

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ (Πάτρα)

Τίτλος Εργασίας: Εφαρμογές του Arduino στο επιχειρησιακό περιβάλλον

Πτυχιακή Εργασία των

Γιαννικούλη Γεώργιου, Παπαχρήστου Χρήστου, Πρέμτο Γιούργκεντ

Επιβλέπων: Στάμος Κωνσταντίνος

ΠΑΤΡΑ, 2015

Περιεχόμενα

Πρόλογος	iii
Εισαγωγή.....	iv
1 ^ο Κεφάλαιο	1
1.1 Επιστήμη και Τεχνολογία	1
1.2 Εξέλιξη ηλεκτρονικών υπολογιστών	2
1.3 Οι μικροελεγκτές.....	9
2 ^ο Κεφάλαιο	18
2.1 Τι είναι το Arduino	18
2.2 Τα μέρη του Arduino	20
2.3 Γλώσσα προγραμματισμού του Arduino.....	24
2.4 Αισθητήρες και το Arduino.....	25
2.5 Σκοπός εργασίας	28
3 ^ο Κεφάλαιο	30
3.1 Προσδιορισμός προβλήματος	30
3.2 Κατασκευή διάταξης	30
3.3 Ανάπτυξη κώδικα	34
3.4 Επεξήγηση κώδικα	39
4 ^ο Κεφάλαιο	44
4.1 Παρουσίαση συστήματος.....	44
4.2 Αποτελέσματα	45
4.3 Μελλοντικοί στόχοι.....	46
Επίλογος.....	48
Πίνακας εικόνων	49
Βιβλιογραφία	51

Πρόλογος

Η παρούσα πτυχιακή εργασία εκπονήθηκε στα πλαίσια του προγράμματος σπουδών του Τμήματος 'Επιχειρηματικού Σχεδιασμού & Πληροφοριακών Συστημάτων' και αποτελεί το επιστέγασμα μιας προσπάθειας της επιτυχούς παρακολούθησης του προγράμματος σπουδών του Τμήματος. Η εργασία μας είναι η πρώτη απόπειρα – ελπίζουμε επιτυχημένη – αξιοποίησης των γνώσεων και των δεξιοτήτων που αποκτήσαμε και θεωρούμε ότι θα αποτελέσει την αφετηρία μιας επιτυχημένης πορείας με προσφορά στην κοινωνία και το επιχειρείν της χώρας.

Θα θέλαμε στο σημείο αυτό να ευχαριστήσουμε αρχικά τον κ. Στάμο Κωσταντίνο καθηγητή του ΑΤΕΙ Πατρών του τμήματος Διοίκησης Επιχειρήσεων για το θέμα που μας εμπιστεύτηκε και τη δυνατότητα να δουλέψουμε πάνω σε ένα θέμα το οποίο συνδυάζει την καινοτομία και την πρακτική εφαρμογή, προσομοιάζοντας πραγματικές συνθήκες εργασίας. Θα θέλαμε, επίσης, να ευχαριστήσουμε το σύνολο των καθηγητών τους οποίους συναναστραφήκαμε καθ' όλη τη διάρκεια των σπουδών μας, καθώς και το υπόλοιπο τεχνικό και διοικητικό προσωπικό του Τμήματος για την καθοδήγηση και στήριξη που μας παρείχαν.

Δεν θα μπορούσαμε, τέλος, παρά να ευχαριστήσουμε τις οικογένειές μας, οι οποίες μας παρείχαν την ολόπλευρη στήριξή τους κατά τη διάρκεια των σπουδών μας, οικονομική και ηθική, αλλά και συνέβαλαν με τις συμβουλές και τις υποδείξεις τους στην ανάπτυξή μας ως προσωπικότητες. Η συμβολή τους στη δύσκολη συγκυρία στην οποία κληθήκαμε να περατώσουμε τις σπουδές μας, ήταν καθοριστική για την ολοκλήρωσή τους.

Πάτρα, Ιούνιος 2015

Γιαννικούλης Γεώργιος

Παπαχρήστος Χρήστος

Πρέμτο Γιούργκεντ

Εισαγωγή

Η παρούσα πτυχιακή εργασία, πραγματοποιήθηκε στα πλαίσια του προγράμματος σπουδών του Τμήματος 'Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων' του Α.Τ.Ε.Ι. Πατρών. Στην εργασία περιγράφεται η κατασκευή ενός συστήματος, το οποίο καταγράφει την κίνηση, τη θερμοκρασία/υγρασία και τη συγκέντρωση αερίων, προκειμένου να αποτρέπεται η μη εξουσιοδοτημένη είσοδος σε μια αποθήκη εύφλεκτων υλικών, ενώ οι συνθήκες θερμοκρασίας/υγρασίας και συγκέντρωσης αερίων να διατηρούνται σταθερές.

Το σύστημα που κατασκευάστηκε, βασίστηκε στην πλακέτα του μικρο-επεξεργαστή Arduino Uno, στον οποίο προσαρμόστηκαν κατάλληλα τρεις αισθητήρες, ένας κίνησης, ένας θερμοκρασίας/υγρασίας και ένας συγκέντρωσης αερίων. Επίσης, προσαρμόστηκαν κατάλληλα χρωματισμένοι λαμπτήρες LED (πράσινοι – κόκκινοι) οι οποίοι θα ειδοποιούν το χρήστη, όταν κάποιος από τους αισθητήρες καταγράψει μια τιμή εκτός του επιτρεπόμενου εύρους λειτουργίας.

Στο πρώτο κεφάλαιο της εργασίας γίνεται μια ιστορική αναδρομή στους υπολογιστές καθώς και ένας διαχωρισμός των εννοιών που πολλές φορές συγχέονται λόγω της καθημερινής τους χρήσης. Μέσα από την ιστορική αναδρομή θα γίνει κατανοητό ότι με τη μείωση των διαστάσεων και την αύξηση των δυνατοτήτων τους, οι υπολογιστές μπορούν να αποτελούν και ισχυρά εργαλεία δουλειάς, αλλά και μια βάση για απλούστερες εφαρμογές οι οποίες θα μπορούσαν να μειώσουν το κόστος λειτουργίας.

Στο δεύτερο κεφάλαιο, γίνεται μια πιο λεπτομερής παρουσίαση του μικροελεγκτή Arduino και των μερών που το αποτελούν. Μέσα από την παρουσίαση αυτή, γίνεται κατανοητό γιατί επιλέχθηκε ο συγκεκριμένος επεξεργαστής για την κατασκευή της διάταξης, μιας και παρουσιάζονται τα πλεονεκτήματα που αφορούν το μέγεθός του και των δυνατοτήτων επέκτασης και παραμετροποίησης με σχετικά εύκολο τρόπο. Η εύκολη παραμετροποίηση, προκύπτει από τη γλώσσα προγραμματισμού που χρησιμοποιεί ο μικρο-επεξεργαστής, για την οποία γίνεται μια σύντομη αναφορά. Παρουσιάζονται, επίσης,

συνοπτικά οι αισθητήρες που χρησιμοποιούνται για την επέκταση των δυνατοτήτων του Arduino. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση του σκοπού της εργασίας μας.

Στο τρίτο κεφάλαιο, εξηγείται αναλυτικά η διαδικασία κατασκευής της διάταξης και η συνδεσμολογία που επιλέξαμε, μέσα από την φωτογραφική παρουσίαση των βημάτων. Στο ίδιο κεφάλαιο παρουσιάζεται και ο κώδικας που αναπτύχθηκε για τη λειτουργία της διάταξης, ενώ με αναλυτικό τρόπο εξηγείται το σύνολο των εντολών που χρησιμοποιήθηκαν.

Στο τέταρτο και τελευταίο κεφάλαιο, παρουσιάζεται η διάταξη που κατασκευάστηκε, ενώ δίνονται μερικά από τα πρώτα αποτελέσματα της λειτουργίας της σε συνθήκες προσομοίωσης πραγματικής λειτουργίας. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση των αδύναμων σημείων της πρωτότυπης αυτής διάταξης, τα οποία κωδικοποιούνται σαν προτάσεις για βελτίωση της λειτουργίας και μελλοντικοί στόχοι.

1^ο Κεφάλαιο

1.1 Επιστήμη και Τεχνολογία

Όταν ο άνθρωπος πριν από αρκετές χιλιάδες χρόνια ξεκίνησε να χτυπά τις πέτρες μεταξύ τους για να τους δώσει κατάλληλο σχήμα ώστε να χρησιμοποιηθούν ως εργαλεία, δε φανταζόταν ότι κάποια στιγμή θα έφτανε στο σημείο να χρησιμοποιεί έναν 3-D εκτυπωτή για να τα κατασκευάσει. Η εντυπωσιακή αυτή εξέλιξη, είχε σαν αποτέλεσμα να μπορεί ο καθένας μας σήμερα να χρησιμοποιεί τα αποτελέσματα της τεχνολογίας για την προσωπική του αλλά και την κοινωνική πρόοδο. Συχνά, στην καθομιλουμένη, ο όρος 'τεχνολογία' συγχέεται με τον όρο 'επιστήμη', με αποτέλεσμα να μη γίνεται σωστή διάκριση των δύο αυτών εννοιών. Για το λόγο αυτό, θα χρειαστεί να αποσαφηνιστούν οι δύο αυτές έννοιες.

Από τη μία μεριά, ο ορισμός της 'επιστήμης' ορίζει ότι *'Επιστήμη είναι η ορθολογική και μεθοδική έρευνα της πραγματικότητας και η συστηματική οργάνωση της γνώσης που προκύπτει από την έρευνα αυτή'* [1]. Σκοπός της Επιστήμης είναι να ερευνά και να ερμηνεύει όλα όσα συμβαίνουν γύρω μας ανεξάρτητα από την θέλησή μας. Τα φαινόμενα αυτά διέπονται από τους φυσικούς νόμους και έχουν σχέση με το φυσικό περιβάλλον.

Από την άλλη μεριά, ο όρος 'τεχνολογία' έχει καταγεγραμμένους πολλούς ορισμούς, σχεδόν ταυτόσημους. Ο επικρατέστερος και πιο περιεκτικός ορισμός, ορίζει ότι *'τεχνολογία είναι το σύνολο των διαθέσιμων γνώσεων, δεξιοτήτων, τεχνικών παραγωγής κατά τη διάρκεια μιας χρονικής περιόδου που ενσωματώνονται στην παραγωγική διαδικασία καθώς και ο τρόπος με τον οποίο συνδυάζονται οι παραγωγικοί συντελεστές ώστε να παραχθεί ένα προϊόν'* [2]. Η τεχνολογία, επίσης, επεκτείνεται σε υπηρεσίες - διοίκηση, εκπαίδευση, τραπεζικά και νομοθετικά συστήματα - καθώς και στη γεωργία και τη βιομηχανία. Σκοπός της Τεχνολογίας είναι η σχεδίαση και κατασκευή τεχνολογικών συστημάτων που θα ικανοποιούν ανθρώπινες ανάγκες.

Από τους ορισμούς των όρων *επιστήμη* και *τεχνολογία* είναι φανερό ότι η δεύτερη ενσωματώνει γνώσεις που προκύπτουν από την πρώτη προκειμένου να τις χρησιμοποιήσει σε

πρακτικές εφαρμογές. Ένα από τα πολλά παραδείγματα είναι η εφαρμογή των νόμων και των γνώσεων της Οπτικής στην κατασκευή και εξέλιξη των μικροσκοπίων ή την ανάπτυξη της τεχνικής της φωτολιθογραφίας που χρησιμοποιείται για την ανάπτυξη των ολοκληρωμένων κυκλωμάτων.

Τα τελευταία χρόνια, η τεχνολογία σχεδόν ταυτίζεται εννοιολογικά με τους ηλεκτρονικούς υπολογιστές. Η ταύτιση αυτή, αν και ατυχής, βρίσκει την εξήγησή της στο γεγονός ότι οι ηλεκτρονικοί υπολογιστές χρησιμοποιούνται σε πάρα πολλούς τομείς και παίζουν πολύ σημαντικό ρόλο στην τεχνολογική εξέλιξη και την ενσωμάτωση της συνεχώς παραγομένης νέας επιστημονικής γνώσης. Η εξέλιξη της τεχνολογίας επέτρεψε την κατασκευή υπολογιστών ολοένα μικρότερων διαστάσεων με συνεχώς αυξανόμενες επιδόσεις. Η εξέλιξη αυτή, επέτρεψε την εύκολη παραμετροποίηση των ηλεκτρονικών υπολογιστών, μια δυνατότητα που είναι πολύ σημαντική για τη συνέχεια και ολοκλήρωση της παρούσας εργασίας.

1.2 Εξέλιξη ηλεκτρονικών υπολογιστών

Μια ιστορική αναδρομή στην εξέλιξη των ηλεκτρονικών υπολογιστών θα μπορούσε να είναι από μόνη της ένα θέμα για μια βιβλιογραφική εργασία. Ωστόσο, ο σκοπός μας είναι μέσα από μια σύντομη ιστορική αναδρομή να αναδειχτεί ο ρόλος του ηλεκτρονικού υπολογιστή ως εργαλείο δουλειάς και όχι η αναλυτική παρουσίαση της ιστορικής εξέλιξης.

Η ιστορία των υπολογιστών πηγαίνει πολύ πίσω στην Ιστορία, κατά την οποία αναπτύχθηκαν συσκευές όπως ο Μηχανισμός των Αντικυθήρων. Για λόγους οικονομίας χώρου, θα επικεντρωθούμε μόνο στις συσκευές οι οποίες θεωρούμε ότι αποτελούν σημαντικά σκαλοπάτια στην εξέλιξη της χρήσης των υπολογιστών.

Άβακας

Ο ηλεκτρονικός υπολογιστής, είναι ένα εργαλείο για την επεξεργασία, αποθήκευση και προβολή πληροφοριών [3]. Η σημερινή του μορφή, προέκυψε μέσα από την εξέλιξη μιας σειράς αναλογικών συσκευών από τον άβακα το 2300 π.Χ. μέχρι το μηχανοκίνητο υπολογιστή

του Pascal το 1645 μ.Χ. που κατά καιρούς χρησιμοποιήθηκαν για την αποτελεσματική καταμέτρηση και την πραγματοποίηση απλών αλγεβρικών πράξεων. Ο άβακας ήταν η πρώτη σοβαρή προσπάθεια του ανθρώπου για να δημιουργήσει ένα εργαλείο υπολογισμού. Στην πραγματικότητα ο άβακας δεν ήταν υπολογιστής αλλά βοηθούσε τον χρήστη του να θυμάται τους υπολογισμούς που έκανε. Ένας ελληνικός άβακας βρέθηκε στην Σαλαμίνα το 1846, χρονολογείται γύρω στο 300 π.χ. και υπολογίζεται ότι είναι ο αρχαιότερος που βρέθηκε μέχρι σήμερα.

Ο άβακας είναι κατασκευασμένος πάνω σε μια πλάκα, με διαστάσεις 149 εκ. μήκος, 75 εκ. πλάτος και 4,5 εκ. πάχος. Η πλάκα περιέχει λευκή άμμο και διαιρείται σε δύο τμήματα από ένα μεγάλο διαχωριστικό που υπάρχει στη μέση. Το επάνω τμήμα απαρτίζεται από 5 παράλληλες γραμμές, εξίσου χωρισμένες από μια κάθετη γραμμή, και ένα ημικύκλιο στο σημείο τομής της κατώτερης παράλληλης γραμμής με την κάθετη. Στο δεύτερο τμήμα υπάρχει άλλη μια ομάδα από 11 παράλληλες γραμμές, που επίσης τέμνονται σε δύο τμήματα από μια γραμμή αλλά το ημικύκλιο βρίσκεται αυτή τη φορά στην κορυφή της τομής. Η τρίτη, η έκτη και ένατη από αυτές τις γραμμές σημειώνονται με ένα σταυρό στο σημείο τομής με την κάθετη γραμμή [4].

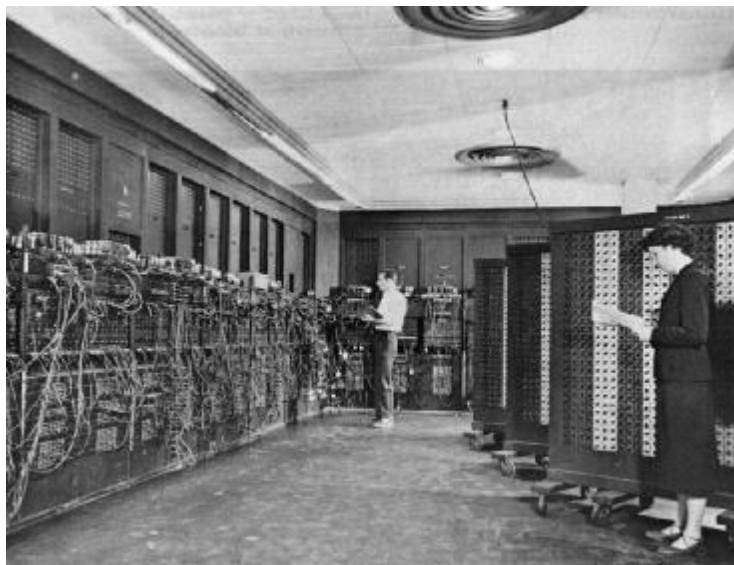


Εικόνα 1: Ο άβακας των Σουμερίων, 2300 π.Χ. (εικόνα από <http://users.sch.gr/mfanarioti/MHXANES/images/abacus/romanabacus1.jpg>)

Η λειτουργία του άβακα βασίζεται στην απλή διαίρεση ή διαχωρισμό αριθμημένων αντικειμένων, έτσι ώστε να κατανοήσει ένα παιδί την έννοια των αριθμητικών πράξεων. Ένα απλό αριθμητάρι αποτελείται από δέκα παράλληλα σύρματα που οι άκρες τους έχουν στερεωθεί σε δύο όρθιους πάσσαλους, με εννέα χάντρες του ίδιου χρώματος περασμένες σε κάθε σύρμα. Κάθε χάντρα πάνω στο σύρμα έχει την ίδια αξία: ίση με το δέκα ή πολλαπλάσια αυτού, έτσι ώστε να γίνονται πλήρεις μαθηματικές πράξεις, ανάλογα με το σύνολο των χαντρών που συγκεντρώνονταν στην αριστερή πλευρά του άβακα, κάθε φορά.

ENIAC

Η εξέλιξη των υπολογιστών επιταχύνθηκε τη δεκαετία του 1940, ακολουθώντας τις απαιτήσεις για αυξανόμενη υπολογιστική ισχύ που προέκυπταν από τις ανάγκες της στρατιωτικής βιομηχανίας και τον 2^ο Παγκόσμιο Πόλεμο. Οι υπολογιστές συνέβαλαν τότε στον ακριβή υπολογισμό των πρώτων υλών για το σχεδιασμό και τη μαζική κατασκευή στρατιωτικών οχημάτων αλλά και στην αποκωδικοποίηση των συνομιλιών των ναζι [5].



Εικόνα 2: Ο υπολογιστής ENIAC, το μέγεθος του οποίου έφτανε να γεμίσει ένα ολόκληρο δωμάτιο (εικόνα από <http://el.wikipedia.org/wiki/ENIAC#mediaviewer/File:Eniac.jpg>)

Ο σημαντικότερος εκπρόσωπος εκείνης της εποχής, ήταν ο ENIAC (ΗΠΑ, 1946) ενώ λιγότερο γνωστός εκπρόσωπος είναι ο 'Κολοσσός' (Ηνωμένο Βασίλειο, 1944). Οι υπολογιστές αυτοί ήταν οι πρώτοι που ήταν δυνατό να προγραμματιστούν για να εκτελούν διαφορετικές λειτουργίες. Χρησιμοποιούσαν εκατοντάδες σωλήνες κενού, μια τεχνολογία που τους διαφοροποιούσε σημαντικά από τους παλαιότερους μηχανοκίνητους προκατόχους τους.

Ο ENIAC είχε περισσότερες από 18.000 λυχνίες κενού και 1500 ηλεκτρονόμους. Ζύγιζε 30 τόνους και καταλάμβανε 163 τετραγωνικά μέτρα χώρο. Κατανάλωνε 140 κιλοβάτ ισχύ. Από αρχιτεκτονικής πλευράς, είχε 20 καταχωρητές (accumulators), κάθε ένας από τους οποίους μπορούσε να αποθηκεύσει έναν αριθμό, του δεκαδικού συστήματος, των 10 ψηφίων. Μία ομάδα από 10 λυχνίες αναπαριστούσε κάθε ένα από τα δέκα ψηφία. Σε κάθε στιγμή, μία μόνο λυχνία βρισκόταν σε κατάσταση λειτουργίας, αναπαριστώντας έτσι την τιμή που αποκτούσε κάθε ένα από τα δέκα ψηφία του αριθμού [6].

Ο προγραμματισμός του γινόταν με τη ρύθμιση 6.000 διακοπών πολλών θέσεων και με την σύνδεση ενός πλήθους υποδοχών με καλώδια (βραχυκυκλωτές). Αυτό αποτελούσε και το μεγάλο του πρόβλημα καθώς έπρεπε να προγραμματίζεται με το χέρι, ανοιγοκλείνοντας διακόπτες και βραχυκυκλώνοντας τις υποδοχές με τα συνδεδετικά καλώδια. Όμως, ήταν σημαντικά ταχύτερος από οποιοδήποτε άλλον ηλεκτρομηχανικό υπολογιστή της εποχής του, καθώς μπορούσε να εκτελέσει 5.000 προσθέσεις ανά δευτερόλεπτο.

Ο ENIAC, μπορούσε να προγραμματιστεί, με χρήση διάτρητων καρτών, έτσι ώστε να εκτελεί περίπλοκες διεργασίες, οι οποίες περιελάμβαναν επαναληπτικές δομές, κλάδους προγραμμάτων και υπορουτίνες [7]. Η διαδικασία εισαγωγής ενός προβλήματος στον ENIAC συνήθως διαρκούσε μερικές εβδομάδες, ενώ μετά την εισαγωγή του ο κατάλληλος χειρισμός των διακοπών και καλωδίων για την επίλυση του προβλήματος, μπορούσε να διαρκέσει αρκετές ημέρες. Κάθε υπολογιστικό βήμα, ακολουθούσαν από μια περίοδο επαλήθευσης και εκσφαλμάτωσης βήμα προς βήμα, μιας και κάθε πρόγραμμα ήταν 'μιας χρήσης'.

Οι πρώτοι οικιακοί υπολογιστές

Η τεχνολογία των σωλήνων κενού, χρησιμοποιήθηκε αργότερα στον πρώτο οικιακό ηλεκτρονικό υπολογιστή, τον Univac I (ΗΠΑ, 1951) [8]. Ο Univac, μπορούσε να χειριστεί αριθμητικούς και αλφαριθμητικούς χαρακτήρες το ίδιο εύκολα ενώ ήταν ο πρώτος υπολογιστής που μπορούσε να αποθηκεύσει απλά υπολογιστικά προγράμματα, ενώ οι οδηγίες για την εκτέλεση του προγράμματος μπορούσαν να αποθηκευτούν σε εσωτερικό χώρο του υπολογιστή και όχι σε διάτρητες κάρτες όπως στον ENIAC.



Εικόνα 3: Ο οικιακός υπολογιστής Univac 1 (εικόνα από

http://upload.wikimedia.org/wikipedia/commons/5/55/Museum_of_Science%2C_Boston%2C_MA_-_IMG_3163.JPG)

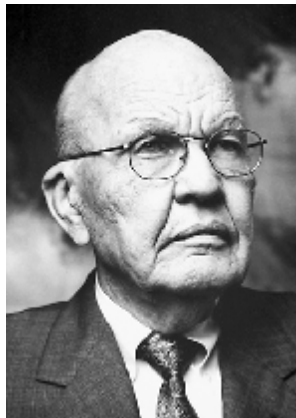
Ο υπολογιστής Univac I καθώς και ο ανταγωνιστής του εκείνη την εποχή, ο IBM 702, αντιπροσώπευαν μια γενιά υπολογιστών οι οποίοι χρησιμοποιήθηκαν κυρίως σε εταιρικά περιβάλλοντα για να καλύψουν τις ανάγκες αποθήκευσης και επεξεργασίας δεδομένων αλλά και για στατιστικές αναλύσεις.



Εικόνα 4: Ο υπολογιστής IBM 702 σε περιβάλλον εργασίας (εικόνα από http://upload.wikimedia.org/wikipedia/commons/1/10/BRL61-IBM_702.jpg)

Στη δεκαετία του 1960, άρχισε να εδραιώνεται η παρουσία των υπολογιστών κυρίως λόγω της δυνατότητας να χρησιμοποιηθούν τα τρανζίστορ για διεργασίες υπολογισμού, αλλά και λόγω της ανάπτυξης των μαγνητικών μέσω αποθήκευσης αντί των σωλήνων κενού. Οι δυνατότητες αυτές, επέτρεπαν τη χρήση υπολογιστών οι οποίοι μίκραιναν συνεχώς σε μέγεθος.

Η ανακάλυψη των ολοκληρωμένων κυκλωμάτων [9] οδήγησε σε ακόμα μεγαλύτερη συρρίκνωση των διαστάσεων των υπολογιστών και τη δυνατότητα για μεγαλύτερη υπολογιστική ισχύ σε μικρότερες διαστάσεις. Το εγχείρημα αυτό ήταν τόσο συναρπαστικό ώστε ο J. Kilby, τιμήθηκε με το βραβείο Nobel το 2000 [10] για αυτή του την ανακάλυψη.



Εικόνα 5: Ο J. Kilby, 'πατέρας' των ολοκληρωμένων κυκλωμάτων (εικόνα από http://www.nobelprize.org/nobel_prizes/physics/laureates/2000/kilby_postcard.jpg)

Μια ακόμη σημαντική παράμετρος, ήταν οι γλώσσες προγραμματισμού που αναπτύχθηκαν και οι οποίες ήταν περισσότερο ευανάγνωστες από τη γλώσσα μηχανής και τις διάτρητες κάρτες. Η γλώσσες προγραμματισμού που χρησιμοποιήθηκαν κυρίως ήταν η COBOL [11] και η FORTRAN [12].

Η γλώσσα FORTRAN είναι μία από τις πρώτες γλώσσες προγραμματισμού υψηλού επιπέδου, η οποία χρησιμοποιήθηκε κυρίως σε επιστημονικές αλλά και σε εμπορικές εφαρμογές. Αρχικά η FORTRAN ήταν προσανατολισμένη στην επίλυση μαθηματικών προβλημάτων, ενώ δημιουργήθηκε τη δεκαετία του 1950 από την IBM και χρησιμοποιείται μέχρι και σήμερα.

Η COBOL είναι γλώσσα προγραμματισμού και αναπτύχθηκε με σκοπό να καλύψει τις δυσχέρειες που παρουσίαζε η FORTRAN ως γλώσσα προγραμματισμού για εμπορικές εφαρμογές. Βασικό χαρακτηριστικό της COBOL είναι η χρησιμοποίηση της Αγγλικής γλώσσας με προτάσεις, παραγράφους και σημεία στίξης. Η COBOL είναι μία γλώσσα με πολλά στοιχεία μιας φυσικής γλώσσας και έντονη την ύπαρξη του συντακτικού της. Για το λόγο αυτό η COBOL έγινε πολύ δημοφιλής στο χώρο των εμπορικών εφαρμογών σε σύντομο χρονικό διάστημα.

Η νεότερη γενιά

Τη δεκαετία του 1970, η ανακάλυψη των μικροεπεξεργαστών, έφερε επανάσταση στους ηλεκτρονικούς υπολογιστές με τη μείωση των διαστάσεων της κεντρικής μονάδας επεξεργασίας ώστε να χωράει σε ένα απλό ολοκληρωμένο κύκλωμα, γνωστή ως 'μικροσίπ'. Τα μικροσίπ χρησιμοποιήθηκαν και σε άλλες συσκευές, από απλούς υπολογιστές τσέπης μέχρι τους προσωπικούς υπολογιστές και αποτελούν το κύριο μέρος διαφόρων ηλεκτρονικών συσκευών ακόμη και σήμερα.



Εικόνα 6: Μικροσίπ της δεκαετίας του '70, από την εταιρία 'Intel' (εικόνα από http://upload.wikimedia.org/wikipedia/commons/f/f8/KL_intel_D3002.jpg)

Η ανάπτυξη της υπολογιστικής ισχύος, οδήγησε και στην ανάπτυξη ισχυρότερων προγραμματιστικών εργαλείων όπως η γλώσσα C, C++ και η Java μεταγενέστερα για επιστημονικές και εμπορικές εφαρμογές.

Τα παραπάνω οδήγησαν μέχρι τους υπολογιστές που γνωρίζουμε σήμερα, είτε τους επιτραπέζιους οικιακούς, είτε τους φορητούς, είτε τους μεγάλους διακομιστές που χρησιμοποιούνται σε βιομηχανικές εφαρμογές και εφαρμογές δικτύων υπολογιστών.

1.3 Οι μικροελεγκτές

Ο μικροελεγκτής είναι μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει [13]. Οι μικροελεγκτές χρησιμοποιούνται σε ενσωματωμένα συστήματα ελέγχου, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, σε προϊόντα όπως ψηφιακές φωτογραφικές μηχανές ή ηλεκτρονικές κονσόλες, άλλες ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα οχήματα.



Εικόνα 7: Σύγχρονος μικροεπεξεργαστής (εικόνα από <http://sdi0700055.wikispaces.com/file/view/Intel-Sandy-Bridge-CPU.jpg/238926147/Intel-Sandy-Bridge-CPU.jpg>)

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του είδους, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση [14]. Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι:

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν. Η ενσωμάτωση περιφερειακών

σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.

- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος), λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών. Το ολοκληρωμένο κύκλωμα που τον αποτελεί, περιέχει μόνο την Λογική και Αριθμητική Μονάδα (ALU), τους στοιχειώδεις καταχωρητές, την προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας και, κάποιες φορές, τον ελεγκτή μνήμης.

Όμως, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι:

- Κύκλωμα συνδετικής λογικής για τη σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων του επεξεργαστή.

- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
- Μεγάλο μέγεθος μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης.
- Διαχειριστή αιτήσεων διακοπής από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Κύκλωμα επιτήρησης λειτουργίας το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος.
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού.
- Έναν ή περισσότερους χρονιστές – απαριθμητές υψηλής ταχύτητας για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι' αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων.

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι, υπάρχουν:

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως ROM-less. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.
- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της (Mask ROM) ή γράφεται μόνο μια φορά (One Time Programmable, OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.
- Μικροελεγκτές με μνήμη FLASH, οι οποία μπορούν συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία (από το ειδικό τζαμάκι).

Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και:

- Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (UART).
- Σύγχρονες σειριακές θύρες επικοινωνίας (πχ I2C, SPI, Ethernet).
- Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικό-λογισμικό (firmware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLC, ISDN, ADSL.

- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (FPU), η οποία είναι πάντοτε πιο γρήγορη από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας σήματος (DSP).
- Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (ADC).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (DAC).
- Ελεγκτή οθόνης υγρών κρυστάλλων (LCD).
- Υποσύστημα προγραμματισμού πάνω στο κύκλωμα. Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στη συσκευή μια εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS-232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.
- Υποσύστημα προγραμματισμού και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο JTAG). Χάρη σε αυτό, είναι δυνατός ο προγραμματισμός της μνήμης προγράμματος χωρίς να προαπαιτείται κάποιο πρόγραμμα υποδοχής. Γι αυτό το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό, πχ κατά τη συναρμολόγηση, ή σε περίπτωση σφάλματος στο λογισμικό υποδοχής το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση.



Εικόνα 8: Διάφορες θύρες επικοινωνίας (εικόνα από http://www.automatica.gr/images/PlcModems/ModemImg2_1.JPG)

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

- Μικροελεγκτές (καμμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κ.ά.)
- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I2C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.

- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερισιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίζει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).
- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware.

Η επιτυχία μιας οικογένειας μικροελεγκτών καθορίζεται σε μεγάλο βαθμό από τη διαθεσιμότητα και την ευχρηστία των σχετικών εργαλείων ανάπτυξης, όπως μεταφραστές από γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από τον μικροελεγκτή, προγραμματιστές της εσωτερικής μνήμης και εργαλεία εκσφαλμάτωσης. Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται ποτέ μόνο λογισμικό, καθώς δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας με αυτούς. Στον τομέα των εργαλείων ανάπτυξης, δραστηριοποιούνται όχι μόνο οι ίδιοι οι κατασκευαστές μικροελεγκτών αλλά και εξειδικευμένες εταιρείες.

Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα η μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.

Μια διάσημη κατηγορία μικροελεγκτών είναι οι PIC. Οι μικρο-ελεγκτές PIC, παρουσιάστηκαν το 1985 από την εταιρία 'General Instruments' [15] και απευθύνονταν κυρίως

σε ερασιτέχνες ή περιστασιακούς χρήστες με περιορισμένη τεχνογνωσία σε ηλεκτρονικούς υπολογιστές. Ο εύκολος και γρήγορος προγραμματισμός τους μέσω απλών εργαλείων όπως η γλώσσα 'PicBASIC' [16], η δυνατότητα αποθήκευσης των προγραμμάτων σε εσωτερική μνήμη αλλά και η δυνατότητα διασύνδεσης με αισθητήρες εισόδου και συσκευές εξόδου, ήταν οι κύριοι λόγοι εξάπλωσής τους.

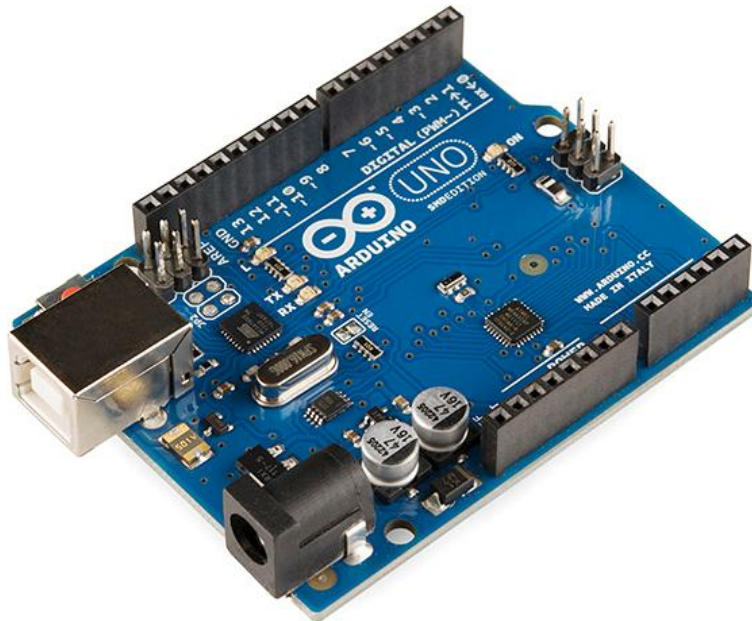
Πρόσφατα η εξέλιξη των μικροελεγκτών οδήγησε στην ανάπτυξη των μικροελεγκτών Arduino, στα πλαίσια μιας φοιτητικής εργασίας στην Ιταλία [17]. Το 2005 μια ομάδα αποτελούμενη από τους H. Barragan, M. Banzi, D. Cuartielles, D. Mellis, G. Marino και N. Zambetti ίδρυσαν την πρώτη ομάδα Arduino [18]. Σκοπός τους ήταν να κατασκευάσουν μια πρωτότυπη ηλεκτρονική πλατφόρμα που θα μπορούσε να χρησιμοποιηθεί και από μη-εξειδικευμένο κοινό. Το αποτέλεσμα ήταν η δημιουργία μιας πλατφόρμας που μπορούσε να ενσωματώσει προγραμματιστικό περιβάλλον, τη δυνατότητα να προγραμματίζεται μέσω μιας θύρας USB και να διατηρεί χαμηλά την τιμή του (περίπου 35 \$) [18].

2^ο Κεφάλαιο

2.1 Τι είναι το Arduino

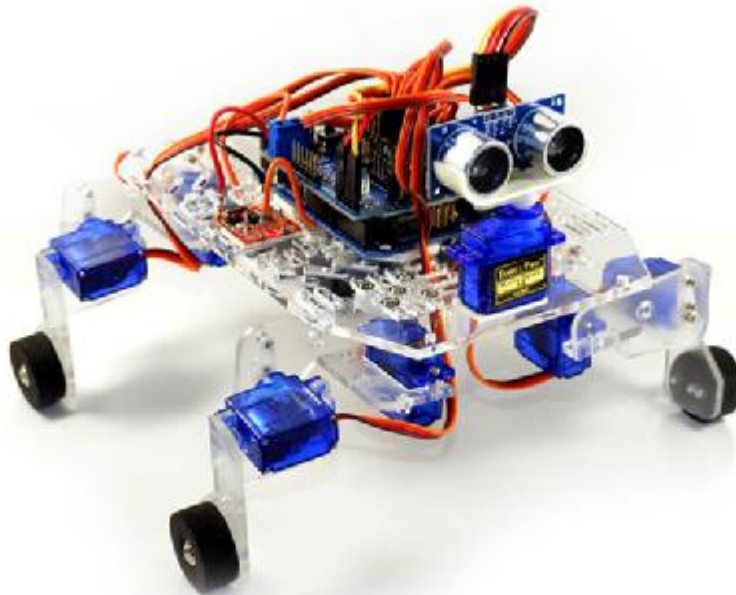
Όπως ήδη φάνηκε από το 1^ο κεφάλαιο, ξεχωριστή σημασία θα δοθεί στους μικροελεγκτές και ιδιαίτερα στους μικροελεγκτές Arduino. Στο 2^ο κεφάλαιο θα γίνει μια λεπτομερέστερη περιγραφή του Arduino, των μερών που το αποτελούν καθώς και των αισθητήρων που συνεργάζονται με αυτό.

Το Arduino είναι ένα εργαλείο το οποίο μπορεί να αλληλεπιδράσει και να ελέγξει παραμέτρους του φυσικού κόσμου, με μεγαλύτερη ευκολία από έναν προσωπικό υπολογιστή. Είναι μια πλατφόρμα που βασίζεται στο ανοιχτό λογισμικό και βασίζεται σε μια πλακέτα η οποία μπορεί να επεκταθεί με χρήση κατάλληλων αισθητήρων και εξόδων [19].



Εικόνα 9: Ένας μικρο-ελεγκτής Arduino (εικόνα από http://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg)

Τα προγράμματα που αναπτύσσονται μπορούν είτε να είναι αυτόνομα είτε να επικοινωνούν μέσω κατάλληλου λογισμικού με άλλους υπολογιστές. Η γλώσσα προγραμματισμού του Arduino είναι μια παραλλαγή της γλώσσας 'Wiring' [20]. Σε σχέση με άλλους μικροελεγκτές, το Arduino έχει επικρατήσει λόγω της χαμηλής τιμής του, της δυνατότητας διεπαφής με άλλες πλατφόρμες (Windows, Linux, Mac OS) και της δυνατότητας εύκολου προγραμματισμού του. Το λογισμικό του, όπως αναφέρθηκε ήδη, είναι ανοιχτού κώδικα που επιτρέπει ακόμη και σε αρχάριους χρήστες να επεξεργαστούν έτοιμο κώδικα, ενώ και το υλικό του επιτρέπεται να αντιγραφεί εύκολα και έτσι επιτρέπει σε πεπειραμένους χρήστες να αναπτύξουν τη δική τους πλακέτα.



Εικόνα 10: Εφαρμογή του Arduino σε ρομπωτικό σύστημα (εικόνα από http://media.apcmag.com/wp-content/uploads/sites/20/2013/09/apcnews2012stompy-project8-sml_mainImage1.jpg1.jpg)

Σήμερα, οι μικροελεγκτές Arduino, έχουν γίνει πολύ δημοφιλείς και χρησιμοποιούνται ολοένα συχνότερα τόσο σε εμπορικές εφαρμογές όσο και σε ερασιτεχνικές/οικιακές εφαρμογές με πολύ χαμηλό κόστος και μεγάλη ευκολία τροποποίησης των προγραμμάτων που αναπτύσσονται. Στη συνέχεια θα παρουσιαστούν τα μέρη του Arduino.

2.2 Τα μέρη του Arduino

Ακολουθεί η παρουσίαση των μερών του **Arduino**, όπως αυτά καταγράφονται αναλυτικά σε άλλη εργασία [21]. Το **Arduino** βασίζεται στην πλακέτα **ATmega 328** η οποία αποτελείται από έναν 8-bit RISC μικροελεγκτή ο οποίος είναι χρονισμένος στα 16MHz. Ο **ATmega 328** διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης **SRAM** που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες και άλλα κατά την εκτέλεσή τους. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο **Arduino** σταματήσει ή αν γίνει επανεκκίνηση.
- 1Kb μνήμης **EEPROM** η οποία μπορεί να χρησιμοποιηθεί για εγγραφή/ανάγνωση δεδομένων ανά byte από τα προγράμματά κατά την εκτέλεσή τους. Σε αντίθεση με την **SRAM**, η **EEPROM** δεν χάνει τα περιεχόμενά της με απώλεια της τροφοδοσίας ή επανεκκίνησης οπότε είναι το ανάλογο του σκληρού δίσκου ενός οικιακού υπολογιστή.
- 32Kb μνήμης **Flash**, από τα οποία τα 2Kb χρησιμοποιούνται από το **firmware** του **Arduino** που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το **firmware** αυτό που στην ορολογία του **Arduino** ονομάζεται **bootloader** είναι αναγκαίο για την εγκατάσταση των προγραμμάτων στον μικροελεγκτή μέσω της θύρας **USB**. Τα υπόλοιπα 30Kb της μνήμης **Flash** χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν. Η μνήμη **Flash**, όπως και η **EEPROM** δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης και ενώ υπό κανονικές συνθήκες δεν προορίζεται για χρήση μέσα από τα προγράμματά, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb **SRAM** + 1Kb **EEPROM**), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει.

Ο μικροελεγκτής **ATmega** υποστηρίζει σειριακή επικοινωνία, την οποία το **Arduino** προωθεί μέσα από έναν ελεγκτή **Serial-over-USB** ώστε να συνδέεται με τον υπολογιστή μέσω της θύρας **USB**. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο **Arduino** αλλά και για αμφίδρομη επικοινωνία του **Arduino** με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται. Επιπλέον, στην

πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.



Εικόνα 11: Η πλακέτα ATmega 328 (εικόνα από <http://www.nkcelectronics.com/assets/images/atmega328p.jpg>)

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμα σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Εάν ένα από αυτά τα pin οριστούν ως ψηφιακή είσοδος μέσα από το πρόγραμμα, μπορεί με κατάλληλη εντολή να διαβαστεί η κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που είναι συνδεδεμένη σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin. Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν κάποιο πρόγραμμα ενεργοποιεί την σειριακή θύρα.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Μπορούν να ρυθμιστούν μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων.

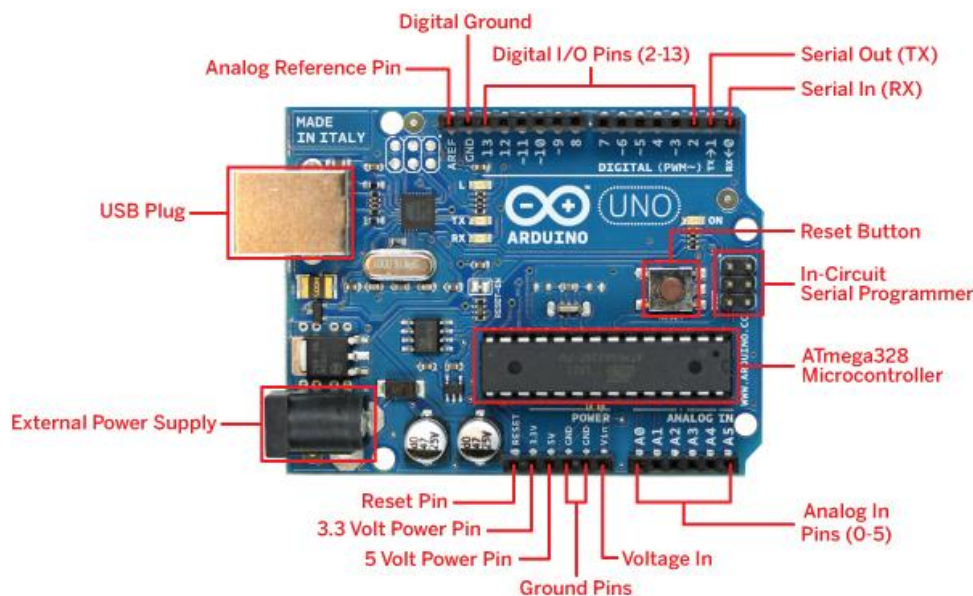
Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

Η τροφοδοσία του Arduino γίνεται με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino. Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει εξαρτήματά με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.

- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino.



Εικόνα 12: Παρουσίαση των μερών του Arduino (εικόνα από http://thietbichetao.com/wp-content/uploads/2014/03/arduino_uno_R3_pinout.jpg)

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι επανεκκίνηση και λειτουργία αντίστοιχα. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας

του Arduino είναι να του ανατεθεί να αναβοσβήνει ένα LED. Για το λόγο αυτό, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, όταν δεν έχει συνδεθεί τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά, θα ανάψει αυτό το ενσωματωμένο LED.

2.3 Γλώσσα προγραμματισμού του Arduino

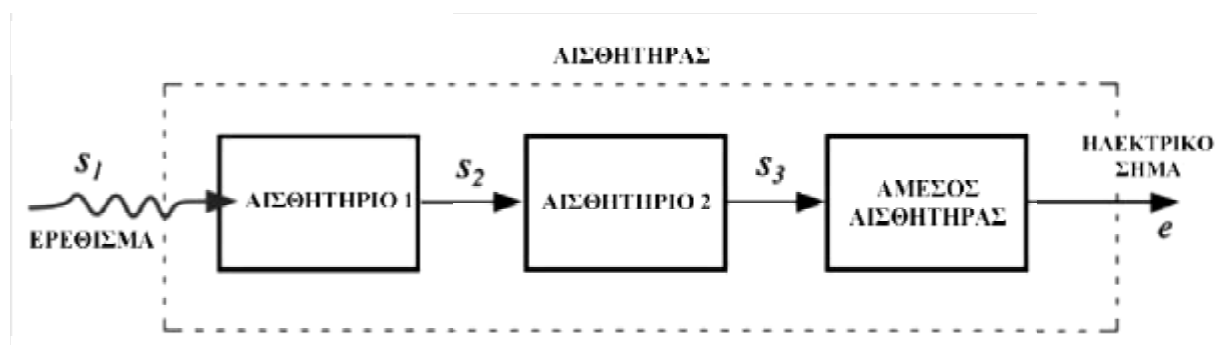
Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch) [22].

Η γλώσσα προγραμματισμού του Arduino βασίζεται στη γλώσσα Wiring [20], μια παραλλαγή της C/C++ για μικροελεγκτές όπως ο ATmega και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για τη μεταγλώττιση χρησιμοποιείται ο μεταγλωττιστής AVR gcc [23] και ως βασική βιβλιοθήκη της γλώσσας προγραμματισμού C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από τη γλώσσα προγραμματισμού C, στην γλώσσα του Arduino χρησιμοποιούνται οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino [24].

2.4 Αισθητήρες και το Arduino

Οι αισθητήρες είναι συσκευές που κατασκευάζει ο άνθρωπος για να μπορέσει ο υπολογιστής να αντιληφθεί μια μεταβολή κάποιας παραμέτρου του φυσικού περιβάλλοντος. Οι αισθητήρες καταγράφουν μια μεταβολή των παραμέτρων του φυσικού περιβάλλοντος και την μετατρέπουν σε ηλεκτρικό σήμα [25] ώστε να μπορεί να γίνει επικοινωνία με τον ηλεκτρονικό υπολογιστή. Μπορούν να τοποθετηθούν είτε εσωτερικά σε ένα σύστημα ώστε να ελέγχουν την απόδοσή του (εγγενής αισθητήρας), είτε εξωτερικά (εξωγενής αισθητήρας) ώστε να λαμβάνουν ερεθίσματα από το εξωτερικό περιβάλλον. Οι αισθητήρες διακρίνονται επίσης σε άμεσους και σύνθετους. Οι άμεσοι αισθητήρες μετατρέπουν ένα ερέθισμα σε ηλεκτρικό σήμα, ενώ οι σύνθετοι έχουν ένα ή περισσότερα αισθητήρια που τοποθετούνται πριν τον άμεσο αισθητήρα που παράγει το ηλεκτρικό σήμα.



Εικόνα 13: Σχηματική αναπαράσταση σύνθετου αισθητήρα

Ένας αισθητήρας μπορεί να είναι παθητικός ή ενεργητικός. Ο παθητικός αισθητήρας δεν χρειάζεται καμία πρόσθετη πηγή ενέργειας και ανταποκρίνεται με το να παράγει άμεσα ένα ηλεκτρικό σήμα. Αντίθετα, ο ενεργητικός αισθητήρας απαιτεί μία εξωτερική τροφοδοσία ρεύματος για τη λειτουργία του, η οποία ονομάζεται σήμα διέγερσης. Αυτό το σήμα τροποποιείται από τον αισθητήρα ώστε να παράγει το σήμα εξόδου. Οι ενεργοί αισθητήρες ονομάζονται μερικές φορές και παραμετρικοί, επειδή οι ιδιότητες τους αλλάζουν, λόγω των μεταβολών στην τιμή του ρεύματος τροφοδοσίας.

Ανάλογα με το σύστημα αναφοράς, οι αισθητήρες μπορούν να ταξινομηθούν σε απόλυτους και σε σχετικούς. Οι απόλυτοι αισθητήρες ανιχνεύουν ένα ερέθισμα σε σχέση με

την απόλυτη φυσική κλίμακα, που είναι ανεξάρτητη από τις συνθήκες μέτρησης, ενώ οι σχετικοί αισθητήρες παράγουν ένα σήμα που σε κάθε περίπτωση δεν ανταποκρίνεται στην απόλυτη φυσική κλίμακα.

Ένας άλλος τρόπος να ταξινομηθούν οι αισθητήρες είναι βάσει ιδιαιτερότητας των χαρακτηριστικών ή των ιδιοτήτων τους:

- Προδιαγραφές: ευαισθησίας, εύρος ερεθίσματος
- Υλικό κατασκευής: οργανικό, ανόργανο
- Μέσα ανίχνευσης: ηλεκτρικά, μαγνητικά
- Φαινόμενα μετατροπής: χημικά, βιολογικά
- Πεδία εφαρμογών: γεωργία, αυτοκινητοβιομηχανία
- Τύπος ερεθίσματος: ακτινοβολία, θερμότητα

Τα στατικά χαρακτηριστικά των αισθητήρων αναφέρονται στην κατάσταση κατά την οποία έχει επέλθει ισορροπία μεταξύ αισθητήρα και του μετρούμενου ερεθίσματος. Για να επιτευχθεί κάτι τέτοιο πρέπει το μετρούμενο ερέθισμα είτε να είναι σταθερό, είτε να μεταβάλλεται πολύ αργά σε σχέση με τη δυνατότητα τον αισθητήρα να αντιληφθεί τη μεταβολή αυτή. Τα παραπάνω συνοψίζονται στα εξής:

- Εύρος: τα όρια στα οποία η συσκευή λειτουργεί αξιόπιστα.
- Ακρίβεια: η εγγύτητα της τιμής εξόδου προς τη τιμή εισόδου.
- Σφάλμα: η διαφορά ανάμεσα στη μετρούμενη τιμή και τη πραγματική τιμή.
- Ανοχή: το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.
- Διακριτική ικανότητα: η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.
- Ευαισθησία: η σχέση της αλλαγής εξόδου προς τη αλλαγή εισόδου, είναι ίση με τη διαφορά των τιμών της εξόδου προς τη διαφορά των αντίστοιχων τιμών εισόδου.
- Βαθμονόμηση: η βαθμολόγηση της κλίμακας σε μονάδες.

- Νεκρή ζώνη: το μέγιστο ποσό αλλαγής της εισόδου που δεν επιφέρει αλλαγή στην έξοδο.
- Γραμμικότητα: ο βαθμός στον οποίο η γραφική παράσταση της εξόδου προσεγγίζει ευθεία ως προς την είσοδο τον αισθητήρα.
- Απόκριση: ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.
- Καθυστέρηση: η καθυστέρηση της αλλαγής της εξόδου ως προς την είσοδο.
- Ευστάθεια: η μεταβολή της εξόδου σε μεγάλη χρονική περίοδο, χωρίς μεταβολή της εισόδου και των συνθηκών.
- Υστέρηση: η διαφορά στην έξοδο όταν η κατεύθυνση της μεταβολής της εισόδου αντιστραφεί.
- Επαναληψιμότητα: η παραγωγή του ίδιου αποτελέσματος, σε διαφορετικές χρονικές στιγμές, με την ίδια είσοδο.
- Ολίσθηση: η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον.
- Στατικό σφάλμα: σταθερό σφάλμα σε όλο το εύρος λειτουργίας, το οποίο μπορεί να αντισταθμιστεί.
- Χρόνος λειτουργίας: ο εκτιμώμενος χρόνος λειτουργίας στα πλαίσια των προδιαγραφών του.

Εκτός από τα στατικά χαρακτηριστικά τα οποία αναφέρονται στην περίπτωση όπου έχει επέλθει ισορροπία μεταξύ σήματος εισόδου και αισθητήρα (δηλαδή όταν πλέον τα σήματα εισόδου και εξόδου δεν μεταβάλλονται με το χρόνο), κάθε αισθητήρας διαθέτει και δυναμικά χαρακτηριστικά. Τα δυναμικά χαρακτηριστικά περιγράφουν τη συμπεριφορά του αισθητήρα μεταξύ της στιγμής κατά την οποία το σήμα εισόδου μεταβάλλεται έως τη στιγμή κατά την οποία το σήμα εξόδου θα σταθεροποιηθεί εκ νέου.

Τα δυναμικά χαρακτηριστικά ισχύουν, όπως και τα στατικά, εντός συγκεκριμένου εύρους συνθηκών λειτουργίας του αισθητήρα. Αν ο αισθητήρας βρεθεί να λειτουργεί εκτός αυτού του εύρους, τότε τα δυναμικά χαρακτηριστικά αναμένεται να μεταβληθούν. Οι

αισθητήρες που αυτή τη στιγμή είναι διαθέσιμοι για το **Arduino** είναι: Ακουστικοί, βιολογικοί, ηλεκτρικοί, μαγνητικοί, ακτινοβολίας, θερμικοί, οπτικοί, μηχανικοί, χημικοί.

2.5 Σκοπός εργασίας

Μια αποθήκη επικίνδυνων και εύφλεκτων υλών και υλικών, επιβάλλεται να έχει πλήρως ελεγχόμενες συνθήκες αποθήκευσης, καθώς και προστασία από μη εξουσιοδοτημένη παρουσία ατόμων στο χώρο. Παράμετροι όπως η θερμοκρασία/υγρασία του χώρου και η συγκέντρωση των αερίων στο χώρο, θα πρέπει να διατηρούνται σε ένα επιτρεπόμενο εύρος τιμών. Στην αγορά υπάρχουν ήδη έτοιμα συστήματα τα οποία καταγράφουν σε πραγματικό χρόνο τις μεταβολές στις παραπάνω παραμέτρους και αναπροσαρμόζουν κατάλληλα ή ειδοποιούν το χρήστη. Τα συστήματα αυτά είναι ιδιαίτερα ακριβά και απαιτούν τακτικά συντήρηση, η οποία αυξάνει το κόστος λειτουργίας τους. Ταυτόχρονα είναι ελάχιστα έως καθόλου ανοιχτά στην παραμετροποίησή τους από το χρήστη, χωρίς την παρουσία ειδικευμένου τεχνικού-προγραμματιστή.

Η ιδέα της εργασίας είναι η κατασκευή μιας διάταξης εφάμιλλης με αυτές που υπάρχουν ήδη στην αγορά, χωρίς όμως τα μειονεκτήματα που παρουσιάζονται σε αυτά. Συγκεκριμένα, επιδιώκουμε την κατασκευή ενός ολοκληρωμένου συστήματος, το οποίο θα έχει σημαντικά χαμηλότερο κόστος από τα ήδη υπάρχοντα, ενώ θα δίνει τη δυνατότητα στο χρήστη να παραμετροποιεί κατά το δοκούν τις λειτουργίες του συστήματος χωρίς την απαίτηση για εξειδικευμένες γνώσεις προγραμματισμού.

Στα πλαίσια της παρούσης εργασίας, θα κατασκευαστεί μια διάταξη βασισμένη στο **Arduino** με κατάλληλη προσαρμογή αισθητήρων κίνησης, θερμοκρασίας/υγρασίας και συγκέντρωσης αερίων. Η διάταξη θα δοκιμαστεί σε λειτουργία προσομοίωσης σε μια εικονική αποθήκη εύφλεκτων υλικών, όπου θα καταγράφει σε πραγματικό χρόνο και θα ενημερώνει κατάλληλα το χρήστη όταν κάποια από τις παραμέτρους μετρηθεί εκτός των προκαθορισμένων ορίων.

Εκτός από την κατασκευή της διάταξης, θα αναπτυχθεί κατάλληλος κώδικας για την επικοινωνία μεταξύ του κάθε αισθητήρα και του **Arduino** και για την επικοινωνία του **Arduino**

με το χρήστη. Μέσα από το λογισμικό, θα μπορεί να καθορίζει ο χρήστης τις επιτρεπόμενες τιμές λειτουργίας, τη συχνότητα καταγραφής των τιμών και τις άλλες παραμέτρους. Τέλος, ιδιαίτερη έμφαση θα δοθεί στην παρουσίαση ενός ολοκληρωμένου συστήματος το οποίο θα μπορεί να χρησιμοποιηθεί άμεσα σε πραγματικές συνθήκες λειτουργίας χωρίς να υπάρχει ανάγκη για νέα κατασκευή από την αρχή.

Για την ολοκλήρωση της εργασίας προμηθευτήκαμε μια πλακέτα Arduino Uno, τους αισθητήρες κίνησης, θερμοκρασίας/υγρασίας και συγκέντρωσης αερίων καθώς και τα απαραίτητα καλώδια σύνδεσης και τους πράσινους και κόκκινους λαμπτήρες LED που χρησιμοποιήθηκαν για την κατασκευή της διάταξης.

3^ο Κεφάλαιο

3.1 Προσδιορισμός προβλήματος

Οι συνθήκες που επικρατούν σε αποθήκες επικίνδυνων υλών και υλικών, όπως η υγρασία και η θερμοκρασία του χώρου, καθώς και λόγοι ασφαλείας επιβάλλουν ο χώρος αυτός να είναι πλήρως ελεγχόμενος από άποψη συνθηκών αλλά και ασφαλής από μη εξουσιοδοτημένες παρουσίες ατόμων στο χώρο. Τα εμπορικά διαθέσιμα συστήματα ικανοποιούν τις παραπάνω απαιτήσεις, ωστόσο έχουν δύο σοβαρά μειονεκτήματα.

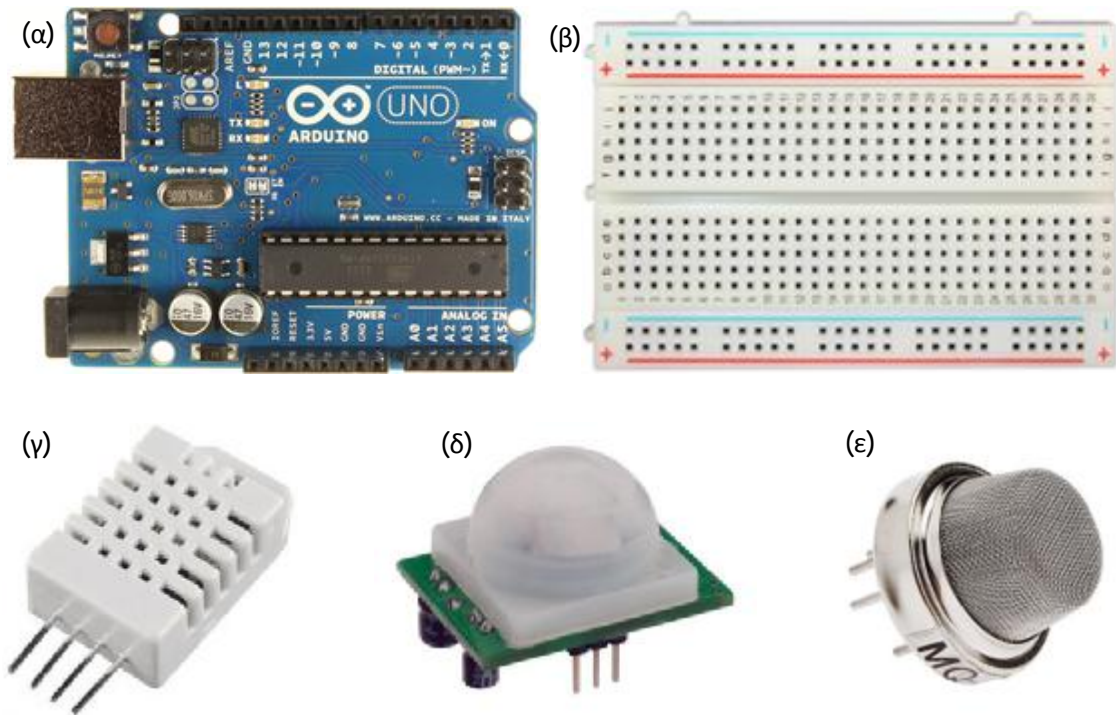
Το πρώτο από τα μειονεκτήματα είναι το κόστος απόκτησης και συντήρησης τέτοιων συστημάτων, πράγμα το οποίο λειτουργεί αποτρεπτικά στην εγκατάστασή τους. Το δεύτερο σημαντικό μειονέκτημα είναι ότι τα συστήματα αυτά είναι 'κλειστά', πράγμα το οποίο σημαίνει ότι δεν υπάρχει η δυνατότητα παραμετροποίησής τους από το χρήστη και απαιτεί την παρουσία ειδικευμένου τεχνικού-προγραμματιστή.

Για το λόγο αυτό θα κατασκευαστεί μια νέα διάταξη βασισμένη στο **Arduino** με την κατάλληλη προσαρμογή αισθητήρων κίνησης, θερμοκρασίας-υγρασίας και αερίων. Εκτός από την κατασκευή της διάταξης, θα αναπτυχθεί και κατάλληλος κώδικας ο οποίος θα δίνει τη δυνατότητα στο χρήστη να παραμετροποιεί το σύστημα κατά το δοκούν, μέσα από μια σειρά εύκολα εκτελέσιμων βημάτων. Η διάταξη θα παραμετροποιηθεί κατάλληλα για να λειτουργήσει εικονικά σε μια αποθήκη εύφλεκτων επικίνδυνων υλών και η οποία θα ειδοποιεί τον υπεύθυνο τόσο με τη χρήση λαμπτήρων LED κατάλληλου χρωματισμού (πράσινο – κόκκινο) αλλά και με κατάλληλο γραπτό μήνυμα στο e-mail του για την εκάστοτε μετρούμενη μεταβολή. Στη συνέχεια παρουσιάζεται η κατασκευή της διάταξης καθώς και η ανάπτυξη του κώδικα.

3.2 Κατασκευή διάταξης

Για την κατασκευή της διάταξης προμηθευτήκαμε μια έτοιμη μητρική πλακέτα **Arduino Uno** (Sparkfun, Ιταλία), μια πλακέτα για την κατασκευή και σύνδεση του ηλεκτρονικού κυκλώματος (**breadboard**) (Sparkfun, ΗΠΑ) καθώς και τους αισθητήρες κίνησης (Sparkfun,

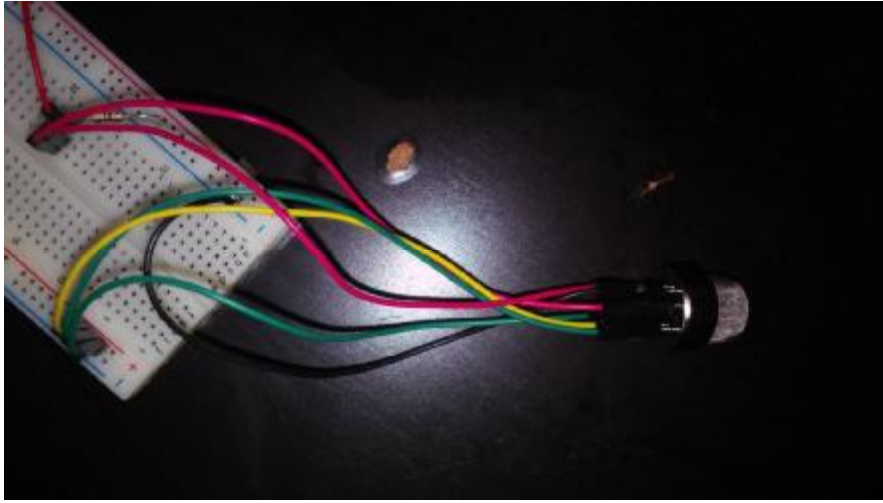
ΗΠΑ), αερίων (Adafruit, ΗΠΑ) και υγρασίας/θερμοκρασίας (Sparkfun, ΗΠΑ). Με τη χρήση καλωδίων και αντιστατών, το κύκλωμα συνδέθηκε αρχικά στο breadboard και στη συνέχεια το breadboard συνδέθηκε στο Arduino Uno. Τα μέρη της διάταξης φαίνονται στην Εικόνα 14.



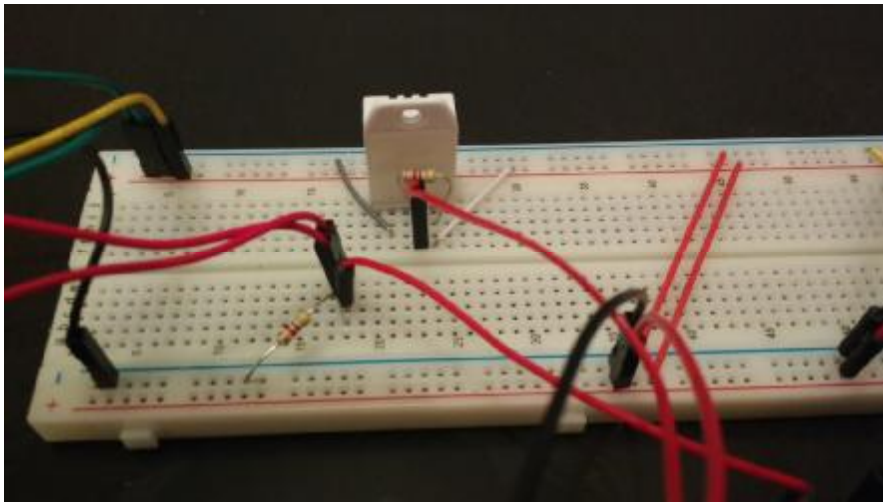
Εικόνα 14: Τα μέρη της διάταξης που κατασκευάστηκε (εικόνες από τις ιστοσελίδες των κατασκευαστών). (α) η μητρική πλακέτα Arduino Uno, (β) το breadboard, (γ) ο αισθητήρας θερμοκρασίας/υγρασίας, (δ) ο αισθητήρας κίνησης και (ε) ο αισθητήρας αερίων.

Στη συνέχεια ξεκίνησε η σύνδεση των αισθητήρων και των LED με το breadboard, η οποία παρουσιάζεται στη συνέχεια αναλυτικά. Αρχικά συνδέθηκε ο αισθητήρας αερίων. Μεταξύ του αισθητήρα και του breadboard παρεμβάλλεται σε σειρά μια αντίσταση 220 Ω για την αποφυγή βραχυκυκλώματος. Ο αισθητήρας συνδέεται στο pin A0 και στο pin για παροχή τάσης +5 V ενώ το μαύρο καλώδιο του αισθητήρα συνδέεται στη γείωση.

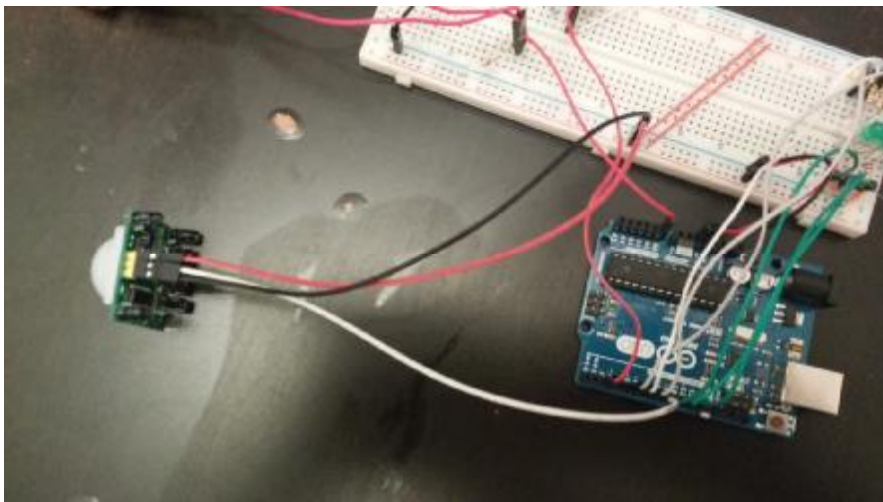
(α)



(β)



(γ)

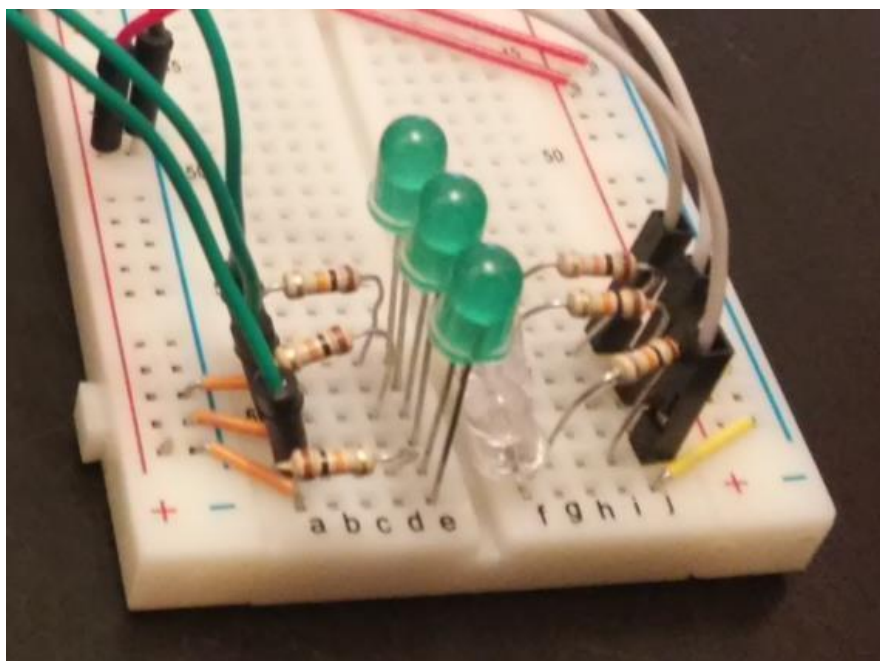


Εικόνα 15: Οι αισθητήρας (α) αερίων, (β) θερμοκρασίας/υγρασίας και (γ) κίνησης συνδεδεμένοι στο breadboard.

Ο αισθητήρας θερμοκρασίας/υγρασίας DHT 22, συνδέθηκε στο breadboard με την παρεμβολή μιας αντίστασης 220 Ω, σε σειρά, για την αποφυγή βραχυκυκλώματος. Ο αισθητήρας γειώθηκε και τροφοδοτήθηκε με τάση +5 V. Ο λόγος σύνδεσής του στην ψηφιακή θύρα στο pin 2 του Arduino είναι ότι η βιβλιοθήκη μέσω της οποίας επικοινωνεί ο αισθητήρας δεν είναι κατάλληλη για σύνδεση σε αναλογική θύρα.

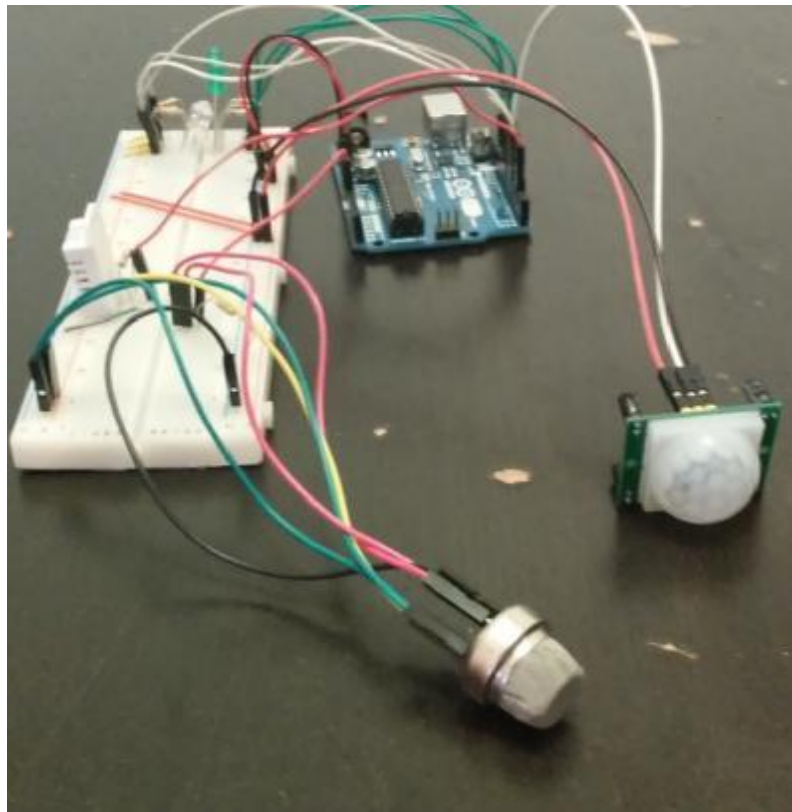
Ο αισθητήρας κίνησης συνδέθηκε με το breadboard στην ψηφιακή θύρα 8 και τροφοδοτήθηκε με τάση +5 V αφού γειώθηκε. Μεταξύ του αισθητήρα και του breadboard παρεμβλήθηκε αντίσταση 220 Ω για την αποφυγή βραχυκυκλώματος. Εικόνες από τη συνδεσμολογία των 3 αισθητήρων παρουσιάζονται στην Εικόνα 15.

Για την ευκολότερη αλληλεπίδραση μεταξύ της διάταξης και του χρήστη, τοποθετήθηκαν 3 λαμπτήρες LED πράσινου και κόκκινου χρώματος οι οποίοι λειτουργούν ως εξής: τα πράσινα LED είναι μόνιμα αναμμένα υποδεικνύοντας μετρήσεις εντός του εύρους τιμών και μη ανίχνευση κίνησης, ενώ όταν ανιχνευτεί κίνηση ή μετρήσεις εκτός των ορισμένων από το χρήστη, σβήνουν τα πράσινα LED και ανάβουν τα κόκκινα .



Εικόνα 16: Οι λαμπτήρες LED της διάταξης

Οι λαμπτήρες αντιπροσωπεύουν σε ζεύγη, τη λειτουργία των αισθητήρων αερίων, θερμοκρασίας/υγρασίας και τον αισθητήρα κίνησης (Εικόνα 16). Τα πράσινα LED γειώνονται χρησιμοποιώντας τα πορτοκαλί καλώδια ενώ τα ίδια τα LED συνδέονται στα ψηφιακά pins 9,10,11. Αντίστοιχα, τα κόκκινα LED συνδέονται στα ψηφιακά pins 7,6,5. Σε κάθε περίπτωση παρεμβάλλονται σε σειρά αντιστάσεις 220 Ω για την αποφυγή βραχυκυκλώματος. Η τελική διάταξη όπως αυτή κατασκευάστηκε φαίνεται στην Εικόνα 17.



Εικόνα 17: Η τελική διάταξη

3.3 Ανάπτυξη κώδικα

Για την επικοινωνία μεταξύ των αισθητήρων και του *Arduino*, καθώς και για την επικοινωνία με τον χρήστη, αναπτύχθηκε κώδικας ο οποίος γράφτηκε στη γλώσσα προγραμματισμού *C++* και παρουσιάζεται στο σημείο αυτό.

```
#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT22

int pirPin = 8;

int GledPin1 = 9;

int GledPin2 = 10;

int GledPin3 = 11;

DHT dht(DHTPIN, DHTTYPE);

void setup() {

  pinMode(pirPin, INPUT);

  Serial.begin(9600);

  dht.begin();

  pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(11, OUTPUT);

}
```



```
void loop() {  
  
    delay(500);  
  
    if (digitalRead(pirPin) == HIGH)  
  
    {  
  
        Serial.println("Kinisi ston xoro!!");  
  
        digitalWrite (9, LOW);  
  
        digitalWrite (7, HIGH);  
  
    }  
  
    else {  
  
        digitalWrite (9, HIGH);  
  
        digitalWrite (7, LOW);  
  
    }  
  
    float h = dht.readHumidity();  
  
    float t = dht.readTemperature();  
  
  
  
    if (t < 29) {  
  
        digitalWrite (10, HIGH);  
  
        digitalWrite (6, LOW);  
  
    }  
  
}
```

```
else {  
  
    digitalWrite (10, LOW);  
  
    digitalWrite (6, HIGH);  
  
}  
  
int val;  
  
val=analogRead(0);  
  
Serial.print(val,DEC);  
  
Serial.println(" parts per milion");  
  
if (val >200) {  
  
    digitalWrite (11, LOW);  
  
    digitalWrite (5, HIGH);  
  
}  
  
else {  
  
    digitalWrite (11, HIGH);  
  
    digitalWrite (5, LOW);  
  
}
```

```
delay(100);
```

```
if (isnan(h) || isnan(t)) {
```

```
    Serial.println("Sfalma anagnosis timon");
```

```
    return;
```

```
}
```

```
Serial.print("Humidity: ");
```

```
Serial.print(h);
```

```
Serial.print(" %\t");
```

```
Serial.print("Temperature: ");
```

```
Serial.print(t);
```

```
Serial.println(" *C ");
```

```
}
```

3.4 Επεξήγηση κώδικα

Ο κώδικας αναπτύχθηκε σε γλώσσα C++, μια σχετικά απλή γλώσσα προγραμματισμού με στόχο την εύκολη παραμετροποίηση των λειτουργιών της διάταξης από το χρήστη. Στο σημείο αυτό θα επεξηγηθεί ο κώδικας που αναπτύχθηκε.

Αρχικά με την εντολή `#include "DHT.h"` δηλώνεται η βιβλιοθήκη που θα χρησιμοποιηθεί από τον μεταφραστή κατά τη μεταγλώττιση του προγράμματος. Οι εντολές `#define DHTPIN 2` και `#define DHTTYPE DHT22` δηλώνουν το pin που είναι συνδεδεμένος ο αισθητήρας θερμοκρασίας και ο τύπος του, αντίστοιχα. Η εντολή `int pirPin = 8`, δηλώνει τη θέση του αισθητήρα κίνησης στο pin 8. Στη συνέχεια αρχικοποιούνται οι ενδείξεις για την κίνηση, τη θερμοκρασία και τη συγκέντρωση των αερίων στο χώρο με τις εντολές `int GledPin1 = 9; int GledPin2 = 10; int GledPin3 = 11;`

Έπειτα γίνεται κατάλληλη δήλωση του αισθητήρα θερμοκρασίας ώστε να εμφανίζονται στην οθόνη οι μετρήσεις που λαμβάνει ο αισθητήρας σε πραγματικό χρόνο. Η δέσμη εντολών είναι:

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
```

```
  pinMode(pirPin, INPUT);
```

```
  Serial.begin(9600);
```

```
  dht.begin();
```

Στη συνέχεια ανάβουν τα πράσινα LED για την ένδειξη μη ανίχνευσης κίνησης, φυσιολογικής θερμοκρασίας/υγρασίας και συγκέντρωσης αερίων:

```
  pinMode(9, OUTPUT);
```

```
  pinMode(10, OUTPUT);
```

```
pinMode(11, OUTPUT);  
  
}
```

Μετά τη δέσμη των παραπάνω εντολών, οι αισθητήρες καταγράφουν με βήμα 0,5 δευτερολέπτων, τη θερμοκρασία/υγρασία, τη συγκέντρωση των αερίων και την κίνηση στο χώρο με τις εντολές

```
void loop() {  
  
  delay(500);
```

Οι λόγοι που επιλέγεται το συγκεκριμένο βήμα, είναι για να μην γεμίζει η οθόνη με μηνύματα, αλλά και ότι η μέτρηση για τη θερμοκρασία διαρκεί περίπου 0,25 δευτερόλεπτα οπότε θα πρέπει να παρέχεται επαρκής χρόνος στον αισθητήρα για την καταγραφή και ενημέρωση των τιμών θερμοκρασίας/υγρασίας.

Όσον αφορά τον αισθητήρα κίνησης, σε περίπτωση που αντιληφθεί κίνηση ο αισθητήρας, σβήνει τα πράσινα LED και ανάβει τα κόκκινα, εμφανίζοντας παράλληλα το μήνυμα "Kinisi sto χορο!!". Ο λόγος που επιλέγονται λατινικοί χαρακτήρες είναι η αδυναμία του Arduino να απεικονίσει τους Ελληνικούς αντίστοιχους:

```
if (digitalRead(pirPin) == HIGH)  
  
{  
  
  Serial.println("Kinisi ston χορο!!");  
  
  digitalWrite (9, LOW);  
  
  digitalWrite (7, HIGH);  
  
}
```

```
else {  
  
    digitalWrite (9, HIGH);  
  
    digitalWrite (7, LOW);  
  
}
```

Όσον αφορά τον αισθητήρα θερμοκρασίας/υγρασίας, ο κώδικας αποθηκεύει στις μεταβλητές υποδιαστολής h, t την υγρασία και τη θερμοκρασία αντίστοιχα, προκειμένου να εμφανίζονται στην οθόνη. Αν η θερμοκρασία υπερβεί τους 29° C τότε τα πράσινα LED σβήνουν και ανάβουν τα κόκκινα:

```
float h = dht.readHumidity();  
  
float t = dht.readTemperature();  
  
  
if (t < 29) {  
  
    digitalWrite (10, HIGH);  
  
    digitalWrite (6, LOW);  
  
}  
  
else {  
  
    digitalWrite (10, LOW);  
  
    digitalWrite (6, HIGH);  
  
}
```

Τέλος, για τον αισθητήρα συγκέντρωσης αερίων, η τιμές αποθηκεύονται στην ακέραια μεταβλητή *val*, η οποία εκφράζει μέρη αερίου στο εκατομμύριο (ppm), και την εμφανίζει στην οθόνη ενώ όταν η τιμή υπερβεί τα 200 ppm για οποιοδήποτε αέριο σβήνουν τα πράσινα LED και ανάβουν τα κόκκινα LED:

```
int val;  
  
val=analogRead(0);  
  
Serial.print(val,DEC);  
  
Serial.println(" parts per milion");  
  
if (val >200) {  
  
    digitalWrite (11, LOW);  
  
    digitalWrite (5, HIGH);  
  
}  
  
else {  
  
    digitalWrite (11, HIGH);  
  
    digitalWrite (5, LOW);  
  
}
```

Σε κάθε βήμα μέτρησης, δίνεται χρόνος 0,1 δευτερολέπτων προτού ξεκινήσει ο νέος κύκλος μετρήσεων. Αυτό οδηγεί σε συνολικό χρόνο του κύκλου μετρήσεων 0,6 δευτερολέπτων (μαζί με το χρόνο αναμονής πριν ξεκινήσει ο κάθε κύκλος μετρήσεων):

```
delay(100);
```

Σε κάθε κύκλο μετρήσεων και για λόγους αδιάλειπτης λειτουργίας, γίνεται έλεγχος για την εύρυθμη λειτουργία ανάγνωσης τιμών στον αισθητήρα θερμοκρασίας/υγρασίας:

```
if (isnan(h) || isnan(t)) {  
  
    Serial.println("Sfalma anagnosis timon");  
  
    return;  
  
}
```

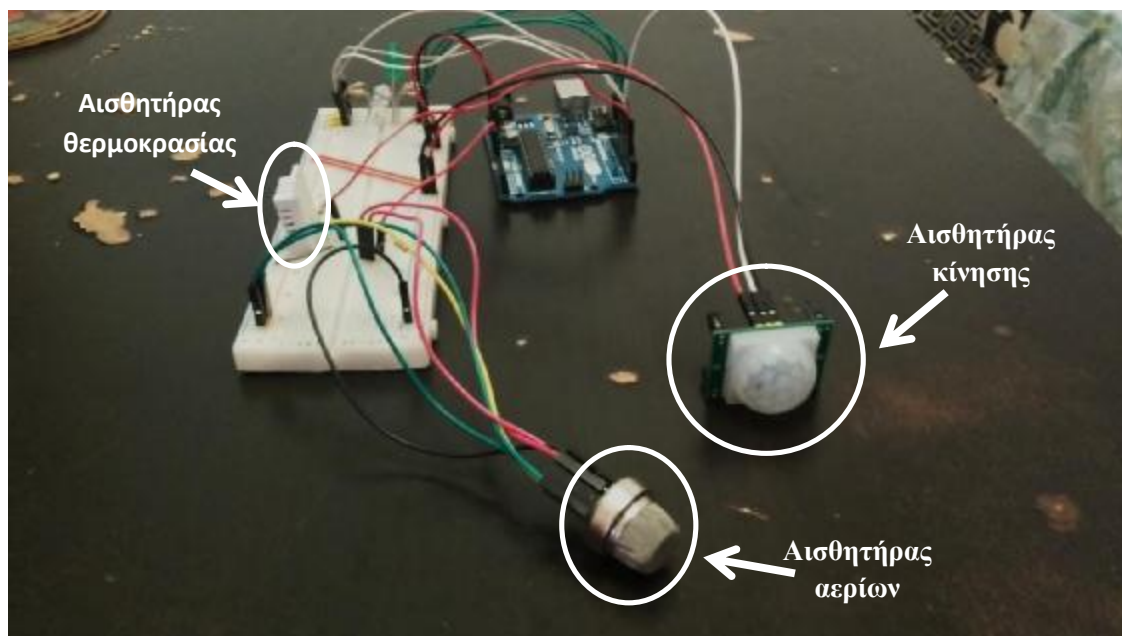
Τέλος, μετά την ολοκλήρωση των μετρήσεων, εμφανίζονται οι ενδείξεις θερμοκρασίας και υγρασίας στην οθόνη του υπολογιστή:

```
Serial.print("Humidity: "); //print stin othoni tou ipologisti tin igrasia  
  
Serial.print(h);  
  
Serial.print(" %\t");  
  
Serial.print("Temperature: "); //print stin othoni tou ipologisti tin thermokrasia  
  
Serial.print(t);  
  
Serial.println(" *C ");  
  
}
```


4^ο Κεφάλαιο

4.1 Παρουσίαση συστήματος

Η τελική μορφή της διάταξης είναι αυτή που παρουσιάζεται στην Εικόνα 18. Διακρίνονται οι 3 αισθητήρες που αποτελούν και τα βασικά στοιχεία της διάταξης. Επιλέχθηκε η συνδεσμολογία των αισθητήρων αερίων και κίνησης να γίνει με χρήση καλωδίων ώστε να είναι ευκολότερος ο προσανατολισμός του αισθητήρα στο σημείο ενδιαφέροντος. Αντίθετα, ο αισθητήρας θερμοκρασίας επιλέχθηκε να τοποθετηθεί σε σταθερό σημείο πάνω στο breadboard μιας και η τιμή της θερμοκρασίας δεν αναμένεται να διαφέρει σημαντικά στο χώρο, με εξαίρεση την περίπτωση πυρκαγιάς.



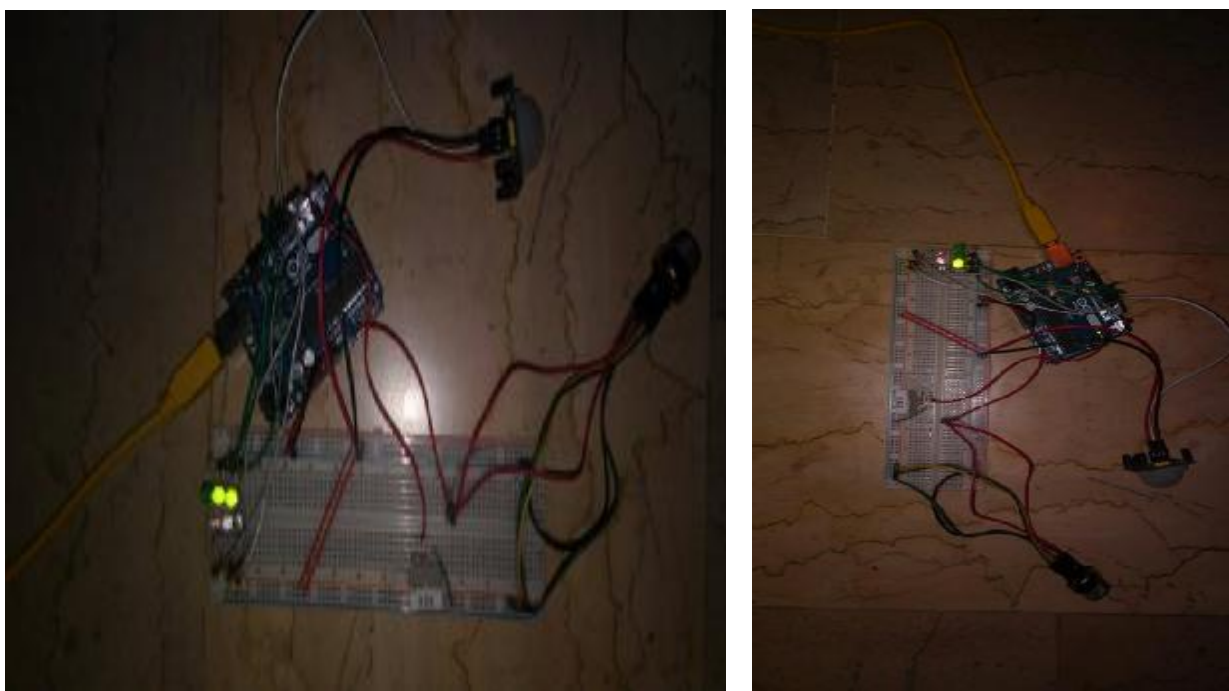
Εικόνα 18: Η τελική διάταξη

Η διάταξη τοποθετείται άνετα σε μια επιφάνεια $20 \times 20 \text{ cm}^2$ ενώ με την προσαρμογή του breadboard μπορεί να μειωθεί περαιτέρω η επιφάνεια που καταλαμβάνει. Σημαντικό στοιχείο είναι ότι η συνολική μάζα της διάταξης δεν ξεπερνά τα 500 γραμμάρια. Τα δύο αυτά στοιχεία επιβεβαιώνουν την αρχική σκέψη για την κατασκευή μιας διάταξης με μικρές διαστάσεις η οποία θα είναι εύκολα μεταφερόμενη.

Με την ολοκλήρωση της κατασκευής στην τελική της μορφή, πραγματοποιήθηκαν προκαταρκτικά πειράματα για τον έλεγχο της λειτουργίας της. Αρχικά επαληθεύτηκαν οι συνδεσμολογίες μεταξύ των αισθητήρων και των LED για τη σωστή επικοινωνία τους με τα αντίστοιχα PIN του Arduino. Στη συνέχεια, η διάταξη λειτούργησε εικονικά εκτελώντας τον αλγόριθμο του κώδικα που αναπτύχθηκε. Η πραγματοποίηση των πειραμάτων έδειξε ότι η διάταξη που κατασκευάστηκε ήταν πλήρως λειτουργική.

4.2 Αποτελέσματα

Στο σημείο αυτό, παρουσιάζονται αποτελέσματα της λειτουργίας της διάταξης σε εικονικό περιβάλλον αποθήκης. Στα στιγμιότυπα που παρουσιάζονται (εικόνες 19 και 20), φαίνεται η οθόνη του υπολογιστή με τις τιμές που απεικονίζονται με βήμα 0,6 δευτερολέπτων.



Εικόνες 19-20: Φωτεινές ενδείξεις LED

Για λόγους επίδειξης, πλησιάσαμε στον αισθητήρα ένα αναμμένο κερί, προκειμένου να καταγραφεί μεταβολή στη θερμοκρασία. Το αποτέλεσμα ήταν να σβήσει το αντίστοιχο πράσινο LED και να ανάψει το κόκκινο, εμφανίζοντας μήνυμα στην οθόνη. Αντίστοιχα, έγινε το πείραμα για λόγους επίδειξης με τον αισθητήρα κίνησης. Με μια απλή κίνηση του χεριού

μπροστά από τον αισθητήρα, έσβησε το αντίστοιχο πράσινο LED και άναψε το κόκκινο, με παράλληλη εμφάνιση του μηνύματος 'Anixnefsi kinisis'.

Αν και η διάταξη λειτούργησε άψογα, δεν κατέστη δυνατό να ενσωματωθεί στον κώδικα η δέσμη εντολών που θα επέτρεπε την αποστολή μηνύματος e-mail στο χρήστη που να τον ειδοποιεί για το ποιος αισθητήρας ανίχνευσε μεταβολή.

4.3 Μελλοντικοί στόχοι

Παρά το γεγονός ότι η αρχική προσπάθεια στα πλαίσια της πτυχιακής εργασίας χαρακτηρίζεται επιτυχημένη, υπάρχουν σημεία στα οποία χρειάζεται να βελτιωθούν σε μια νεότερη εκδοχή της διάταξης:

- Στην αρχική της μορφή η διάταξη είναι απλά δυο πλακέτες ενωμένες με μερικά καλώδια μεταξύ τους. Χρειάζεται να βελτιστοποιηθεί η συνδεσμολογία ώστε να μπορεί η διάταξη να χωρέσει σε ένα κουτί συγκεκριμένων διαστάσεων για ευκολότερη μεταφορά.
- Ο αισθητήρας θερμοκρασίας/υγρασίας είναι ένας. Σε μια νεότερη εκδοχή οι αισθητήρες θερμοκρασίας/υγρασίας θα μπορούσαν να είναι περισσότεροι, ώστε να υπάρχει αμεσότερη απόκριση σε περίπτωση τοπικής αύξησης θερμοκρασίας (π.χ. φωτιά), με αντίστοιχη μεταβολή της συνδεσμολογίας ώστε να τοποθετούνται στα σημεία ενδιαφέροντος.
- Ο αισθητήρας κίνησης είναι ένας. Η αύξηση του αριθμού τους, μπορεί να προσδώσει καλύτερη εποπτεία της κίνησης στο χώρο, ακόμη και σε 'τυφλά' σημεία, όπου ο ένας αισθητήρας αδυνατεί να 'δει'.
- Ο ένας αισθητήρας συγκέντρωσης αερίων, μπορεί να ανιχνεύσει ένα είδος αερίου. Ωστόσο σε μια αποθήκη, τα αέρια είναι πολλά περισσότερα. Είναι σημαντικό να αυξηθεί ο αριθμός των αισθητήρων αερίων προκειμένου να καλύπτεται ο αριθμός των αερίων που είναι σημαντικές οι συγκεντρώσεις τους στο χώρο, είτε πρόκειται για διαρροή, είτε για επικίνδυνα για τον άνθρωπο αέρια.

- Απαιτείται βελτιστοποίηση του κώδικα ο οποίος θα πρέπει να ενσωματώνει τον μεγαλύτερο αριθμό αισθητήρων που θα υιοθετηθεί, καθώς και τη δυνατότητα επικοινωνίας με το χρήστη, ώστε να μη χρειάζεται η παρουσία του μπροστά από την οθόνη και τα LED.

Επίλογος

Στην παρούσα πτυχιακή εργασία, παρουσιάστηκε η κατασκευή μιας ολοκληρωμένης διάταξης βασισμένη στον μικρο-ελεγκτή Arduino, στον οποίο έγινε κατάλληλη προσαρμογή αισθητήρων κίνησης, θερμοκρασίας/υγρασίας και συγκέντρωσης αερίων. Παράλληλα με την κατασκευή της διάταξης, αναπτύχθηκε και ο κατάλληλος κώδικας που επιτρέπει την παραμετροποίηση της λειτουργίας της, τα οποία παρουσιάζονται αναλυτικά μαζί με τις κατάλληλες επεξηγήσεις.

Η διάταξη είναι δυνατό να χρησιμοποιηθεί άμεσα σε πραγματικές συνθήκες λειτουργίας για την προστασία μιας αποθήκης εύφλεκτων υλών. Η διάταξη που κατασκευάστηκε, φιλοδοξούμε να συμβάλει στην ελαχιστοποίηση του κόστους λειτουργίας τέτοιων συστημάτων και την δυνατότητα παραμετροποίησης τους από το χρήστη.

Πίνακας εικόνων

Εικόνα 1: Ο άβακας των Σουμερίων, 2300 π.Χ. (εικόνα από http://users.sch.gr/mfanarioti/MHXANES/images/abacus/romanabacus1.jpg).....	3
Εικόνα 2: Ο υπολογιστής ENIAC, το μέγεθος του οποίου έφτανε να γεμίσει ένα ολόκληρο δωμάτιο (εικόνα από http://el.wikipedia.org/wiki/ENIAC#mediaviewer/File:Eniac.jpg)	4
Εικόνα 3: Ο οικιακός υπολογιστής Univas 1 (εικόνα από http://upload.wikimedia.org/wikipedia/commons/5/55/Museum_of_Science%2C_Boston%2C_MA_-_IMG_3163.JPG).....	6
Εικόνα 4: Ο υπολογιστής IBM 702 σε περιβάλλον εργασίας (εικόνα από http://upload.wikimedia.org/wikipedia/commons/1/10/BRL61-IBM_702.jpg).....	7
Εικόνα 5: Ο J. Kilby, 'πατέρας' των ολοκληρωμένων κυκλωμάτων (εικόνα από http://www.nobelprize.org/nobel_prizes/physics/laureates/2000/kilby_postcard.jpg)	7
Εικόνα 6: Μικροσίπ της δεκαετίας του '70, από την εταιρία 'Intel' (εικόνα από http://upload.wikimedia.org/wikipedia/commons/f/f8/KL_intel_D3002.jpg).....	9
Εικόνα 7: Σύγχρονος μικροεπεξεργαστής (εικόνα από http://sdi0700055.wikispaces.com/file/view/Intel-Sandy-Bridge-CPU.jpg/238926147/Intel-Sandy-Bridge-CPU.jpg)	10
Εικόνα 8: Διάφορες θύρες επικοινωνίας (εικόνα από http://www.automatica.gr/images/PlcModems/ModemImg2_1.JPG)	15
Εικόνα 9: Ένας μικρο-ελεγκτής Arduino (εικόνα από.....)	18
Εικόνα 10: Εφαρμογή του Arduino σε ρομποτικό σύστημα (εικόνα από http://media.apcmag.com/wp-content/uploads/sites/20/2013/09/apcnews2012stompy-project8-sml_mainImage1.jpg1.jpg).....	19
Εικόνα 11: Η πλακέτα ATmega 328 (εικόνα από http://www.nkcelectronics.com/assets/images/atmega328p.jpg).....	21
Εικόνα 12: Παρουσίαση των μερών του Arduino (εικόνα από http://thietbichetao.com/wp-content/uploads/2014/03/arduino_uno_R3_pinout.jpg)	23
Εικόνα 13: Σχηματική αναπαράσταση σύνθετου αισθητήρα.....	25
Εικόνα 14: Τα μέρη της διάταξης που κατασκευάστηκε (εικόνες από τις ιστοσελίδες των κατασκευαστών). (α) η μητρική πλακέτα Arduino Uno, (β) το breadboard, (γ) ο αισθητήρας θερμοκρασίας/υγρασίας, (δ) ο αισθητήρας κίνησης και (ε) ο αισθητήρας αερίων.....	31

Εικόνα 15: Οι αισθητήρας (α) αερίων, (β) θερμοκρασίας/υγρασίας και (γ) κίνησης συνδεδεμένοι στο breadboard.....	32
Εικόνα 16: Οι λαμπτήρες LED της διάταξης.....	33
Εικόνα 17: Η τελική διάταξη	34
Εικόνα 18: Η τελική διάταξη	44
Εικόνες 19-20: Φωτεινές ενδείξεις LED.....	45

Βιβλιογραφία

- [1] Σ. Α. Δρακόπουλος, Σημειώσεις μαθήματος 'Μεθοδολογία κοινωνικών επιστημών', Αθήνα: ΕΚΠΑ, 2001.
- [2] Γ. Βουτσινός και Ν. Ηλιάδης, Τεχνολογία και Ανάπτυξη, Αθήνα: Οργανισμός Εκδόσεως Διδακτικών Βιβλίων, 1999.
- [3] Britannica, «Britannica,» [Ηλεκτρονικό]. Available: <http://www.britannica.com/EBchecked/topic/130429/computer>. [Πρόσβαση 3 Μάρτιο 2015].
- [4] Βικιπαίδεια, «wikipedia.org,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Αβακας>. [Πρόσβαση 21 Μάρτιος 2015].
- [5] «History Channel,» [Ηλεκτρονικό]. Available: <http://www.history.co.uk/study-topics/history-of-ww2/code-breaking>. [Πρόσβαση 18 Ιανουάριος 2015].
- [6] wikipedia.org, «wikipedia.org,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/ENIAC>. [Πρόσβαση 1 Απρίλιος 2015].
- [7] «Wikipedia - ENIAC,» [Ηλεκτρονικό]. Available: <http://en.wikipedia.org/wiki/ENIAC>. [Πρόσβαση 16 Ιανουαρίου 2015].
- [8] «The History of Computing Project,» [Ηλεκτρονικό]. Available: <http://www.thocp.net/hardware/univac.htm>. [Πρόσβαση 25 Φεβρουάριος 2015].
- [9] «EDN Network,» [Ηλεκτρονικό]. Available: <http://www.edn.com/electronics-blogs/edn-moments/4412805/Noyce-receives-1st-IC-patent--April-25--1961->. [Πρόσβαση 14 Δεκέμβριος 2014].
- [10] Nobelprize, «Jack S. Kilby - Facts,» Nobelprize.org, [Ηλεκτρονικό]. Available: http://www.nobelprize.org/nobel_prizes/physics/laureates/2000/kilby-facts.html. [Πρόσβαση 9 Μάρτιος 2015].
- [11] B. Bemer, «A View of the History of COBOL,» *Honeywell Computer Journal*, τόμ. 5, αρ. 3, 1971.
- [12] F. E. Allen, «A History of Language Processor Technology in IBM,» *IBM Journal of Research and Development*, τόμ. 25, αρ. 5, 1981.

- [13] «Wikipedia - μικροελεγκτής,» [Ηλεκτρονικό]. Available: <http://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>. [Πρόσβαση 19 Ιανουαριος 2015].
- [14] K. Arnold, Embedded controller hardware design, Newnes, 2001.
- [15] «Wikipedia - PIC microcontroller,» [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/PIC_microcontroller. [Πρόσβαση 18 Φεβρουαριος 2015].
- [16] «MikroElektronika,» [Ηλεκτρονικό]. Available: <http://www.mikroe.com/chapters/view/75/pic-basic-book-chapter-2-programming-microcontrollers/#c2v2>. [Πρόσβαση 11 Μάρτιος 2015].
- [17] «The History Of The Arduino Microcontroller Information Technology Essay,» UK Essays, [Ηλεκτρονικό]. Available: <http://www.ukessays.com/essays/information-technology/the-history-of-the-arduino-microcontroller-information-technology-essay.php>. [Πρόσβαση 17 Δεκέμβριος 2014].
- [18] A. M. Gibb, «New Media Art, Design, And The Arduino Microcontroller: A Malleable Pool,» Master's Thesis, Pratt Institute, 2010.
- [19] «Arduino - Introduction,» Arduino, [Ηλεκτρονικό]. Available: <http://arduino.cc/en/Guide/Introduction>. [Πρόσβαση 13 Ιανουάριος 2015].
- [20] «Wiring Co,» [Ηλεκτρονικό]. Available: <http://wiring.org.co/>. [Πρόσβαση 10 Μάρτιος 2015].
- [21] «DeltaHacker,» [Ηλεκτρονικό]. Available: <http://deltahacker.gr/arduino-intro/>. [Πρόσβαση 8 Μάρτιος 2015].
- [22] S. Monk, Programming Arduino: Getting Started With Sketches, Tab Books, 2011.
- [23] «GNU - AVR,» [Ηλεκτρονικό]. Available: <http://www.nongnu.org/avr-libc/>. [Πρόσβαση 2 Μάρτιος 2015].
- [24] «Arduino - Language Reference,» [Ηλεκτρονικό]. Available: <http://arduino.cc/en/Reference/HomePage>. [Πρόσβαση 17 Δεκέμβριος 2014].
- [25] H. N. Norton, Handbook of Transducers, Prentice Hall, 1989.