- Κασάπογλου Ευάγγελος (ΑΜ 628)
- Οικονομίδης Ρουσλάν (ΑΜ 847)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

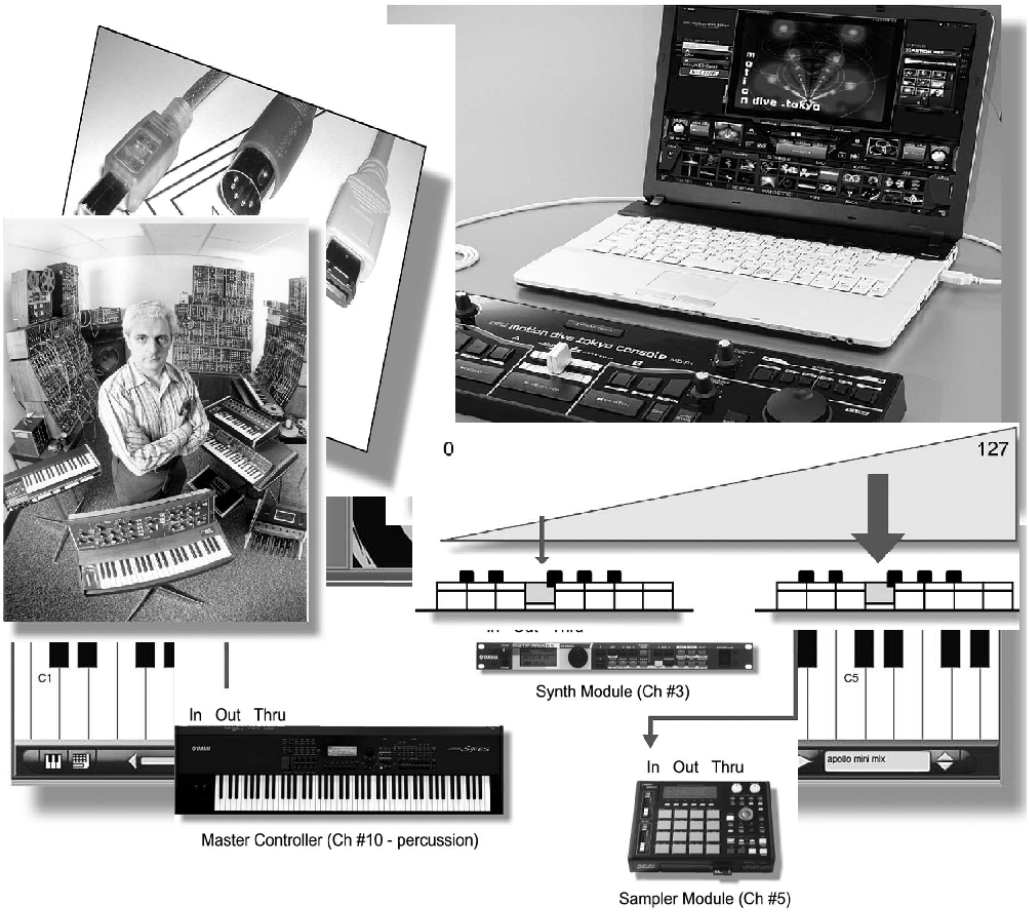
ΘΕΜΑ : ΚΑΤΑΣΚΕΥΗ ΣΥΣΚΕΥΗΣ ΕΛΕΓΧΟΥ ΠΡΩΤΟΚΟΛΟΥ MIDI (MIDI CONTROLLER) ΜΕ 16 ΠΕΡΙΣΤΡΕΦΟΜΕΝΑ ΡΥΘΜΙΣΤΙΚΑ (pots)

Στην εργασία αυτή προσπαθήσαμε να φτιάξουμε έναν midi controller, μια συσκευή δηλαδή που στέλνει midi μηνύματα και θα μας επιτρέπει να ελέγχουμε οποιαδήποτε άλλη συσκευή που υποστηρίζει το midi πρωτόκολλο. Ο λόγος που αποφασίσαμε να φτιάξουμε την συγκεκριμένη συσκευή είναι γιατί θέλαμε να φτιάξουμε κάτι που θα μας μείνει και μιας και έτυχε να χρειαζόμαστε έναν midi controller σκεφτήκαμε να φτιάξουμε έναν.

Την εργασία την χωρίσαμε σε δυο κομμάτια. Το κομμάτι της κατασκευής και το θεωρητικό κομμάτι , καθένα από τα οποία το ανέλαβε ο καθένας μας χωριστά. Αναλυτικά η όλη διαδικασία της κατασκευής ακολουθεί στις επόμενες σελίδες.

ΠΕΡΙΕΧΟΜΕΝΑ

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ



MIDI

Musical Instrument Digital Interface
(Ψηφιακή Διασύνδεση Μουσικών Οργάνων)

Το midi είναι ένα πρωτόκολλο που επιτρέπει μια μεγάλη γκάμα από ψηφιακά μουσικά όργανα, υπολογιστές και άλλες συσκευές να συνδέονται και να επικοινωνούν μεταξύ τους.

ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ

Μέχρι και το 1970 οι ηλεκτρονικές μουσικές συσκευές είχαν αρχίσει να εξαπλώνονται και είχαν γίνει αρκετά προσιτές στην Β. Αμερική, Ευρώπη και Ιαπωνία. Το 1979, κάποια καινούργια κλαβιέ από τις εταιρίες Oberheim, Rhodes και Roland κατασκευάσθηκαν με βύσματα διασύνδεσης {interface plugs} στο πίσω μέρος τους, μέσω του οποίου μπορούσαν να συνδεθούν μεταξύ τους. Αυτή η σύνδεση γινόταν μόνο μεταξύ οργάνων της ίδιας εταιρίας. Όταν λοιπόν παίζαμε στο ένα κλαβιέ, ακουγόταν, εκτός από αυτό και όλα τα άλλα κλαβιέ που είχαμε συνδέσει με αυτό. Αυτή η λύση έλυνε κατά ένα μέρος το πρόβλημα αλλά δεν έδινε απάντηση στο μεγάλο ερώτημα του πως να συνδεθούν μεταξύ τους διάφορα όργανα από διαφορετικές εταιρίες.

Ο μηχανικός ήχου και σχεδιαστής synthesizer Dave Smith πρόεδρος της **Sequential Circuits**, μιας γνωστής εταιρίας synthesizer αυτής της εποχής σκέφτηκε ένα πρωτόκολλο για την ηλεκτρονική μουσική όταν δημιουργούσε ένα νέου είδους synthesizer και αργότερα αυτό το πρωτόκολλο εξελίχθηκε στο MIDI. Ο Smith είχε την ιδέα ότι μια σύνδεση δεδομένων με την λογική εισόδων-εξόδων θα ήταν εφικτή μεταξύ των συσκευών αν υπήρχε ένα καθορισμένο πρωτόκολλο μεταξύ των διάφορων κατασκευαστών και θα έδινε έτσι το μέσο για διάφορα όργανα κι συσκευές να επικοινωνούν ελέγχοντας η μια την άλλη σε ψηφιακό επίπεδο.

Δύο φορές το χρόνο, η **NAMM** (National Association of Music Merchandisers) ή Εθνική Ένωση Μουσικών Εμπόρων, οργανώνει ένα διεθνές συνέδριο για την επίδειξη καινούργιων μουσικών προϊόντων και την εύρεση νέων μεθόδων για τη προώθηση των μουσικών οργάνων και των εξαρτημάτων. Στη διάρκεια ενός τέτοιου συνεδρίου το 1982, έγινε μια συνάντηση από μία μικρή ομάδα κατασκευαστών synthesizer με την καθοδήγηση του **Dave Smith**. Κατά τη συνάντηση αυτή συζητήθηκε μία πρόταση για την αποδοχή, από όλους τους κατασκευαστές synthesizer και εν

γένει ηλεκτρονικών μουσικών οργάνων, ενός παγκοσμίου πρωτόκολλου για την μετάδοση και λήψη πληροφοριών σχετικών με τη μουσική εκτέλεση, μεταξύ όλων των ειδών των ηλεκτρονικών μουσικών οργάνων. Η αρχική αυτή πρόταση ονομάσθηκε UMI (Universal Musical Interface) ή Παγκόσμια Μουσική Διασύνδεση.

Στη συνέχεια, αυτή η πρόταση πέρασε από ένα σημαντικό αριθμό αναθεωρήσεων πριν να γίνει γνωστή με το όρο **MIDI** ή **MIDI Standard** {Πρωτόκολλο MIDI}. Τελικά, το 1983 η **Sequential Circuits** από την Αμερική και η **Roland** από τη Ιαπωνία παρουσίασαν τα πρώτα κλαβιέ με MIDI και σύντομα ακολούθησαν ουσιαστικά όλες οι άλλες εταιρίες synthesizer. Μέχρι σήμερα το Πρωτόκολλο MIDI δεν έχει υποκατασταθεί από άλλο σύστημα διασύνδεσης ηλεκτρονικών μουσικών οργάνων, πάνω στο οποίο θα ήταν δυνατή η δόμηση ενός studio – κάτι σαν αυτό που σήμερα συνηθίζουμε να ονομάζουμε MIDI Home Studio. Και όπως και οι υπολογιστές, έτσι και το MIDI χρησιμοποιείται σήμερα από εκατομμύρια μουσικούς είτε αυτοί είναι επαγγελματίες είτε ερασιτέχνες για πολλές εφαρμογές που έχουν να κάνουν με τη σύνθεση, τον αυτοσχεδιασμό, την εκπαίδευση, την επεξεργασία παρτιτούρας κλπ. Ακόμα το Πρωτόκολλο MIDI θα το συναντήσουμε και σε χώρους εκτός της μουσικής, όπως ο φωτισμός θεάτρων, στις επικοινωνίες, τα στούντιο ηχοληψίας, κλπ.

Το MIDI έφερε την επανάσταση στην συνδεσιμότητα μεταξύ των συσκευών δίνοντας στους μουσικούς μια νέα γενιά από ευέλικτα ψηφιακά μουσικά όργανα που είχαν την δυνατότητα να ελέγχουν ταυτόχρονα πολλές διαφορετικές συσκευές.

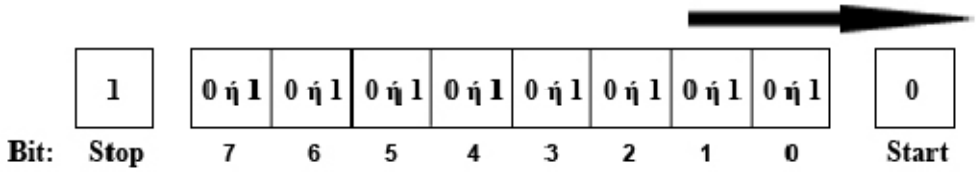
Το πρωτόκολλο MIDI και η τεχνολογία γύρω από αυτό διατηρείται και εξελίσσεται από την MMA (Midi Manufacturers Association).

ΠΡΩΤΟΚΟΛΛΟ MIDI

Το πρωτόκολλο MIDI είναι ένα σειριακό ασύγχρονο πρωτόκολλο με ροή δεδομένων 31.250 bits/sec (bps). Είναι ένα δικατευθυντικό πρωτόκολλο που σημαίνει ότι για κάθε κατεύθυνση χρησιμοποιεί και μια ανεξάρτητη ξεχωριστή γραμμή. Η ψηφιακή πληροφορία μεταδίδεται σε δυαδική μορφή κωδικοποιημένη σε λέξεις που αποτελούνται από τα ψηφία 0 και 1. Η κάθε λέξη (byte) αποτελείται από 8 ψηφία (bits). Επίσης κάθε byte περιλαμβάνει ένα start bit (λογικό 0) και ένα stop bit (λογικό 1) (ένα bit με τιμή 0 μεταδίδεται ουσιαστικά με ένα λογικό 1 και το αντίθετο).

Τα start και stop bit είναι απαραίτητα γιατί επιτρέπουν στην μονάδα υποδοχής να αναγνωρίζει την αρχή και το τέλος του byte καθώς αυτό έρχεται bit προς bit. Με άλλα λόγια αυτά τα bit είναι η αρχή και το τέλος του κάθε byte.

Όπως αναφέραμε και πιο πάνω τα δεδομένα μεταδίδονται σε *δυαδική μορφή* {**binary form**}, όπου κάθε bit είναι κωδικοποιημένο ως 0 και 1. Τα δεδομένα είναι ομαδοποιημένα σε **bytes** τα οποία είναι σύνολο των 8 **bits** (βλέπε εικόνα). Κάθε byte συνοδεύεται από ένα **start bit** {αρχικό – υπό τύπο προθέματος – bit} και ένα **stop bit** {τελικό bit}. Τα start και stop bit επιτρέπουν στη συσκευή αποδέκτης {slave} να αναγνωρίσει την αρχή και το τέλος ενός byte όπως αυτό φτάνει σε αυτήν, bit ανά bit. Αυτά τα MIDI bytes των 10 bit είναι ομαδοποιημένα σε «πακέτα» και συνθέτουν τα MIDI μηνύματα {**MIDI messages**}. Το κάθε «πακέτο bytes» περιλαμβάνει τόσα bytes όσα

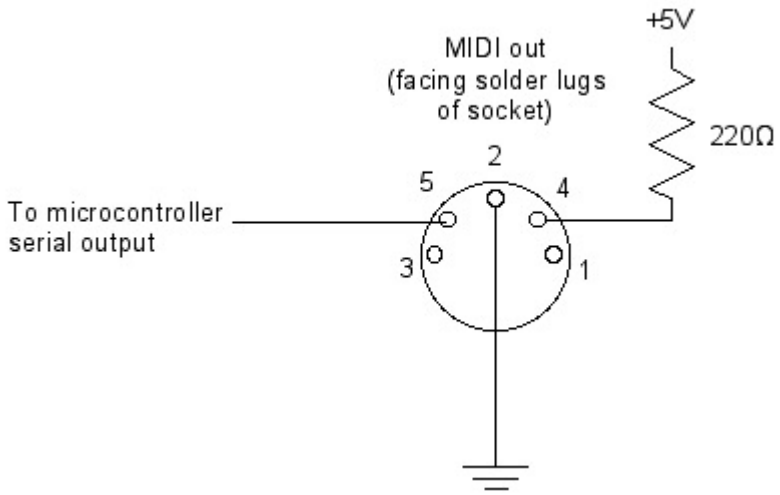


είναι αναγκαία για την αναπαράσταση του κάθε μηνύματος.

MIDI Byte / Start και Stop Bit

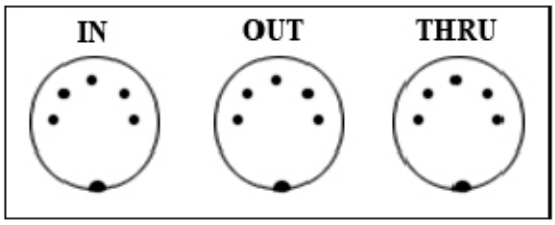
Η έννοια της **σειριακής διασύνδεσης** δηλώνει ότι τα bit μεταδίδονται το ένα μετά το άλλο, μέσα από ένα και μοναδικό καλώδιο. Σε ρυθμό των 31,250 bps, χρειάζονται 320 κλάσματα του δευτερολέπτου {microseconds} για τη μετάδοση ενός byte (ή 10 bit) ή με άλλα λόγια μπορούμε να στείλουμε περισσότερα από 3000 byte το δευτερόλεπτο.

Η έννοια του **ασύγχρονου πρωτόκολλου** στο MIDI σημαίνει οτι start και stop bit επιτρέπουν στη συσκευή αποδέκτης {slave} να εντοπίσει την παρουσία των δεδομένων και να αντιδράσει ανάλογα, χωρίς να απαιτείται από τον αποδέκτη να γνωρίζει την ακριβή χρονική στιγμή άφιξης των δεδομένων. Η ύπαρξη της ασυγχρόνιστης διασύνδεσης, στον κώδικα MIDI, συμπορεύεται απόλυτα με τις ανάγκες της μουσικής εκτέλεσης μιας και η «συμπεριφορά» του μουσικού / εκτελεστή είναι πάντοτε απρόβλεπτη. Όλα τα όργανα και τα καλώδια που είναι συμβατά με το MIDI είναι εξοπλισμένα με ένα βύσμα των 5 ακίδων {pin}, που ονομάζεται DIN βύσμα {**DIN plug**}. Τα βύσματα των οργάνων είναι «θηλυκά» με 5 μικρές τρύπες, ενώ τα καλώδια είναι «αρσενικά» με 5 ταιριαστές ακίδες στις δύο άκρες τους. Σε όλα τα MIDI καλώδια και βύσματα απέναντι από τις 5 μικρές ακίδες υπάρχει μία *μικρή εγκοπή* που μας βοηθάει να ευθυγραμμίσουμε τις μικρές ακίδες στο καλώδιο με τις αντίστοιχες τρύπες στην υποδοχή του θηλυκού βύσματος. Από τις πέντε ακίδες και τις αντίστοιχες υποδοχές χρησιμοποιούνται μόνο οι 2 , 4 και η 5 (βλ. παρακάτω εικόνα). Οι ακίδες 3 και 1 έχουν μείνει προς το παρόν ελεύθερες για μελλοντική χρησιμοποίηση από το πρωτόκολλο MIDI.



Όπως φαίνεται στην παραπάνω εικόνα (midi out) το pin 2 είναι η γείωση , το pin 4 είναι η τροφοδοσία 5volt , το pin 5 φέρει την πληροφορία midi.

Οι MIDI θύρες για τη MIDI διασύνδεση {MIDI Interface Ports} υπάρχουν στο πίσω μέρος όλων των MIDI συσκευών (βλ. παρακάτω εικόνα). Αυτές οι θύρες είναι:



• **OUT** *(στέλνει τα δεδομένα)*

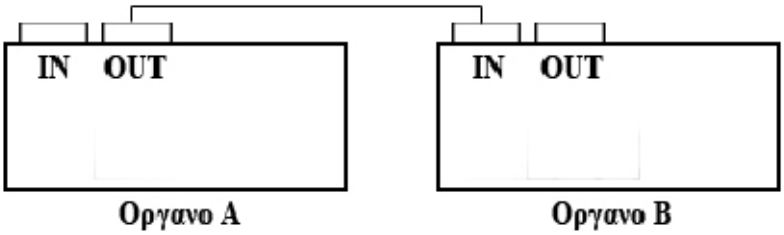
• **IN** *(λαμβάνει τα δεδομένα)*

• **THRU** *(επιτρέπει στα εισερχόμενα δεδομένα να περάσουν σε κάποιο άλλο όργανο)*

Πως λειτουργεί το MIDI:

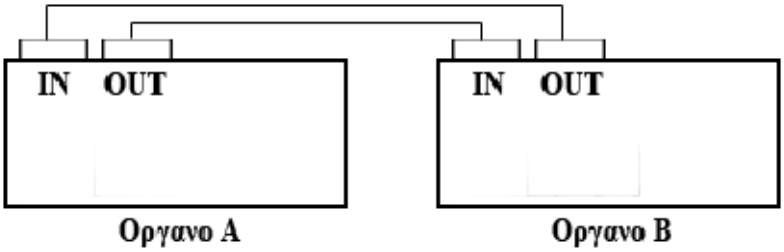
• Σύνδεση μονής κατεύθυνσης {**one way connection**}:

Η θύρα MIDI OUT του οργάνου Α, που ονομάζεται Ελεγκτής {**Master**}, μεταδίδει τα δεδομένα στο MIDI IN του οργάνου Β, που ονομάζεται Αποδέκτης {**Slave**}. Η αντίστροφη σύνδεση είναι επίσης πιθανή.



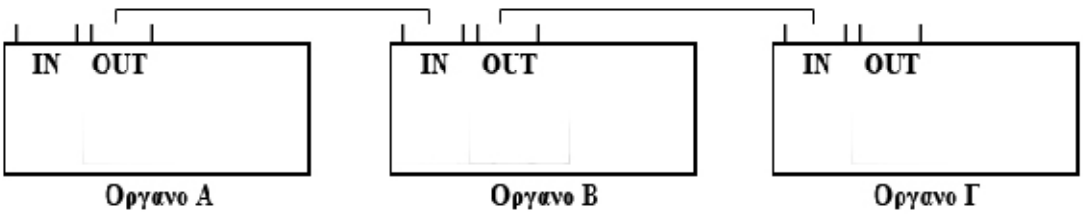
• Σύνδεση διπλής κατεύθυνσης {**bidirectional connection**} ή σύνδεση αμφίδρομης επικοινωνίας {**handshake connection**}:

Το MIDI OUT του οργάνου Α συνδέεται με το MIDI IN του οργάνου Β και αντίστροφα. Με αυτό τον τρόπο τα MIDI bytes μπορούν να ταξιδέψουν και στις δύο κατευθύνσεις.



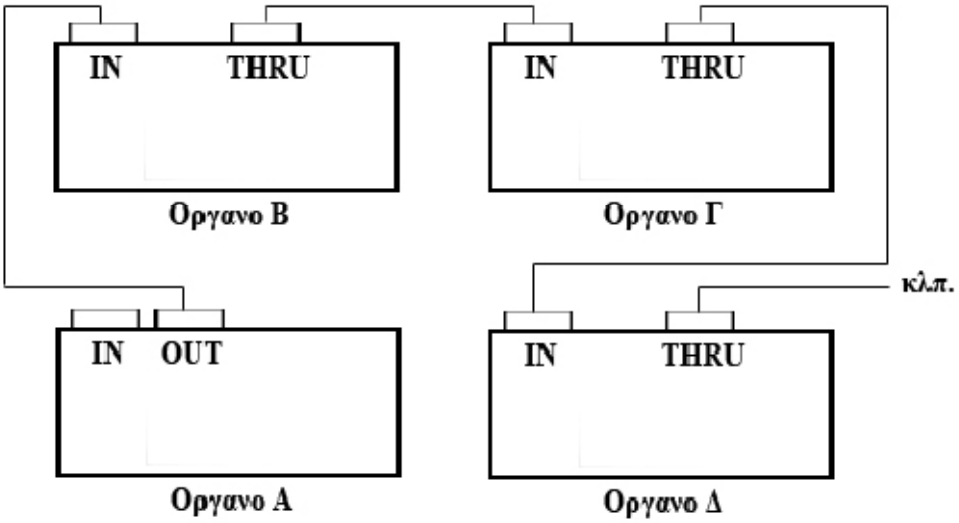
• Σύνδεση τριών MIDI οργάνων:

Το όργανο Α ενεργοποιεί το όργανο Β και το όργανο Β ενεργοποιεί το όργανο Γ, αλλά το Α δεν ενεργοποιεί το Γ. Το όργανο Γ θα μπορούσε να ενεργοποιεί το Α με μία ακόμη σύνδεση.



• Σύνδεση **Daisy Chain :**

Τα δεδομένα που στέλνονται από το όργανο Α στο Β αντιγράφονται και στέλνονται αμέσως στο όργανο Γ, το οποίο με τη σειρά του τα μεταφέρει στο όργανο Δ, και πάει λέγοντας.



Το MIDI σήμα έχει μία έντονη τάση να διακόπτεται όταν περνάει από ένα όργανο σε ένα άλλο. Για αυτό συνιστάται να μη υπάρχουν πάνω από 4 όργανα σε μία σύνδεση daisy chain.

Η Γλώσσα του MIDI / Status και Data Bytes

Το MIDI Πρωτόκολλο { MIDI Standard} διαχωρίζει τις MIDI πληροφορίες σε δύο κατηγορίες byte:

1. Byte ιδιότητας {**Status Byte**}, που ορίζει τη μουσική εντολή.

2. Byte δεδομένων {**Data Byte**}, που ορίζει την αξία της εντολής.

Γενικότερα, το status byte αντιπροσωπεύει τις ενέργειες που εκτέλεσε ο μουσικός – δηλαδή το πάτημα ή η απελευθέρωση ενός πλήκτρου ή η χρήση του πεντάλ {sustain pedal}. Όμως, στις περισσότερες περιπτώσεις μία ενέργεια του μουσικού εκτελεστή περικλείει περισσότερες πληροφορίες. Για παράδειγμα, τον αριθμό της νότας που παίχθηκε (οι νότες είναι αριθμημένες στο MIDI, π.χ. το μεσαίο Ντο είναι αριθμός 64) ή την αρχή και το τέλος του κρατήματος ενός πλήκτρου, δηλαδή η διάρκεια μίας νότας. Η πληροφορία αυτή είναι το data byte που συνοδεύει το status byte.

Στο δυαδικό σύστημα, το κάθε byte αναπαριστάνει μία από τις 256 πιθανές και διαφορετικές αξίες (από 0 έως _____255). Για να μπορέσει ένα MIDI όργανο να ξεχωρίσει ένα status byte από ένα data byte, το MIDI Πρωτόκολλο χρησιμοποιεί το έβδομο στοιχείο ενός byte, δηλαδή το έβδομο bit ως δείκτη. Το έβδομο bit ονομάζεται «το σημαντικότερο bit» {**most significant bi**t ή **MSB**}. Αν το bit αυτό εμφανισθεί ως χαρακτήρας 1, το byte είναι πληροφορία status, αν όμως εμφανισθεί ως 0, το byte είναι πληροφορία data (βλ. παρακάτω).

Στη γλώσσα του MIDI υπάρχουν 7 bit (τα bit 0 έως 6) που μπορούν να χρησιμοποιηθούν για την αναπαράσταση των πληροφοριών. Έτσι συνεπάγεται ότι ένα MIDI byte μπορεί να εκφράσει αντί για 256 διαφορετικές αξίες, μόνο **128** (από 0 έως 127).

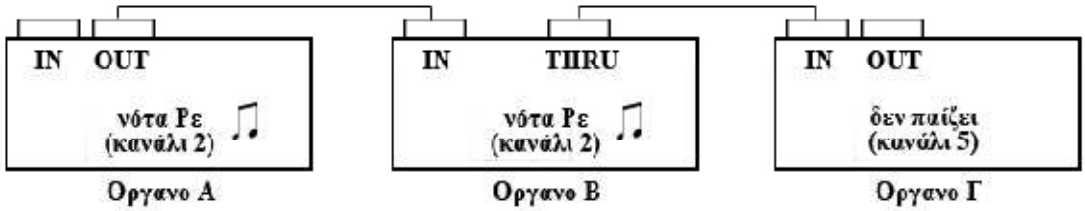
Status Byte:	**1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**
Bit:	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data Byte:	**0**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**	**0 ή 1**
Bit:	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Δεκαδική μορφή:		**(64)**	**(32)**	**(16)**	**(8)**	**(4)**	**(2)**	**(1)**

Status Byte και Data Byte

Όλες οι πληροφορίες MIDI μεταδίδονται δομημένες ως «status συν data». Δομημένοι συνδυασμοί «status συν data» πληροφοριών αποτελούν τα μηνύματα MIDI {**MIDI Messages**}.

Τα MIDI Κανάλια (MIDI Channels)

Ας θεωρήσουμε ότι ένα μουσικό σύστημα ή δίκτυο αποτελείται από ένα κλαβιέ συνδεμένο με άλλα 3 MIDI όργανα (συνθεσάιζερ, sound module, κλπ.) σε σύνδεση daisy chain. Κάθε μήνυμα που στέλνεται μέσω ελεγκτή {master}, όπως το πάτημα ενός πλήκτρου, περνάει από όλο το δίκτυο. Κάτω από κανονικές συνθήκες, όλα τα όργανα αποδέκτες {slaves} θα πρέπει να αρχίσουν να παίζουν αυτή τη νότα σε ταυτοφωνία. Τι μπορούμε να κάνουμε, ώστε χωρίς την αλλαγή των συνδέσεων να ενεργοποιήσουμε τις γεννήτριες ήχων {sound generators} «διαδοχικά» και όχι ταυτόχρονα. Ο τρόπος της τηλεοπτικής μετάδοσης φαίνεται να είναι μία καλή απάντηση. Μία τηλεόραση μπορεί να μεταδίδει κυριολεκτικά εκατοντάδες σήματα τηλεοπτικών καναλιών μέσω ενός μοναδικού καλωδίου ταυτόχρονα. Ένας ενσωματωμένος συντονιστής μέσα στη συσκευή της τηλεόρασης μας επιτρέπει να επιλέξουμε ένα μόνο κανάλι και να αγνοήσουμε όλα τα υπόλοιπα. Κάθε MIDI μήνυμα που μεταβιβάζει κάποιο γεγονός, χρησιμοποιεί ένα μέρος του status byte για να υποδείξει το κανάλι του γεγονότος. Αυτό επιτρέπει στο κάθε όργανο (αποδέκτη) το οποίο έχει ορισθεί σε ένα συγκεκριμένο κανάλι, να αποφασίσει αν κάποιο μήνυμα το αναφορά ή όχι.



Παίζοντας μία νότα στο όργανο Α (Ελεγκτής), που έχει ορισθεί στο κανάλι 2, οι πληροφορίες τις νότας μεταδίδονται στα άλλα όργανα. Το όργανο Β (κανάλι 2) παίζει τη νότα ενώ το όργανο Γ (κανάλι 5) δεν τη παίζει. Αν θέλουμε να ακουστεί το Όργανο Γ αντί για το Β, αλλάζουμε το κανάλι του Οργάνου Α από 2 σε 5. Ο αριθμός των MIDI καναλιών είναι περιορισμένος σε **16**. Η αναπαράσταση 16 αξιών σε δυαδική μορφή απαιτεί 4 bit. Το status byte ενός MIDI μηνύματος, bit 0 έως 3, μεταφέρει τον κώδικα για το MIDI κανάλι (βλ. εικόνα). Επειδή το έβδομο bit χρησιμοποιείται ως ενδεικτικό για την ιδιότητα του δεδομένου, μόνο τα bit 4 έως 6 είναι διαθέσιμα για την αναπαράσταση του περιεχομένου της μουσικής εντολής.

Για ευκολία, καλά είναι να σκέφτεστε τα 16 MIDI κανάλια ως 16 όργανα μιας ορχήστρας όπου το κάθε κανάλι είναι και ένας εκτελεστής.

MIDI ΜΗΝΥΜΑΤΑ

Στον παρακάτω πίνακα φαίνεται περιληπτικά το σύνολο των midi μηνυμάτων όπως αναγράφονται στο επίσημο site του midi (midi.org).

MIDI 1.0 Specification Message Summary

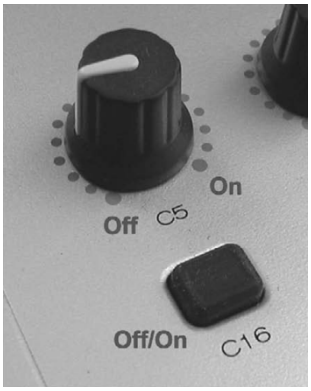
Status D7----D0	**Data Byte(s) D7----D0**	**Description**
Channel Voice Μηνύματα [nnnn = 0-15 (MIDI Channel Number 1-16)]		
1000nnnn	0kkkkkkk 0vvvvvvv	Note Off event. Αυτό το μήνυμα στέλνεται όταν αφήνουμε το πλήκτρο της νότας. (kkkkkkk) Είναι ο αριθμός της νότας. (vvvvvvv) Είναι το velocity.
1001nnnn	0kkkkkkk 0vvvvvvv	Note On event. Αυτό το μήνυμα στέλνεται όταν πατάμε το πλήκτρο της νότας. (kkkkkkk) Είναι ο αριθμός της νότας. (vvvvvvv) Είναι το velocity.
1010nnnn	0kkkkkkk 0vvvvvvv	Polyphonic Key Pressure (Aftertouch). Το μήνυμα αυτό σχετίζεται με την πίεση που ασκούμε στο πλήκτρο ενός midi keyboard όταν αυτό φτάσει στην κάτω θέση. (kkkkkkk) είναι ο αριθμός της νότας. (vvvvvvv) είναι η τιμή της πίεσης.
1011nnnn	0ccccccc 0vvvvvvv	Control Change. Αυτό το μήνυμα στέλνεται όταν αλλάζει η τιμή κάποιου controller (π.χ κάποιο pot). (ccccccc) είναι ο αριθμός του controller (0-119). (vvvvvvv) είναι η τιμή του controller (0-127).
1100nnnn	0ppppppp	Program Change. Αυτό το μήνυμα στέλνεται όταν θέλουμε να αλλάξουμε το ήχο σε κάποιο synthesizer . (ppppppp) είναι ο αριθμός του καινούργιου ήχου.
1101nnnn	0vvvvvvv	Channel Pressure (After-touch). Αυτό το μήνυμα στέλνεται συνήθως όταν πατάμε το πλήκτρο ενός midi keyboard και διαφέρει από το polyphonic aftertouch καθώς στέλνει την μεγαλύτερη τιμή της πίεσης που ασκούμε. (vvvvvvv) είναι η τιμή της πίεσης.

1110nnnn	0lllllll 0mmmmmmm	Pitch Wheel Change. 0mmmmmmm Αυτό το μήνυμα δείχνει την θέση του pitch-bend τροχού που βρίσκεται στα midi keyboard και δείχνει την μετατόπιση του τροχού σε πραγματικό χρόνο. (llllll) είναι το LSB . (mmmmmm) είναι το MSB.
Channel Mode Μηνύματα		
1011nnnn	0ccccccc 0vvvvvvv	Channel Mode Messages. Είναι ο ίδιος κώδικας με τα control change,αλλά συμπεριλαμβάνει το mode control καθώς κ ειδικά προκαθορισμένα μηνύματα με αριθμούς από 120-127. Οι εντολές είναι:
		All Sound Off. Όταν πατηθεί αυτό το κουμπί όλοι οι ταλαντωτές σε ένα synthesizer κλείνουν και οι εντάσεις τους μηδενίζονται. c = 120, v = 0: All Sound Off
		Reset All Controllers. Όταν δοθεί αυτή η εντολή οι τιμές όλων των ελεγκτών ρυθμίζονται στις default τιμές. c = 121, v = x: η τιμή αυτή πρέπει να είναι 0.
		Local Control. Όταν το Local Control είναι στην θέση Off, όλα οι συσκευές στο συγκεκριμένο κανάλι ανταποκρίνονται στον έλεγχο από midi. Όταν το Local Control είναι στην θέση On επανέρχεται και η λειτουργία των υπόλοιπων controllers. c = 122, v = 0: Local Control Off c = 122, v = 127: Local Control On
		All Notes Off. Όταν λαμβάνονται το μήνυμα All Notes Off απενεργοποιούνται όλοι οι ταλαντωτές. c = 123, v = 0: All Notes Off c = 124, v = 0: Omni Mode Off c = 125, v = 0: Omni Mode On c = 126, v = M: Mono Mode On (Poly Off) όπου M είναι ο αριθμός των καναλιών.(Omni Off) ή 0 (Omni On) c = 127, v = 0: Poly Mode On (Mono Off)

System Common Μηνύματα

11110000		System Exclusive. Αυτού του τύπου μηνύματα επιτρέπουν στους κατασκευαστές να δημιουργούν δικά τους μηνύματα.
11110001	0nnndddd	MIDI Time Code Quarter Frame. nnn = Είδος μηνύματος dddd = Τιμές
11110010	0lllllll 0mmmmmmm	Song Position Pointer. Αυτός είναι ένας εσωτερικός καταχωρητής (14 bit) ο οποίος κρατάει τον αριθμό των midi beat (1 beat= six MIDI clocks) από την αρχή του τραγουδιού. l είναι το LSB, m το MSB.
11110011	0sssssss	Song Select. Προσδιορίζει πια ακολουθία θα παιχτεί.
11110100		Undefined. (Κενό)
11110101		Undefined. (Κενό)
11110110		Tune Request. Όταν λαμβάνεται ένα μήνυμα tune request όλα τα αναλογικά synthesizers προχωρούν σε συντονισμό των ταλαντωτών τους.
11110111		End of Exclusive. Χρησιμοποιείται για να τερματιστεί το System Exclusive.

System Real-Time Μηνύματα		
11111000		Timing Clock. Στέλνεται 24 φορές ανά ένα τέταρτο της νότας όταν απαιτείται συγχρονισμός.
11111001		Undefined. (Κενό)
11111010		Start. Ξεκινάει την αναπαραγωγή του επιλεγμένου κομματιού (αυτό το μήνυμα ακολουθείται με Timing Clocks).
11111011		Continue. Συνέχεια της αναπαραγωγής από το σημείο της διακοπής.
11111100		Stop. Διακοπή της αναπαραγωγής.
11111101		Undefined. (Κενό)
11111110		Active Sensing. Χρησιμοποίηση αυτού του μηνύματος είναι προαιρετική. Όταν λαμβάνεται, ο δέκτης περιμένει να λάβει και άλλο active sensing κάθε 300 ms (max), αλλιώς υποθέτει ότι η σύνδεση έχει τερματιστεί. Όταν τερματιστεί ο δέκτης θα τερματίσει όλες τις φωνές και θα επιστρέψει σε κανονική λειτουργία.
11111111		Reset. Τοποθέτηση όλων των δεκτών στο σύστημα σε default κατάσταση.



CONTROL CHANGE ΜΗΝΥΜΑΤΑ

Παρακάτω ακολουθεί μια περιγραφή των control change μηνυμάτων καθώς αυτός είναι ο τύπος μηνυμάτων που θα χρησιμοποιήσουμε για την κατασκευή του midi controller.

Τα Control Change μηνύματα έχουν σαν σκοπό να παρέχουν έναν έλεγχο σε πραγματικό χρόνο σε διάφορες παραμέτρους. Εξαιτίας της πολύ ευέλικτης κατασκευής τους αυτά τα μηνύματα έγιναν το πιο ζωντανό κομμάτι της ιστορίας του midi. Τα Control Change μηνύματα δεν έπαψαν ποτέ να επεκτείνονται. Τα μηνύματα αυτά στην ουσία εκφράζουν και προέρχονται από έναν ελεγκτή (controller) που έχει σαν σκοπό να μετατρέψει μια εκφραστική ενέργεια του μουσικού σε midi πληροφορία. Υπάρχουν ωστόσο control change μηνύματα που δεν εκφράζουν απόλυτα μουσικές ενέργειες αλλά γενικές εκφραστικές διαδικασίες.

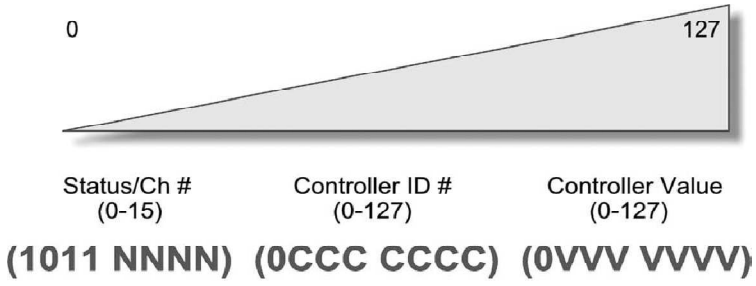
Υπάρχουν τρείς τύποι control change μηνυμάτων που μεταδίδονται μέσω MIDI:

- *Continuous Controllers* – Η πληροφορία που μεταφέρουν αυτοί οι controllers θεωρητικά περιγράφει την κίνηση ενός τροχού ή ενός πεντάλ, ή την μετακίνηση ενός ποτενσιόμετρου. Το αποτέλεσμα είναι ένα ασταμάτητο ρεύμα δεδομένων, γι αυτό και η ονομασία συνεχείς controllers.
- *Switch controllers* – Είναι controllers που μπορούν να βρίσκονται είτε σε “off” είτε σε “on” κατάσταση (buttons / κουμπιά).
- *Channel mode message controllers* – Είναι controllers που αριθμούνται από 120 μέχρι 127 και χρησιμοποιούνται για να ρυθμίσουν τον ήχο μιας νότας, να κάνουν reset στο όργανο, να ρυθμίζουν το local control (on/off), all notes off και το MIDI mode status μιας συσκευής ή ενός οργάνου.

Παρακάτω βλέπουμε την ανάλυση ενός control change messege:

- Όνομα ….....Control Change
- Format…….. (1011 NNNN) (0CCC CCCC) (0VVV VVVV)
- Τύπος ……...Channel Voice Message από 0-120

  Channel Mode Message από 120-127
- NNNN……. …….Αριθμός καναλιού
- CCC CCCC.. ……... Αριθμός Controller
- VVV VVVV…….. Τιμή Controller



ΤΙΜΕΣ ΤΩΝ CONTROLLERS

Το τρίτο byte ενός control change μηνύματος χρησιμοποιείται για να δηλώσει την πραγματική τιμή του controller. Η τιμή αυτή χρησιμοποιείται για να δείξει την θέση, το βάθος ή το επίπεδο μιας παραμέτρου. Παρακάτω ακολουθεί ένα παράδειγμα πως αυτές οι τιμές μπορούν να χρησιμοποιηθούν για να ελέγξουνε διάφορες παραμέτρους.

- Στην περίπτωση που μια μεταβλητή παράμετρος δεν απαιτεί πολύ υψηλή ακρίβεια ένας controller τον 7-bit επιτρέπει 128 διακριτές τιμές (με ελάχιστη τιμή το 0 και μέγιστη το 127).
- Το εύρος τιμών ενός pan-pot κυμαίνεται μεταξύ του 0 (τέρμα αριστερά) και του 127 (τέρμα δεξιά) και η τιμή 64 δηλώνει την κεντρική θέση.
- Το εύρος τιμών ενός switch (κουμπί) contoller είναι συνήθως 0 (για off) και 127 (για on). Ωστόσο μερικές φορές οι τιμές από 0 έως 63 αναγνωρίζονται σαν “off” και οι τιμές από 64 έως 127 αναγνωρίζονται σαν “on”.

MIDI Controllers

Midi controllers είναι συσκευές που παράγουν midi μηνύματα και μπορούν να συνδεθούν με οποιαδήποτε άλλη συσκευή που υποστηρίζει midi ή με τον υπολογιστή.

Υπάρχουν διάφορα είδη midi controllers αλλά ίσως ο ποιο συνηθισμένος τύπος είναι αυτός που φαίνεται στην παρακάτω εικόνα και είναι ένα midi keyboard. Χρησιμοποιείται κυρίως για τον έλεγχο synthesizers, είτε σε hardware είτε σε software (VST) επίπεδο.

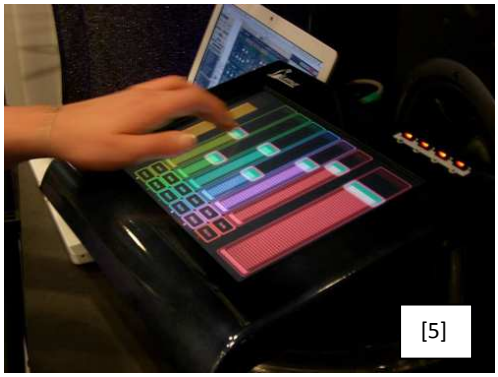
Κάποια άλλα είδη φαίνονται παρακάτω.


[1]

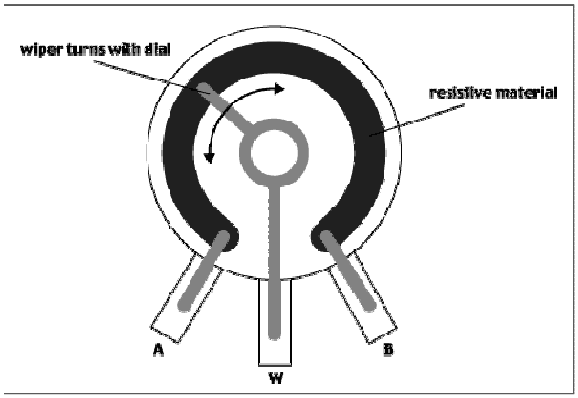

[2]

[3]

[4]


[5]

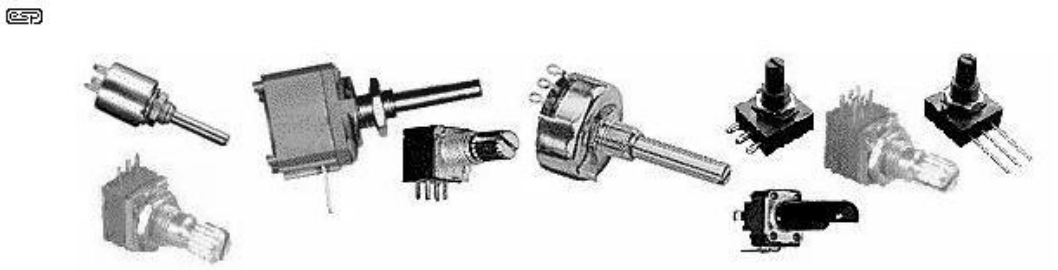
Στις εικόνες 1 και 2 είναι controllers που απευθύνονται σε dj και για έλεγχο dj εφαρμογών. Στην 3 φαίνεται ένας controller για τύμπανα και κρουστά (drum pad) . Στην 4 ένας controller με πολλά κουμπιά για διάφορες εφαρμογές ενώ στην 5 φαίνεται ένας σύγχρονος controller αφής που μπορεί να προσομοιώσει faders,pots,buttons και άλλα.



ΠΟΤΕΝΣΙΟΜΕΤΡΑ

Αφού τελειώσαμε την σύντομη αναφορά μας στο midi και τα midi μηνύματα θα πρέπει να δούμε κάποια βασικά στοιχεία όσων αναφορά τα ποτενσιόμετρ καθώς αυτός είναι ο τύπος ελεγκτών που θα χρησιμοποιήσουμε στην κατασκευή μας.

Το ποτενσιόμετρο (ή pot, όπως είναι γνωστό) είναι ένα απλός ήλεκτρο-μηχανικός μετατροπέας. Μετατρέπει την κυκλική ή γραμμική κίνηση του χειριστή σε μεταβολή της αντίστασης και η μεταβολή αυτή μπορεί να χρησιμοποιηθεί για να ελέγχει οτιδήποτε , από την ένταση σε ένα hi-fi μέχρι την πλοήγηση ενός πλοίου.

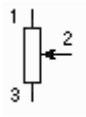


Βασικά είδη ποτενσιόμετρων

Στην παραπάνω εικόνα φαίνονται κάποιοι συνηθισμένοι τύποι ποτενσιόμετρων. Το πιο συνηθισμένο είναι αυτό στο κέντρο της φωτογραφίας, ένα panel mount ποτενσιόμετρο με διάμετρο 25 mm. Το pot αυτό στηρίζεται

σε μια τρύπα 10 mm και έχει άξονα 6,33 millimetre. Αυτού το είδος ποτενσιόμετρο παραμένει σχεδόν ίδιο εδώ και σαράντα χρόνια.

Τα περισσότερα ποτενσιόμετρα έχουν γωνία περιστροφής 270 μοίρες από την μια άκρη στην άλλη (ποτενσιόμετρα μιας στροφής) υπάρχουν ωστόσο και pots με γωνία περιστροφής 200 μοίρες και λιγότερο.



Το standard σύμβολο σε ένα κύκλωμα για τα pots φαίνεται στα αριστερά.

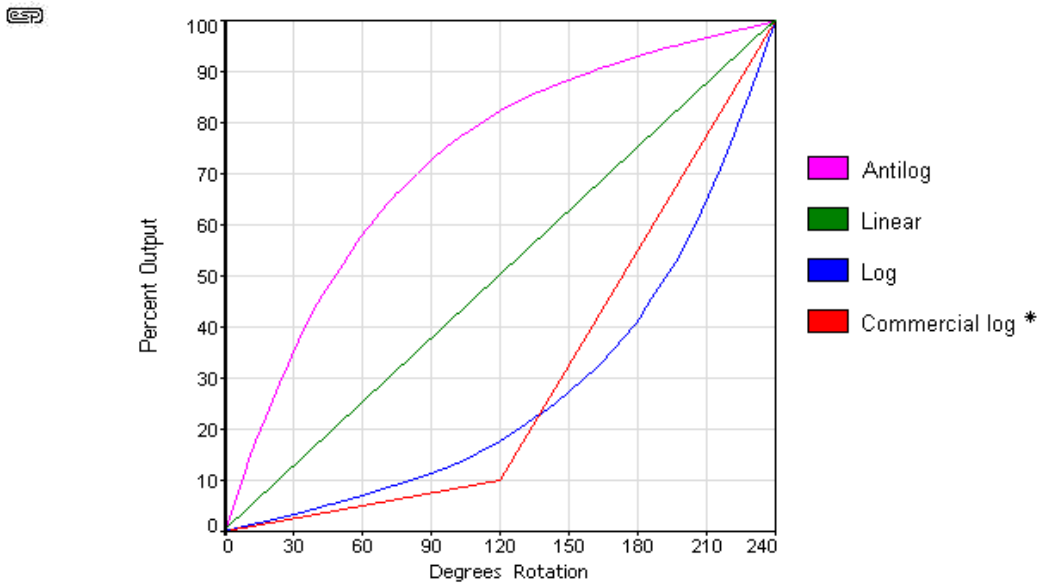
Knobs

Πριν δούμε όμως τα είδη των ποτενσιόμετρων ας ρίξουμε μια ματιά και στα knobs. Τα Knobs εφαρμόζουν πάνω στον άξονα του pot και χρησιμεύουν σαν «χερούλια» κάνοντας πιο βολική την χρήση τους.

Καμπύλες ποτενσιόμετρων

Η καμπύλη (taper) ενός ποτενσιόμετρου είναι πολύ σημαντική. Κυρίως για τα panel pots (αυτά που στηρίζονται σε panel) πρέπει να γνωρίζουμε συγκεκριμένα την λειτουργία που θα έχει το pot και να επιλέξουμε ανάλογα τον τύπο που θα χρησιμοποιήσουμε.

Πιο συνηθισμένη χρήση του pot είναι για audio εφαρμογές π.χ. για έλεγχο εντάσεων. Από την στιγμή που η ακοή μας έχει λογαριθμική απόκριση στην ηχητική πίεση είναι σημαντικό για ένα pot που ελέγχει την ένταση να έχει ομαλή μεταβολή ώστε μια αλλαγή της θέσης του pot να προκαλεί την ίδια μεταβολή στην ένταση σε όλα τα επίπεδα.



Στο παραπάνω διάγραμμα φαίνονται κάποιες χαρακτηριστικές καμπύλες ποτενσιόμετρων.

- Antilog : Αντίστροφη λογαριθμική – η τάση στην έξοδο σε συνάρτηση με την περιστροφή του pot μεταβάλλεται όπως φαίνεται από την μωβ καμπύλη.
- Linear : Γραμμική – η τάση στην έξοδο του pot μεταβάλλεται γραμμικά σε σχέση με την περιστροφή (πράσινη καμπύλη). Αυτός είναι ο τύπος ποτενσιόμετρου που θα χρησιμοποιήσουμε για την κατασκευή.
- Log : Λογαριθμική – η τάση στην έξοδο του pot μεταβάλλεται λογαριθμικά σε σχέση με την περιστροφή. Αυτός είναι ο πιο συνηθισμένος τύπος για audio εφαρμογές (μπλε καμπύλη).
- Commercial log : είναι ειδικός τύπος pot που δημιουργείται χρησιμοποιώντας δυο στοιχεία διαφορετικής αντίστασης με αποτέλεσμα η τάση στην έξοδο να μεταβάλλεται όπως φαίνεται στην κόκκινη καμπύλη.

Πάνω σε κάθε pot υπάρχουν σύμβολα που δηλώνουν τα χαρακτηριστικά του. Όσον αφορά την καμπύλη τα σύμβολα αυτά φαίνονται στον παρακάτω πίνακα.

Taper (καμπύλη)	**Old Code (παλαιότερος συμβολισμός)**	**New Code (νέος συμβολισμός)**	**Alternate(εναλλακτικά)**
Linear	A	B	LIN
Log (Audio)	C	A	LOG
Antilog	F	N/A	N/A

Όσον αφορά την αντίσταση πάνω σε κάθε pot αναγράφεται η αντίστοιχη τιμή π.χ σε ένα pot των 100 ΚΩ θα αναγράφεται 100K. Οι συνηθισμένες τιμές αντίστασης για τα ποτενσιόμετρα είναι 1K , 5K , 10K , 25K , 50K , 100K , 500K και ένα 1ΜΩ.

Τα ποτενσιόμετρα διαφέρουν και ως προς το υλικό της αντίστασης που χρησιμοποιούν. Κάποια είδη φαίνονται παρακάτω:

Υλικό	**Common uses (συνήθεις χρήσεις)**	**Power (Ισχύς) (Typ)**
Carbon (άνθρακας)	Είναι το πιο συνηθισμένο υλικό, ειδικά για φθηνά κ μέτριας ποιότητας pots. Έχει καλή διάρκεια ζωής και η στάθμη θορύβου είναι αρκετά αποδεκτή στις περισσότερες περιπτώσεις.	0.1 to 0.5W
Cermet (κεραμικό)	Πολύ υψηλής ποιότητας ποτενσιόμετρο με πολύ χαμηλή στάθμη θορύβου. Πολύ σταθερά αλλά με περιορισμένη διάρκεια ζωής.	0.25 to 2W (or more)
Conductive Plastic (αγώγιμο πλαστικό)	Πολύ υψηλή ποιότητα(για επαγγελματικές audio εφαρμογές) είτε για περιστρεφόμενα pots είτε για fader. Εξαιρετική διάρκεια ζωής, χαμηλή στάθμη θορύβου και πολύ καλή μηχανική αίσθηση.	0.25 to 0.5W
Wire wound (Σύρματος)	Για εφαρμογές υψηλής ισχύος με σχεδόν άπειρη διάρκεια ζωής.	5 to 50W (or more)

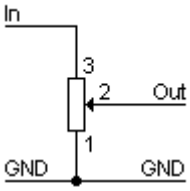
Τα ποτενσιόμετρα διαχωρίζονται και ως προς την μηχανική και ηλεκτρική φύσης τους όπως φαίνεται στον παρακάτω πίνακα.

Χειρισμός	**Configuration**	**Τύπος**	**Συνήθεις χρήσεις**
Περιστροφικός (rotary)	Μονό κανάλι (Single gang)	Μιας στροφής	Έλεγχος ενός καναλιού όπως σε μονοφωνικούς ενισχυτές, ενισχυτές κιθάρας.
Περιστροφικός (rotary)	Μονό κανάλι (Single gang)	Πολλών στροφών	Μεγάλης ακρίβειας ποτενσιόμετρα για κρίσιμες εφαρμογές. Το εύρος της αντίστασης καλύπτεται σε 10 έως 25 στροφές.
Περιστροφικός (rotary)	Διπλό κανάλι (Dual gang)	Μιας στροφής	Για stereo εφαρμογές ή όπου αλλού χρειάζεται ο ταυτόχρονος έλεγχος δυο αντιστάσεων. Συνήθως οι δυο αντιστάσεις σε όλα τα dual gang pots είναι ίδιες.
Περιστροφικός (rotary)	Διπλό ομόκεντρο (Dual concentric)	Μιας στροφής	Συνήθως χρησιμοποιείται σε στερεοφωνικά αυτοκινήτου και άλλες οικιακές συσκευές. Αυτά τα pot έχουν δύο ομόκεντρους άξονες επιτρέποντας έτσι τον ταυτόχρονο έλεγχο π.χ. της έντασης της συχνότητας και του τόνου.
Fader (linear)	Μονό κανάλι (Single gang)	Ολίσθησης	Συνήθως χρησιμοποιούνται σαν faders (ποτενσιόμετρα ολίσθησης). Διατίθενται σε διάφορα μήκη από 30 mm μέχρι 100 mm ή και περισσότερο και συνήθως κατασκευάζονται από αγώγιμο πλαστικό υλικό.
Fader (linear)	Διπλό κανάλι (Dual gang)	Ολίσθησης	Όπως και το παραπάνω αλλά για stereo εφαρμογές (stereo κανάλια, stereo μίκτες).

Στην παρακάτω φωτογραφία φαίνεται το πιο συνηθισμένο είδος ποτενσιόμετρου και είναι αυτό που χρησιμοποιήθηκε και στην κατασκευή. Το ποτενσιόμετρο αυτό έχει τρείς ακίδες (terminals) αριθμημένα σαν 1, 2 και 3 από αριστερά προς τα δεξιά. Η ακίδα 2 είναι η έξοδος (wiper). Η ακίδα 1 συνδέεται στην γείωση και η 3 στην τάση τροφοδοσίας. Έτσι η τάση στην έξοδο μεταβάλλεται από το μηδέν μέχρι την τιμή της τάσης τροφοδοσίας.

Συνδεσμολογία ποτενσιόμετρου

Στο παρακάτω σχήμα φαίνεται η απλή σύνδεση ενός pot σε ένα κύκλωμα ελέγχου της έντασης, που είναι και ο πιο συνηθισμένος τρόπος σύνδεσης.



Γυρίζοντας τον άξονα με την φορά του ρολογιού στην ουσία το pin 2 (έξοδος) έρχεται πιο κοντά στο pin 3 (τάση τροφοδοσίας) με αποτέλεσμα την αύξηση της τάσης στην έξοδο. Η περιστροφή του άξονα στην ουσία επιλέγει διαφορετικό σημείο κατά μήκος του υλικού της αντίστασης δημιουργώντας έτσι έναν διαιρέτη τάσης, οπότε η εξασθένηση του σήματος είναι ανάλογη με την περιστροφή του άξονα. Στην θέση τέρμα δεξιά η αντίσταση του pot είναι σχεδόν μηδενική (σε σειρά με το σήμα) και έτσι σε αυτό το σημείο έχουμε την μέγιστη τάση εξόδου (και αντίστροφα).

ARDUINO

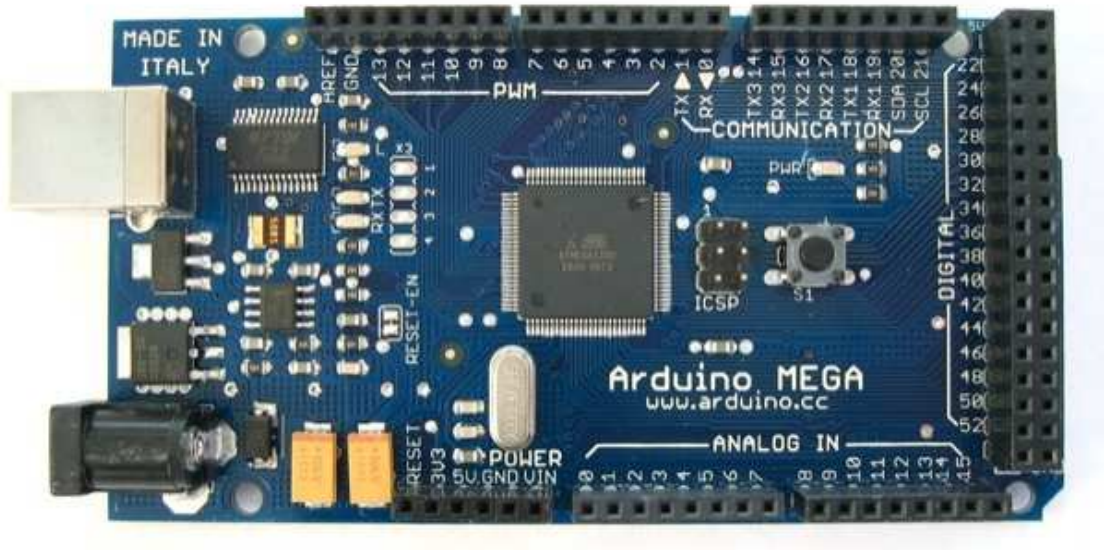
Το arduino είναι μια πλατφόρμα δημιουργίας ηλεκτρονικών πρωτοτύπων ανοιχτού κώδικα (open-source) βασισμένη σε ευέλικτο και εύκολο στην χρήση hardware και software. Απευθύνεται σε καλλιτέχνες, σχεδιαστές, χομπίστες και γενικά σε όποιον ενδιαφέρεται να δημιουργήσει interactive εφαρμογές ή περιβάλλοντα.

Το arduino μπορεί να διαβάσει το περιβάλλον λαμβάνοντας σήματα από διάφορους αισθητήρες και μπορεί να επηρεάσει το περιβάλλον ελέγχοντας φώτα, μηχανισμούς και άλλα. Ο μικροελεγκτής που βρίσκεται σε κάθε πλακέτα arduino μπορεί να προγραμματιστεί σε γλώσσα arduino programming language (βασισμένη στην Wiring) και το προγραμματιστικό περιβάλλον του arduino (βασισμένο στο Proccesing). Οι arduino εφαρμογές μπορούν να λειτουργούν μόνες τους (stand-alone) ή μπορούν να επικοινωνούν με διάφορα προγράμματα σε ένα υπολογιστή (π.χ Flash,Processing, MaxMSP).

Υπάρχουν διάφορες πλακέτες arduino βασισμένες σε διαφορετικούς μικροελεγκτές οι οποίες ανάλογα με τα χαρακτηριστικά κοστίζουν και ανάλογα. Το πρόγραμμα compiler του arduino είναι δωρεάν και διαθέσιμο από το επίσημο site (arduino.cc).

ARDUINO MEGA

Για την κατασκευή του midi controller χρησιμοποιήσαμε την πλακέτα ARDUINO MEGA που φαίνεται παρακάτω.



Η πλακέτα Arduino Mega είναι μια πλακέτα μικροελεγκτή βασισμένη στον Atmega1280. Έχει 54 ψηφιακές εισόδους-εξόδους (από τις οποίες οι 14 μπορούν να χρησιμοποιηθούν σαν PWM έξοδοι), 16 αναλογικές εισόδους, 4 UARTs (Hardware serial ports), έναν ταλαντωτή (crystall oscillator) στα 16 MHz, σύνδεση USB, μια υποδοχή για τροφοδοσία, υποδοχές (Header) ICSP, και ένα κουμπί reset. Περιέχει όλα όσα χρειάζονται για να υποστηρίξη τον μικροελεγκτή. Συνδέεται στον υπολογιστή μέσο USB και είναι συμβατό με τα περισσότερα shields (πλακέτες επέκτασης) σχεδιασμένα για τον Arduino Duemilanove ή Diecimila.

Χαρακτηριστικά

Μικροελεγκτής	Atmega1280
Τάση λειτουργίας	5V
Τάση εισόδου (προτεινόμενη)	7-12V
Τάση εισόδου (όρια)	6-20V
Ψηφιακές είσοδοι-έξοδοι	54 (από τις οποίες οι 15 παρέχουν PWM έξοδο)

Αναλογικές είσοδοι	16
DC ρεύμα για κάθε είσοδο-έξοδο	40mA
DC ρεύμα για το PIN 3,3V	50mA
Flash μνήμη	128Kb από τα οποία τα 4kb χρησιμοποιούνται από τον bootloader
SRAM	8Kb
EEPROM	4Kb
Clock Speed	16 MHz

Τροφοδοσία

Η πλακέτα Arduino Mega τροφοδοτείται είτε μέσω USB είτε από εξωτερικο τροφοδοτικό. Η πηγή τροφοδοσίας επιλλέγεται αυτόματα.

Η πλακέτα μπορεί να λειτουργήσει με παρεχόμενη τάση από 6 εως 20 Volt. Ωστόσο όταν τροφοδοτείται με λιγότερα από 7V το pin των 5V μπορεί να παρέχει λιγότερα από 5V και το κύκλωμα μπορεί να μην είναι σταθερό. Ενώ αν τροφοδοτείται με περισσότερα από 12V ο σταθεροποιητής τάσης μπορεί να υπερθερμανθεί και να προκαλέσει ζημιά στην πλακέτα. Γιαυτό η προτεινόμενη τάση τροφοδοσίας είναι από 7 εως 12V.

Τα pin τροφοδοσίας φαίνονται παρακάτω:

- VIN . Είναι η τάση εισόδου για την πλακέτα όταν χρησιμοποιούμε εξωτερική τροφοδοσία. Μπορούμε να τροφοδοτήσουμε την πλακέτα μέσω αυτού του pin ή αν χρησιμοποιούμε την ειδική υποδοχή για την τροφοδοσία. Μπορούμε να έχουμε πρόσβαση στην τάση τροφοδοσίας μέσω αυτού του pin.
- 5V. Είναι η σταθεροποιημένη τάση τροφοδοσίας που χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή και των άλλων στοιχείων της πλακέτας.
- 3V3. Παρέχει τάση 3,3V που δημιουργείται από το FTDI chip με μέγιστη παροχή ρεύματος 50mA.
- GND. Έιναι η γείωση.

Μνήμη

Ο Atmega1280 έχει 128 kb flash μνήμη για την αποθήκευση του κώδικα, 8 Kb SRAM και 4 kb EEPROM.

Είσοδοι και έξοδοι

Η κάθε μια από τις 54 ψηφιακές εισόδους-εξόδους μπορούν να χρησιμοποιηθούν σε συνδυασμό με τις εντολές pinMode(), digitalWrite(), digitalRead(). Λειτουργούνε στα 5V και κάθε pin μπορεί να παρέχει ή να λάβει μέχρι και 40mA και έχει εσωτερική αντίσταση από 20-50ΚΩ. Συγκεκριμένα κάποια pins έχουν ξεχωριστές ιδιότητες:

- Τα pin Serial : 0 (RX) και 1 (TX); Serial 1: 19 (RX) και 18 (TX); Serial 2: 17 (RX) και 16 (TX); Serial 3: 15 (RX) και 14 (TX) χρησιμοποιούνται για να λαμβάνουν (RX) ή να στέλνουν (TX) σειριακή πληροφορία (TTL 5V).
- Εξωτερικά Interrupts (τα interrupts είναι διακοπές της ομαλής ροής τους προγράμματος): 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), και 21 (interrupt 2). Αυτά τα pin μπορούν να ρυθμιστούν για να ξεκινήσουν κάποιο interrupt χρησιμοποιώντας την εντολή attachInterrupt() .
- PWM : Τα pin 2 εως 13 και 44 εως 46. Παρέχουν 8 bit PWM (Pulse-width modulation-είναι μια τεχνική για τον έλεγχο της τάσης) έξοδο χρησιμοποιώντας την εντολή analogWrite().
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) . Αυτά τα pin υποστηρίζουν επικοινωνία τύπου SPI .
- LED : το pin 13 είναι συνδεδεμένο με ένα LED που είναι ενσωματωμένο στην πλακέτα.
- I2C: 20 (SDA) και 21 (SCL).

Ο Atmega1280 έχει 16 αναλογικές εισόδους κάθε μια από τις οποίες έχει ανάλυση 10 bits (δηλαδή 1024 διαφορετικές τιμές). Σαν προεπιλεγμένη κατάσταση διαβάζουν από 0-5 V , ωστόσο υπάρχει η δυνατότητα αλλαγής του εύρους χρησιμοποιώντας το pin AREF και την εντολή analogReference().

Προγραμματισμός

Η πλακέτα Arduino Mega μπορεί να προγραμματιστεί με το πρόγραμμα arduino στη γλώσσα που φαίνεται παρακάτω:

Σύνολο Εντολών

Τα arduino προγράμματα μπορούν να χωριστούν σε 3 κύρια μέρη : δομή (structure), τιμές (μεταβλητές και σταθερές), και οι εντολές (functions).

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /* */ (multi-line comment)
- #define
- #include

Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long
- float
- double
- string - char array
- String - object
- array

Conversion

- char()
- byte()
- int()
- word()
- long()
- float()

Variable Scope & Qualifiers

- variable scope
- static

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

Time

- millis()
- micros()
- delay()
- delayMicroseconds()

Math

- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

Trigonometry

- sin()
- cos()

- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

- && (and)
- || (or)
- ! (not)

Pointer Access Operators

- * dereference operator
- & reference operator

Bitwise Operators

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

Compound Operators

- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)
- &= (compound bitwise and)
- |= (compound bitwise or)

- volatile
- const

Utilities

- sizeof()

- tan()

Random Numbers

- randomSeed()
- random()

Bits and Bytes

- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

External Interrupts

- attachInterrupt()
- detachInterrupt()

Interrupts

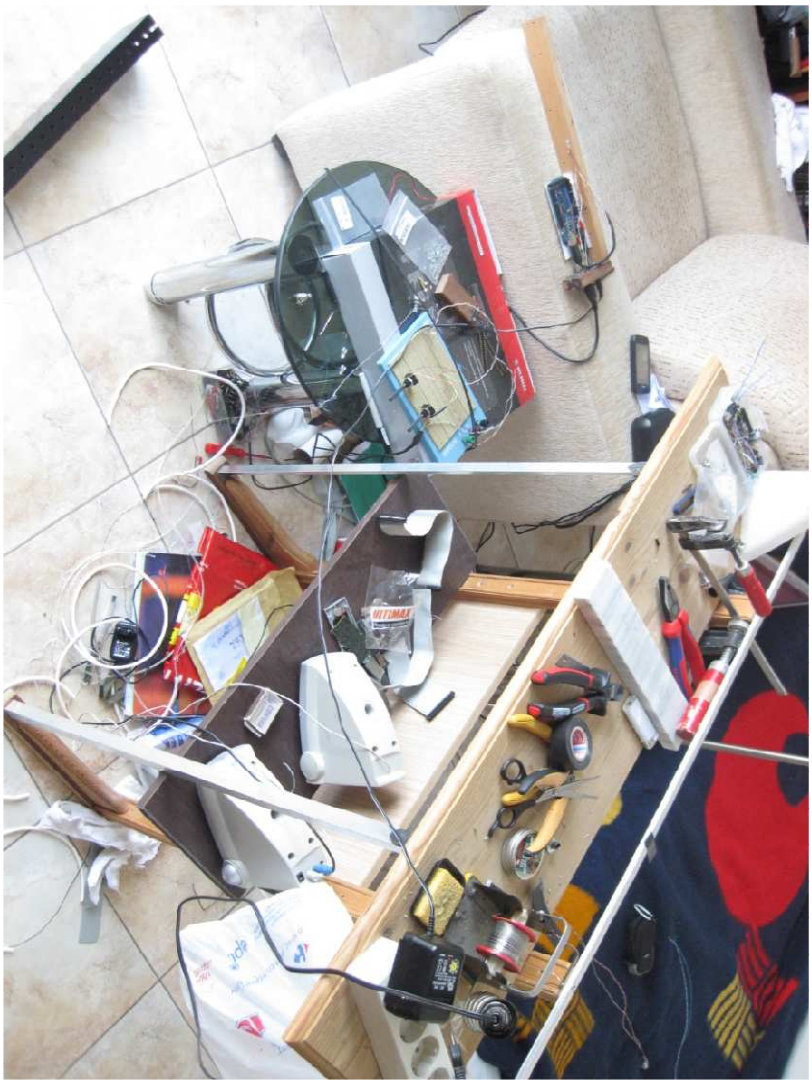
- interrupts()
- noInterrupts()

Communication

- Serial
- Stream

Τις εντολές που θα χρησιμοποιήσουμε στην κατασκευή θα τις αναλύσουμε στο πειραματικό μέρος της εργασίας.

ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ



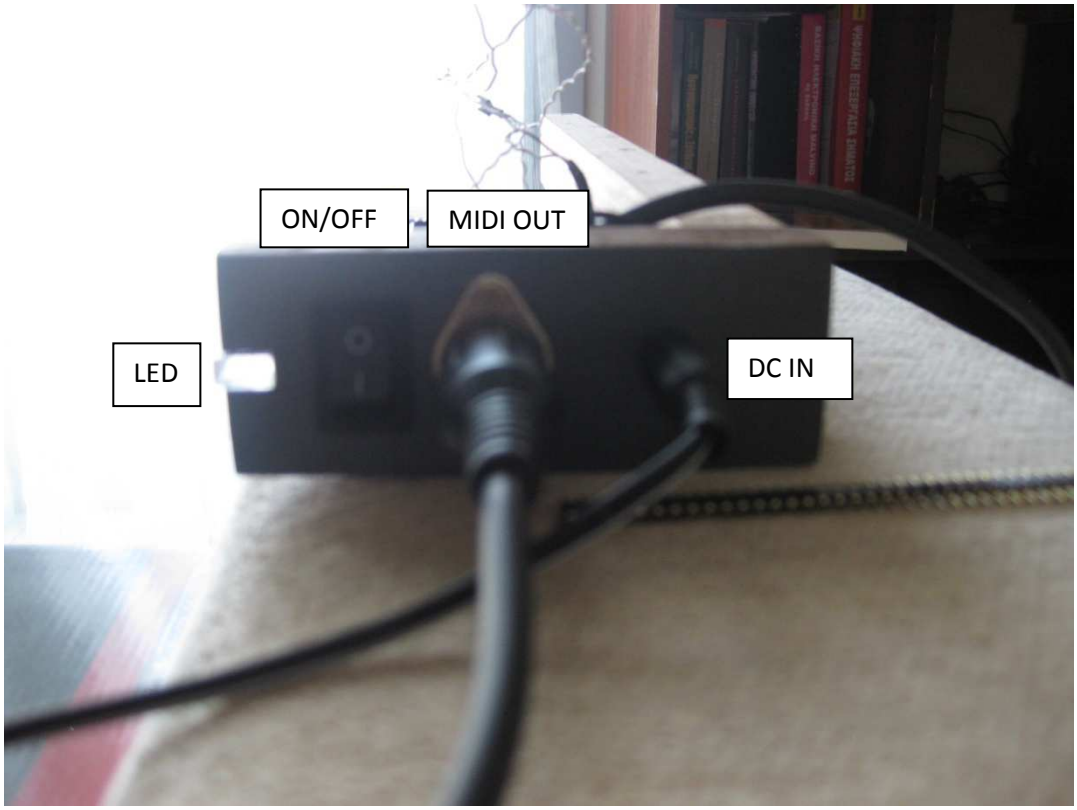
Στο κεφάλαιο αυτό θα δούμε αναλυτικά την όλη διαδικασία της κατασκευής του midi controller.

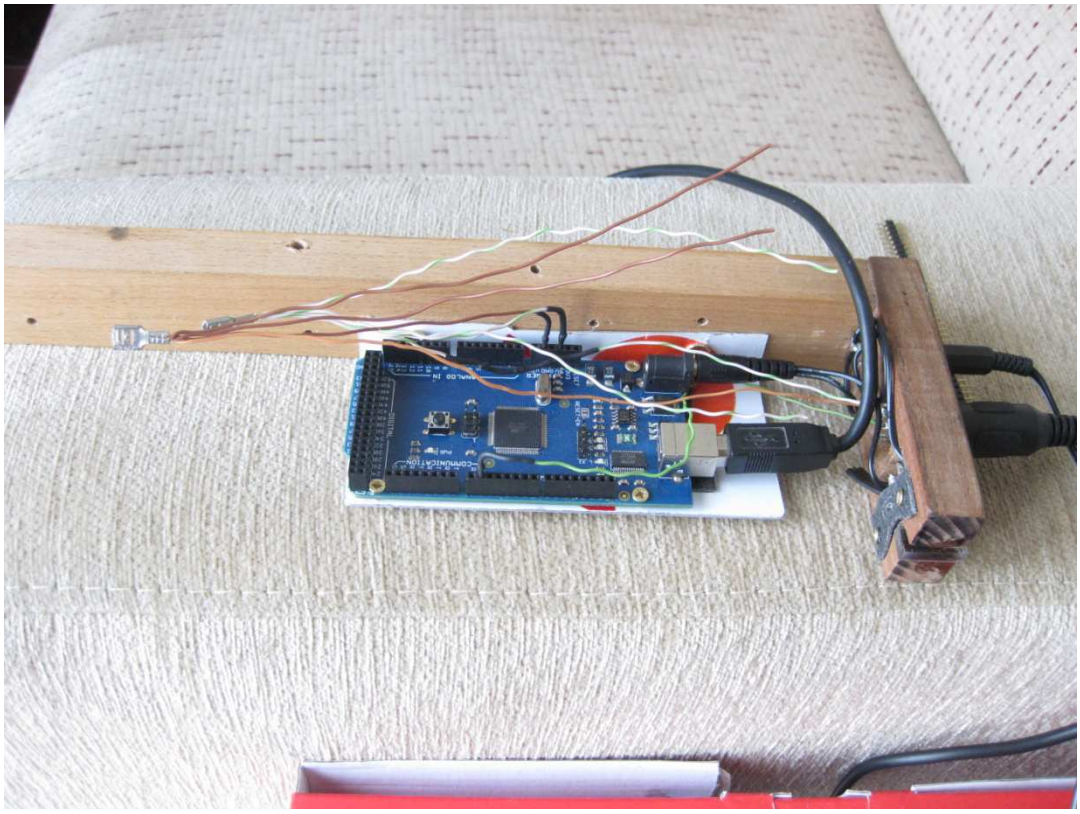
❖ ***Το κουτί***

Θα ξεκινήσουμε την περιγραφή μας με την κατασκευή του κουτιού.

Το κουτί που φαίνεται στην παραπάνω φωτογραφία το φτιάξαμε από ένα κομμάτι αλουμινίου με διαστάσεις 10x4x42cm. Το κομμάτι αυτό ήταν από λευκό αλουμίνιο πάχους 1mm με κενό εσωτερικό. Αφού το κόψαμε στις διαστάσεις που το θέλαμε χρησιμοποιήσαμε τρυπάνι διαμέτρου 6mm για να κάνουμε τις 16 τρύπες για τα ποτενσιόμετρα. Μετά χρησιμοποιώντας τροχό κάναμε τις εγκοπές για τα LED. Τέλος το κομμάτι βάφτηκε με μαύρο χρώμα.

΄Υστερα φτιάξαμε τον πάτο και τα δύο πλαϊνά καπάκια από ξύλο πάχους 2 cm. Στο αριστερό πλαϊνό καπάκι κάναμε τρύπες για να ταιριάξουμε την υποδοχή midi out , την υποδοχή της τροφοδοσίας όπως επίσης και τον διακόπτη on/off.


ON/OFF
MIDI OUT
LED
DC IN



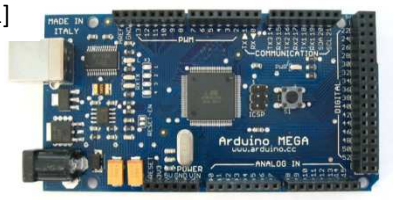
Στον ξύλινο πάτο του κουτιού στηρίξαμε την arduino πλακέτα. Ενώ στα δύο πλαϊνά κολλήσαμε τα LED (βλέπε Φώτο).

❖ ***Κύκλωμα***

Για την κατασκευή του κυκλώματος χρησιμοποιήσαμε τα παρακάτω ηλεκτρονικά εξαρτήματα:

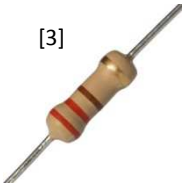
- Μια πλακέτα arduino Mega [1]
- 16 ποτενσιόμετρα γραμμικά 4,7KΩ (μοντέλο: COM-09941, από την εταιρία SparkFun) [2]
- 3 αντιστάσεις 220Ω [3]
- 2 LED 2,5V μπλε χρώματος [4]
- 2 μέτρα τηλεφωνικού καλωδίου για τις εσωτερικές συνδέσεις [5]
- Μια υποδοχή midi-out [6]
- Μια υποδοχή τροφοδοσίας [7]
- Έναν διακόπτη on/off [8]
- Δύο τεμάχια connectors με 40 pins [9]

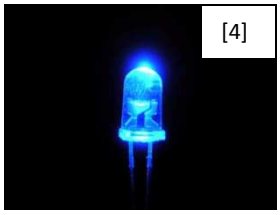
[1]

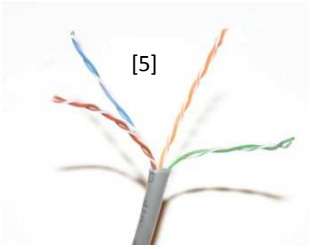


[2]

[3]



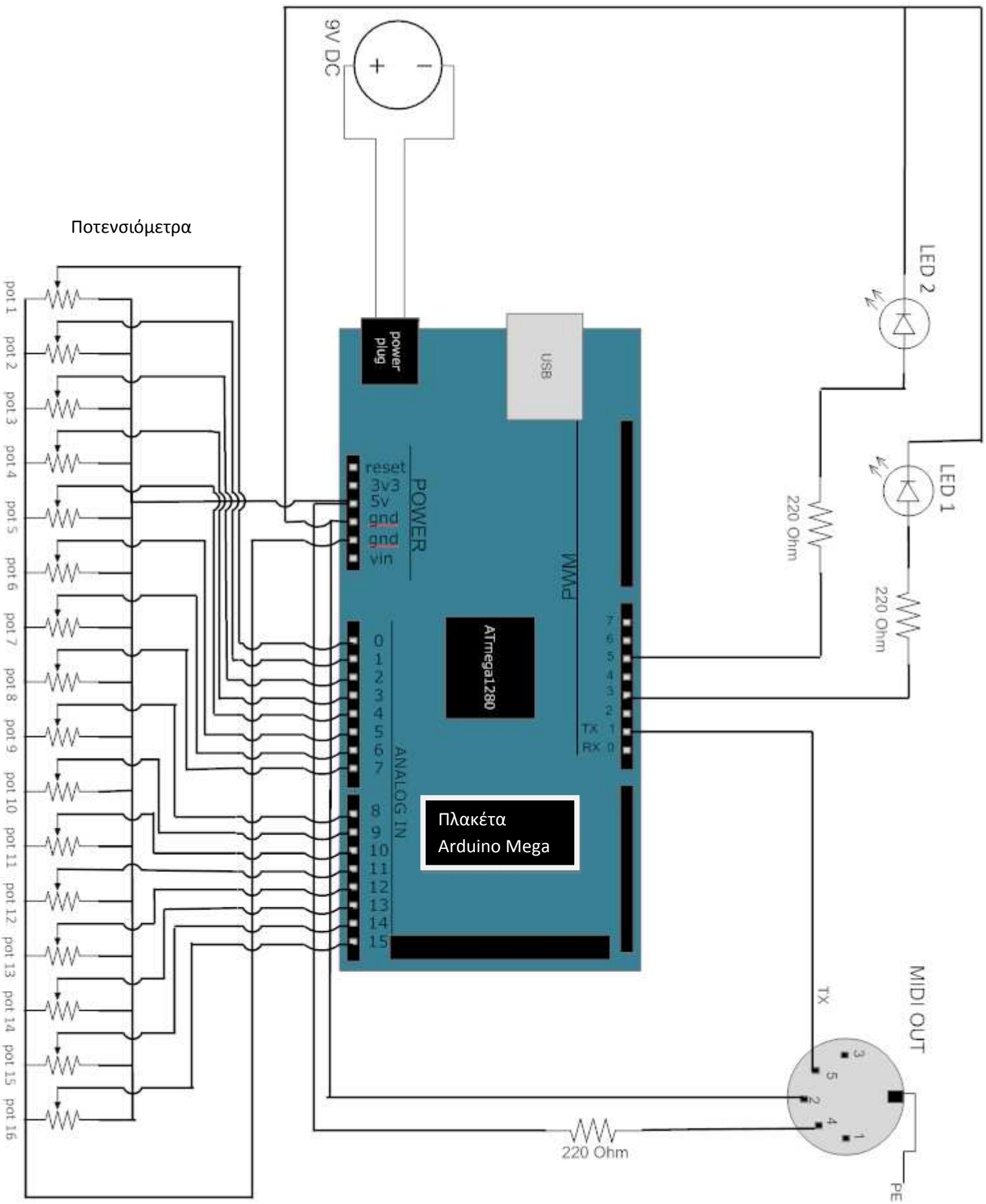




[7]

[8]

[9]

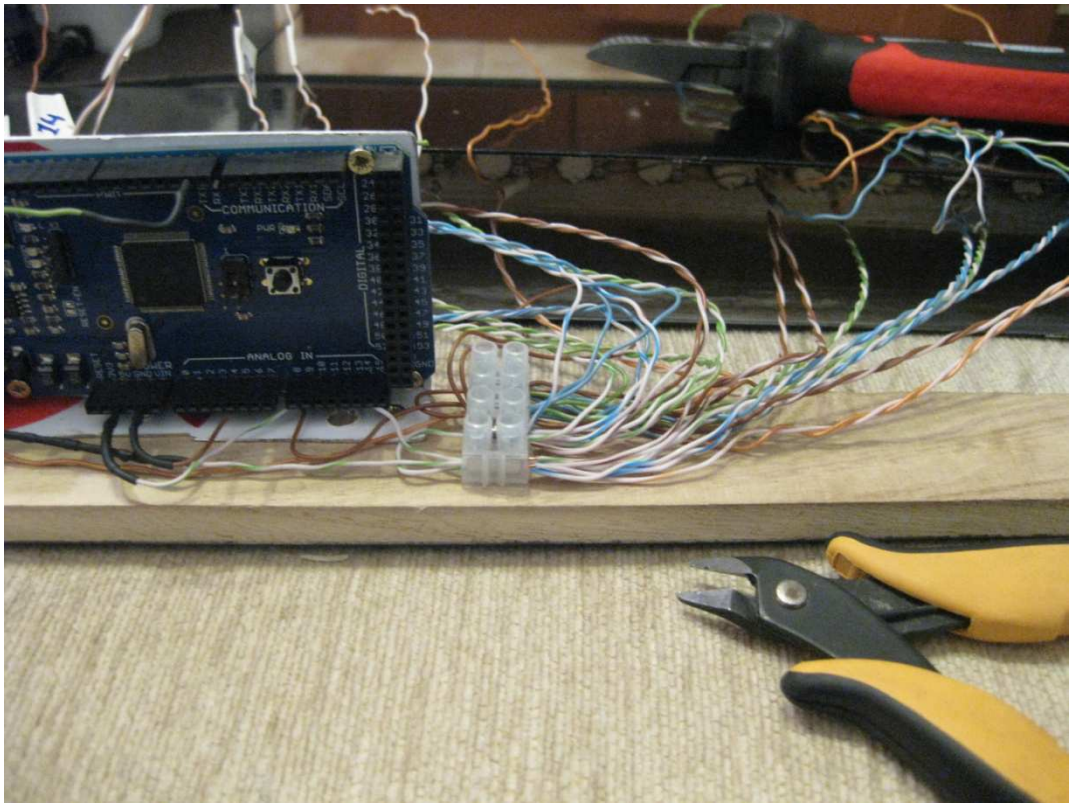

Ποτενσιόμετρα
9V DC
power plug
USB
LED 2
LED 1
220 Ohm
220 Ohm
220 Ohm
pot 1
pot 2
pot 3
pot 4
pot 5
pot 6
pot 7
pot 8
pot 9
pot 10
pot 11
pot 12
pot 13
pot 14
pot 15
pot 16
reset
3v3
5v
gnd
gnd
vin
POWER
ANALOG IN
PWM
ATmega1280
TX
RX
Πλακέτα
Arduino Mega
MIDI OUT
TX
PE

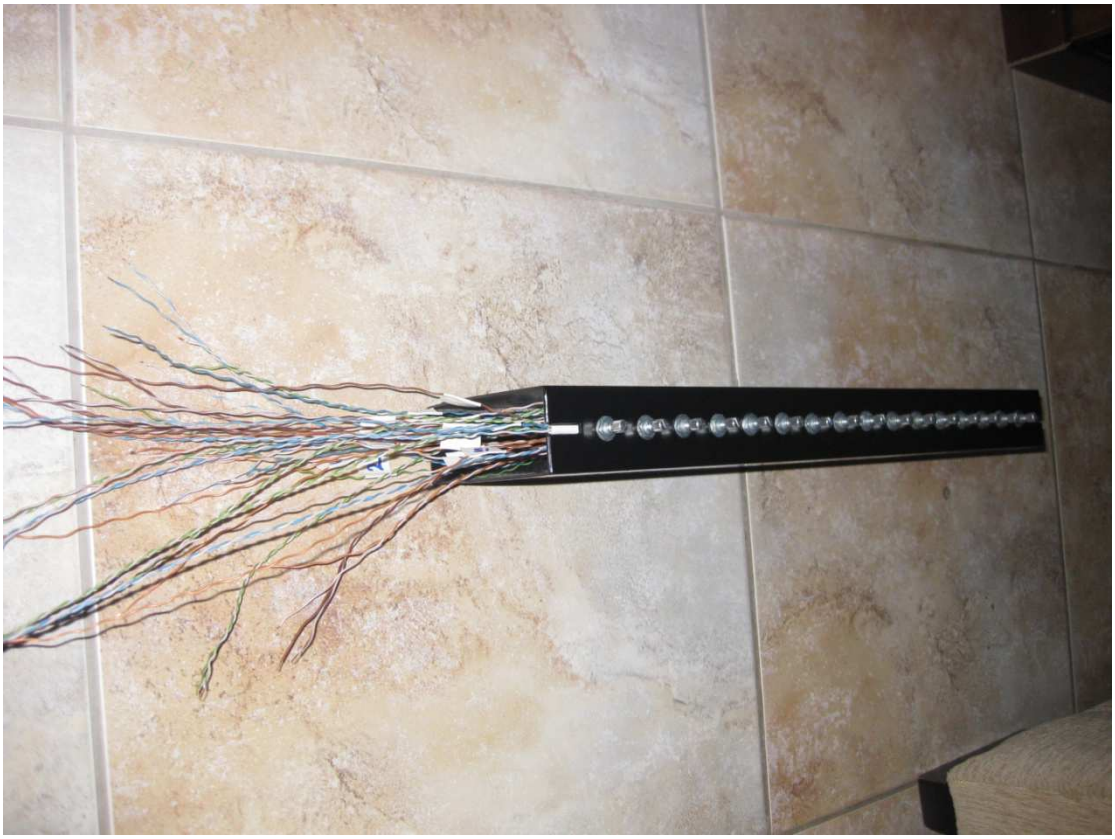
Κύκλωμα

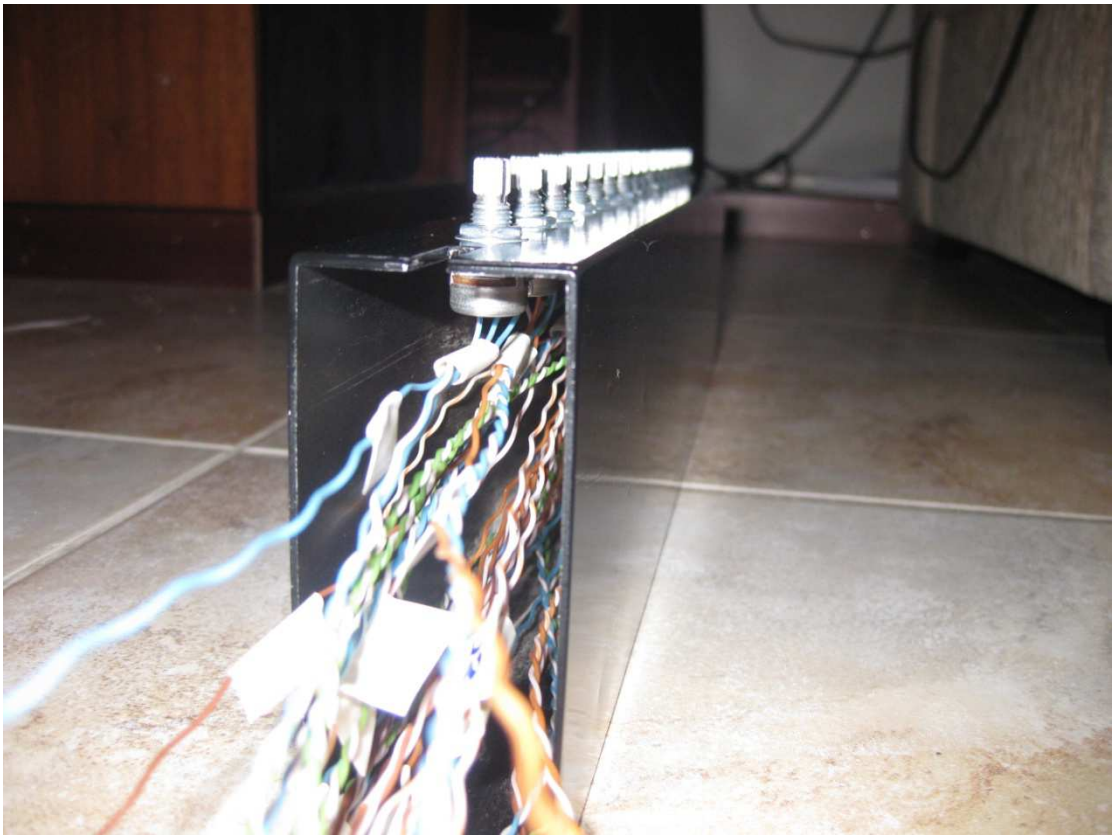
Στο παραπάνω σχεδιάγραμμα φαίνεται η συνδεσμολογία του κυκλώματος. Η πλακέτα τροφοδοτείται με 9V μέσω ενός AC-DC αντάπτορα στην ειδική υποδοχή. Τα ποτενσιόμετρα τροφοδοτούνται με 5V μέσω του 5V pin της πλακέτας. Η έξοδος του κάθε pot πηγαίνει στην αντίστοιχη υποδοχή analog in της πλακέτας (το pot 1 συνδέεται στο pin 0, το pot 2 συνδέεται στο pin 1, το pot 3 συνδέεται στο pin 2, κτλ.). Κάθε ένα από τα pin 0-15 (analog-in) συνδέεται με τον ενσωματωμένο ADC (analog to digital) μετατροπέα ο οποίος μετατρέπει την μεταβολή τάσης από τα pot σε 1024 διακριτές τιμές (ανάλυση 10 bit) τις οποίες μπορεί να διαβάσει ο μικροελεγκτής της πλακέτας.

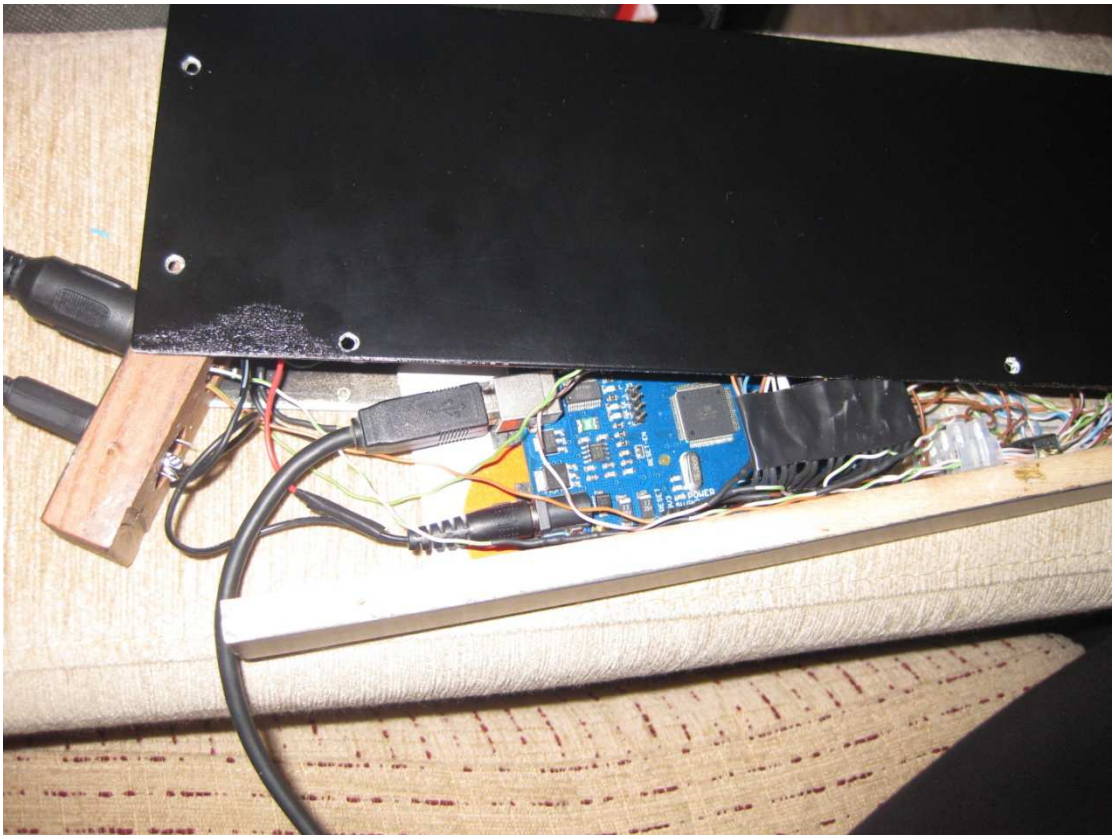
Ο μικροελεγκτής (Atmega 1280) αφού διαβάσει την τιμή του κάθε pot δημιουργεί Control Change midi μηνύματα (127 διακριτών τιμών) τα οποία στέλνει στο pin 5 του midi-out μέσω της σειριακής θύρας TX 1(UART). Το pin 2 του midi-out συνδέεται στην γείωση ενώ το pin 4 συνδέεται στο pin των 5V της πλακέτας μέσω μιας αντίστασης 220Ω (σύμφωνα με τις προδιαγραφές του MIDI).

Επίσης την στιγμή που γυρνάμε κάποιο pot ο μικροελεγκτής στέλνει την τιμή του pot αυτού(σε 127 διακριτές τιμές) στο LED 1 μέσω του pin 3 (PWM). Δηλαδή όσο περισσότερο γυρνάμε το pot τόσο περισσότερο ανάβει το LED 1. Ενώ όταν γυρίσουμε ένα pot στην μέση (κέντρο, δηλαδή midi τιμή 64) ο μικροελεγκτής στέλνει την τιμή 127 (μέγιστο) στο LED 2 μέσω του pin 5 (PWM) . Έτσι όταν γυρίσουμε κάποιο pot στην μέση ανάβει το LED 2. Στην ουσία τα PWM pins έχουν ενσωματωμένους DCA (digital to analog) μετατροπείς οι οποίοι μετατρέπουν διακριτές τιμές (0-127) σε ανάλογη μεταβολή τάσης (Pulse Width Modulation ή PWM). Παρακάτω φαίνονται κάποιες φωτώ της διαδικασίας της κατασκευής.









❖ ***Πρόγραμμα***

Για να προγραμματίσουμε τον μικροελεγκτή συνδέουμε την arduino πλακέτα με τον υπολογιστή μέσω θύρας USB. Ο προγραμματισμός γίνεται με το ειδικό πρόγραμμα του arduino με την γλώσσα Wiring που είναι παρόμοια με την C++, απλά με κάποιες προσθήκες.

Ο τελικός κώδικάς στη γλώσσα Wiring είναι αρκετά απλός και φαίνεται παρακάτω:

```
void setup()
 {
  Serial.begin(31250);          // to baud rate twn Midi serial ports
  analogReference(5);
 }

int iAn0_Val, iAn1_Val, iAn2_Val, iAn3_Val, iAn4_Val, iAn5_Val, iAn6_Val, iAn7_Val, iAn8_Val,
iAn9_Val, iAn10_Val, iAn11_Val, iAn12_Val, iAn13_Val, iAn14_Val, iAn15_Val;

void loop()
 {
  //Pot[1]
  int iAn0_Val_Prev = iAn0_Val;    //Pernoume thn prohgoumenh timh tou pot[1]
  iAn0_Val = analogRead(0)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(1, iAn0_Val, iAn0_Val_Prev); //TX me thn timh tou pot[1]


  //Pot[2]
  int iAn1_Val_Prev = iAn1_Val;    //Pernoume thn prohgoumenh timh tou pot[2]
  iAn1_Val = analogRead(1)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(2, iAn1_Val, iAn1_Val_Prev); //TX me thn timh tou pot[2]


  //Pot[3]
  int iAn2_Val_Prev = iAn2_Val;     //Pernoume thn prohgoumenh timh tou pot[3]
  iAn2_Val = analogRead(2)/8;      //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(3, iAn2_Val, iAn2_Val_Prev); //TX me thn timh tou pot[3]


  //Pot[4]
  int iAn3_Val_Prev = iAn3_Val;    //Pernoume thn prohgoumenh timh tou pot[4]
  iAn3_Val = analogRead(3)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(4, iAn3_Val, iAn3_Val_Prev); //TX me thn timh tou pot[4]


  //Pot[5]
  int iAn4_Val_Prev = iAn4_Val;    //Pernoume thn prohgoumenh timh tou pot[5]
  iAn4_Val = analogRead(4)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(5, iAn4_Val, iAn4_Val_Prev); //TX me thn timh tou pot[5]
```

```
//Pot[6]
int iAn5_Val_Prev = iAn5_Val;    //Pernoume thn prohgoumenh timh tou pot[6]
iAn5_Val = analogRead(5)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(6, iAn5_Val, iAn5_Val_Prev); //TX me thn timh tou pot[6]


//Pot[7]
int iAn6_Val_Prev = iAn6_Val;    //Pernoume thn prohgoumenh timh tou pot[7]
iAn6_Val = analogRead(6)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(7, iAn6_Val, iAn6_Val_Prev); //TX me thn timh tou pot[7]


//Pot[8]
int iAn7_Val_Prev = iAn7_Val;    //Pernoume thn prohgoumenh timh tou pot[8]
iAn7_Val = analogRead(7)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(8, iAn7_Val, iAn7_Val_Prev); //TX me thn timh tou pot[8]


//Pot[9]
int iAn8_Val_Prev = iAn8_Val;    //Pernoume thn prohgoumenh timh tou pot[9]
iAn8_Val = analogRead(8)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(9, iAn8_Val, iAn8_Val_Prev); //TX me thn timh tou pot[9]


//Pot[10]
int iAn9_Val_Prev = iAn9_Val;    //Pernoume thn prohgoumenh timh tou pot[10]
iAn9_Val = analogRead(9)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(10, iAn9_Val, iAn9_Val_Prev); //TX me thn timh tou pot[10]


//Pot[11]
int iAn10_Val_Prev = iAn10_Val;    //Pernoume thn prohgoumenh timh tou pot[11]
iAn10_Val = analogRead(10)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(11, iAn10_Val, iAn10_Val_Prev); //TX me thn timh tou pot[11]


//Pot[12]
int iAn11_Val_Prev = iAn11_Val;    //Pernoume thn prohgoumenh timh tou pot[12]
iAn11_Val = analogRead(11)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(12, iAn11_Val, iAn11_Val_Prev); //TX me thn timh tou pot[12]


//Pot[13]
int iAn12_Val_Prev = iAn12_Val;    //Pernoume thn prohgoumenh timh tou pot[13]
iAn12_Val = analogRead(12)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
analogPinMidiTX(13, iAn12_Val, iAn12_Val_Prev); //TX me thn timh tou pot[13]
```

```
  //Pot[14]
  int iAn13_Val_Prev = iAn13_Val;    //Pernoume thn prohgoumenh timh tou pot[14]
  iAn13_Val = analogRead(13)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(14, iAn13_Val, iAn13_Val_Prev); //TX me thn timh tou pot[14]


  //Pot[15]
  int iAn14_Val_Prev = iAn14_Val;    //Pernoume thn prohgoumenh timh tou pot[15]
  iAn14_Val = analogRead(14)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(15, iAn14_Val, iAn14_Val_Prev); //TX me thn timh tou pot[15]


  //Pot[16]
  int iAn15_Val_Prev = iAn15_Val;    //Pernoume thn prohgoumenh timh tou pot[16]
  iAn15_Val = analogRead(15)/8;     //Dieroume me to 8 gia na paroume times apo 0 eos 127
  analogPinMidiTX(16, iAn15_Val, iAn15_Val_Prev); //TX me thn timh tou pot[16]

  }

 void analogPinMidiTX(int iChan, int iVal, int iVal_Prev)

 {
 if(iVal_Prev != iVal)     //Medadosh tou mhnhmatos mono an an h timh tou pot exei alaksei gia na
mhn yperfortonoume thn midi thhra
 {
  MidiTX(176, iChan, iVal);  //176 = Control/Mode Change mhnhma [kanali 1]
  analogWrite(3, iVal);
  if(iVal == 64)
  {
   analogWrite(5, 127);
  }
   else
    {
    analogWrite(5, 0);
    }
 }
 }

void MidiTX(unsigned char MESSAGE, unsigned char CONTROL, unsigned char VALUE) //Stelnoume
thn timh tou pot meso midi mhnhmatos
 {
  Serial.print(MESSAGE);
  Serial.print(CONTROL);
  Serial.print(VALUE);
 }
```

Ξεκινώντας το πρόγραμμα έχουμε την εντολή setup (void setup()). Η εντολή setup() μπαίνει στην αρχή κάθε προγράμματος και χρησιμοποιείται για την αρχικοποίηση των μεταβλητών, των pins, βιβλιοθηκών κτλ. Η εντολή setup εκτελείται μόνο μια φορά όταν τίθεται σε λειτουργία η πλακέτα.

Ακολουθεί η εντολή Serial.Begin (31250bit/sec για το MIDI) η οποία δηλώνει την ταχύτητα με την οποία θα μεταδίδονται τα δεδομένα από την σειριακή έξοδο του μικροελεγκτή.

Ύστερα έχουμε την εντολή analogReference(5) η οποία χρησιμοποιείται για να δηλώσουμε την τάση αναφοράς για τις αναλογικές (ADC) εισόδους του μικροελεγκτή. Συγκεκριμένα βάζουμε την τιμή 5 γιατί τα ποτενσιόμετρα τροφοδοτούνται με 5V και συνεπώς η τάση στην έξοδο τους μεταβάλλεται από 0 έως 5V.

Ακολουθεί η δήλωση μεταβλητών χρησιμοποιώντας την εντολή int. στην οποία έχουμε βάλει όλες τις μεταβλητές που θα χρησιμοποιήσουμε.

Ακολουθεί η εντολή void loop () η οποία όπως δηλώνει και το όνομα της είναι ένας βρόγχος επανάληψης (λούπα) και εκτελεί επανειλημμένα τις εντολές που περιέχονται στις αγκύλες που ακολουθούν.

Μέσα στην εντολή void loop () διαβάζουμε ξεχωριστά την τιμή του κάθε pot. Για παράδειγμα για το pot 1 έχουμε:

την εντολή **int iAn0_Val_Prev = iAn0_Val;** με την οποία αποθηκεύουμε την προηγούμενη τιμή του pot στην μεταβλητή **iAn0_Val_Prev.**

Μετά έχουμε την εντολή **iAn0_Val = analogRead(0)/8;** με την οποία διαβάζουμε την τιμή του pot1 (analog pin0) μέσω της εντολής **analogRead(0)** , η οποία αποθηκεύει την τιμή της τάσης του analog pin0 στη μεταβλητή **iAn0_Val** αφού την μετατρέψει σε ψηφιακή ακολουθία 10bit δηλαδή 1024 διακριτές τιμές. Ωστόσο για το midi χρειαζόμαστε 128 διακριτές τιμές (από 0 εως 127) και γιαυτό διαιρούμε την τιμή που μας δίνει η analogRead με το 8. Το αποτέλεσμα το αποθηκεύουμε στην μεταβλητή **iAn0_Val** . Υστερα έχουμε την εντολή **analogPinMidiTX(1, iAn0_Val, iAn0_Val_Prev)** με την οποία αποθηκεύουμε τις τιμές 1 (που είναι ο αριθμός του controller για τον pot1), **iAn0_Val** και **iAn0_Val_Prev** στην μεταβλητή **analogPinMidiTX**. Ιδια διαδικασία έχουμε και για τα υπόλοιπα pot.

Αφού διαβάσουμε τις τιμές και των 16 pot τις αποθηκεύουμε όλες στην μεταβλητή **analogPinMidiTX** . Όμως επειδή θέλουμε τον μικροελεγκτή να στέλνει midi μηνύματα μόνο όταν μεταβάλλεται η τιμή κάποιου pot φτιάχνουμε μια «λούπα» στην οποία θα ελέγχουμε αν η προηγούμενη τιμή του pot είναι ίδια με την παρούσα τιμή. Αυτό γίνεται με την εντολή **if(iVal_Prev != iVal)** .

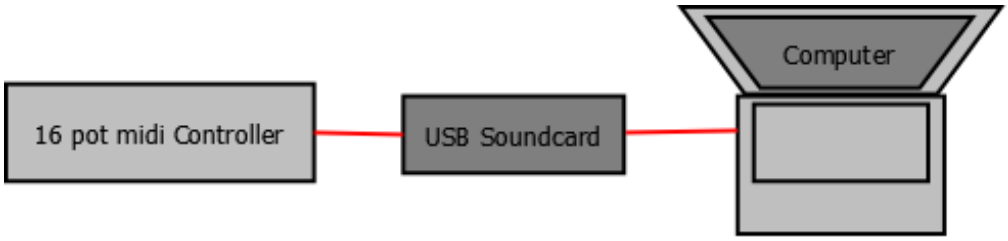
Αν οι δύο τιμές είναι διαφορετικές τότε και μόνο τότε θα «δημιουργείται» midi μήνυμα με την εντολή **MidiTX(176, iChan, iVal)** με την οποία αποθηκεύουμε τις τιμές midi στην μεταβλητή MidiTX. Όπου το 176 αν μετατραπεί σε δυαδική μορφή δηλώνει Control Change μήνυμα με αριθμό καναλιού 1, iChan είναι ο αριθμός του controller pot και iVal είναι η τιμή του controller pot. Επίσης μέσα στον βρόγχο (λούπα) 'if' έχουμε την εντολή **analogWrite(3, iVal)** με την οποία στέλνουμε την τιμή του pot στο pin3 (PWM) που συνδέεται με το LED 1(βλέπε κύκλωμα). Μέσα στον βρόγχο έχουμε και άλλη «λούπα» που ελέγχει αν το pot βρίσκεται στην μέση (**if(iVal == 64)**) και αν ναι στέλνουμε την τιμή 127 (μέγιστο) στο pin5 που είναι συνδεδεμένο με το LED 2 ώστε να ανάβει. Αλλιώς στέλνεται η τιμή 0 και το LED 2 μένει σβηστό .

Τέλος με την εντολή **void MidiTX(unsigned char MESSAGE, unsigned char CONTROL, unsigned char VALUE)** μετατρέπουμε τις τιμές 176, iChan, iVal σε δυαδική μορφή (με τιμές από 0 εως 255) με τις ονομασίες message, control, value αντίστοιχα και τις στέλνουμε με την σειρά στην σειριακή έξοδο του μικροελεγκτή με τις εντολές **Serial.print(MESSAGE); Serial.print(CONTROL); Serial.print(VALUE);** . Αυτό που βγαίνει από την σειριακή έξοδο είναι πλέον Control Change MIDI μηνύματα τα οποία μπορούν να ελέγχουν οποιαδήποτε συσκευή που υποστηρίζει MIDI.

Σύνδεση και Λειτουργία

Με την ολοκλήρωση και του κώδικα και αφού τον αντιγράψουμε στην μνήμη του μικροελεγκτή ο Midi Controller μας είναι πλέον έτοιμος και μπορούμε να τον χρησιμοποιήσουμε.

Μπορούμε για παράδειγμα να συνδέσουμε τον controller με τον υπολογιστή για να ελέγξουμε κάποιο μουσικό πρόγραμμα, όπως το ableton live (πρόγραμμα που απευθύνεται κυρίως σε dj και μουσικούς παραγωγούς). Ωστόσο επειδή οι περισσότεροι υπολογιστές δεν έχουν on-board MIDI υποστήριξη , θα χρειαστούμε κάποιο hardware όπως μια εξωτερική κάρτα ήχου που έχει MIDI θύρες. Έτσι συνδέουμε την έξοδο (midi out) του controller μας με την είσοδο (midi in) της κάρτας ήχου ,ενώ η κάρτα ήχου συνδέεται με τον υπολογιστή μέσω USB.



Συγκεκριμένα η κάρτα ήχου που χρησιμοποιήσαμε είναι η Audio 8 DJ της εταιρίας Native Instruments η οποία έχει 2 midi θύρες: μια MIDI IN και μια MIDI OUT. Έτσι ανοίγοντας το ableton και επιλέγοντας για midi in την είσοδο της Audio 8 DJ μπορούμε πλέον να ελέγχουμε διάφορες παραμέτρους του προγράμματος. Ιδιαίτερα τα ποτενσιόμετρα είναι βολικά για τον έλεγχο παραμέτρων όπως το Equalizer, το Pan, το Filter αλλά και άλλες. Γενικά οι δυνατότητες ελέγχου που προσφέρει το MIDI είναι σχεδόν απεριόριστες. Έτσι αφού δοκιμάσαμε την συσκευή μας και βλέποντας ότι όλα λειτουργούν κανονικά μπορούμε να πούμε πως κάπου εδώ η πτυχιακή μας εργασία έχει ολοκληρωθεί.

Συμπεράσματα

Η διαδικασία της κατασκευής κράτησε περίπου 3 με 4 μήνες και κατά την διάρκεια της συναντήσαμε κάποιες δυσκολίες όπως είναι φυσιολογικό σε τέτοιου είδους εργασίες. Όσων αφορά την κατασκευή του κουτιού μπορούμε να πούμε ότι ήταν αρκετά χρονοβόρα διαδικασία. Μας πήρε περίπου 1 με 2 εβδομάδες μόνο για να σκεφτούμε το κατάλληλο υλικό κατασκευής καθώς στην αρχή δοκιμάσαμε να φτιάξουμε το κουτί από ξύλο. Ωστόσο το μειονέκτημα του ξύλινου κουτιού είναι το βάρος του και το ότι η κατασκευή θα γινόταν ποιο ογκώδης και λιγότερο συμπαγής. Τελικά αποφασίσαμε να δοκιμάσουμε αλουμίνιο για υλικό κατασκευής κάτι που ήταν και η σωστή κίνηση, καθώς το αλουμίνιο είναι πολύ πιο συμπαγές και πιο λεπτό απ ότι το ξύλο. Έτσι προχωρήσαμε στην κατασκευή του κουτιού από αλουμίνιο. Ευτυχώς για εμάς είχαμε όλα τα απαραίτητα εργαλεία για την κατασκευή όπως γωνιακοί τροχοί, τρυπάνια, πάγκο κοπής, κολλητήρια, πολύμετρα και άλλα.

Όσων αφορά το κύκλωμα αντιμετωπίσαμε κάποιες δυσκολίες με τα ποτενσιόμετρα. Αυτό γιατί στην αρχή είχαμε συνδέσει τα pot σε σειρά και το πρόβλημα ήταν πως η τάση στην έξοδο των pot έμοιαζε να ταλαντεύεται με αποτέλεσμα ο μικροελεγκτής να στέλνει συνεχώς midi μηνύματα αχρηστεύοντας έτσι την λειτουργία του controller. Το πρόβλημα λύθηκε όταν, μετά από πολύ ταλαιπωρία και σκέψη, συνδέσαμε τα pot παράλληλα μεταξύ τους και πλέον το κύκλωμα λειτουργούσε άψογα χωρίς κανένα απολύτως πρόβλημα. Βεβαίως συναντήσαμε και άλλα μικροπροβλήματα αλλά δεν χρειάζεται να αναφερθούν εδώ.

Όσων αφορά το κόστος τις κατασκευής , κυμαίνεται περίπου στα 60 με 70 ευρώ με τα οποία κατασκευάσαμε έναν πολύ καλής ποιότητας midi controller και αρκετά φθηνότερο από αντίστοιχους controllers της αγοράς (συνήθως πάνω από 100 ευρώ).

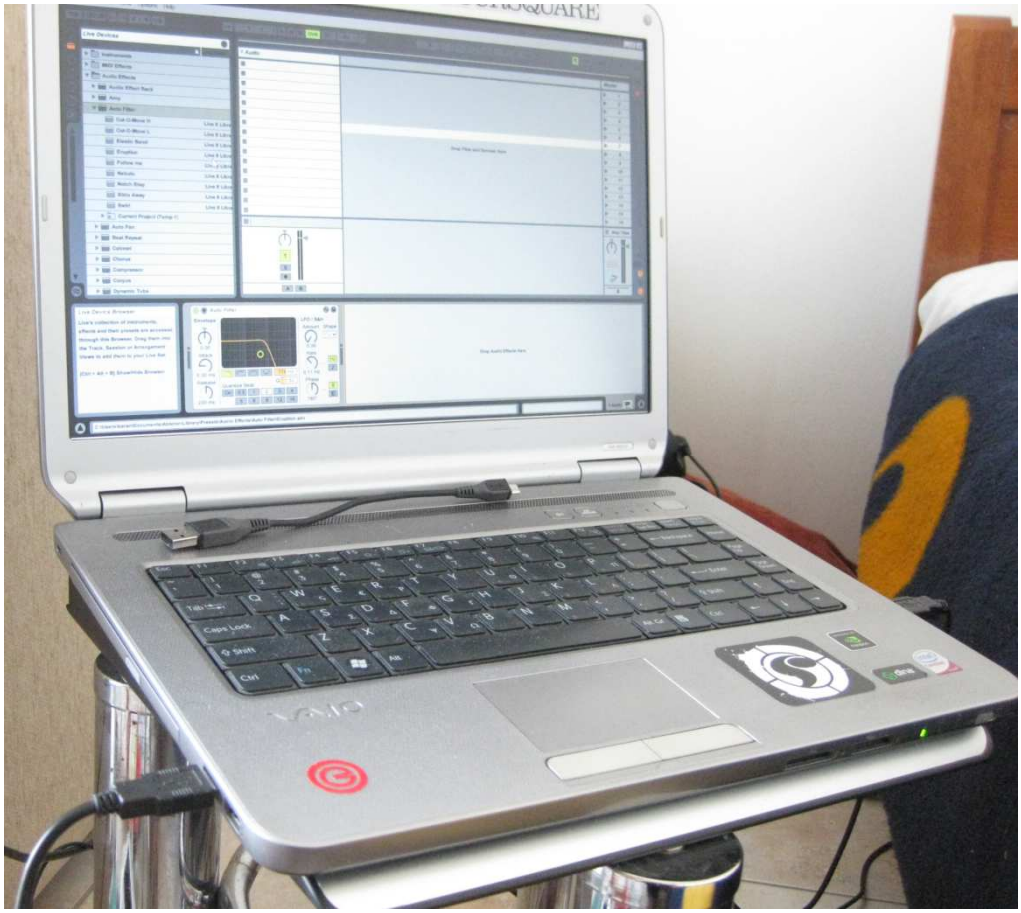
Επίσης θα μπορούσαμε στο μέλλον να βελτιώσουμε τον controller μας ανεβάζοντας ίσως την ανάλυση από 7bit σε 10bit . Έτσι η θέση του κάθε pot θα κωδικοποιείται σε 1024 διακριτές τιμές αντί για 128 που είναι τώρα. Αυτό θα έδινε πολύ μεγαλύτερη ακρίβεια στον έλεγχο και υποστηρίζεται από το πρωτόκολλο midi με την πρόσθεση ενός ακόμη byte στο τέλος κάθε midi μηνύματος. Για την ακρίβεια το midi υποστηρίζει 16384 τιμές (14bit) για τα control change μηνύματα , αλλά ο Atmega1280 παρέχει ανάλυση 10bit στις αναλογικές (ADC) εισόδους. Ωστόσο για την ώρα η default ανάλυση 7bit είναι υπέρ αρκετή για τις περισσότερες εφαρμογές. Επίσης είναι δυνατόν στο

μέλλον να προστεθούν και άλλα εξαρτήματα στον controller όπως: κουμπιά, προσθήκη layers (pages, δηλαδή με το πάτημα ενός κουμπιού να έχουμε 16 επιπλέων pot με controller number απο17 έως 32) , οθόνες, faters και άλλα.

Γενικά κατά την διαδικασία της κατασκευής αποκομίσαμε πολύτιμες εμπειρίες καθώς μας έδωσε την δυνατότητα να πειραματιστούμε πάνω στο MIDI και τις δυνατότητες του αλλά και στο κομμάτι του προγραμματισμού μικροελεγκτών που είναι ένας εξαιρετικά ενδιαφέρον κλάδος που προσφέρει άπειρες δυνατότητες ως προς το τι μπορεί να φτιάξει κάποιος.

ΤΕΛΟΣ


MIC/LINE
IN 5|6
IN 7|8
LINE
CH. D
INPUT MODE
VOLUME
HEADPHONES



ΒΙΒΛΙΟΓΡΑΦΙΑ & SITES

- http://en.wikipedia.org
- The MIDI Manual Third Edition , David Miles Huber
- http://arduino.cc
- http://sound.westhost.com/pots.htm
- http://www.midi.org/
- http://arduino.cc/forum/
- http://www.avrfreaks.net/
- http://www.atmel.com/