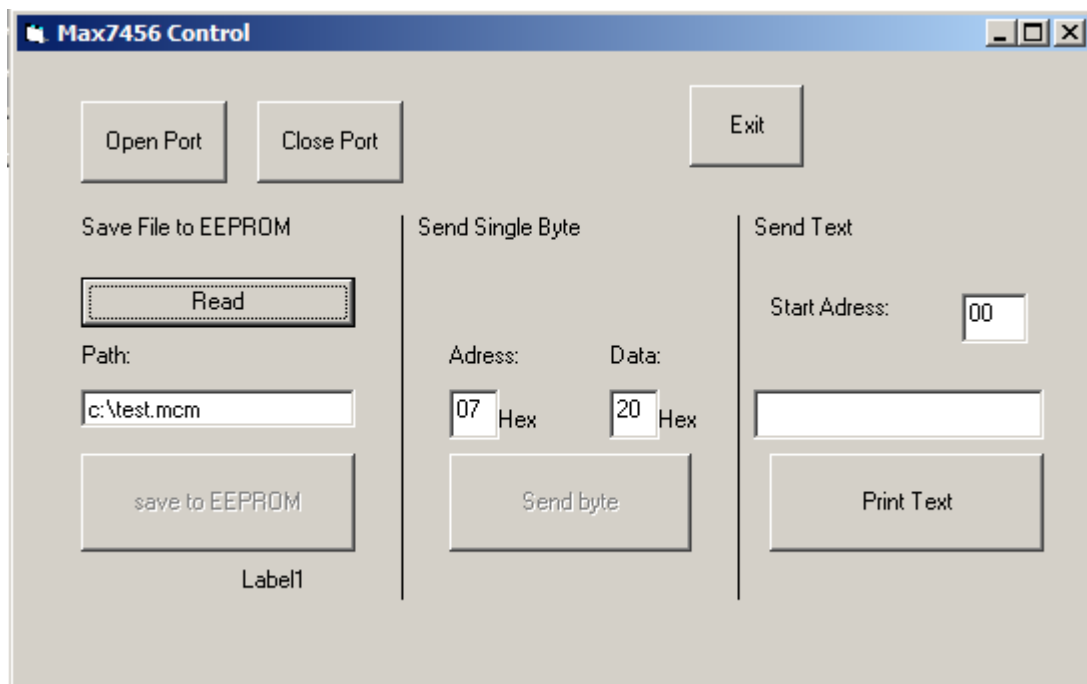
Από το πρόγραμμα αυτό μπορούμε να φορτώσουμε αρχείο χαρακτήρων στην EEPROM, να στείλουμε 1 Byte σε μία διεύθυνση ή να στείλουμε κείμενο για να τυπωθεί στην οθόνη.

Ο κατασκευαστής του ολοκληρωμένου Max7456, δίνει ένα software, επεξεργασίας και δημιουργίας χαρακτήρων. Το software αυτό συνεργάζεται με το Evaluation Kit Board, το οποίο είναι διαθέσιμο από τον κατασκευαστή. Μπορεί όμως να αποθηκεύσει σε αρχείο τα δεδομένα των χαρακτήρων. Το πρόγραμμα που έφτιαξα διαβάζει αυτό το αρχείο και μέσω της σειριακής, και την βοήθεια του μικροελεγκτή το αποθηκεύει στην EEPROM.

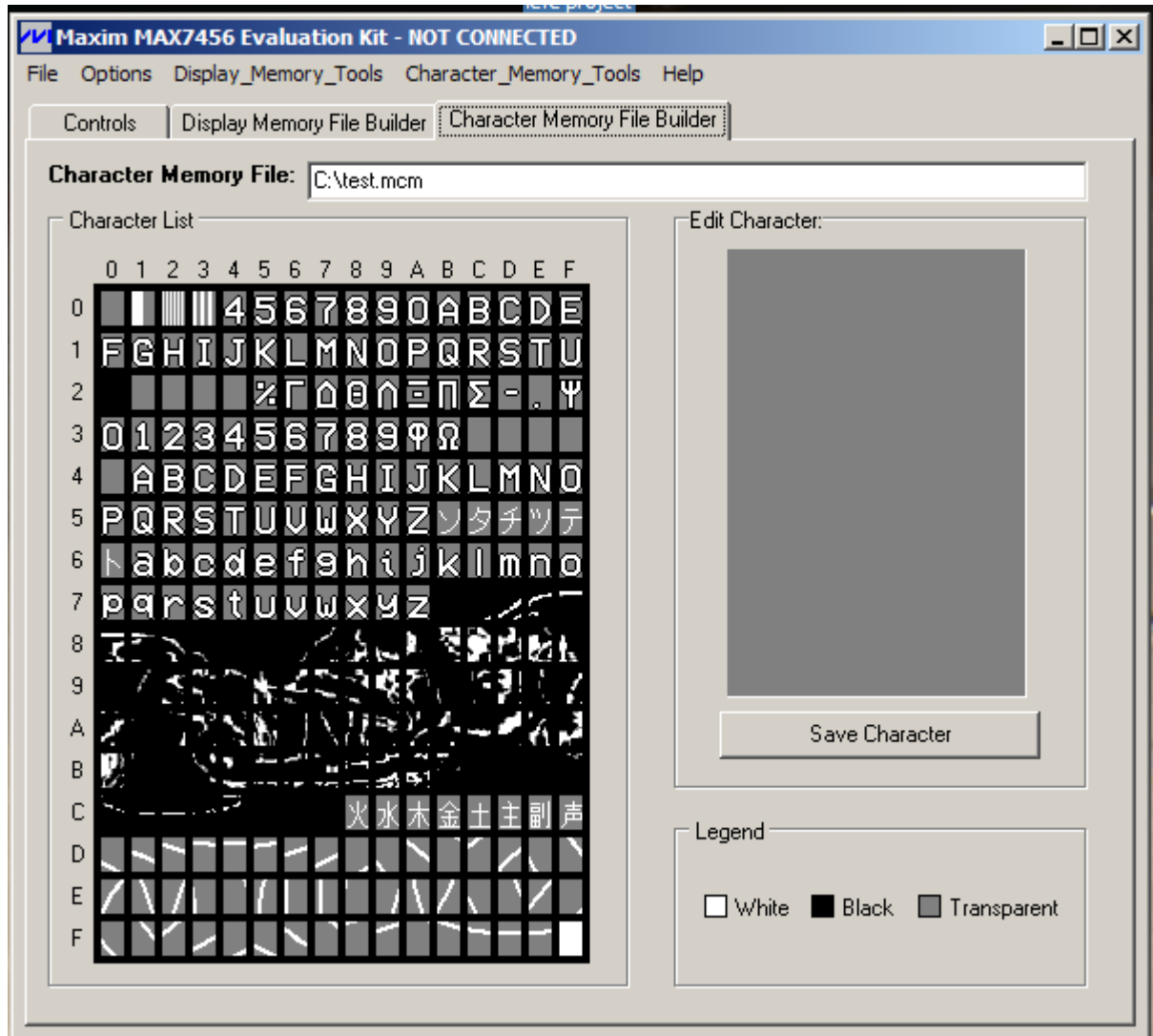
Το Baud Rate για την σειριακή επικοινωνία έχει ρυθμιστεί εσωτερικά στο πρόγραμμα καθώς και στον μικροελεγκτή στο 1200 .

Αρχικά πατάμε Open port για να ανοίξουμε την σειριακή.

Για να αποθηκεύσουμε αρχείο χαρακτήρων, γράφουμε την θέση (Path) που βρίσκεται το αρχείο, και πατάμε read. Ένα message box μας ειδοποιεί οτι το αρχείο ανοίχτηκε, οπότε μπορούμε να πατήσουμε Save to EEPROM. Όταν ολοκληρωθεί η διαδικασία ένα message box μας ειδοποιεί με ένα μήνυμα OK. Αν θέλουμε να στείλουμε ένα Byte γράφουμε την διεύθυνση και τα δεδομένα, σε δεκαεξαδικά μορφή και πατάμε send byte. Για αποστολή κειμένου γράφουμε την θέση μνήμης στην οθόνη που θα ξεκινά το κείμενο και πατάμε Print Text.



Στιγμιότυπο απο το Πρόγραμμα στην VisualBasic



Στιγμιότυπο απο το πρόγραμμα της Maxim.

Κώδικας του προγράμματος:

```
Private Sub Command1_Click()
MSComm1.PortOpen = True
Command2.Enabled = True
Command3.Enabled = True
End Sub
Private Sub Command2_Click()
MSComm1.Output = Chr(0)
MSComm1.Output = Chr(64)
Dim kstart, i, k, j, l As Integer
Dim MyArr
Dim ch As String
i = 2
ch = ""
For j = 0 To 255
MSComm1.Output = Chr(9)
MSComm1.Output = Chr(j)
For l = 0 To 63
MSComm1.Output = Chr(10)
MSComm1.Output = Chr(l)
```

```
MSComm1.Output = Chr(11)
MSComm1.Output = Chr(Datas(i))
i = i + 1
Next
MSComm1.Output = Chr(8)
MSComm1.Output = Chr(170)
label1.Caption = j
Wait 1
'MsgBox "ok char " & j
Next
MsgBox "OK"
End Sub
```

```
Private Sub Command3_Click()
Dim str, ch As String
Dim i, a, b, c As Integer
str = Text1.Text
a = 0
b = 0
c = 0
If Len(str) > 2 Then
MsgBox "not valid Hex 0 to 255"
Exit Sub
End If
If Len(Text2.Text) > 2 Then
MsgBox "not valid Hex 0 to 255"
Exit Sub
End If
If Len(str) = 1 Then
str = 0 & str
End If
For i = 1 To Len(str)
ch = Mid(str, i, 1)
If ch = "1" Then
a = 1
ElseIf ch = "2" Then
a = 2
ElseIf ch = "3" Then
a = 3
ElseIf ch = "4" Then
a = 4
ElseIf ch = "5" Then
a = 5
ElseIf ch = "6" Then
a = 6
ElseIf ch = "7" Then
a = 7
ElseIf ch = "8" Then
a = 8
ElseIf ch = "9" Then
a = 9
ElseIf ch = "A" Then
a = 10
ElseIf ch = "B" Then
a = 11
ElseIf ch = "C" Then
a = 12
ElseIf ch = "D" Then
a = 13
ElseIf ch = "E" Then
a = 14
```

```
ElseIf ch = "F" Then
a = 15
ElseIf ch = "0" Then
a = 0
Else
MsgBox "not valid characeter " & ch
Exit Sub
End If
If i = 1 Then
a = a * 16
End If
b = b + a
Next i
str = Text2.Text
If Len(str) = 1 Then
str = 0 & str
End If
For i = 1 To Len(str)
ch = Mid(str, i, 1)
If ch = "1" Then
a = 1
ElseIf ch = "2" Then
a = 2
ElseIf ch = "3" Then
a = 3
ElseIf ch = "4" Then
a = 4
ElseIf ch = "5" Then
a = 5
ElseIf ch = "6" Then
a = 6
ElseIf ch = "7" Then
a = 7
ElseIf ch = "8" Then
a = 8
ElseIf ch = "9" Then
a = 9
ElseIf ch = "A" Then
a = 10
ElseIf ch = "B" Then
a = 11
ElseIf ch = "C" Then
a = 12
ElseIf ch = "D" Then
a = 13
ElseIf ch = "E" Then
a = 14
ElseIf ch = "F" Then
a = 15
ElseIf ch = "0" Then
a = 0
Else
MsgBox "not valid characeter " & ch
Exit Sub
End If
If i = 1 Then
a = a * 16
End If
c = c + a
Next i
MSComm1.Output = Chr(b)
MSComm1.Output = Chr(c)
```

```
End Sub
Private Sub Command4_Click()
MSComm1.PortOpen = False
Command2.Enabled = False
Command3.Enabled = False
End Sub
```

```
Private Sub Command5_Click()
Dim nFileNum As Integer, sText As String, sNextLine As String, lLineCount, line As Long
Dim col, row, i, res As Integer
Dim ch As String
nFileNum = FreeFile
Open Text3.Text For Input As nFileNum
lLineCount = 2
For line = 1 To 16385
Line Input #nFileNum, sNextLine
Datas(line) = sNextLine
Next
Close nFileNum
col = 1
row = 7
res = 0
For
col = 2 To 16385
For i = 1 To 8
ch = Mid(Datas(col), i, 1)
If ch = "1" Then
res = res + (2 ^ row)
End If
row = row - 1
Next
Datas(col) = res
res = 0
row = 7
Next
MsgBox "OK"
End Sub
```

```
Private Sub Command6_Click()
End
End Sub
Private Sub Command7_Click()
Dim str, ch As String
Dim i, a, b As Integer
str = Text5.Text
a = 0
b = 0
If Len(str) > 2 Then
MsgBox "not valid Hex 0 to 255"
Exit Sub
End If
Form1 - 4
If Len(str) = 1 Then
str = 0 & str
End If
For i = 1 To Len(str)
ch = Mid(str, i, 1)
If ch = "1" Then
a = 1
ElseIf ch = "2" Then
```

```
a = 2
ElseIf ch = "3" Then
a = 3
ElseIf ch = "4" Then
a = 4
ElseIf ch = "5" Then
a = 5
ElseIf ch = "6" Then
a = 6
ElseIf ch = "7" Then
a = 7
ElseIf ch = "8" Then
a = 8
ElseIf ch = "9" Then
a = 9
ElseIf ch = "A" Then
a = 10
ElseIf ch = "B" Then
a = 11
ElseIf ch = "C" Then
a = 12
ElseIf ch = "D" Then
a = 13
ElseIf ch = "E" Then
a = 14
ElseIf ch = "F" Then
a = 15
ElseIf ch = "0" Then
a = 0
Else
MsgBox "not valid characeter " & ch
Exit Sub
End If
If i = 1 Then
a = a * 16
End If
b = b + a
Next i
str = Text4.Text
For i = 1 To Len(str)
ch = Mid(str, i, 1)
PrintScreen (b - 1 + i), ch
Next i
End Sub

Private Sub Command8_Click()
MsgBox Len(Text1.Text)
Dim i As Integer
For i = 0 To 16384
For j = 1 To 8
Next
Datas(i) = 3
label1.Caption = i
Next
End Sub
Private Sub Command9_Click()
MsgBox Datas(15690)
End Sub
Private Sub Read_Click()
End Sub
```

```
Public Function PrintScreen(ByVal pos As Integer, dat As String)
'MsgBox pos & dat
Dim i As Integer
i = 0
If pos > 255 Then
i = 1
pos = pos - 256
End If
Form1.MSComm1.Output = Chr(5)
Form1.MSComm1.Output = Chr(i)
Form1.MSComm1.Output = Chr(6)
Form1.MSComm1.Output = Chr(pos)
Form1.MSComm1.Output = Chr(7)
Form1.MSComm1.Output = dat
End Function
```


ΒΛΙΟΓΡΑΦΙΑ

BIBLIA

1. “Αναλογική-ψηφιακή τηλεόραση και βίντεο” - Παντελής Χ. Βαφειάδης
2. “Προγραμματίζοντας τον μικροελεγκτή 8051” - Myke Predko
3. “Έγχρωμη Τηλεόραση Θεωρία – Βλάβες – Επισκευή” - Δημήτριος Τζιόλας

DATASHEET

1. DS89C430 Datasheet – Dallas Semiconductors
2. DS89C430 User Guide – Dallas Semiconductors
3. MAX7456 Datasheet – Maxim IC
4. LM1881 Datasheet – National
5. ADV7125 Datasheet – Analog Devices
6. HITACHI HD44780U LCD Controller -Hitachi
7. AD8044 – Analog Devices

ONLINE

1. Wikipedia: Test card
http://en.wikipedia.org/wiki/Test_card
[προσβάσιμο στις 28 Νοεμβρίου 2010]
2. Wikipedia: Monoscope
<http://en.wikipedia.org/wiki/Monoscope>
[προσβάσιμο στις 28 Νοεμβρίου 2010]
3. Chalk Hill Media's Virtual Museum - RCA TK-1A Monoscope camera
<http://www.chalkhillmedia.org/Museum/RCA2.htm>
[προσβάσιμο στις 28 Νοεμβρίου 2010]
4. Wikipedia: Phillips PM5544
http://en.wikipedia.org/wiki/Philips_PM5544
[προσβάσιμο στις 28 Νοεμβρίου 2010]
5. PAL video timing specification - Martin Hinner's homepage
<http://martin.hinner.info/vga/pal.html>
[προσβάσιμο στις 28 Νοεμβρίου 2010]
6. Max7456 chars made easy
<http://www.mylifesucks.de/tools/max7456/>
[προσβάσιμο στις 28 Νοεμβρίου 2010]
7. SCART RGB interfacing – ePanorama.net
<http://www.epanorama.net/documents/vga2rgb/scart.html>
[προσβάσιμο στις 28 Νοεμβρίου 2010]
8. Television Test Cards, Tuning Signals, Idents and Clocks
<http://www.pembers.freemove.co.uk/Test-Cards/index.html>
[προσβάσιμο στις 28 Νοεμβρίου 2010]

ΠΑΡΑΡΤΗΜΑ 1: Datasheet's