

HPY 411 – PINBALL

Αναφορά Project

Πολυτεχνείο 21/12/2017

Βρουβάκης Γιάννης : 2014030122

Φελέκης Παναγιώτης : 2014030198



ΠΕΡΙΕΧΟΜΕΝΑ

1.1	Περιγραφή Project.....	page 3
1.2	Κατανομή Εργασίας και δημιουργία χρονοδιαγράμματος.....	page 3
2.1	Κατασκευή Διαγραμμάτων.....	page 4
2.2	Work Breakdown Graph.....	page 5
2.3	Product flow.....	page 6
2.4	P.E.R.T. chart.....	page 7
2.5	Gantt chart.....	page 8
3.1	Υλικά και Περιγραφή.....	page 9
3.2	Microcontroller.....	page 10
3.3	Αισθητήρες.....	page 11
3.4	LEDs Display.....	page 13
4.1	Προγραμματισμός.....	page 14
4.2	Ενσωμάτωση.....	page 15
5.1	Προβλήματα και αντιμετώπιση.....	page 16
5.2	Μελλοντικές προσθήκες και αλλαγές.....	page 17
6.1	Βιβλιογραφία.....	page 17

1.1 Περιγραφή Project

Ως project στο μάθημα των ενσωματωμένων συστημάτων, επιλέξαμε την υλοποίηση και κατασκευή ενός κλασσικού pinball.

Το Pinball είναι ένα παιχνίδι ευρέως γνωστό από τη δεκαετία του 30'. Σκοπός του παιχνιδιού είναι να μαζεύσεις όσο το δυνατόν περισσότερους πόντους χτυπώντας πάνω σε εμπόδια, χωρίς να αφήσεις τη μπίλια να περάσει πέρα από τους βραγχίωνες που την εκτοξεύουν.

Για τη κατασκευή λοιπόν μιας τέτοιας μηχανής, απαραίτητη ήταν υλοποίηση ενός μετρητή score καθώς και διάφοροι αισθητήρες για να δίνουν τιμές στο παραπάνω μετρητή.

1.2 Κατανομή Εργασίας και δημιουργία χρονοδιαγράμματος

Από τα πρώτα βήματα για την υλοποίηση μας, ήταν ο καθορισμός milestones και η κατανομή εργασίας, τόσο για τις απαιτήσεις του μαθήματος, όσο και για την καλύτερη διαχείριση του χρόνου.

Αρχικά η κατανομή εργασίας ήταν κοινή, με την έννοια πως και τα 2 άτομα της ομάδας μας ασχολήθηκαν με τη κατανόηση των εργαλείων που μας είχαν δοθεί, ώστε να αποκτήσουμε το υπόβαθρο που χρειαζόταν για την αξιοποίηση κάθε δυνατότητας που μπορούσαν να προσφέρουν τα εργαλεία μας.

Όπως αναφέραμε προηγουμένως η κατασκευή του pinball απαιτεί ένα μετρητή καθώς και αισθητήρες, έτσι σε αυτό το σημείο έγινε η πρώτη κατανομή με την ανάθεση των αισθητήρων και την υλοποίησης ενός Score Display, στον καθένα αντίστοιχα.

Για τελευταίο μέρος, τη σύνδεση των περιφεριακών με το πλαίσιο και το μικροελεγκτή μας, χρειάστηκε να κατασκευάσουμε το πλαίσιο και στη συνέχεια να τοποθετήσουμε με λεπτομέρεια το κάθε υλικό πάνω.

Πέραν των επίσημων milestones, υπήρχαν ιδιωτικές συντήσεις για την πρόοδο της κατασκευής μας, επίσης κάθε ένα milestone χωριζόταν σε επιμέρους κομμάτια και ανατέθονταν εργασία αντίστοιχα σε κάθε μέλος.

Στο πρώτο απο αυτά τα meeting αποφασίσαμε τι πρόοδος πρέπει να έχει γίνει εντός του εξαμήνου. Χωρίσαμε το εξάμηνο σε 3 κομμάτια ίσης διάρκειας και θέσαμε εργασία. Συγκεκριμένα :

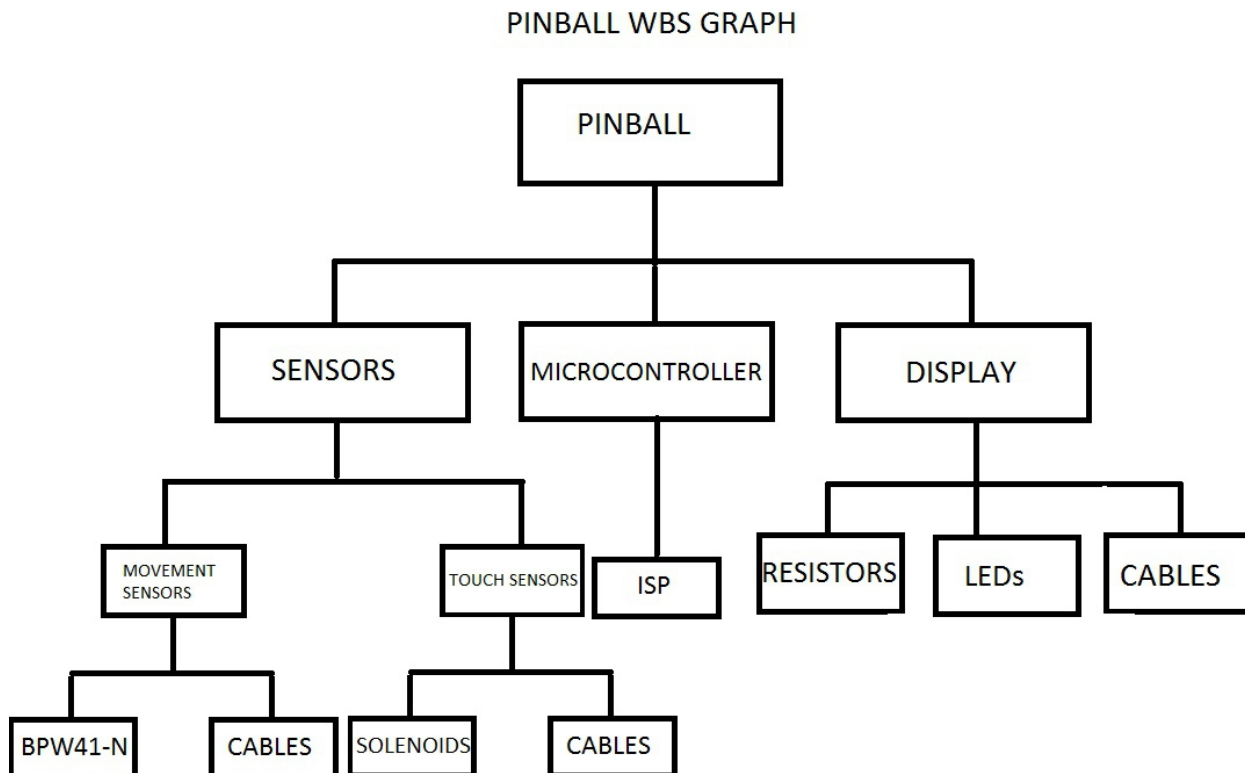
	Milestone 1	Milestone 2	Milestone 3
Dates	11/10/17	23/11/17	14/12/17
Description	Κατανόηση του avr&stk και προγραμματισμός	Κατανόηση και χρήση αισθητήρων και κατασκευή display	Κατασκευή πλαισίου, τοποθέτηση περιφεριακών και testing

2.1 Κατασκευή Διαγραμμάτων

Για την καλύτερη διαχείριση του project μας, κατασκευάσαμε διαγράμματα όπως work breakdown graph, product flow, PERT & Gantt charts. Με τη κατασκευή αυτών γινόταν ανάλυση απαιτήσεων αλλά και κατανομή χρόνου, ώστε να έχουμε συνεχώς υπό έλεγχο τη πρόοδο, να προγραμματίζουμε τα ρεπό μας και να υπάρχει μια καλύτερη και πιο οργανωμένη παρουσίαση.

2.2 Work Breakdown Graph

Με το work breakdown graph θα γινόταν η ανάλυση του project μας σε επιμέρους τμήμα, ώστε να γνωρίζουμε τι υλικά χρειάζεται να παραγγείλουμε ή και να κατασκευάσουμε.

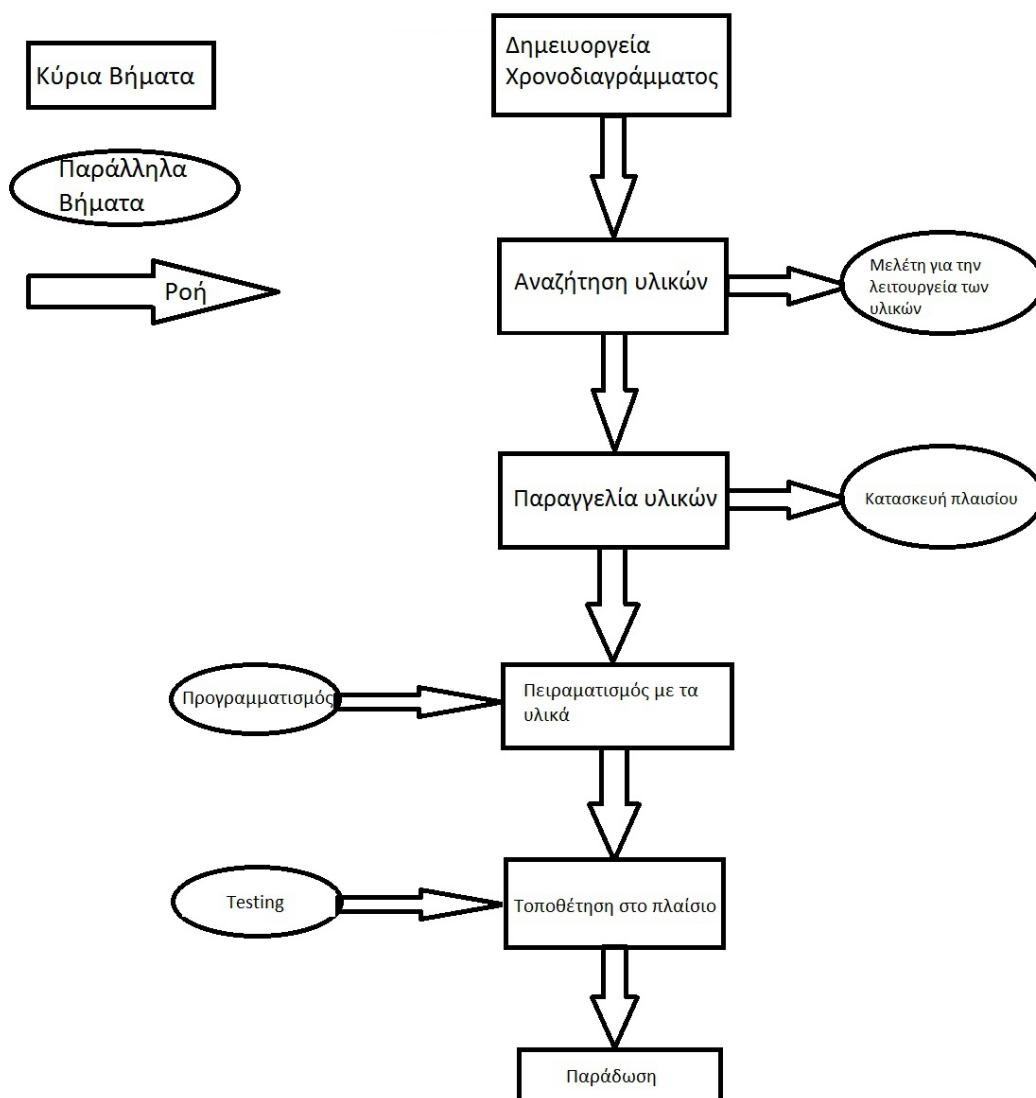


Με τη δημιουργία του παραπάνω διαγραμμάτος γνωρίζαμε εξ αρχής τα προβλήματα που θα αντιμετωπίζαμε με την εύρεση υλικού, έτσι κινηθήκαμε αντίστοιχα.

Έπειτα από ένα οικονομικό έλεγχο που πραγματοποιήσαμε διαπιστώσαμε πως η αγορά των Solenoids που χρειάζεται για τους αισθητήρες επαφής θα μας ανέβαζε σημαντικά το κοστολόγιο, έπειδή όμως θα θέλαμε να διατηρήσουμε το κόστος όσο το δυνατόν χαμηλότερα, μετά από μελέτη για τη λειτουργία τους, επιλέξαμε να τα αντικαταστήσουμε με δικούς μας αυτοσχέδιους αισθητήρες.

2.3 Product flow

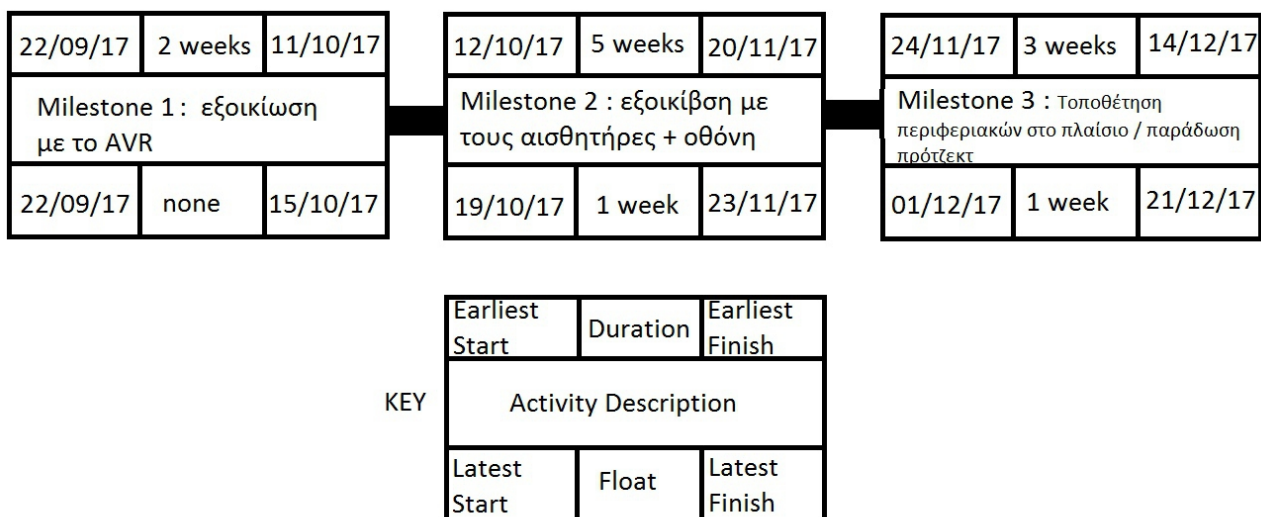
Με τη δημιουργία του product flow διατυπώσαμε τη σειρά που θα έπρεπε να έχουν τα γεγονότα καθώς και τις εξαρτήσεις μεταξύ αυτών. Όπως αναφέρει και το όνομα, ήταν η ροή για τη παραγωγή του project μας.



Με τη βοήθεια του παραπάνω διαγράμματος είχαμε συνεχώς γνώση για το που βρισκόμαστε και ποιά θα είναι τα επόμενα βήματα. Το διάγραμμα αυτό έπαιξε καθοριστικό ρόλο για το πόσο χρόνο θα αναθέταμε σε κάθε βήμα και πού θα χωρίζαμε το project για τη παράδοση των milestones.

2.4 PERT Chart (Program Evaluation and Review Technique)

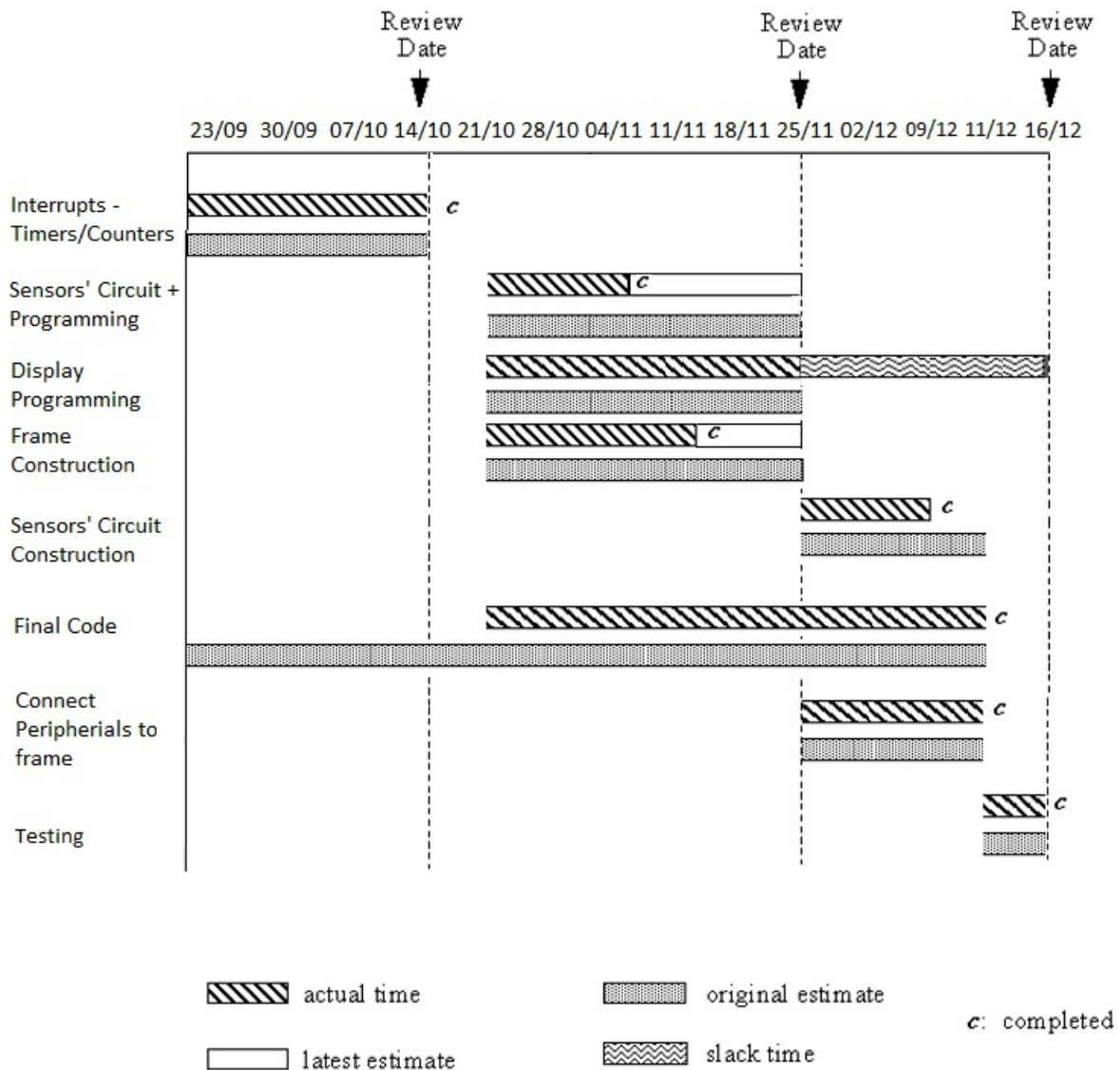
Το διάγραμμα PERT αποτελείται τη τεχνική με την οποία εργαστήκαμε πάνω στο χρονοδιάγραμμα, ώστε να μην έχουμε καθυστερήσεις τόσο στη παράδοση των milestones όσο και γενικότερα στην υλοποίησή μας.



Όπως φαίνεται, δηλώσαμε κανονικά τις ημερομηνίες παράδοσης και τη διάρκεια κάθε milestone καθώς συμπεριλάβαμε και τις καθυστερήσεις (διαλείμματα) μετά από κάθε παράδοση milestone.

2.5 Gantt chart

Αντίστοιχα με τα διαγράμματα PERT, έτσι τα διαγράμματα Gantt μας βοήθησαν στη κατανομή χρόνου-εργασίας. Με όμοιο τρόπο δηλώσαμε τα tasks που έπρεπε να υλοποιηθούν εντός των milestones. Με αυτό το τρόπο είχαμε μια καλύτερη αναπαράσταση των εργασιών που έπρεπε να ολοκληρωθούν σε μια λιγότερο λεπτομερή περιγραφή.



Το διάγραμμα ενημερωνόταν εβδομαδιαία με βάση τη πρόοδο που είχε πραγματοποιηθεί, με αυτό το τρόπο μπορούσαμε και είχαμε ένα ημερολόγιο με τις εργασίες μας, και στο τέλος του εξαμήνου μπορούσαμε να αναφέρουμε που παρουσιάστηκαν προβλήματα και κατά πόσο μας επηρέασαν.

3.1 Υλικά και Περιγραφή

Έπειτα από τα αποτελέσματα που πήραμε χάρη στο WBS graph καταλήξαμε στα υλικά μας. Για αρχή το εργαστήριο MHL μας υποστήριξε με τη χορήγηση ενός AVR microcontroller AT90s8515 που στη συνέχεια αντικαταστήθηκε από το Atmega8515 καθώς και το εργαλείο STK500, απαραίτητο για το προγραμματισμό του microcontroller.

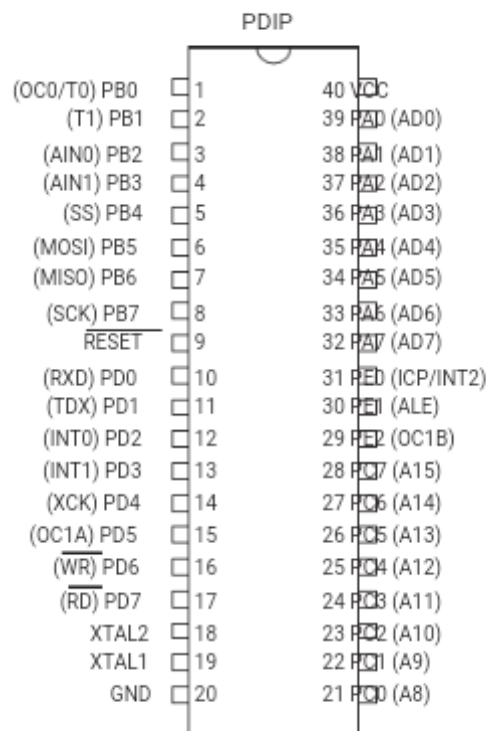
Έτσι από εμάς χρειάστηκε να αποφασίσουμε τους αισθητήρες που θα χρησιμοποιούσαμε. Αυτό που απαιτούσε η κατασκευή μας ήταν κάτι το οποίο θα μπορούσε να στείλει αντίστοιχο σήμα όταν αντιληφθεί κίνηση, τέτοιο χαρακτηριστικό αποτέλεσαι ο αισθητήρας hbrw421n. Στη συνέχεια θα επιθυμούσαμε ένα αισθητήρα ο οποίος θα έστελνε σήμα σε περίπτωση επαφής. Αρχικά εστιάσαμε σε solenoids, πρόκειται για πηνία τα οποία όταν νιώσουν επαφή εκτοξεύουν ένα πυράκι και το επαναφέρουν, όμως λόγω του μεγάλου κόστους που θα πρόσφεραν, καταλήξαμε στην επιλογή της κατασκευής δικών μας αυτοσχέδιων αισθητήρων που μέσα από επαφή θα μας έδιναν το σήμα που χρειαζόμασταν.

Από πλευρά hardware, το τελευταίο υλικό που θα έπρεπε να λάβει μέρος, ήταν εκείνο του display, μια αρχική σκέψη για τη χρήση μιας οθόνης LCD 2x16 δε παρέμεινε, καθώς επιλέξαμε τη χρήση LEDs σε σχηματισμό BCD (binary code decimal) για 2 ψηφία.

Έτσι φτάσαμε στο τέλος, με τα μοναδικά υλικά που να μένουν ακόμα να είναι εκείνα των καλωδίων για τις συνδέσεις αλλά και τα ξύλα των μηχανισμών και του πλαισίου.

3.2 Microcontroller

Για το πεδίο του μικροελεγκτή χρησιμοποιήσαμε τον Atmega8515, πρόκειται για ένα 8-bit microcontroller με αρχιτεκτονική RISC, περιέχει κοντά στις 130 εντολές με 32x8 καταχωρητές, λειτουργία κοντά στα 16MHz και τροφοδοσία στα 5 volt.



Βλέποντας παράνω το PINOUT του μικροελεγκτή μας παρατηρούμε πέραν των χαρακτηριστικών, πως διαθέτει 3 καταχωρητές για interrupts, όσα ακριβώς χρειαζόμαστε και θα εξηγηθούν παρακάτω λεπτομερώς.

Για τον επιτυχή προγραμματισμό του παραπάνω μικροελεγκτή χρησιμοποιήσαμε το STK500, το οποίο μέσα από σύνδεση με τον υπολογιστή με τη χρήση του πρωτοκόλλου RS232 και τη λειτουργία του προγράμματος Atmel Studio 7.0 δημιουργούσε ένα .hex file το οποίο φορτωνόταν στο microcontroller.

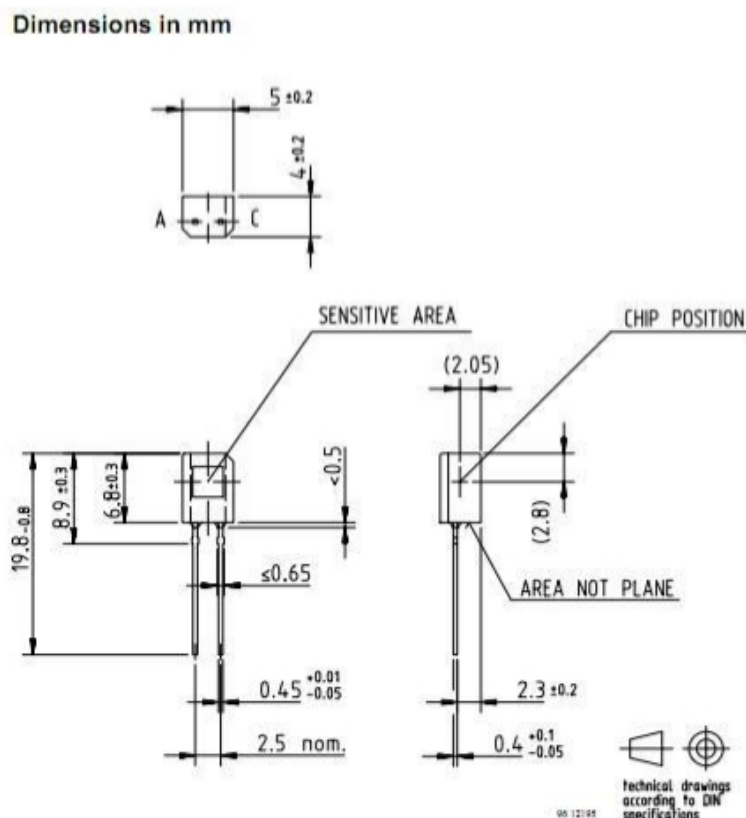
Αξιοσημείωτο είναι να δηλώσουμε πως το STK είχε αρνητική λογική.

3.3 Αισθητήρες

Όπως αναφέραμε προηγουμένως έγινε χρήση 2 ειδών αισθητήρων , εκείνους για ανίχνευση κίνησης και αυτούς για επαφή.

- BPW41n

Πρόκειται για ένα αισθητήρα ο οποίος λειτουργεί σαν μεταβλητή αντίσταση, της τάξης λίγων των kiloOhm ($k\Omega$) σε κατάσταση αδράνειας, όταν όμως ανιληφθεί κίνηση η αντίσταση σου εκτοξεύεται στις τάξεις των MegaOhm ($M\Omega$) με αποτέλεσμα να περιορίζει πλήρως το ρεύμα που το διαρρέει έχοντας ως έξοδο μια τάση κοντά στα 0 volt.

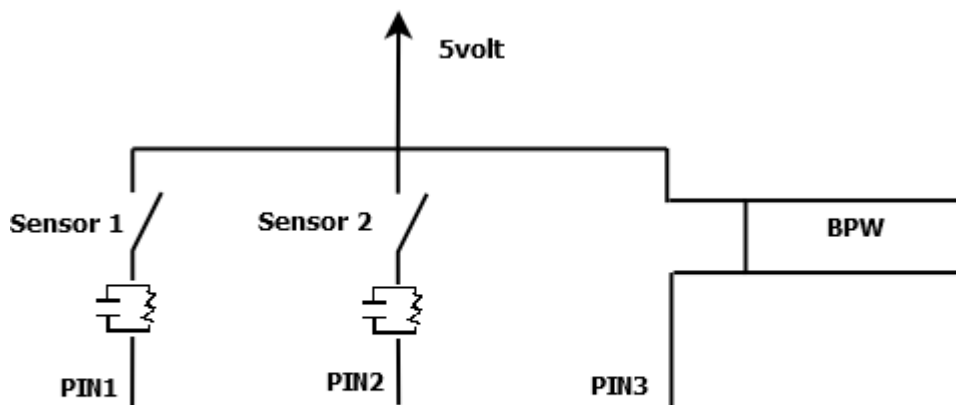


Στο παραπάνω διάγραμμα φαίνονται οι χαρακτηριστικές διαστάσεις καθώς και η περιοχή ευαισθησίας του αισθητήρα μας.

- **Αισθητήρας επαφής**

Αυτός ο αισθητήρας αποτελεί κομμάτι αυτοσχέδιας κατασκευής, αυτό που επιθυμούσαμε είναι όταν η μπίλια χτυπήσει είτε τα τρίγωνα είτε τα bumpers να έρχεται ένα αντίστοιχο σήμα για την ενημέρωση του score, συνεπώς, με το κύκλωμα που σχεδιάσαμε θέλαμε μέσω της σύγκρουσης να γίνεται μια επαφή – βραχυκύκλωμα ώστε η τάση από το ένα πόδι να μεταφέρεται στο άλλο, το οποίο είναι συνδεδεμένο σε ένα από τα pin εισόδου του μικροελεγκτή.

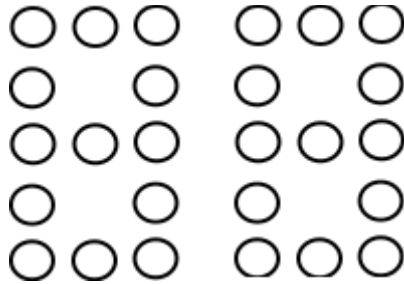
Με βάση τη κατασκευή μας η επαφή θα ήταν στιγμιαία, επομένως, με το να δίνουμε σταθερή τροφοδοσία στα 5 volt όταν θα πραγματοποιηθεί η ακαριαία επαφή θα περάσει το σήμα που επιθυμούμε.



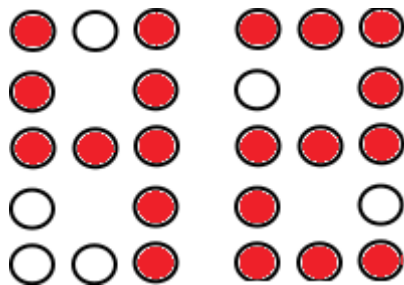
Παραπάνω φαίνεται το ολοκληρωμένο κύκλωμα των αισθητήρων στο πρότζεκτ μας. Όπως παρατηρούμε έχουμε τοποθετήσει RC για τη λειτουργία του διακόπτη, ώστε να μην έχουμε συνεχόμενη μετάφορά σήματος και να τη περιορίσουμε με τη χρήση της σταθεράς $\tau=RC$, για να μην μετρά συνεχώς πόντους στο score. Για τη σύνδεση του brw δε χρειάστηκε να υλοποιήσουμε αντίστοιχη λογική.

3.4 LEDs Display

Στο κομμάτι του Display σκεφτήκαμε να χρησιμοποιήσουμε 13 LEDs σε σχηματισμό BCD (binary code decimal) για την αναπαράσταση 2 ψηφίων. Η ιδέα ήταν να έχουμε το Score Display στη παρακάτω μορφή :



Έτσι ώστε όταν δεχόμαστε σήματα για το score μέσα από μία συνάρτηση που φτιάξαμε χωρίζει το score ανα ψηφίο και δίνει έξοδο στα κατάλληλα ports ώστε να ανάψουν τα αντίστοιχα LEDs, παράδειγμα οθόνης :



Τα LEDs που χρησιμοποιήσαμε δεχόντουσαν τροφοδοσία στα 2.3V έτσι έγινε χρήση αντίστοιχης αντίστασης ώστε να κατέβουμε στην επιθυμητή μας τάση.

4.1 Προγραμματισμός

Πλέον στο μέρος της υλοποίησης, το χωρίσαμε στο κομμάτι του προγραμματισμού και εκείνο της σχεδίασης – ενσωμάτωσης.

Ο προγραμματισμός αποτέλεσαι το πιο εκτενές έργο καθ'όλη τη διάρκεια του πρότζεκτ, καθώς ήδη από τις πρώτες μέρες ξεκινήσαμε να εργαζόμαστε πάνω σε αυτό, ωστόσο ο τελικός κώδικας δεν είναι αρκετά μεγάλος, ούτε περίπλοκος, όμως ως μέρος του μαθήματος ασχοληθήκαμε με σχεδόν όλες τις δυνατότητες που πρόσφερε ο μικροελεγκτής μας, ακόμα και αν δεν τις χρησιμοποιήσαμε στο τέλος.

Οι συναρτήσεις που υλοποιήθηκαν έχουν ως εξής :

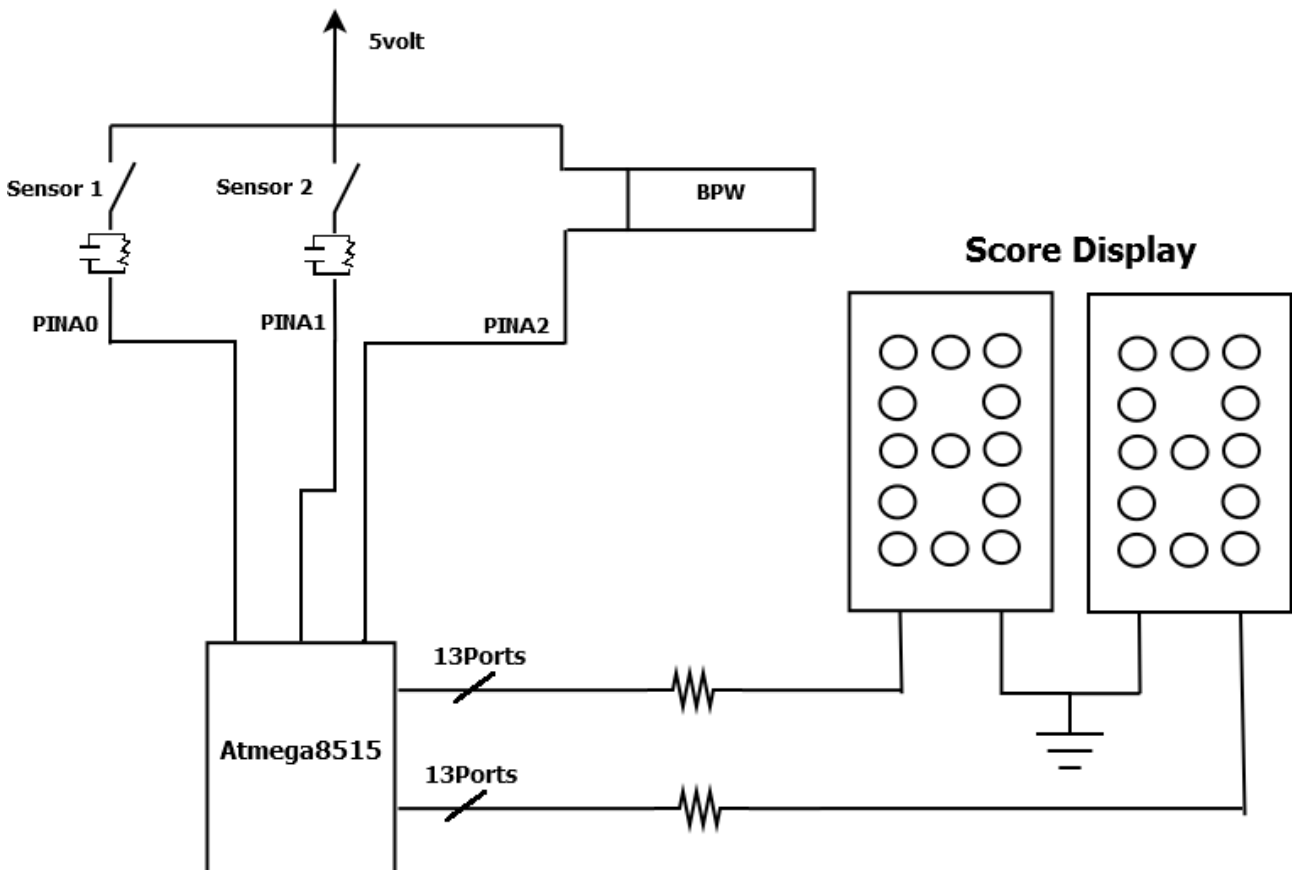
Interrupts, Initialize, Score Counter(ISR), BCD calculator, Checkloss.

- **Initialize** : Στη συνάρτηση αυτή γινόταν δήλωση των εισόδων/εξόδων του μικροελεγκτή αλλά και ενεργοποιούσαμε τα enables των interrupts
- **Interrupts** : Το κομμάτι των interrupts ήταν ιδιαίτερα ευνοϊκό, καθώς τα εμπόδια που είχαμε τοποθετήσει χωρίζονται σε 2 ειδών, και λαμβάνοντας υπ'όψη τον αισθητήρα για την ήττα, έχουμε τριών ειδών εισόδων, 1) τρίγωνα INT0(2 πόντοι) , 2) bumpers INT1(5 πόντοι) , 3) τρύπα INT2(ήττα).
- **Score Counter - ISRs** : Για το ISR(INT0_vect) είχαμε ως αποτέλεσμα το άθροισμα 2 πόντων στο score, αντίστοιχα για το ISR(INT1_vect) για 5 πόντους, και στη συνέχεια για το ISR(INT2_vect) μείωνε κατά ένα το counter για τις ζωές του παίκτη.
- **BCD calculator** : Έπαιρνε σαν όρισμα το score και χώριζε τα ψηφία του με τις εντολές $msb = score/10$, $lsb = score \% 10$, στη συνέχεια για το κάθε ψηφίο ξεχωριστά υπολόγιζε μέσω του αριθμού τα ports που έπρεπε να στέλνουν σήμα στα LEDs που χρειάζεται.
- **CheckLoss** : Πρόκειται για μια απλή if συνθήκη που εξετάζει αν οι ζωές του παίκτη είναι μικρότερες από 1, κάνει όλα τα LEDs να αναβοσβήνουν, και κάνει reset τις ζωές και το score του παίκτη.

4.2 Ενσωμάτωση

Η ενσωμάτωση αποτελείται το τελευταίο βήμα από την σκοπιά της κατασκευής. Ήταν η εργασία κατά την οποία τοποθετήσαμε όλα όσα είχαμε καταφέρει στο πλαίσιο και φτιάξαμε το τελικό μας προϊόν.

Από μια ανασκόπηση στα προηγούμενα, βλέπουμε πως είχαμε στη διάθεση μας, ένα προγραμματισμένο μικροελεγκτή, 6 αισθητήρες (2 για τα τρίγωνα, 3 για τα bumpers, και ένα για την τρύπα), μια συστοιχία LEDs και ένα πλαίσιο με τους μηχανισμούς τοποθετημένους. Πρέπει λοιπόν, να σχεδιάσουμε το τελικό κύκλωμα :



Να σημειωθεί για λόγους διευκρίνησης, χρησιμοποιήθηκαν 26 αντιστάσεις, μία για κάθε LED που υπάρχει στο display. Επιπλέον, τα 26 ports που χρησιμοποιήθηκαν για τις εξόδους ήταν τα 8 pins από τα B,C,D αλλά και τα pin6 & pin7 του A.

5.1 Προβλήματα και αντιμετώπιση

Όσον αφορά τα προβλήματα που αντιμετωπίσαμε, ήταν κυρίως η εύρεση των solenoids, καθώς εκείνα που πρόσφερε η ελληνική αγορά δεν ήταν αρκετά δυνατά, και όσα μπορούσαμε να προμηθευτούμε από μέσα από το διαδίκτυο ήταν αρκετά ακριβά, όπως αναφέραμε και προηγουμένως το πρόβλημα αυτό αντιμετωπίστηκε με τη κατασκευή δικών μας αισθητήρων που πήραν θέση τους.

Ένα άλλο πρόβλημα αποτέλεσαι ο χρόνος ο οποίος χάθηκε στην αναζήτηση αξιόπιστου μικροελεγκτή και συσκευής που να το προγραμματίσει, καθώς χάθηκαν περίπου 2 βδομάδες, για μπορέσουμε να βρούμε κατάλληλο μικροελεγκτή, και στο τέλος να έχουμε ακόμα πρόβλημα με το προγραμματισμό του για τη λειτουργία των interrupts.

Παράλληλα, είχαμε δυσκολία με το STK500 καθώς λειτουργούσε με αρνητική λογική (π.χ. Η εντολή PORTout = 0x00; ανάβε τα LEDs, ενώ η PORTout = 0xff; τα κρατούσε σβηστά), το πρόβλημα αυτό επιλύθηκε γρήγορα, καθώς ο προβληματισμός μας πάνω στο θέμα, μας έκανε να εξετάσουμε τις εξόδους και τις εισόδους του STK με το πολύμετρο, ώστε να καταλάβουμε πώς λειτουργεί και με αυτό το τρόπο προσαρμόσαμε κατάλληλα το κώδικα μας, για να λειτουργεί ορθά.

Τέλος, το μεγαλύτερο και πιο σημαντικό πρόβλημα ήταν η κατασκευή του display, καθώς δε καταφέραμε να διατηρήσουμε αυστηρά το χρονοδιάγραμμα, είχαμε καθυστέρηση στην υλοποίηση της, με αποτέλεσμα να βρεθούμε προ εκπλήξεων. Το πρόβλημα που παρουσιάστηκε ήταν ότι δεν είχαμε αρκετά LEDs για την κατασκευή της συστοιχίας μας, έτσι χρησιμοποιήσαμε λαμπάκια από χριστουγεννιάτικα στολίδια, ελέγξαμε όλες τις προδιαγραφές λειτουργίας τους, αλλά δε δώσαμε βάση στη κατανάλωση, έτσι λοιπόν όταν έγινε η σύνδεση με το μικροελεγκτή μας, δε μπορούσε να το υποστηρίξει.

5.2 Μελλοντικές προσθήκες και αλλαγές

Ήδη από τη δήλωση των milestones είχαμε αναφέρει προσθήκες που θα κάναμε αν βρισκόμασταν μπροστά σχετικά με τη διορία παράδοσης των milestones. Τέτοιες προσθήκες θα ήταν :

- Λειτουργία αποθήκευσης highscore, να αποθηκεύει δηλαδή στη μνήμη του μικροελεγκτή τις 5 καλύτερες τιμές score που έχει λάβει και να της αναπαριστά.
- Ενσωμάτωση ηχείων για Sound Effects.
- Προσθήκη επιπλέον διακοσμητικών LEDs που θα αναβοσβήνουν με patterns καθ'όλη τη διάρκεια του παιχνιδιού.
- Επιπλέον εμπόδια για ποικιλία παιχνιδιού.
- Μεγαλύτερο μέγεθος, διαφορετικό σχέδιο.

Τώρα όσον αφορά τις αλλαγές, εάν δεν είχαμε το οικονομικό όριο, θα προσαρμόζαμε solenoids στους μηχανισμούς των βραγχίωνων και θα αντικαθιστούσαμε τους αισθητήρες μας με αυτά.

Μία ακόμα αλλαγή, είναι σε περίπτωση που κρατούσαμε τα λαμπάκια αντί των LEDs να σχεδιάζαμε ένα κύκλωμα ενίσχυσης, ώστε να μπορούσε να το υποστηρίξει ο μικροελεγκτής, πράγμα που ίσως θα ήταν απαραίτητο εάν πραγματοποιήσαμε αρκετές προσθήκες κατανάλωσης ισχύος.

6.1 ΒΙΒΛΙΟΓΡΑΦΕΪΑ

- http://www.atmel.com/webdoc/avrassembler/avrassembler.wb_instruction_list.html
- <http://www.atmel.com>
- <http://www.atmel.com/images/io.pdf>
- <http://www.avrfreaks.net/forum>
- <http://www.alldatasheet.com/view.jsp?Searchword=Bpw41n>