



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΊΔΡΥΜΑ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΗΛΕΚΤΡΟΝΙΚΕΣ ΠΛΑΤΦΟΡΜΕΣ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ: Η ΠΕΡΙΠΤΩΣΗ ΤΟΥ ARDUINO

Πτυχιακή εργασία

της

ΝΙΚΗΣ ΚΥΡΙΑΚΟΠΟΥΛΟΥ

Επιβλέπων: Γιάννης Λιαπέρδος

Επίκουρος Καθηγητής

Σπάρτη Νοέμβριος 2017



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Ηλεκτρονικές Πλατφόρμες Ανοικτού Κώδικα:

Η Περίπτωση του Arduino

Πτυχιακή εργασία

της

Νίκης Κυριακοπούλου

Επιβλέπων: Γιάννης Λιαπέρδος

Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την – Νοεμβρίου 2017

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

.....

.....

Γιάννης Λιαπέρδος

Επίκουρος Καθηγητής

Σάββατο Νοέμβριος 2017

Copyright ©-All rights reserved. Με την επιφύλαξη παντός δικαιώματος.

Νίκη Κυριακοπούλου, 2017.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας πτυχιακής εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....

Ν. Κυριακοπούλου

Νοέμβριος 2017

Περίληψη

Στη συγκεκριμένη πτυχιακή εργασία ασχολούμαστε με τις ηλεκτρονικές πλατφόρμες και συγκεκριμένα με την περίπτωση του Arduino. Το Arduino, είναι μία πλατφόρμα ανοικτού κώδικα το οποίο έχει αρκετά χαμηλό κόστος και μπορεί να χρησιμοποιηθεί εύκολα από τον καθένα, χωρίς να χρειάζεται κάποια εξειδικευμένη γνώση.

Στην εργασία αναδεικνύονται οι διαφορές που έχει το ελεύθερο λογισμικό από το λογισμικό ανοικτού κώδικα. Οι έννοιες ελεύθερο και ανοιχτό λογισμικό συχνά χρησιμοποιούνται σαν μία έννοια, αλλά όπως θα δούμε στη συνέχεια είναι δύο διαφορετικοί ορισμοί.

Ακολουθούν κάποιες από τις βασικές εφαρμογές του Arduino και η αναλυτική τους περιγραφή.

Λέξεις κλειδιά : Arduino, πλατφόρμες, ανοιχτό λογισμικό, ελεύθερο λογισμικό, εφαρμογές

Abstract

In this thesis we will deal with electronic platforms, and in particular with the case of Arduino. Arduino, therefore, is an open source platform that has a fairly low cost and can be easily used by everyone, without the need for some specific knowledge.

Also this thesis, deals with the differences between free software and open source software. The concepts of free and open source software are often used as a concept, but as we will see below are two different definitions.

Then some of the basic applications of Arduino are extensively analyzed.

Keywords: Arduino, platforms, open source software, free software, applications

Αφιερώνεται
στους υπέροχους ανθρώπους που είναι γύρω μου
και με υποστηρίζουν

Ευχαριστίες

Ως αφορμή την εκπόνηση της πτυχιακής μου εργασίας θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες σε όλους εκείνους που συνέβαλλαν με τον δικό τους ξεχωριστό τρόπο στην ολοκλήρωση με επιτυχία των προπτυχιακών μου σπουδών.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Ιωάννη Λιαπέρδο, για την αμέριστη και πολύτιμη βοήθεια και τις συμβουλές που μου προσέφερε καθ' όλη τη διάρκεια της εκπόνησης της πτυχιακής μου εργασίας, διότι χωρίς εκείνον δεν θα είχε πραγματοποιηθεί αυτό το αποτέλεσμα.

Ακόμη, θα ήθελα να ευχαριστήσω τον Νίκο Γόγια για την απεριόριστη συμπαράσταση, βοήθεια και υπομονή που επέδειξε κατά την διάρκεια των σπουδών μου.

Βεβαίως, θα ήθελα να αφιερώσω την πτυχιακή μου εργασία στου γονείς μου, Χρήστο και Παρασκευή, και να τους ευχαριστήσω για ότι μου προσέφεραν και συνεχίζουν να μου προσφέρουν όλα αυτά τα χρόνια, δίνοντας μου τα εφόδια να επιτύχω τους στόχους μου και να τους κάνω υπερήφανους.

Επίσης, οφείλω να ευχαριστήσω τον αδερφό μου, Πέτρο, για τη συμπαράσταση και την ψυχολογική βοήθεια που μου προσέφερε έτσι ώστε να καταφέρω να φέρω εις πέρας τη συγκεκριμένη εργασία.

Πίνακας Περιεχομένων

Περίληψη	vii
Abstract	ix
1. Εισαγωγή.....	4
2. Βιβλιογραφική Ανασκόπηση	6
2.1 Ανοικτός Κώδικας.....	6
2.1.1 Ορισμός Ανοικτού Κώδικα	8
2.1.2 Ορισμός Λογισμικού Ανοικτού Κώδικα.....	9
2.1.3 Ορισμός Ελεύθερου Λογισμικού	13
2.2 Ηλεκτρονικές πλατφόρμες ανοιχτού κώδικα	14
2.2.1 Ιστορική Αναδρομή.....	19
3. Εφαρμογές.....	22
3.1 MSP430	22
3.2 Nanode.....	22
3.3 Penguin PIC32.....	23
3.4 AVR DRAGON	23
3.5 RASBERRY.....	24
3.6 SIMPP-απλή πλατφόρμα πρωτοτύπων.	25
4 Arduino	30
4.1 Ιστορική αναδρομή	30
4.2. Μοντέλα Arduino	33
4.2.1 Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino	33
4.2.3. Βασική δομή και λειτουργίες	36
4.2.3.1 Μέθοδοι Τροφοδοσίας	37
4.2.3.2 Προγραμματισμός	38
4.2.3.3 Βασικές μνήμες.....	38
4.2.3.4 Παροχές επικοινωνίας.....	39
4.2.3.5 Η δομή ενός προγράμματος και η γλώσσα	40
4.2.3.6 Ασπίδες του Arduino- Arduino Shields.....	41
4.2.3.7 Ακροδέκτες	43
5. Ανάλυση	45
5.1 Μειονεκτήματα.....	45
5.2.1 Έτοιμο για χρήση	45
5.2.2 Βιβλιοθήκες.....	45
5.2.3 Εύχρηστες Λειτουργίες	45

5.2.4 Μεγάλη κοινότητα.....	46
5.3 Προοπτική Εξέλιξης.....	46
6. Συμπεράσματα.....	47
7. Βιβλιογραφία	49

Πίνακας περιεχομένων πινάκων

Πίνακας 3. 1 Raspberry Pi Model B	25
---	----

Πίνακας περιεχομένων εικόνων

Εικόνα 2. 1 Chip TMS-1802-NC Ολοκληρωμένο κύκλωμα μικροελεγκτή	19
Εικόνα 3. 1MSP43.....	22
Εικόνα 3. 2 Nanode	22
Εικόνα 3. 3 Pinguin PIC32.....	23
Εικόνα 3. 4 AVR DRAGON.....	24
Εικόνα 3. 5Δομή προγραμματισμού MPLABΛειτουργική βιβλιοθήκη	27
Εικόνα 3. 6 Λειτουργική βιβλιοθήκη για διασύνδεση UART	27
Εικόνα 3. 7 Διαθέσιμη βιβλιοθήκη SIMPP	28
Εικόνα 4. 1Μοντέλα Arduino	33
Εικόνα 4. 2 Arduino Ethernet.....	37
Εικόνα 4. 3 Arduino Wifi Shield.....	41
Εικόνα 4. 4 Arduino Ethernet Shield	41
Εικόνα 4. 5 Arduino Wifi Shield.....	42
Εικόνα 4. 6 Arduino Proto Shield	42
Εικόνα 4. 7 Arduino Motor Shields	42

1. Εισαγωγή

Είναι γνωστό πως στις σύγχρονες κοινωνίες η τεχνολογία εξελίσσεται ραγδαία και επηρεάζει κάθε πτυχή της ανθρώπινης ζωής. Μέσα σε λίγες δεκαετίες οι υπολογιστές από δύσκλητα ογκώδη αντικείμενα που χρησιμοποιούνταν από περιορισμένες ομάδες ανθρώπων, αποκλειστικά για επιστημονικούς και στρατιωτικούς λόγους, έχουν καταλήξει να είναι διαθέσιμα στην συντριπτική πλειοψηφία των πολιτών για οποιαδήποτε βελτίωση της ποιότητας ζωής τους. Μεγάλες εμπορικές εταιρίες, όπως η Microsoft, κατάφεραν να εφοδιάσουν κάθε σπίτι με έναν επιτραπέζιο υπολογιστή. Το διαδίκτυο έδωσε απεριόριστες δυνατότητες στον ανθρώπινο πολιτισμό και έγινε τόσο απαραίτητο σε κάθε δραστηριότητα, ώστε να θεωρείται πια από πολλούς ως αναγκαίο αγαθό. Πρόσφατα, οι υπολογιστές έχουν σμικρυνθεί και βρίσκονται στα χέρια μας, συνοδεύοντας και βοηθώντας μας σε μια πληθώρα από τις πράξεις μας. Αδιαμφισβήτητα, οι υπολογιστές έχουν εισχωρήσει σε βάθος μέσα στις ανθρώπινες κοινωνίες και το γεγονός αυτό προκαλεί πολλά ερωτήματα για το μέλλον, καθώς η σχέση ατόμου και υπολογιστή συνεχώς εξελίσσεται (Schultz and van Vugt, 2015).

Παράλληλα όμως με την ανάπτυξη μεγάλων εταιριών διάθεσης υπολογιστικών συστημάτων και λογισμικού, αναπτύχθηκαν πολλά κινήματα ελεύθερου και ανοιχτού κώδικα που έχουν με την σειρά τους προσφέρει απαραίτητες συνθήκες για αυτήν την πρωτόγνωρη τεχνολογική εξέλιξη. Η ιδέα του λογισμικού ανοιχτού κώδικα ξεκίνησε στις αρχές του 1980, βασισμένη στο όραμα του Richard Stallman, ιδρυτή του GNU Project, για την ελευθερία σε σχέση με κάθε πτυχή του λογισμικού. Ο όρος ‘ανοιχτός κώδικας’ εισήχθη από τον Eric S. Raymond, προγραμματιστής και συγγραφέας του *The Cathedral and the Bazaar*, στα τέλη του 1990 στην προσπάθεια το κίνημα του ελεύθερου λογισμικού να συμβαδίσει περισσότερο με τις οικονομικές και επιχειρησιακές συνθήκες. Η άδεια του ανοιχτού κώδικα επιτρέπει την ελεύθερη χρήση, επεξεργασία και διανομή του λογισμικού, που υπόκειται σε αυτήν. Ανάμεσα στις άδειες ανοιχτού κώδικα υπάρχουν διαφοροποιήσεις, αλλά και τα τρία προηγούμενα κριτήρια είναι υποχρεωτικό να πληρούνται. Κάποια παραδείγματα από ευρέως χρησιμοποιημένες άδειες ανοιχτού κώδικα είναι οι GNU General Public License, Mozilla Public License και η Apache License.

Όλες οι σκέψεις γύρω από τον ανοιχτό κώδικα συνοδεύτηκαν από την ανάπτυξη καινούργιων καινοτόμων επιχειρησιακών μοντέλων και μιας τεράστιας κοινότητας εθελοντών, που χαρακτηρίζεται από ανεξάντλητη δημιουργικότητα. Οι

προγραμματιστικές κοινότητες αποτελούν υπόδειγμα μαζικής συνεργασίας, ανθρώπων που βρίσκονται σε διαφορετικά μέρη στον πλανήτη και μοιράζονται κοινές ιδέες και οράματα (Peatman, 2003).

Σήμερα δεν θα μας προκαλούσε έκπληξη η είδηση ότι ένα μικρό παιδί έχει αποκτήσει εξελιγμένες γνώσεις προγραμματισμού. Θεωρείται δεδομένο ότι ο καθένας έχει την δυνατότητα να προγραμματίσει τον ηλεκτρονικό του υπολογιστή και αυτό οφείλεται στην εξάπλωση λογισμικού ανοιχτού κώδικα. Το ερώτημα που προκύπτει είναι αν είναι εφικτό, με ευκολία ο καθένας να μπορεί να κατασκευάσει μόνος του ηλεκτρονικές υπολογιστικές συσκευές.

Οι ηλεκτρονικές πλατφόρμες ανοιχτού κώδικα έχουν στόχο να απαντήσουν σε αυτή την ερώτηση, προσπαθώντας να παρέχουν στο ευρύ κοινό όσο γίνεται πιο εύκολη πρόσβαση στην δημιουργία και στον προγραμματισμό ηλεκτρικών και ηλεκτρονικών κυκλωμάτων. Όσο αυξάνεται η πολυπλοκότητα των εφαρμογών προκύπτει η ερώτηση τι κρύβεται πίσω από το μαύρο κουτί. Μπορούμε να πάρουμε ενδείξεις για αυτά τα ερωτήματα από τις ηλεκτρονικές πλατφόρμες ανοιχτού λογισμικού, στις οποίες πρωτοστατεί το Arduino. Ο σκοπός της εργασίας είναι μια εκτενής επισκόπηση στις ηλεκτρονικές πλατφόρμες ανοιχτού κώδικα, στις οποίες πρωτοστατεί το Arduino (McRoberts, 2013).

2. Βιβλιογραφική Ανασκόπηση

2.1 Ανοικτός Κώδικας

Δεν υπάρχει μια ενιαία φιλοσοφία ανοιχτού κώδικα την οποία να συνυπογράφουν όλοι οι προγραμματιστές. Ορισμένοι είναι της άποψης ότι όλη η ανάπτυξη λογισμικού πρέπει να είναι ανοιχτή και ότι η χρήση του εμπορικού λογισμικού πρέπει να αποθαρρύνεται ενεργά. Άλλοι παίρνουν μια πιο χαλαρή στάση και θέλουν το λογισμικό τους να ωφελήσει τους άλλους με οποιοδήποτε τρόπο μπορεί, είτε είναι εμπορικό είτε όχι. Είναι αδύνατο να καταγράψουμε όλες τις διαφορετικές απόψεις, αλλά είναι δυνατόν να μελετήσουμε τα πνευματικά τους δώρα, τις άδειες λογισμικού ανοιχτού κώδικα. Σε γενικές γραμμές, ανοίγει τις άδειες λογισμικού προέλευσης σε τρεις κατηγορίες ανάλογα με την άδεια για εμπορική χρήση. Πρώτα υπάρχουν οι επιτρεπτές άδειες, όπως η άδεια MIT και οι άδειες BSD. Αυτές οι άδειες δημιουργούν πολύ λίγους περιορισμούς στη χρήση του πηγαίου κώδικα, είναι αρκετά συγκρίσιμες με την απελευθέρωση του πηγαίου κώδικα στον δημόσιο τομέα. Η άδεια χρήσης αναφέρει είτε, "ο δημιουργός έχει ευθύνη απολύτως", είτε ότι, "οι αρχικοί συγγραφείς πρέπει να πιστώνονται σε παράγωγα έργα".

Απέναντι από αυτές τις άδειες υπάρχουν οι ισχυρές άδειες copyleft, όπως η δημοφιλής άδεια GPL. Αυτές οι άδειες έχουν αμοιβαίο χαρακτήρα, οποιοσδήποτε παράγωγες εργασίες πρέπει επίσης να αποδεσμευτούν υπό τους ίδιους όρους άδειας. Όταν οι άδειες επιτρέπουν την ενσωμάτωση της εργασίας σε ένα εμπορικό πακέτο λογισμικού, η ισχυρή άδεια copyleft απαιτεί το συγκεκριμένο πακέτο λογισμικού να κυκλοφορήσει και με ισχυρή άδεια χρήσης copyleft. Η άδεια είναι σχεδιασμένη να απαγορεύει τη χρήση σε εμπορικό λογισμικό. Εκτός από την ρήτρα περί αμοιβαίας κατανομής, μερικές άλλες δημοφιλείς ρήτρες αποτελούν ρήτρα αντιποίνων, η οποία ανακαλεί την άδεια μόλις ο χρήστης χρησιμοποιήσει τα διπλώματα ευρεσιτεχνίας κατά τρόπο που μπορεί να βλάψει το έργο και ρήτρα περιορισμού DRM που ανακαλεί την άδεια όταν το τελικό προϊόν περιορίζει τον τελικό χρήστη στην ελεύθερη χρήση του προϊόντος με άλλα μέσα από την τροποποίηση του λογισμικού (όπως η τροποποίηση του υλικού).

Για ορισμένους προγραμματιστές οι άδειες είναι ελεύθερες, επειδή επιτρέπουν σε άλλους συγγραφείς να χρησιμοποιούν ένα κομμάτι της τεχνολογίας χωρίς να συμβάλλουν ποτέ τις βελτιώσεις τους στον αρχικό συντάκτη και οι ισχυρές άδειες copyleft είναι ισχυρές, επειδή εμποδίζουν τους χρήστες της τεχνολογίας την απελευθέρωση του σύνθετου προϊόντος υπό διαφορετικούς όρους άδειας. Οι αδύναμες

άδειες copyleft είναι ένας συμβιβασμός, επιτρέπεται στον χρήστη να χρησιμοποιεί την τεχνολογία ως συστατικό σε ένα μεγαλύτερο προϊόν το οποίο εκδίδεται με διαφορετική άδεια, αλλά οι τυχόν αλλαγές στο στοιχείο πρέπει να αποδεσμευτούν υπό την ίδια αδύναμη άδεια copyleft. Μια πρόσθετη ρήτρα ορίζει ότι παρόλο που ο πηγαίος κώδικας των σύνθετων έργων δεν χρειάζεται να απελευθερωθεί, πρέπει να γίνουν διατάξεις ώστε ο χρήστης να μπορεί να δει, να τροποποιήσει ή να αντικαταστήσει το αδύναμο συστατικό copyleft.

Στην πραγματικότητα, μόνο το μέρος που ήταν ελεύθερο από την πηγή πρέπει πρώτα να γίνει ανοιχτό. Αυτή η άδεια χρήσης είναι δημοφιλής στους εμπορικούς προγραμματιστές, ο μηχανισμός WebKit, ο οποίος χρησιμοποιείται τόσο από την Apple όσο και από τον Google ως πυρήνα των προγραμμάτων περιήγησης ιστού, υπάγεται στην άδεια LGPL. Αυτή η άδεια χρήσης τους επιτρέπει να αναπτύξουν τα δικά τους αποκλειστικά προγράμματα περιήγησης ιστού, αλλά και τους απαιτεί να μοιράζονται τις βελτιώσεις στον WebKit με τον κόσμο (και επομένως ο ένας τον άλλον). Είναι σημαντικό να σημειωθεί ότι η παράδοση του πηγαίου κώδικα κάτω από μια συγκεκριμένη άδεια χρήσης δεν αλλάζει το γεγονός ότι ο συγγραφέας κώδικα εξακολουθεί να κατέχει τα πνευματικά δικαιώματα.

Η κατοχή δικαιωμάτων πνευματικής ιδιοκτησίας επιτρέπει στον ιδιοκτήτη να απελευθερώσει εκ νέου τον κώδικα κάτω από διαφορετικές άδειες χρήσης. Στο μοντέλο επιχειρήσεων διπλής άδειας, μια εταιρεία κυκλοφορεί ένα προϊόν σε δύο εκδόσεις, μία σε άδεια ανοιχτού κώδικα και μία σε εμπορική άδεια. Εάν η άδεια ανοιχτού κώδικα είναι της ισχυρού συστατικού copyleft τότε οι εμπορικοί χρήστες υποχρεούνται να πληρώσουν για μια εμπορική άδεια. Αλλά ακόμα κι αν η άδεια ανοιχτού κώδικα είναι ανεκτική, η εμπορική έκδοση μπορεί να είναι ενδιαφέρουσα λόγω ιδιόκτητων επεκτάσεων ή εμπορικής υποστήριξης.

Το έργο που οργανώνεται σε μη κερδοσκοπικό ή κερδοσκοπικό οργανισμό συνήθως θέλει να διατηρήσει όλα τα πνευματικά δικαιώματα σε αυτόν τον κεντρικό οργανισμό. Αυτό απαιτείται όταν ο οργανισμός πρέπει να αλλάξει τους όρους άδειας χρήσης του κώδικα ή για το σύστημα διπλής αδείας. Για να διατηρηθεί το δικαίωμα πνευματικής ιδιοκτησίας σε όλο τον κώδικα, ο οργανισμός καλείται να αποκτήσει απαλλαγές από δικαιώματα πνευματικής ιδιοκτησίας από όλους τους προγραμματιστές σε όλο τον κόσμο. Πρόκειται για ένα πολύ περίπλοκο νομικό παράθυρο που πολύ λίγοι προγραμματιστές λογισμικού ενδιαφέρονται πραγματικά (McRoberts, 2013).

2.1.1 Ορισμός Ανοικτού Κώδικα

Γενικά, ο Ανοικτός κώδικας αναφέρεται σε οποιοδήποτε πρόγραμμα του οποίου ο πηγαίος κώδικας είναι διαθέσιμος για χρήση ή τροποποίηση, όπως οι χρήστες ή άλλοι προγραμματιστές θεωρούν κατάλληλα. Το λογισμικό ανοιχτού κώδικα αναπτύσσεται συνήθως ως δημόσια συνεργασία και διατίθεται ελεύθερα (Peatman, 2003).

Επίσης είναι ένα σήμα πιστοποίησης που ανήκει στην πρωτοβουλία Open Source Initiative (OSI). Οι προγραμματιστές του λογισμικού που προορίζονται να μοιραστούν ελεύθερα και ενδεχομένως να βελτιώσουν και να αναδιανέμουν σε άλλους, μπορούν να χρησιμοποιήσουν το εμπορικό σήμα Open Source αν οι όροι διανομής τους συμφωνούν με τον ορισμό του Open Source της OSI. Συνοψίζοντας, το μοντέλο ορισμών των όρων διανομής απαιτεί:

- Το λογισμικό που διανέμεται πρέπει να αναδιανεμηθεί σε οποιονδήποτε άλλον χωρίς κανένα περιορισμό.
- Ο πηγαίος κώδικας πρέπει να είναι διαθέσιμος (έτσι ώστε το μέρος που το λαμβάνει να μπορέσει να το βελτιώσει ή να το τροποποιήσει).
- Η άδεια χρήσης μπορεί να απαιτεί βελτιωμένες εκδόσεις του λογισμικού για να φέρει διαφορετικό όνομα ή έκδοση από το αρχικό λογισμικό
- Η κοινότητα λογισμικού ανοιχτού κώδικα προσφέρει μια πολύ ενδιαφέρουσα και ως επί το πλείστον ανεξερεύνητη ευκαιρία στην έρευνα της καινοτομίας. Η κοινότητα του λογισμικού ανοιχτού κώδικα έχει αποδειχθεί ότι αποτελεί σημαντική κινητήρια δύναμη της καινοτομίας στον κλάδο της πληροφορικής, με ορισμένα βασικά κομμάτια τεχνολογίας πληροφορικής που αναπτύσσονται από έργα ανοιχτού κώδικα και εργατικό δυναμικό ανάπτυξης λογισμικού που ξεπερνάει το σύνολο του εργατικού δυναμικού του κατασκευαστή εμπορικών λογισμικών των Η.Π.Α.
- Το μοντέλο λογισμικού ανοιχτού κώδικα έχει εμπνεύσει παρόμοια μοντέλα σε, μεταξύ άλλων, την τέχνη, την ανάπτυξη υλικού, τη βιοτεχνολογία.

Ένα παράδειγμα είναι το ανοιχτό έργο cola, το οποίο έχει σχεδιαστεί για να εξηγήσει την έννοια του ανοιχτού κώδικα και μπερδεύει τη χρήση εμπορικού απορρήτου από την Coca Cola αναπτύσσοντας μια κόλα εντελώς ανοιχτή. Το μοντέλο ανοικτής πηγής έχει μια ενδιαφέρουσα αλληλεπίδραση με την εμπορική ανάπτυξη. Πολλές μεγάλες και μικρές εμπορικές οντότητες χρησιμοποιούν ή / και επενδύουν σε ανάπτυξη ανοιχτού κώδικα.

Υπάρχουν όμως και συγκρούσεις, όπως για παράδειγμα όταν οι εμπορικές οντότητες παραβιάζουν τις συμφωνίες παραχώρησης άδειας εκμετάλλευσης των έργων ανοιχτού κώδικα ή όταν τα έργα ανοιχτού κώδικα καταργούν τα διπλώματα ευρεσιτεχνίας που κατέχουν εμπορικές οντότητες. Επί του παρόντος, υπάρχει μια ενδιαφέρουσα και σημαντική συζήτηση σχετικά με τη φύση και την αξία των διπλωμάτων ευρεσιτεχνίας λογισμικού, η οποία έχει ευρείες συνέπειες τόσο για την εμπορική όσο και για την ανοιχτή ανάπτυξη. Είναι σημαντικό να δοθεί αυτή η συζήτηση με τα επιστημονικά στοιχεία που απαιτούνται για να βρεθεί μια βέλτιστη, δίκαιη και ορθολογική λύση. Λόγω της φύσης της ανοιχτής πηγής, πολλές πληροφορίες μπορούν να συγκεντρωθούν με αυτοματοποιημένο τρόπο, με σχετικά μικρή προσπάθεια, όμως αυτή η περιοχή εξακολουθεί να είναι πολύ ανεξερεύνητη.

Στην ανάπτυξη ανοιχτού κώδικα μεγάλα έργα λαμβάνονται από άτομα που μπορούν να ζήσουν σε αντίθετες πλευρές του κόσμου, συνήθως ο μόνος τρόπος συντονισμού τους είναι μέσω του Διαδικτύου. Η ανάπτυξη μπορεί να συμβεί μέσω διαφόρων καναλιών, ένα κοινό πρότυπο είναι να έχουμε έναν ιστότοπο για να παρέχουμε στους χρήστες και τους νέους συνδρομητές πληροφορίες, έναν κατάλογο αλληλογραφίας και να συζητήσουμε τη διαδικασία ανάπτυξης, ένα σύστημα εντοπισμού ζητημάτων για να διαχειριστεί τι πρέπει να γίνει και μια διαχείριση αναθεώρησης σύστημα παρακολούθησης όσων έχουν γίνει στο παρελθόν.

Και εδώ είναι το καλό μέρος: στον ανοιχτό κώδικα, όλα αυτά τα συστήματα είναι προσιτά στο κοινό, επιτρέποντας στους ερευνητές καινοτομίας να έχουν σχεδόν τέλεια πληροφόρηση σχετικά με την καινοτόμο διαδικασία καθόλη τη διάρκεια του έργου. Σε αυτό το κεφάλαιο γίνεται μια σύντομη εισαγωγή στη δομή της κοινότητας ανοιχτού κώδικα, εισάγονται μερικοί από τους σημαντικότερους οργανισμούς και έργα, εξηγούνται οι μέθοδοι ανάπτυξης και συνεργασίας τους και εξηγούνται οι έννοιες των πακέτων και των εξαρτήσεων (Bolor, Shah, Shah, & Schwartz, 2016).

2.1.2 Ορισμός Λογισμικού Ανοικτού Κώδικα

Ο Bruce Perens, αποτελούσε μέλος του έργου Debian και έκανε την διατύπωση των βασικών αρχών του ορισμού του ανοικτού κώδικα στους “κανονισμούς Debian για το ελεύθερο λογισμικό” (Debian Project, 2003e). Μετά την εξαίρεσή τους από τις συγκεκριμένες συμβάσεις του έργου Debian, τα πρότυπα αυτά χρησιμοποιήθηκαν από την Open Source Initiative ή OSI για να διαμορφώσουν την προσαρμογή 1.0 (2003c) της Πρωτοβουλίας Ανοικτού Κώδικα. Η έννοια της ανοιχτής πηγής στην παραλλαγή

1.9 καταγράφει τα συνοδευτικά δέκα βασικά στοιχεία για τον προγραμματισμό που ονομάζεται ανοιχτή πηγή:

➤ Ελεύθερη κυκλοφορία: Η άδεια δεν μπορεί να περιορίζει τη πίστωση (δώρο ή διαπραγμάτευση) του προϊόντος στις δέσμες προγραμμάτων που προέρχονται από διάφορες πηγές. Δεν είναι δυνατόν να απαιτούνται υποχρεώσεις για τη μεταφορά.

➤ Πηγαίος κώδικας: Η διάδοση του λογισμικού πρέπει να επιτρέπεται τόσο στον προγραμματισμό του πηγαίου κώδικα όσο και στον συναρμολογημένο σχεδιασμό. Με την πιθανότητα να αποκλειστεί ο πηγαίος κώδικας στη διασπορά, τότε θα έπρεπε να θεωρηθεί ότι είναι δωρεάν (π.χ. μέσω του Διαδικτύου) ή με το δικό του κόστος. Ο πηγαίος κώδικας μπορεί να μην συντίθεται με τεράστιο τρόπο ή να βρίσκεται σε ένα μισό πλαίσιο (π.χ. προπαραγγεληθέν).

➤ Προγραμματισμός που παραδίδεται: Αλλαγές και υφιστάμενοι προγραμματισμοί πρέπει να επιτρέπονται. Αυτά θα έπρεπε να ανοίξουν με παρόμοια άδεια.

➤ Ειλικρίνεια του πηγαίου κώδικα του δημιουργού: Η αναδιανομή του προσαρμοσμένου πηγαίου κώδικα ενεργοποιείται μόνο περιορίζοντας την άδεια χρήσης των υποτιθέμενων patches². Η άδεια επιτρέπεται να εξασφαλίζει το όνομα του προγράμματος και να επιτρέπει τη διάδοση του προσαρμοσμένου πηγαίου κώδικα ακριβώς κάτω από ένα εναλλακτικό όνομα.

➤ Καμία διάκριση εις βάρος προσώπων ή ομάδων: Η άδεια δεν μπορεί να βλάψει κανένα άτομο ή συγκέντρωση ατόμων.

➤ Δεν υπάρχει εμπόδιο στο πεδίο εφαρμογής: Η έκταση του προϊόντος δεν επιτρέπεται να περιορίζεται από την άδεια.

➤ Αναδιανείμετε την άδεια: Τα παραχωρηθέντα δικαιώματα πρέπει να μεταβιβάζονται σε όλους τους πολίτες που ασφαλίζουν το προϊόν χωρίς να λάβουν επιπλέον άδεια.

➤ Η άδεια δεν μπορεί να περιοριστεί σε συγκεκριμένη δέσμη προγραμματισμού: Τα δικαιώματα που αποκομίζονται από ένα πρόγραμμα δεν πρέπει να βασίζονται στο εάν το πρόγραμμα είναι ένα κομμάτι μιας δέσμης προϊόντων. Με την πιθανότητα να καταργηθεί το πρόγραμμα από το πακέτο, τότε αυτό ως αποτέλεσμα δεν πρέπει να έχει τον περιορισμό των δικαιωμάτων.

➤ Η άδεια δεν μπορεί να παρεμποδίσει την κατανομή του προϊόντος με άλλο προγραμματισμό: Η άδεια δεν πρέπει να περιορίζει την κυκλοφορία του προϊόντος σε συνδυασμό με διαφορετικά έργα (π.χ. να μην κάνει τα εναλλακτικά έργα ανοιχτά).

➤ Η άδεια πρέπει να είναι αμερόληπτη για την καινοτομία: κανένας όρος δεν πρέπει να βασίζεται σε μια συγκεκριμένη καινοτομία, τρόπο ή διεπαφή.

➤ Κάθε άδεια που υποβάλλεται βάσει της Πρωτοβουλίας Ανοικτού Κώδικα εξετάζεται σε περίπτωση που συμμορφώνεται με τον "ορισμό ανοικτού πηγαίου κώδικα". Από την πιθανότητα ότι η άδεια συμπίπτει με τον ορισμό, αναγνωρίζεται στην περίληψη των φερόμενων "OSI-επιβεβαιωμένων αδειών" (Banzi, n.d.).

Η ύπαρξη διαφορετικών ορισμών ως προς την έννοια του Ελεύθερου Λογισμικού ή Λογισμικού Ανοικτού Κώδικα δεν αναιρεί την ύπαρξη των θεμελιωδών αρχών επί των οποίων βασίζεται. Βάσει αυτών των αρχών ο χρήστης έχει τη δυνατότητα ελεύθερα να χρησιμοποιήσει λογισμικά προγράμματα για όποιο λόγο εκείνος επιθυμεί, να πραγματοποιήσει μελέτη ή και τροποποίηση αυτών με τη δυνατότητα πρόσβασης του πηγαίου κώδικα τους, καθώς επίσης μπορεί και να κάνει διανομή αντιγράφων των προγραμμάτων τροποποιημένων ή μη. Ακολούθως θα αναφέρουμε τους επικρατέστερους ορισμούς βάσει της διεθνούς βιβλιογραφίας. Σύμφωνα με τον Gorling η έννοια του Λογισμικού Ανοικτού Κώδικα ορίζεται ως το σύνολο των μεθόδων που ενισχύει τη σύμπραξη για το στόχο ανάπτυξης του λογισμικού έτσι ώστε να αυξηθούν οι ωφέλειες από τη χρησιμοποίησή του. Σύμφωνα με τους Johnston και Waring η έννοια του Λογισμικού Ανοικτού Κώδικα ορίζεται η ύπαρξη του λογισμικού εκείνου το οποίο δίδεται συμπεριλαμβανομένου του πηγαίου κώδικα του δίνοντας τη δυνατότητα στο χρήστη να το βελτιώσει όταν χρειάζεται, να ανιχνεύσει και να διορθώσει όσα λάθη εντοπίσει ενώ μπορεί και να κοινοποιήσει την όποια βελτίωση και σε άλλους χρήστες. Ο Gitoc θεωρεί πως οι δυνατότητα του διαμερισμού των γνώσεων μεταξύ των χρηστών γεγονός το οποίο επιτρέπει την ενίσχυση και βελτίωση του λογισμικού καθώς επίσης και την επιπλέον βελτίωση του και από άλλους χρήστες είναι αποτέλεσμα της βασικής ιδέας του Λογισμικού Ανοικτού Κώδικα.

Η διαφορά μεταξύ του Λογισμικού Ανοικτού Κώδικα και του λογισμικού public domain είναι ότι το Λογισμικό Ανοικτού Κώδικα παρέχεται κατόπιν αδείας και υπάρχει η δυνατότητα αντιγραφής μόνο με τη διατήρηση των πνευματικών δικαιωμάτων από τον δημιουργό. Αναφορικά με την έννοια του ιδιόκτητου λογισμικού

αυτό έχει αναπτυχθεί από κάποιους για παράδειγμα από το χώρο των εμπορικών επιχειρήσεων, απαιτείται επιβολή τελών για την άδεια ενώ δεν υπάρχει δημοσιοποίηση του πηγαίου κώδικα.

Υπάρχει διαφορά μεταξύ των Ανοιχτών Προτύπων και αυτών που γίνεται χρήση όταν αναφερόμαστε σε ανοιχτό λογισμικό. Τα πρότυπα ανοιχτού λογισμικού μπορούν να ορισθούν ως πρότυπα, τα οποία χρησιμοποιούνται δημοσίως και με διαφάνεια προκειμένου να διασφαλίσουν τη διαλειτουργικότητα μεταξύ διαφορετικών συστημάτων. Τα Ανοιχτά Πρότυπα μπορούν να εφαρμοστούν σε περιβάλλοντα τόσο ιδιόκτητου όσο και Ανοιχτού Λογισμικού.

Συνήθως η διεθνής βιβλιογραφία αναφέρεται στο Λογισμικό Ανοικτού Κώδικα ως ακολούθως :

FLOSS : Ελεύθερο λογισμικό ανοικτού κώδικα

Η ικανότητα που παρέχεται στο χρήστη της ελεύθερης ανάγνωσης της μετατροπής και της νέας κατανομής όσον αφορά τον πηγαίο κώδικα στο λογισμικό αποτελεί ίσως το σπουδαιότερο χαρακτηριστικό του OSS. Είναι αλήθεια πως η χρήση του Ελεύθερου Λογισμικού λειτουργεί ανεξαρτήτως πλατφορμών και συστημάτων για παράδειγμα συστημάτων GNU/ Linux και BSD δίνοντας ροή σε μεγαλύτερο αριθμό πλατφορμών σε σχέση με άλλα ιδιόκτητα λειτουργικά συστήματα.

Επιπλέον η ανακατανομή δεν έχει κανένα εμπόδιο δεν χρεώνονται οι χρήστες ενώ παρέχεται η ευκαιρία να ενσωματώνεται ο τροποποιημένος πηγαίος κώδικας σε αυτόν που είχε δημιουργηθεί αρχικά και εκ του αποτελέσματος αναδεικνύεται η κατά πολύ γρηγορότερη πρόοδος συγκρινόμενο το λογισμικό αυτό με το λογισμικό των τυπικών κλειστών συστημάτων.

Επιπροσθέτως με το OSS δεν υφίσταται διαχωρισμός προσώπων ή ομάδων ενώ η παροχή αδειών χρήσης δεν λειτουργεί περιοριστικά σε άλλα λογισμικά καθώς επίσης δεν γίνεται αναφορά μόνο συγκεκριμένου προϊόντος. Ο ορισμός των αρχών αυτών καθώς επίσης και των ελευθεριών πραγματοποιείται μέσω του Ιδρύματος Ελεύθερου Λογισμικού.

Η δημοφιλία του Ελεύθερου Λογισμικού είναι εξαιρετικά μεγάλη στη σύγχρονη εποχή με ραγδαία εξέλιξη μη γνωστή εκτός του κόσμου της τεχνολογίας της Πληροφορικής επιστήμης (IT).

Δεν είναι πολύς ο καιρός που αυτό το λογισμικό συγκρινόμενο με το φαινόμενο των “γιγάντων της πληροφορικής” αποτελούσε μία μη δημοφιλή λύση ενώ τη σημερινή εποχή λειτουργεί ανταγωνιστικά και εναλλακτικά έναντι του ιδιόκτητου

λογισμικού γεγονός το οποίο οδήγησε στην εκδήλωση ενδιαφέροντος και από τους πολιτικούς. Οι αιτίες έχουν σαν βάση τους διαφορετικούς παραμέτρους. Οι εμπορικοί σκοποί που οδηγούν στη χρήση του γίνονται ολοένα και περισσότεροι ενώ λειτουργεί ανεξάρτητα των παραγωγών λογισμικού λειτουργεί χωρίς χρέωση και είναι αντίθετο με τα μονοπώλια.

Επιπρόσθετα ασκεί επιρροή στην πολιτική της καθημερινότητας σε όλα τα επίπεδα είτε εθνικά είτε διεθνή. Ουσιαστικά το Ελεύθερο Λογισμικό έχει καταξιωθεί στις περισσότερες χώρες αντικαθιστώντας τα εμπορικά και κλειστά λογισμικά και εξαπλώνεται τάχιστα, μολονότι η ανάπτυξη των προγραμμάτων είχε ξεκινήσει εντός του ακαδημαϊκού περιβάλλοντος.

Δεν είναι εύκολη η ακριβής εκτίμηση των εργασιών του ΕΛΛΑΚ (Το Ελεύθερο Λογισμικό/Λογισμικό Ανοικτού Κώδικα (ΕΛ/ΛΑΚ) είναι το λογισμικό που ο καθένας μπορεί ελεύθερα να χρησιμοποιεί, να αντιγράψει, να διανέμει και να τροποποιεί ανάλογα με τις ανάγκες του)(ellak.gr) επειδή διατίθενται ελεύθερα. Έρευνα που πραγματοποιήθηκε το 2009 μέτρησε μεγαλύτερο αριθμό των 230.000 έργων σε αριθμό χρηστών μεγαλύτερο των 2.000.000.

2.1.3 Ορισμός Ελεύθερου Λογισμικού

Το “ελεύθερο λογισμικό” είναι μια φράση που συχνά παίρνει την έννοια του ανοιχτού λογισμικού. Ορίζεται από το έργο GNU και έχει τις βάσεις της σε 4 είδη ελευθερίας που πρέπει να χρησιμοποιούνται σε κάθε περίπτωση στο λογισμικό. Την ελευθερία να χρησιμοποιεί κανείς το πρόγραμμα για οποιοδήποτε σκοπό, την ελευθερία να κατανοεί κανείς πώς λειτουργεί το πρόγραμμα και πώς μπορεί να το προσαρμόσει στις δικές του απαιτήσεις. Η πρόσβαση στον πηγαίο κώδικα είναι για την απόκτηση αυτής της ελευθερίας βασική προϋπόθεση, την ελευθερία να διανέμει κανείς αντίγραφα με στόχο να βοηθήσει τους συνανθρώπους του. Επίσης σε πολλές περιπτώσεις την ελευθερία να βελτιώνει κανείς το πρόγραμμα και να δημοσιοποιεί τις βελτιώσεις έτσι ώστε να ωφελείται ολόκληρη η κοινότητα από αυτές. Η πρόσβαση στο πηγαίο κώδικα είναι για την απόκτηση αυτής της ελευθερίας βασική προϋπόθεση (Boloor, Shah, Shah, & Schwartz, 2016).

2.1.3.1 Διαφορές Λογισμικού Ανοικτού Κώδικα και Ελευθέρου

Ο προγραμματισμός ανοιχτού κώδικα με την πρώτη ματιά δεν φαίνεται να διαφέρει από τον ελεύθερο προγραμματισμό. Μόνο αν εξετάσουμε τους ορισμούς τους με την αυστηρή τους σημασία, όπως αυτοί έχουν διατυπωθεί από την Πρωτοβουλία Ανοικτού Κώδικα (Open Source Initiative ή OSI) και το Ίδρυμα Ελεύθερου Λογισμικού (Free

Software 12 Foundation ή FSF), αντίστοιχα, διαπιστώνουμε ότι στην πραγματικότητα πρόκειται για διαφορετικά πράγματα (σύμφωνα με το άρθρο της Γερμανικής Wikipedia, http://de.wikipedia.org/wiki/Open_Source#Open_Source_und_Freie_Software).

Η επιλογή να χρησιμοποιηθεί η έκφραση "open source" που προτείνει το Ινστιτούτο Προοπτικής Διερεύνησης της Christine Peterson νομιμοποιείται με τον τρόπο που η λέξη "free" στα αγγλικά έχει δυο έννοιες ("ελεύθερος" και "δωρεάν"). Το Ίδρυμα του FSF κατανοεί τη λέξη με την έννοια της ελευθερίας, παρά το γεγονός ότι αυτή η λέξη συνδέεται συχνά με την ιδέα ότι κάτι είναι δωρεάν. Το πρόβλημα οξύνεται περισσότερο από το γεγονός ότι το ελεύθερο λογισμικό στις περισσότερες περιπτώσεις είναι όντως δωρεάν. Η χρησιμοποίηση της ανάθεσης ανοικτής πηγής θα καταστρέφονταν από αυτήν την ασυμφωνία και θα καθιστούσε λιγότερο απαιτητική τη "διάδοση" του προγραμματισμού ανοιχτού κώδικα από τους νομισματικούς φορείς.

Σύντομα μετά την παρουσίαση του νέου όρου, το FSF καταδίκασε την ανάπτυξη ανοιχτού κώδικα, που κατηγορήθηκε για νέα αναστάτωση, δεδομένου ότι οι στόχοι του δεν έλαβαν κοινωνικά και ηθικά κριτήρια αλλά είχαν εξειδικευμένο χαρακτήρα. Ομοίως, ο τρόπος με τον οποίο δεν είναι σημαντικό να προστατεύεται ο ελεύθερος προγραμματισμός από τη μελλοντική κατάχρηση των εξεταστών, η οποία θα μπορούσε να αλλάξει πέρα από τις δωρεάν άδειες χωρίς να υπάρχουν άδειες (π.χ. άδεια BSD).

Όντως, το FSF επιδιώκει την ανάπτυξη ανοιχτού κώδικα για να τονίσει την ευκαιρία να εξετάσει τον πηγαίο κώδικα, αλλά όχι την απεριόριστη ευελιξία της αναδιανομής και αλλαγής του. Για παράδειγμα, για το PGP (Pretty Good Privacy), το οποίο είναι προγραμματισμός ανοιχτού κώδικα, η αναδιανομή και η τροποποίηση του πηγαίου κώδικα είναι ταμπού για την πιο πρόσφατη προσαρμογή.

Ο ορισμός του ελεύθερου λογισμικού είναι, επομένως, περισσότερο περιοριστικός από τον ορισμό του ανοικτού κώδικα. Με άλλα λόγια, ενώ το ελεύθερο λογισμικό είναι λογισμικό ανοικτού κώδικα, το λογισμικό ανοικτού κώδικα δεν είναι οπωσδήποτε "ελεύθερο". Στην καθημερινή πρακτική όμως, αυτοί οι δύο όροι χρησιμοποιούνται σαν συνώνυμα, κάτι που θα κάνουμε κι εμείς συχνά στη συνέχεια της εργασίας αυτής (Banzi, n.d.).

2.2 Ηλεκτρονικές πλατφόρμες ανοιχτού κώδικα

Υπάρχουν πολλά διαφορετικά εργαλεία ανάπτυξης μικροελεγκτών διαθέσιμα για χρήση στη διδασκαλία και τη δημιουργία πρωτοτύπων. Εκείνοι που είναι πιο

δημοφιλείς εκτός της κοινότητας της ηλεκτρομηχανολογίας λειτουργούν για να προσφέρουν κάποια ισορροπία μεταξύ κόστους, επεκτασιμότητας και ευκολίας χρήσης. Το Arduino επιδιώκει επίσης την εξισορρόπηση αυτών των παραγόντων, καλύπτοντας παράλληλα ορισμένες από τις ελλείψεις των υφιστάμενων πλατφορμών. Αυτή η ενότητα παρουσιάζει μια έρευνα για μερικά από τα πιο δημοφιλή εργαλεία στην αγορά, ακολουθούμενη από μια ανάλυση του τρόπου με τον οποίο τα πλεονεκτήματα και τα μειονεκτήματα τους επηρεάζουν τις επιλογές σχεδιασμού πίσω από το Arduino.

Στο υψηλότερο επίπεδο αφαίρεσης υπάρχουν εργαλεία μικροελεγκτή όπως το MicroDig Systems Infusion, τα Phidgets και το daemon tools του Stanford. Τα δομοστοιχεία σε αυτό το επίπεδο δεν είναι γενικά προγραμματιζόμενα από τον τελικό χρήστη. Αντίθετα, έχουν ρυθμιστεί χρησιμοποιώντας ένα εργαλείο επιφάνειας εργασίας. Αυτά τα εργαλεία γενικά δεν είναι αυτόνομες συσκευές, αλλά για να είναι χρήσιμα πρέπει να συνδεθούν με έναν προσωπικό υπολογιστή. Το Infusion Systems' MicroDig είναι ένα κιβώτιο διεπαφής αισθητήρα με μια διεπαφή MIDI. Η διασύνδεσή του υλικού αποτελείται από έναν αναλογικό προς MIDI ελεγκτή με 8 αναλογικές εισόδους και διάφορες μονάδες αισθητήρων που συνδυάζονται με τον ελεγκτή.

Οι χρήστες τοποθετούν προσυσκευασμένους αισθητήρες στις εισόδους και συνδέουν τον ελεγκτή σε μια συσκευή εξόδου MIDI. Οι εκτιμήσεις των αισθητήρων εξάγονται ως εκτιμήσεις MIDI. Το MicroDig είναι χρήσιμο για τη διδασκαλία των μαθητών με κάποια γνώση του MIDI, αλλά ελάχιστες γνώσεις προγραμματισμού ή ηλεκτρονικών συστημάτων για το σχεδιασμό διεπαφών υλικού, επειδή απαιτούν λίγες νέες γνώσεις.

Είναι όμως μια ακριβή πλατφόρμα, με το βασικό πακέτο να κοστίζει περίπου € 350. Το Phidgets είναι ένα αρθρωτό σύστημα ελεγκτών αισθητήρων, ελεγκτών κινητήρων, αναγνωστών RFID και άλλων ειδικών συσκευών λειτουργίας, που όλες μαζί έχουν μια κοινή διασύνδεση USB και ένα σετ λογισμικού επιφάνειας εργασίας API. Κάθε συσκευή Phidget είναι μια αυτόνομη ηλεκτρονική συσκευή, είτε πρόκειται για αισθητήρα, μοτέρ ή ελεγκτή LED είτε για μια πιο περίπλοκη συσκευή όπως μια οθόνη LCD. Ο χρήστης δεν χρειάζεται να έχει γνώσεις ηλεκτρονικής για να χρησιμοποιήσει Phidgets.

Κάθε συσκευή συνδέεται με έναν επιτραπέζιο υπολογιστή για να αποκτήσει πρόσβαση στα δεδομένα αισθητήρων ή για να τον ελέγξει. Η δημιουργική ομάδα κυκλοφόρησε διασυνδέσεις προγραμματισμού εφαρμογών για το σύστημα σε πολλές

γλώσσες, όπως οι Visual Basic, VBA (Microsoft Access και Excel), LabView, Java, Delphi, C, C ++ και Max / MSP.

Τα δομοστοιχεία λογισμικού, οι ενότητες είναι σχετικά φθηνές, που κυμαίνονται από € 9 έως € 90 . Ωστόσο, οι συσκευές δεν μπορούν να χρησιμοποιηθούν ως αυτόνομες μονάδες και πρέπει να συνδεθούν με έναν προσωπικό υπολογιστή για να λειτουργήσουν. Η καμπύλη εκμάθησης για Phidgets είναι πιο δυσχερής απ' ό, τι για το MicroDig αλλά είναι χρήσιμη για όσους είναι εξοικειωμένοι με την ανάπτυξη λογισμικού που θέλουν να αρχίσουν να κάνουν διεπαφές υλικού.

Το daemon tools είναι ένα εργαλείο υλικού και λογισμικού υψηλού επιπέδου που αναπτύχθηκε στο πανεπιστήμιο Stanford στο τμήμα HCI , το οποίο καλύπτει τις ελλείψεις που παρουσιάζουν τα άλλα της κλάσης του. Καταρχήν, το daemon tools είναι ένα πιο ευέλικτο σύστημα και μπορεί να χρησιμοποιηθεί με άλλες πλατφόρμες υλικού, εφόσον το υλικό αυτό χρησιμοποιεί ένα λογισμικό που μπορεί να επικοινωνεί με το πρωτόκολλο του. Τα Wiring, Arduino και Phidgets κάνουν χρήση Daemon tools.

Το λογισμικό είναι γραμμένο σε Java ως πρόσθετο στοιχείο για την πλατφόρμα εργαλείων Eclipse και μπορεί θεωρητικά να τρέχει σε οποιοδήποτε λειτουργικό σύστημα με δυνατότητα Java. Όπως τα Phidgets, το υλικό αποτελείται από μια σειρά μονάδων USB σύνδεσης και λειτουργίας, κάθε μία από τις οποίες επικοινωνεί με το λογισμικό Daemon tools.. Το λογισμικό Daemon tools προσφέρει επίσης μια σουίτα εργαλείων ανάλυσης που επιτρέπουν στους χρήστες να βλέπουν τα αποτελέσματα των συσκευών τους σε ετικέτες στην οθόνη και να αναπροσαρμόζονται κατά τη διάρκεια ενός βίντεο του ατόμου που χρησιμοποιεί τη συσκευή.

Το Daemon tools είναι μια πλατφόρμα ανοιχτού κώδικα αλλά αυτή τη στιγμή , το υλικό δεν είναι εμπορικά διαθέσιμο. Προχωρώντας ένα επίπεδο αφαίρεσης, εκεί υπάρχουν ορισμένοι μικροελεγκτές μεσαίου επιπέδου. Οι ελεγκτές της σειράς αυτής διαθέτουν έναν μικροελεγκτή με τα απαραίτητα ηλεκτρονικά υποστήριξης δηλαδή κρύσταλλο, ρυθμιστής ισχύος κ.λπ. σε μια μικρή μονάδα. Αυτά τα δομοστοιχεία υποθέτουν ότι οι χρήστες μπορούν να δημιουργήσουν κυκλώματα εισόδου και εξόδου για να προσαρτηθούν στη μονάδα.

Συνήθως προγραμματίζονται σε BASIC ή σε κάποια παραλλαγή του C και προσαρτώνται στο περιβάλλον προγραμματισμού σε έναν προσωπικό υπολογιστή χρησιμοποιώντας σειριακή σύνδεση ή σύνδεση USB. Η Parallax Basic stamp είναι το πιο γνωστό από αυτά τα δομοστοιχεία.

Επίσης σε αυτήν την οικογένεια ανήκουν οι βασικοί επεξεργαστές της BasicMate , BasicX και της BASMIC ATOM. Το Arduino μοιάζει περισσότερο με αυτά. Το κύριο πλεονέκτημα που προσφέρουν οι ελεγκτές μέσου επιπέδου είναι η δυνατότητα προγραμματισμού σε γλώσσα υψηλού επιπέδου, με απλή διεπαφή προγραμματισμού.

Το μειονέκτημα είναι γενικά ότι οι γλώσσες προγραμματισμού είναι πολύ περιορισμένες και ότι τα χαμηλότερα επίπεδα του ίδιου του ελεγκτή δεν είναι καθόλου προσιτά στον χρήστη.

Τα περιβάλλοντα προγραμματισμού είναι σχεδόν όλα διαθέσιμα μόνο για τα λειτουργικά συστήματα των Windows, αν και υπάρχουν κάποιες εξαιρέσεις. Οι ίδιοι οι επεξεργαστές κοστίζουν συνήθως περίπου € 40 ενώ τα πακέτα πειραματιστών, συμπεριλαμβανομένου ενός επεξεργαστή, τροφοδοτικού και πρωτοτύπου, κοστίζουν συνήθως μεταξύ €65 και € 85. Δεδομένου ότι απευθύνονται σε αρχάριους, αυτή η τιμή, ενώ είναι μικρότερη από τους ελεγκτές υψηλού επιπέδου, εξακολουθεί να είναι υψηλή. Οι χρήστες συχνά δεν σκέφτονται να χρησιμοποιούν πολλαπλά δομοστοιχεία σε ένα έργο επειδή η τιμή είναι απαγορευτική. Είναι δεδομένο πως όταν κάποιο άτομο μαθαίνει θα κάνει λάθη. Με μικροελεγκτές μέσου επιπέδου, η πιθανότητα λάθους στην καλωδίωση ενός κυκλώματος και η καταστροφή του επεξεργαστή είναι υψηλή. Δύο από τα πιο δημοφιλή είναι η οικογένεια επεξεργαστών PIC της Microchip και οι επεξεργαστές AVR της Atmel. Αυτά είναι προγραμματισμένα σε C ή Assembly, ή BASIC, ενώ μεγάλο μέρος του προγραμματισμού περιλαμβάνει άμεση πρόσβαση στα μητρώα του επεξεργαστή.

Συνήθως απαιτείται ένας ξεχωριστός προγραμματιστής υλικού για την επικοινωνία μεταξύ του περιβάλλοντος προγραμματισμού και του επεξεργαστή. Ο προγραμματιστής πρέπει να κατασκευάσει το απαραίτητο κύκλωμα υποστήριξης για τον επεξεργαστή εκτός από οποιοδήποτε κύκλωμα αισθητήρα ή ενεργοποιητή. Η καμπύλη εκμάθησης για αυτή την κατηγορία ελεγκτών είναι η πιο απότομη από τις αναφερόμενες εδώ. Το πλεονέκτημα που προσφέρουν οι ελεγκτές χαμηλότερου επιπέδου είναι το κόστος. Σε \$ 1 - \$ 15 ανά τεμάχιο, μπορούν να χρησιμοποιηθούν πολλές φορές. Ωστόσο, υπάρχει συχνά κάποιο κόστος αρχικής εγκατάστασης για το περιβάλλον ανάπτυξης. Οι προγραμματιστές κυμαίνονται από € 35 έως € 120 και πιο πλήρως εξοπλισμένα περιβάλλοντα προγραμματισμού μπορούν να υπερβαίνουν τα € 350. Στο κόστος αυτό γίνεται απόσβεση από τη συνεχιζόμενη χρήση δηλαδή όσο περισσότεροι επεξεργαστές χρησιμοποιούνται το κόστος μειώνεται.

Υπάρχουν μερικά καλά περιβάλλοντα ανάπτυξης ανοιχτού κώδικα, ειδικά για τους ελεγκτές AVR, αλλά δεν έχουν σχεδιαστεί για αρχάριους χρήστες. Οι διασυνδέσεις είναι συνήθως πολύπλοκες και οι διεπαφές γραμμής εντολών καθώς και οι βιβλιοθήκες κώδικα απαιτούν γνώσεις εργασίας C ή C ++.

Τέλος, υπάρχουν και οι προκάτοχοι του Arduino, το Programma 2003 και το Wiring. Ένας από τους συγγραφείς ο Massimo ανέπτυξε το 2003 μια απλή πλατφόρμα μικροελεγκτών που ονομάζεται "Programma 2003" με βάση ένα chip PIC και τη γλώσσα ανοιχτού κώδικα Jai.

Ο στόχος του σχεδιασμού αυτής της πλατφόρμας ήταν να έχει κάτι όσο το δυνατόν φθηνότερο, το οποίο θα ήταν ανοικτού κώδικα και θα τρέχει σε Windows, Mac OS X και Linux. Το πρόγραμμα 2003, ωστόσο, δεν διέθετε καλή τεκμηρίωση μια ευρεία κοινότητα και χρησιμοποίησε μια σχετικά άγνωστη γλώσσα προγραμματισμού που δεν είχε ορισμένα βασικά χαρακτηριστικά.

Το Wiring είναι μονάδα ενδιάμεσου επιπέδου βασισμένη σε έναν από τους μικροελεγκτές AVR. Το Wiring επιχειρεί να αντιμετωπίσει τους περιορισμούς της διεπαφής προγραμματισμού των παραπάνω οικογενειών επεξεργαστών. Το περιβάλλον προγραμματισμού για το Wiring στο οποίο βασίζεται το περιβάλλον Arduino, έχει την προέλευσή του στο Processing , ένα περιβάλλον προγραμματισμού πολυμέσων. Το Wiring όπως το Processing, έγιναν για να διδάξουν στους φοιτητές προγραμματισμό. Το ίδιο το περιβάλλον είναι απλό και κατανοητό. Η γλώσσα χρησιμοποιεί σαφείς όρους για τα ονόματα εντολών όπως το Analog Read και το Digital. Η διεπαφή για τη μεταφόρτωση προγραμμάτων στον μικροελεγκτή είναι ελάχιστη, επιτρέποντας στους χρήστες να επικεντρωθούν στο έργο προγραμματισμού. Τα ονόματα στοιχείων μενού είναι ξεκάθαρα και διατηρούνται στο ελάχιστο. Το περιβάλλον είναι γραμμένο σε Java και είναι διαθέσιμο για Windows, OSX και Linux, σε αντίθεση με τα περισσότερα περιβάλλοντα ανάπτυξης μικροελεγκτών.

Η διαθεσιμότητά του μόνο για το OSX έχει οδηγήσει σε αυξημένη χρήση από σχεδιαστές και καλλιτέχνες που προτιμούν αυτή την πλατφόρμα. Το Wiring είναι ένα ισχυρό εργαλείο, αλλά υπάρχει περιορισμός στο ότι δεν μπορεί να κατασκευαστεί από έναν αρχάριο. Η ενότητα που πωλείται από τους προγραμματιστές διατίθεται σε μεσαίου επιπέδου τιμές, περίπου € 50 - €70, πολύ δαπανηρό για να επιτρέψει τον εύκολο πειραματισμό (Schultz and van Vugt, 2015).

2.2.1 Ιστορική Αναδρομή

Ο μικροελεγκτής είναι ένα ενσωματωμένο τσιπ του υπολογιστή που ελέγχει τα περισσότερα από τα ηλεκτρονικά εξαρτήματα και τις συσκευές που χρησιμοποιούν οι άνθρωποι σε καθημερινή βάση, από τα πλυντήρια ρούχων μέχρι τα συστήματα αντιμεπλοκής κατά το φρενάρισμα των αυτοκινήτων. Παρακάτω θα δούμε πώς προέκυψε αυτή η ιδέα ενός μικροελεγκτή και ποια είναι η ιστορία εφεύρεσης αυτού του ισχυρού τσιπ (Boxall, 2013).



Εικόνα 2. 1 Chip TMS-1802-NC Ολοκληρωμένο κύκλωμα μικροελεγκτή

Κατά τη διάρκεια των ετών 1970 και του 1971, όταν η Intel εργάζονταν στην επινοήση του πρώτου μικροεπεξεργαστή στον κόσμο, ο Gary Boone της Texas Instruments δούλευε σε μια παρόμοια ιδέα και τότε εφευρέθηκε ο μικροελεγκτής. Ο Boone σχεδίασε ένα ενιαίο τσιπ ολοκληρωμένου κυκλώματος το οποίο θα μπορούσε να κρατήσει σχεδόν όλα τα βασικά κυκλώματα για να σχηματίσει μια αριθμομηχανή. Τα μόνα που δεν ενσωματώθηκαν είναι η οθόνη και το πληκτρολόγιο.

Παραδόξως, αυτή η εξαιρετική πρόοδος στον τομέα της ηλεκτρονικής και της επικοινωνίας έχει την παγκόσμια ονομασία TMS1802NC. Ωστόσο, η συσκευή δεν ήταν συνηθισμένη. Είχε πέντε χιλιάδες τρανζίστορ που παρείχαν 3000bit μνήμης προγράμματος και 128bit μνήμης πρόσβασης. Έτσι, ήταν δυνατό να προγραμματιστεί για να εκτελέσει μια σειρά λειτουργιών (Lita, Cioc, Visan, 2006).

➤ Ιστορία του πρώτου μικροελεγκτή

Είναι γνωστό ότι οι απαιτήσεις και οι ανάγκες οδηγούν σε εφευρέσεις και ανακαλύψεις.

Η επιχείρηση Texas Instruments MOS ήταν ως επί το πλείστον μια συνηθισμένη επιχείρηση όπου εταιρείες όπως οι κατασκευαστές υπολογιστών γραφείου

απευθύνονταν με ένα σύνολο συγκεκριμένων απαιτήσεων. Η TI μετέτρεψε τις προδιαγραφές αυτές σε ένα σύνολο ολοκληρωμένων κυκλωμάτων τέσσερα, πέντε ή έξι κυκλώματα για να εκτελέσουν ή να εφαρμόσουν τις προδιαγραφές που απαιτούσαν δημοφιλείς εταιρείες όπως η Canon, η Olympia και η Olivetti. Για αυτές τις εταιρείες, ήταν πραγματικά εκπληκτικό το γεγονός ότι η TI συμπίεσε τόσες μονάδες σε μόλις τέσσερις, πέντε ή έξι μάρκες επειδή η προηγούμενη τεχνολογία που ονομαζόταν TTL χρησιμοποιούσε από εκατό έως διακόσια ολοκληρωμένα κυκλώματα. Δεν υπήρχε καμία αμφιβολία πως η δουλειά ήταν εξαιρετική (Peatman, 2003).

Κατά τη διάρκεια αυτών των χρόνων ο Gary Boone μαζί με άλλους ανθρώπους πραγματοποιούσαν εκτεταμένες έρευνες στο εσωτερικό της χώρας, καθώς επίσης στη Γερμανία, την Ιταλία και άλλες χώρες. Προσπαθούσαν να κατανοήσουν τις ανάγκες των νέων πελατών και να εργαστούν πάνω σε αυτές. Εξάλλου, καθώς τα προηγούμενα έργα τους ήταν επιτυχημένα και είχαν έναν πολύ μεγάλο αριθμό ικανοποιημένων πελατών, νέοι πελάτες ζητούσαν να πραγματοποιήσουν και σε αυτούς την ίδια δουλειά που έκαναν την προηγούμενη χρονιά σε άλλους πελάτες. Έτσι, υπήρξε μια σημαντική ζήτηση για όλο και περισσότερα από αυτά τα έργα (Boloor, Shah, Shah, & Schwartz, 2016).

Ένας κοινός κανόνας στο Τέξας ήταν "μία ταραχή, ένας φύλακας" που σημαίνει "ένα ολοκληρωμένο κύκλωμα, ένας μηχανικός". Έτσι, η Texas Instruments που είχε περίπου είκοσι μηχανικούς σχεδίασης MOS ανέπτυξε τρία ή τέσσερα από αυτά σε ένα έργο κάθε φορά. Και χρειάστηκαν περίπου έξι μήνες για να ολοκληρωθεί ένα έργο. Ως εκ τούτου, η επιχειρηματική ικανότητα ήταν ο αριθμός των μηχανικών διαιρούμενος με τον αριθμό των κυκλωμάτων, μία φορά κάθε έξι μήνες. Αν και οι απαιτήσεις αυτών των έργων διέφεραν στις λεπτομέρειες, η κύρια αρχή και η συνολική λειτουργία ήταν εντελώς πανομοιότυπες. Έτσι, οι μηχανικοί ήταν πραγματικά κουρασμένοι να κάνουν αυτά τα μονότονα έργα. Εργάζονταν για πολλές ώρες και θεώρησαν ότι θα έπρεπε να διερευνηθεί ένας καλύτερος τρόπος για την επίτευξη αυτών των έργων.

Ως αποτέλεσμα, ήρθε το πρώτο τσιπ μικροελεγκτή TMS 100. Μια βασική τεχνική πτυχή ήταν ότι η τεχνολογία σχεδιασμού εκείνων των ημερών ήταν αρκετά αναποτελεσματική όσον αφορά τον τρόπο με τον οποίο χρησιμοποιήθηκε το πυρίτιο. Έπρεπε να επιλεγούν διαφορετικές αρχιτεκτονικές, ώστε να τονιστούν οι τακτικές δομές και η μνήμη.

Επιπλέον, έπρεπε να δοθεί έμφαση στο να πραγματοποιηθούν τα αντίστοιχα βήματα έτσι ώστε η ιδέα bit-slice όπου τα πάντα που ταιριάζουν με κάθε ένα από τα

τέσσερα κομμάτια ενός αθροιστή τεσσάρων δυαδικών ψηφίων έχει σχεδιαστεί να έχει την ίδια διάσταση σε φυσικούς όρους. Μόνο τότε θα ταιριάζει μαζί και θα ταιριάζει τέλεια. Τελικά, επιτεύχθηκε ένας άλλος παράγοντας έναντι τριών ή τεσσάρων ως προς την πυκνότητα πυριτίου.

Παρά τον περιορισμό των αρχιτεκτονικών που ήταν κυρίως προσανατολισμένες στη μνήμη και στο pitch-match, ο τότε μικροελεγκτής ήταν πολύ καλύτερος από την προηγούμενη τεχνολογία TTL και κατέληξε σε πολύ αποτελεσματική χρήση του πυριτίου. Αυτός ο μικροελεγκτής χρησιμοποιήθηκε στην Texas Instruments στα είδη αριθμομηχανής μεταξύ των ετών 1972 και 1974. Η συσκευή βελτιώθηκε περαιτέρω με την πάροδο των ετών. Η TI προσέφερε αυτόν τον μικροελεγκτή για πώληση στην ηλεκτρονική βιομηχανία το 1974. Διατέθηκε σε διάφορες διαμορφώσεις μεγεθών RAM και ROM. Μέχρι το 1983, σχεδόν 100 εκατομμύρια TMS 1000 είχαν ήδη πωληθεί (Hintz, Tabak, 1992).

➤ Οι επαναστατικές ηλεκτρικά διαγράψιμες μνήμες

Ήταν κατά τη διάρκεια της δεκαετίας του 1990 που προηγμένοι μικροελεγκτές με ηλεκτρικά διαγράψιμες και προγραμματιζόμενες ROM μνήμες όπως η μνήμη flash άρχισαν να κατακλύζουν την αγορά ηλεκτρονικών. Το μοναδικό χαρακτηριστικό αυτών των μικροελεγκτών είναι ότι μπορούν να προγραμματιστούν, να διαγραφούν και να επαναπρογραμματιστούν με τη βοήθεια μόνο ηλεκτρικών σημάτων. Πολλοί μικροελεγκτές που χρησιμοποιούνται σήμερα, όπως αυτοί που διατίθενται από την Atmel και τη Microchip, χρησιμοποιούν την τεχνολογία μνήμης flash (Boloor, Shah, Shah, & Schwartz, 2016).

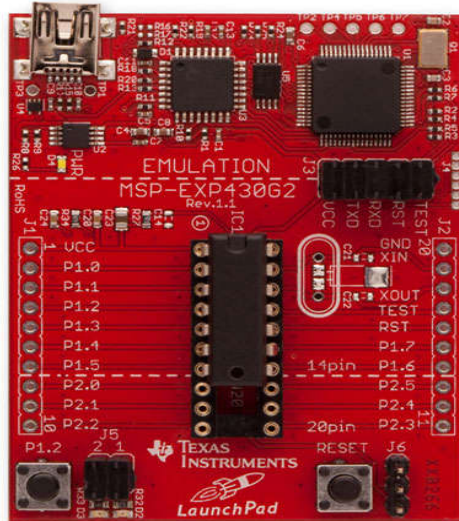
➤ Οι τελευταίοι μικροελεγκτές

Σήμερα, εκτός από τα gadgets γενικού σκοπού, δημιουργούνται μοναδικοί μικροελεγκτές για τομείς όπως ο φωτισμός, η αυτοκινητοβιομηχανία, οι επικοινωνίες και τα καταναλωτικά αγαθά χαμηλής ισχύος. Οι σημερινοί μικροελεγκτές όπως το AVR και το PIC έχουν γίνει μικρότεροι και ακόμα πιο ισχυροί. Για παράδειγμα, υπάρχουν μικροσκοπικοί μικροελεγκτές διαθέσιμοι μικροί και φθηνοί για να χρησιμοποιηθούν σε απλά προϊόντα όπως οδοντόβουρτσες και παιχνίδια (Stewart, 2016).

3. Εφαρμογές

Πιο κάτω θα δούμε μερικές από τις εφαρμογές του Arduino :

3.1 MSP430



Εικόνα 3. 1MSP43

Το MSP430 LaunchPad είναι ένας μικροελεγκτής χαμηλού κόστους, χαμηλής κατανάλωσης ενέργειας από την Texas Instruments, κατασκευαστή του BeagleBone. Το Value Line LaunchPad είναι διαθέσιμο για \$ 4.30 σε ένα κιτ που έρχεται με ένα δεύτερο τσιπ. Το τσιπ MSP430 προσφέρει επίσης μια λειτουργία εξοικονόμησης ενέργειας που ξυπνάει σχεδόν αμέσως, κάτι που μπορεί να είναι ιδανικό για απομακρυσμένους αισθητήρες. Σε ένα κλάσμα της τιμής του Arduino, το LaunchPad μοιάζει με μια καλή εναλλακτική λύση, τουλάχιστον για σχετικά απλά έργα. Ωστόσο, περιλαμβάνει 512 bytes μνήμης RAM σε σύγκριση με τα 2 kB του Arduino Uno, οπότε ο Arduino μπορεί ακόμα να κερδίσει για πιο περίπλοκες διεργασίες (Hintz, Tabak, 1992).

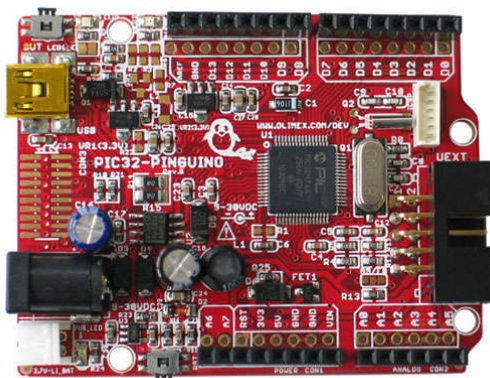
3.2 Nanode



Εικόνα 3. 2 Nanode

Το Nanode έχει σχεδιαστεί για να λειτουργεί σαν ένα Arduino, αλλά είναι ειδικά σχεδιασμένο για έργα που συνδέονται με το Διαδίκτυο. Διαθέτει τον ίδιο επεξεργαστή ATmega328 που χρησιμοποιεί το Arduino Uno και μπορείτε να το προγραμματίσετε με το IDE του Arduino. Το Nanode χρησιμοποιεί το API ανοικτών δεδομένων API Cosm για να συνδεθεί στο Web για εργασίες που περιλαμβάνουν την αποστολή δεδομένων στο σύννεφο, μετά από ηλεκτρονικές πηγές ή ως διακομιστής για έναν μικρό, απλό ιστότοπο. Κάνει ένα καλό εργαλείο ανάπτυξης για αισθητήρες, οθόνες ή χειριστήρια συνδεδεμένα στο Web. Το Nanode RF πωλείται για \$ 56,57, υψηλότερη τιμή από την Arduino (McRoberts, 2013).

3.3 Pinguin PIC32



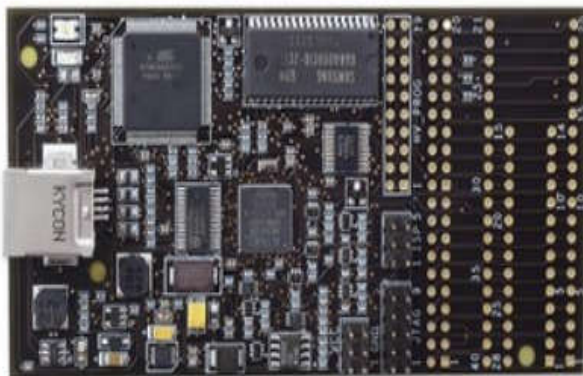
Εικόνα 3.3 Pinguin PIC32

Πρόκειται για ένα στερεό πρωτότυπο εργαλείο που σχεδιάστηκε αρχικά για φοιτητές τέχνης. Το Pinguino έχει το ίδιο μέγεθος και σχήμα με το Arduino Uno και όπως το Arduino, το Pinguino διαθέτει υλικό ανοικτού κώδικα για να τρέξει ένα IDE ανοικτού κώδικα. Παρά τις ομοιότητες, όμως, τα φόρουμ Arduino δεν παρέχουν υποστήριξη για το Pinguino και η εταιρεία συμβουλεύει τους αγοραστές ότι το Pinguino μπορεί να μην λειτουργεί με βιβλιοθήκες ή σκίτσα Arduino. Η Pinguino πωλείται για \$ 25.99, αλλά η εταιρεία συνιστά στους αγοραστές να εξασφαλίσουν ότι είναι πεπειραμένοι και άνετοι με την πρώτη τεχνολογία - το Pinguino δεν είναι μικροελεγκτής για αρχάριους (Hintz, Tabak, 1992).

3.4 AVR DRAGON

Με το Atmel AVR Dragon, η Atmel έχει θέσει ένα νέο πρότυπο για εργαλεία ανάπτυξης χαμηλού κόστους. Το AVR Dragon υποστηρίζει όλες τις λειτουργίες προγραμματισμού για τις οικογένειες συσκευών Atmel AVR. Περιλαμβάνει επίσης πλήρη υποστήριξη εντοπισμού σφαλμάτων για τις περισσότερες συσκευές AVR. Σε

ένα κλάσμα της τιμής που συνδέεται παραδοσιακά με αυτό το είδος εργαλείων, το AVR Dragon θα ικανοποιήσει όλες τις ανάγκες προγραμματισμού και εντοπισμού σφαλμάτων. Η ευέλικτη και ασφαλής δυνατότητα αναβάθμισης του υλικολογισμικού επιτρέπει στο front-end του λογισμικού να αναβαθμίσει εύκολα το AVR Dragon για να υποστηρίξει νέες συσκευές (Margolis, 2016).



Εικόνα 3. 4 AVR DRAGON

Το Atmel AVR Dragon μπορεί να χρησιμοποιηθεί με εξωτερικό πίνακα στόχου. Ωστόσο, η πλατφόρμα πρωτοτύπου επιτρέπει τον απλό προγραμματισμό και τον εντοπισμό σφαλμάτων χωρίς πρόσθετο υλικό εκτός από τα καλώδια σύνδεσης. Το AVR Dragon τροφοδοτείται από το καλώδιο USB και μπορεί επίσης να προμηθεύσει εξωτερικό στόχο έως και 300mA (από τον σύνδεσμο VCC, 5V) κατά τον προγραμματισμό ή τον εντοπισμό σφαλμάτων. Εάν ο στόχος τροφοδοτείται ήδη από μια εξωτερική πηγή ενέργειας, το AVR Dragon θα προσαρμόσει και θα μετατρέψει όλα τα σήματα μεταξύ του στόχου και του AVR Dragon. Το AVR αποτελεί μια από τις εμπορικές λύσεις με δυναμικό χαρακτήρα.

3.5 RASBERRY

Μόλις πάνω από επτά εκατομμύρια Raspberry Pis έχουν πωληθεί σε όλο τον κόσμο. Αυτό κάνει αυτό τον υπολογιστή μεγέθους πιστωτικής κάρτας έναν από τους πιο επιτυχημένους υπολογιστές που έχει κάνει ποτέ το Ηνωμένο Βασίλειο. Είναι πραγματικά εκπληκτικό το γεγονός ότι ένα μικροσκοπικό φιλανθρωπικό ίδρυμα του Ηνωμένου Βασιλείου, αφιερωμένο στη διάθεση υπολογιστών και επιστημών πληροφορικής σε προσιτές τιμές και διαθέσιμο σε ανθρώπους από όλα τα κοινωνικά στρώματα, θα πρέπει να καταλήξει σε μια τόσο απίστευτη ιστορία επιτυχίας στα χέρια τους (Stewart, 2016).

	Raspberry Pi Model B
Price	\$35
Size	8.6cm x 5.4cm x 1.7cm
Memory	512MB
Clock Speed	700 MHz
On Board Network	10/100 wired Ethernet RJ45
Multitasking	Yes
Input voltage	5 V
Flash	SD Card (2 to 16G)
USB	Two, peripherals OK
Operating System	Linux distributions
Integrated Development Environment	Scratch, IDLE, anything with Linux support

Πίνακας 3. 1 Raspberry Pi Model B

Όπως και το «πρότυπο» Raspberry Pi, το Raspberry Pi 2 Μοντέλο B, υπάρχει επίσης μια μικρότερη έκδοση του Raspberry Pi που μπορούν να χρησιμοποιηθούν. Το Pi Model A + είναι μια περιορισμένη έκδοση του το αρχικό Raspberry Pi, με λίγο λιγότερη ενέργεια στη διάθεσή της και λιγότερα συνδέσεις σε αυτό.

3.6 SIMPP-απλή πλατφόρμα πρωτοτύπων.

Η SIMPP ή (απλή πλατφόρμα πρωτοτύπων) είναι μια ηλεκτρονική πλατφόρμα πρωτότυπου ανοικτού κώδικα που χρησιμοποιείται για την κατασκευή ηλεκτρονικών κυκλωμάτων ή έργων που χρησιμοποιούν κυρίως μικροελεγκτή. Το SIMPP είναι φυσικό προγραμματιζόμενο υλικό πλακέτας (μικροελεγκτής) και χρησιμοποιεί ένα λογισμικό ή ένα IDE (Integrated Development Environment) που εκτελείται σε υπολογιστή στον οποίο γράφεται κώδικας υπολογιστή ή μικροελεγκτής και από το IDE που μεταφορτώνεται στο Hardware (SIMPP). Το SIMPP χρησιμοποιεί "μικροελεγκτή μικροεπεξεργαστή PIC Microchip" καθώς είναι ελεγκτής πυρήνα και Microchip IDE (MPLAB) για το λογισμικό. SIMPP Χρησιμοποιεί κύκλωμα προγραμματιστή PICKIT-2 και ενσωματωμένο υλικό λογισμικού για την άμεση σύνδεση του SIMPP με το MPLAB IDE μέσω USB. Το SIMPP μπορεί να λειτουργήσει σαν πίνακας ανάπτυξης και σαν πίνακας προγραμματιστών. Ακολουθούν τα βασικά χαρακτηριστικά του SIMPP:

3.1 Κύρια τμήματα

Εξωτερική τροφοδοσία ρεύματος - Απαιτείται ρυθμιζόμενη τροφοδοσία προσαρμογέα dc με τάση από 9v έως 12V και μέγιστο ρεύμα 1amp. Απαιτείται εξωτερική τροφοδοσία για έργα που απαιτούν πηγή τάσης υψηλότερη από 5V.

Διεπαφή USB με τον υπολογιστή. Παρέχει επίσης την πηγή ισχύος για το υλικό και τις εφαρμογές κυκλώματος που λειτουργούν σε τροφοδοσία 5V.

Ενότητα ισχύος - πηγή τροφοδοσίας εξωτερικού κυκλώματος. Το εξωτερικό κύκλωμα μπορεί να τροφοδοτήσει σε αυτή την περιοχή τάσης ενότητας της εξωτερικής τροφοδοσίας, 5v και 3.3V.

GPIO είσοδος και έξοδος γενικής χρήσης, είναι μια θηλυκή εσοχή όπου ένα καλώδιο ή ένα αρσενικό καλώδιο DuPont μπορούν να συνδεθούν για να συνδεθούν με τον μικροελεγκτή.

Ρυθμίσεις Λειτουργίας - Το SIMPP μπορεί να ρυθμιστεί σε πίνακα ή προγραμματιστή. Οι ρυθμίσεις ανάπτυξης είναι εκεί όπου ένα νέο κύκλωμα εφαρμογών μπορεί να δημιουργηθεί χρησιμοποιώντας το SIMPP. Ενώ οι ρυθμίσεις του προγραμματιστή είναι όπου το SIMPP χρησιμοποιείται για τον προγραμματισμό άλλων μικροελεγκτών PIC. Από τη στιγμή που το SIMPP χρησιμοποιεί τις πλατφόρμες Pickit2 και MPLAB, υποστηρίζεται πολύς μικροελεγκτής από το SIMPP κατά τη διάρκεια των ρυθμίσεων του προγραμματιστή.

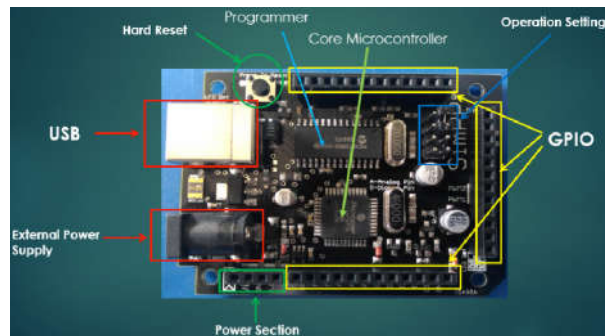
Hard Reset - Το κουμπί χρησιμοποιείται για την επαναφορά του μικροελεγκτή στην αρχική του διεύθυνση προγραμματισμού.

Προγραμματιστής - Αυτός που επικοινωνεί μεταξύ του MPLAB και του μικροελεγκτή, για να κατεβάσει το πρόγραμμα που έχει συνταχθεί από το MPLAB στον κεντρικό μικροελεγκτή. Έχει τη διασύνδεση USB και μέρος του κυκλώματος pickit2 με προ-φορτωμένο firmware pickit2.

Μικροελεγκτής πυρήνα - Είναι ο εγκέφαλος του SIMPP. Αυτό γίνεται όταν εκτελούνται όλες οι υπολογιστικές διαδικασίες.

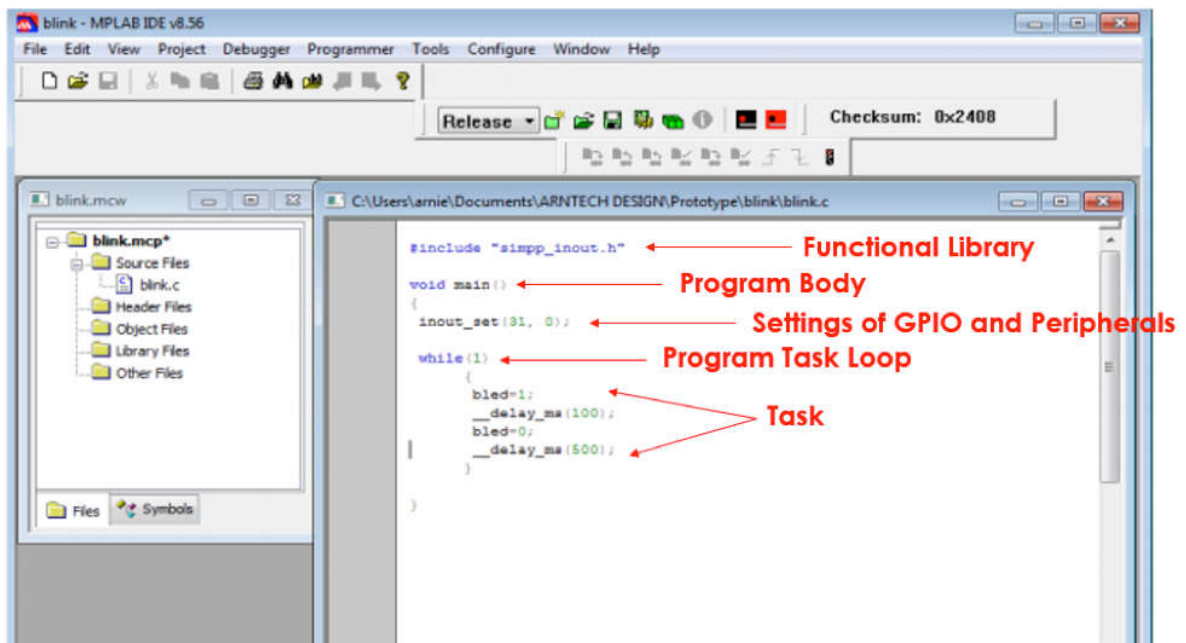
Το SIMPP χρησιμοποιεί το MPLAB IDE και προγραμματίζει σε γλώσσα C. Προκειμένου να προγραμματιστεί ο υπολογιστής σε γλώσσα C στο MPLAB, απαιτείται επίσης να εγκατασταθεί ένα άλλο όνομα λογισμικού PICC για τον προγραμματισμό C που θα καταρτιστεί στο MPLAB. Τα λογισμικά MPLAB και PICC μπορούν να μεταφορτωθούν από το διαδίκτυο. Δημιουργήθηκε δομή προγραμματισμού και λειτουργικές βιβλιοθήκες για την ανάπτυξη έργων. Αυτό θα

βοηθήσει τον προγραμματισμό με διεξάγεται με απλό και εύκολο τρόπο. Στο σχήμα 2, εμφανίζεται η δομή προγραμματισμού (Schultz and van Vugt, 2015).



Εικόνα 3. 5 Δομή προγραμματισμού MPLAB Λειτουργική βιβλιοθήκη

Μορφή αρχείου κεφαλίδας που δημιουργήθηκε για να βοηθήσει το πρόγραμμα να γίνει με ευκολία. Σύνθετη εργασία ή υπολογισμός δημιουργήθηκε και ορίστηκε με λειτουργικό τρόπο και ο χρήστης θα χρησιμοποιήσει απλά τη λειτουργία. Στο σχήμα 3 παρουσιάζεται μια λειτουργία δείγματος (Hintz, Tabak, 1992).



Εικόνα 3. 6 Λειτουργική βιβλιοθήκη για διασύνδεση UART

Το ανώτερο τμήμα της λειτουργικής βιβλιοθήκης χρησιμοποιήθηκε για τον καθορισμό των λειτουργιών που περιλαμβάνονται στη βιβλιοθήκη. Η παραπάνω εικόνα είναι για το χρήστη για να κατανοήσει σωστά τη σημασία και τη χρήση των λειτουργιών.

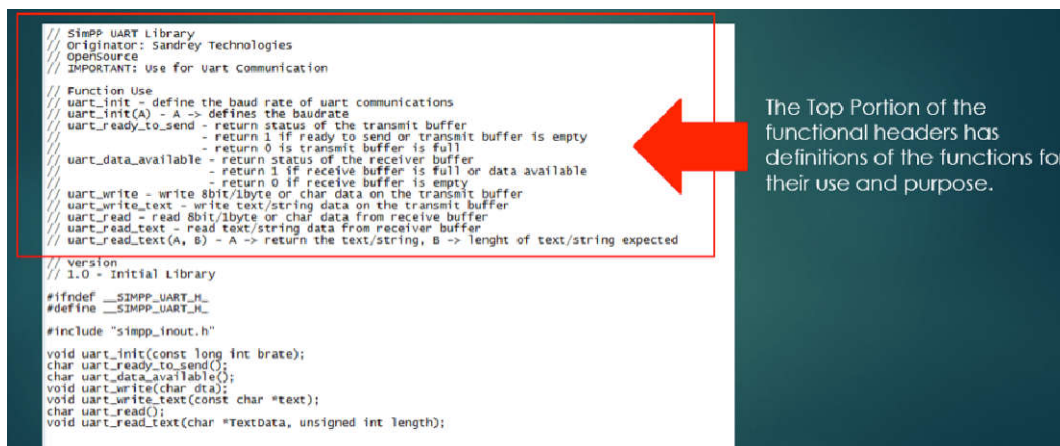
Βασικός κορμός προγράμματος - ορισμός για ολόκληρη την κάλυψη του προγράμματος. Η εκτέλεση της εργασίας του μικροελεγκτή καλύπτεται στον κορμό του προγράμματος. Περιλαμβάνει τις ρυθμίσεις και την εργασία του μικροελεγκτή.

Ρυθμίσεις του GPIO και των περιφερειακών συστημάτων σε αυτή την ενότητα ή το μέρος της δομής του προγράμματος, ο χρήστης ορίζει τις αρχικές ρυθμίσεις του GPIO και των περιφερειακών που θα χρησιμοποιηθούν στην εργασία μικροελεγκτή. Αυτό θα πρέπει να οριστεί πρώτα πριν από την πραγματική εργασία.

Πρόγραμμα βρόχου εργασιών - αυτό που περιλαμβάνεται σε αυτή την ενότητα είναι ότι ορίζεται η εργασία του μικροελεγκτή. Ο κύκλος εργασιών του προγράμματος είναι απλώς μια συνάρτηση βρόχου όπου η εργασία μικροελεγκτή γίνεται με επαναλαμβανόμενο τρόπο.

Η διαθέσιμη βιβλιοθήκη είναι οι οδηγίες που δίνονται στον κεντρικό μικροελεγκτή για το τι πρέπει να εκτελέσει και αυτό θα αντικατοπτρίζει στην εφαρμογή του εξωτερικού κυκλώματος.

Το σχήμα 4 δείχνει τη λίστα της διαθέσιμης βιβλιοθήκης για τη χρήση του SIMPP κατά την ανάπτυξη του έργου (Warren, Adams, & Molle, 2011).



```

SIMPP UART Library
Originator: Sandrey Technologies
OpenSource
IMPORTANT: Use for uart Communication

Function Use
uart_init - define the baud rate of uart communications
uart_init(A) - A -> defines the baudrate
uart_ready_to_send - return status of the transmit buffer
                  - return 1 if ready to send or transmit buffer is empty
                  - return 0 if transmit buffer is full
uart_data_available - return status of the receiver buffer
                  - return 1 if receive buffer is full or data available
                  - return 0 if receive buffer is empty
uart_write - write 8bit/1byte or char data on the transmit buffer
uart_write_text - write text/string data on the transmit buffer
uart_read - read 8bit/1byte or char data from receive buffer
uart_read_text - read text/string data from receiver buffer
uart_read_text(A, B) - A -> return the text/string, B -> length of text/string expected

Version
1.0 - Initial Library
#ifdef _SIMPP_UART_H_
#define _SIMPP_UART_H_
#include "simpl_uart.h"

void uart_init(const long int brate);
char uart_ready_to_send();
char uart_data_available();
void uart_write(char data);
void uart_write_text(const char *text);
char uart_read();
void uart_read_text(char *textdata, unsigned int length);

```

Εικόνα 3. 7 Διαθέσιμη βιβλιοθήκη SIMPP

Πιο συγκεκριμένα, πληθώρα διαφορών σημειώνονται ανάμεσα στο SIMPP από την άλλη πλατφόρμα. Ειδικότερα:

1. το υψηλό GPIO Count με μικρό αποτύπωμα.
2. υπάρχει πιθανότητα να λειτουργήσει ως πλατφόρμα ανάπτυξης και προγραμματισμού.
3. Χρησιμοποιείται το IDE του κατασκευαστή μικροελεγκτών που είναι φιλικό προς το χρήστη και υποστηρίζει πολλούς μικροελεγκτές.

4. Χρησιμοποιεί τον προγραμματιστή- κατασκευαστή (pickit2-τροποποιημένο για simpp) που θα συνδεθεί εύκολα με άλλους μικροελεγκτές (χρησιμοποιώντας το ICSP) και αυτό δεν θα απαιτεί μικροελεγκτή με bootloaders.

5. Δεν χρειάζεται να εγκατασταθεί πρόγραμμα οδήγησης USB / patch ή dll για επικοινωνία με υπολογιστή. Just Plug,

6.εύκολη και άμεση σύνδεση και αναπαραγωγή.

7. Ο προγραμματισμός μπορεί να γίνει ακόμα και με σύνδεση εξωτερικής τροφοδοσίας.

8. Απλή και ευέλικτη δομή προγραμματισμού

Λόγοι χρήσης και προτερήματα του SIMPP

Μερικά δοκιμασμένα έργα που χρησιμοποιούν το SIMPP:

- RFID με σύστημα παρακολούθησης SMS
- Αλληλογραφία με ειδοποίηση SMS
- Σύστημα άρδευσης και παρακολούθησης φυτών
- Φυγόκεντρο σύστημα χαμηλού κόστους
- Θερμοστάτης (χρησιμοποιείται σε εκκολαπτήρια αυγών)
- Σύστημα στάθμευσης με κλείδωμα και ειδοποίηση SMS
- Σύστημα πράσινου σπιτιού
- Κλείδωμα πόρτας με ειδοποίηση SMS
- Ανίχνευση παρουσίας αυτοκινήτων με αισθητήρα υπερήχων (Hintz, Tabak, 1992).

4 Arduino

4.1 Ιστορική αναδρομή

Υπάρχουν πολλά εργαλεία για τη δημιουργία πρωτοτύπων με ηλεκτρονικά μέσα, τα οποία χρησιμοποιούνται για όλα, από τα νέα μουσικά όργανα έως τα ευφυή δωμάτια, τις προσαρμοσμένες συσκευές εισόδου και τα έργα τέχνης. Αυτά τα εργαλεία προσπαθούν να μειώσουν τη δυσκολία εργασίας με ηλεκτρονικά και να επεκτείνουν τον αριθμό των ατόμων που μπορούν να πειραματιστούν με το μέσο αυτό.

Πολλά από αυτά, ωστόσο, είναι εμπορικά προϊόντα - ακριβά και με δύσκολη πρόσβαση - ή ερευνητικά έργα που δεν είναι διαθέσιμα για χρήση από τους περισσότερους ανθρώπους. Άλλα αποτελούνται μόνο από κυκλώματα, που δεν παρέχουν εργαλεία για την απλοποίηση του προγραμματισμού τους. Η κίνηση ανοιχτού κώδικα, εν τω μεταξύ, έχει δείξει ότι χρήσιμο και λειτουργικό λογισμικό μπορεί να δημιουργηθεί από μια κατανεμημένη ομάδα εθελοντών που μοιράζονται ελεύθερα τα αποτελέσματα των προσπαθειών τους.

Τα έργα ανοιχτού κώδικα συχνά συγκεντρώνουν ισχυρές κοινότητες ανθρώπων που εργάζονται σε πολλά επίπεδα: μερικές εργασίες για τον βασικό κώδικα, άλλοι συνεισφέρουν με μικρές επεκτάσεις, ακόμα άλλοι εγγράφουν τεκμηρίωση ή προσφέρουν υποστήριξη, με την πλειοψηφία να χρησιμοποιεί απλά ένα ποιοτικό προϊόν.

Μπορούμε να εφαρμόσουμε τις αρχές της ανοιχτής πηγής στο υλικό και τα ηλεκτρονικά; Τι σημαίνει να κάνεις μια πλακέτα που είναι ανοιχτή και επεκτάσιμη, αλλά μπορεί να χρησιμοποιηθεί με λίγη προσπάθεια; Πώς μπορούμε να κάνουμε την εργασία με τα ηλεκτρονικά εύκολα, φθηνά και γρήγορα; Αυτά είναι μερικά από τα ερωτήματα που οδήγησαν στη δημιουργία της πλατφόρμας πρωτοτύπων Arduino. Ασχολείται με το σχετικό έργο, το εκπαιδευτικό και το σχεδιαστικό πλαίσιο μέσα στο οποίο αναπτύχθηκε το Arduino, τη φιλοσοφία πίσω από αυτό, την ίδια την πλατφόρμα, τόσο το υλικό όσο και το λογισμικό, καθώς και την κοινότητα που διαμορφώθηκε γύρω από το Arduino.

Το Arduino γεννήθηκε από το συνδυασμό πολλών πρωτότυπων πολιτισμών. Αναπτύχθηκε στο Ινστιτούτο Σχεδίασης Αλληλεπίδρασης "Inrea", ένα μικρό σχολείο στη βόρεια Ιταλία που προσφέρει πτυχίο Μάστερ στο σχεδιασμό αλληλεπίδρασης. Το πρόγραμμα σπουδών της Inrea επικεντρώθηκε σε διασυνδέσεις οθόνης, φυσικά αντικείμενα και υπηρεσίες, τόνισε την πρακτική εργασία, ενθαρρύνοντας τους σχεδιαστές να δημιουργούν γρήγορα πρωτότυπα των ιδεών τους, στη συνέχεια έπρεπε

να τα δοκιμάζουν και να τα τελειοποιούν επανειλημμένα. Το ινστιτούτο υποστήριξε επίσης τη δημιουργία εργαλείων που θα χρησιμοποιηθούν στη διαδικασία του πρωτοτύπου.

Ανηκε για αρκετά χρόνια στον Casey Reas, ο οποίος εργάστηκε για την ανάπτυξη της Επεξεργασίας. Συγκεκριμένα μια σειρά μαθημάτων σχετίζονται με την αλληλεπίδραση-Invrea που χρησιμοποιούνται σήμερα. Η επεξεργασία για τη δημιουργία πρωτοτύπων των διασυνδέσεων λογισμικού είναι ένα σημαντικό επίτευγμα. Το πρόγραμμα που διεξαχθεί το 2003 δημιουργήθηκε ενώ ο Massimo ήταν στο Interaction-Invrea και εφαρμόστηκε σε μαθήματα σχεδιασμού φυσικής αλληλεπίδρασης.

Η καλωδίωση αναπτύχθηκε ως έργο διατριβής στο ινστιτούτο και επίσης χρησιμοποιήθηκε στο εκπαιδευτικό πρόγραμμα. Οι αλληλεπιδράσεις της Invrea διέθεταν μια φιλοσοφία της εκμάθησης μέσω της λειτουργίας, πιστεύοντας ότι οι δεξιότητες, όπως ο προγραμματισμός ή η ηλεκτρονική είναι καλύτερα κατανοητές κατά τη διάρκεια γενικότερων έργων.

Οι σπουδαστές έχουν κίνητρο να αποκτήσουν τις πρακτικές γνώσεις που χρειάζονται για να εκφράσουν τις ιδέες του σχεδίου τους. Αυτό, με τη σειρά του, δίνει έμφαση σε εργαλεία που επιτρέπουν γρήγορο πειραματισμό και γρήγορη επανάληψη. Παρόλο που δημιουργήθηκαν σε εκπαιδευτικά περιβάλλοντα, αυτές οι πλατφόρμες προορίζονταν κυρίως για να βοηθήσουν τους ανθρώπους να δημιουργήσουν πρωτότυπα λογισμικού ή υλικού, όχι ως πειραματικά βοηθήματα σε ερευνητική μελέτη.

Από την αρχή κύριο μέλημα ήταν η διάδοση τεχνικών πρωτοτύπων στις εκπαιδευτικές εφαρμογές ως τρόπος επικοινωνίας των αξιών των νέων διαδραστικών συσκευών. Ο στόχος ήταν πάντα να ενθαρρύνουν τους παραδοσιακούς κλάδους του σχεδιασμού να υπερβαίνουν την οθόνη, προσπαθώντας να δώσουν νόημα στην ανθρώπινη διαδραστικότητα μέσω της σχεδίασης του ίδιου του μηχανήματος. Δεν είναι ρεαλιστικό να περιμένουμε οι σχεδιαστές να γίνουν μηχανικοί. Αντ' -'αυτού πιστεύουμε σε πρωτότυπα, με τα οποία οι σχεδιαστές μπορούν να δημιουργήσουν οι ίδιοι ένα αντικείμενο που εκφράζει τις επιθυμητές σχεδιαστικές προθέσεις. Αυτό τους δίνει ένα πολύτιμο εργαλείο επικοινωνίας με τους μηχανικούς στην περαιτέρω ανάπτυξη και υλοποίηση μιας ιδέας.

Επιδιώκονται να παρέχονται στους σχεδιαστές τα εργαλεία που χρειάζονται για πρωτότυπες αλληλεπιδράσεις, όχι μόνο μορφές ή υλικά - μετατρέποντας την ίδια τη

διαδραστικότητα σε μέσο έκφρασης. Η ενασχόληση με την εκπαίδευση σημαίνει συμβιβασμούς σχετικά με τη γλώσσα που πρέπει να χρησιμοποιείται κατά την επικοινωνία, το βάθος του περιεχομένου κλπ. Η φιλοσοφία του Arduino αποφεύγει να απομακρύνει όλη την πολυπλοκότητα πίσω από την ηλεκτρονική, όπως κάνουν κάποιες άλλες πλατφόρμες. Τα ηλεκτρονικά στοιχεία είναι κατασκευασμένα από εξαρτήματα, τα οποία είναι φυσικές συσκευές που αντιπροσωπεύουν λογικές λειτουργίες και πιστεύουμε ότι πρέπει να τη διατηρήσουμε ως μέρος της εκπαιδευτικής διαδικασίας.

Ως εκ τούτου, υπάρχει ανάγκη να γίνεται συζήτηση για να εφαρμοστούν πράγματα "αρκετά εύκολα" για τους μαθητές ώστε να κατανοήσουν πώς λειτουργούν τα συστήματα, δοκιμάζοντάς τα. Αντί να προσπαθούμε να αποκρύψουμε την πολυπλοκότητα από τους χρήστες, προτιμούμε να την απλοποιήσουμε στο σημείο που μπορούν να τα αντιμετωπίσουν άμεσα.

Αργότερα, θα έχουν την ευκαιρία να δουν το κόστος, το οποίο ήταν ένα σημαντικό στοιχείο κατά το σχεδιασμό του Arduino. Εάν ένα εργαλείο υλικού δεν είναι φθηνό, οι άνθρωποι διστάζουν να το αγοράσουν, επιβραδύνοντας τη διανομή και διατηρώντας το απρόσιτο για πολλούς ανθρώπους.

Επιπλέον, αν η πλατφόρμα είναι ακριβή, οι άνθρωποι δεν θα χρησιμοποιήσουν πολλά από αυτά, πράγμα που σημαίνει ότι μπορεί να χρειαστεί να αποσυναρμολογήσουν μια εργασία για να χτίσουν την επόμενη. Ένας απροσδόκητος παράγοντας είναι τα έξοδα αποστολής, καθώς ένα κομμάτι υλικού που είναι φθηνό στις ΗΠΑ μπορεί να διπλασιάσει την τιμή εάν πρέπει να μεταφερθεί, για παράδειγμα, στην Κίνα. Από την αρχή, το Arduino γεννήθηκε ως συνεργατικό έργο μεταξύ διαφόρων πανεπιστημίων και ατόμων.

Ο στόχος ήταν πάντα να μεγιστοποιηθεί ο αντίκτυπος εντός του ακαδημαϊκού κόσμου, προσπαθώντας να τεθούν ερωτήματα σχετικά με τον τρόπο σχεδιασμού διαδραστικών αντικειμένων. Ένα από τα βασικά θέματα που πρέπει να αντιμετωπιστούν είναι αυτά που σχετίζονται με την πνευματική ιδιοκτησία. Ο σχεδιασμός της αλληλεπίδρασης είναι μια νέα πρόκληση.

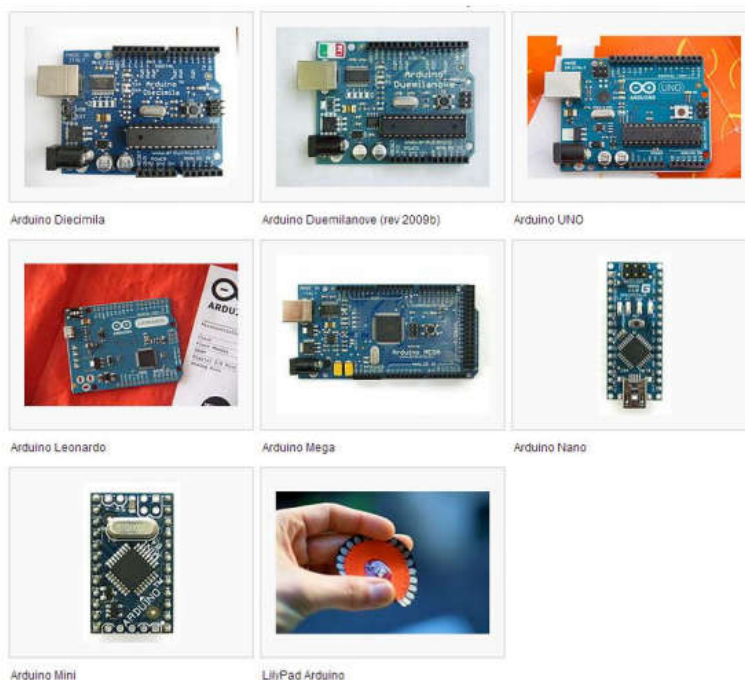
Πιστεύεται ότι ένας καλός τρόπος για να γίνει αυτό να ανταποκριθεί στις ανάγκες της κοινωνίας είναι να αναζητηθεί ένας τρόπο αδειοδότησης των αποτελεσμάτων που τα καθιστά διαθέσιμα για χρήση από άλλους ανθρώπους. Επιλέχθηκε να γίνει ολόκληρη η πλατφόρμα μέρος του ελεύθερου κινήματος της κουλτούρας,

απελευθερώνοντάς την υπό επιτρεπόμενες άδειες. Το Arduino έχει μείνει ανοιχτό με πολλούς διαφορετικούς τρόπους.

Πρώτον, το σύστημα είναι έτοιμο να πειραματιστεί με διαφορετικά μέσα τόσο από άποψη λογισμικού όσο και υλικού. Δεύτερον, οι άδειες επιτρέπουν την επαναχρησιμοποίηση του σχεδίου σε άλλα πλαίσια. Τρίτον, η εκπαιδευτική διαδικασία δεν είναι κλειστή, οι άνθρωποι δημιουργούν δικά τους υλικά, εργαστήρια, παραδείγματα και σεμινάρια (Warren, Adams, & Molle, 2011).

4.2. Μοντέλα Arduino

Το μοντέλο Arduino έχει κυκλοφορήσει με διαφορετικούς τύπους στις αγορές, οι οποίοι απεικονίζονται στην εικόνα που ακολουθεί. Η πτυχιακή εργασία, επικεντρώθηκε στην μελέτη στις ηλεκτρονικές πλατφόρμες του Arduino, καθώς παρέχει κάλυψη για το σύνολο των αναγκών των εργασιών αλλά και των άρθρων από τα οποία αποτελείται είναι ικανοποιητικά για το εύρος των λειτουργιών του, για παράδειγμα από την άποψη της μνήμης των μικροελεγκτών έχει απόλυτη συμβατότητα για να αποθηκευτεί ο κωδικός ο οποίος έχει κατασκευαστεί (Margolis, 2016).



Εικόνα 4. 1 Μοντέλα Arduino

4.2.1 Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino

Το Arduino επιτρέπει στους χρήστες να δημιουργούν ηλεκτρονικά πρωτότυπα, είτε αυτόνομα, είτε αντικείμενα που είναι εγκατεστημένα σε έναν υπολογιστή. Μπορεί να διαβάσει από ένα ευρύ φάσμα αισθητήρων, να ελέγξει ένα ευρύ φάσμα συσκευών

εξόδου και να επικοινωνήσει με λογισμικό που τρέχει σε υπολογιστή ή μιλάει σε δίκτυο.

Υπάρχουν πολλά βήματα που απαιτούνται για την εκτέλεση των πιο βασικών εργασιών με έναν μικροελεγκτή. Πρώτον, διενεργείται η συλλογή ενός συγκεκριμένου μικροελεγκτή, η καταγραφή του κυκλώματος που απαιτείται για τη χρήση του, η παραγγελία των απαραίτητων εξαρτημάτων, η συναρμολόγησή τους, η λήψη του λογισμικού που απαιτείται για τον προγραμματισμό του μικροελεγκτή, να εγκαταστήσει τους απαραίτητους οδηγούς, να αγοράσει ή να κατασκευάσει μια εξωτερική συσκευή για να προγραμματίσει τον μικροελεγκτή, να μάθει πώς να γράψει κώδικα για τον μικροελεγκτή (που μπορεί να απαιτεί ανάγνωση ενός φύλλου δεδομένων που έχει εκατοντάδες σελίδες κώδικα), κλπ. Το Arduino προσπαθεί να εξαλείψει ή να ελαφρύνει όσο το δυνατόν περισσότερα βήματα με ένα συνδυασμό υλικού και λογισμικού (Schwartz, 2014).

4.2.2 Η δομή του προγράμματος

Το λογισμικό Arduino είναι μια προσπάθεια απλοποίησης της διαδικασίας εγγραφής κώδικα χωρίς να περιορίζεται αδικαιολόγητα η ευελιξία του χρήστη. Βασίζεται σε πολλά άλλα έργα ανοιχτού κώδικα, προσαρμόζοντάς τα στο υλικό του Arduino και κρύβοντας την περιττή πολυπλοκότητά τους.

Το λογισμικό Arduino αποτελείται από δύο βασικά μέρη: το αναπτυξιακό περιβάλλον και μια κεντρική βιβλιοθήκη ανοιχτού κώδικα. Το περιβάλλον ανάπτυξης του Arduino είναι ένας ελάχιστος αλλά πλήρης επεξεργαστής πηγαίου κώδικα. Πρόκειται για μια εφαρμογή πολλαπλών πλατφορμών γραμμένη σε Java και χρησιμοποιήσιμη σε Windows, Mac OS X και Linux. Σε αυτό, οι χρήστες μπορούν να διαχειρίζονται, να επεξεργάζονται, να μεταγλωττίζουν και να ανεβάζουν τα προγράμματά τους (που ονομάζονται σκίτσα).

Όλες οι λειτουργίες μπορούν να αποκτήσουν πρόσβαση από ένα σύνολο επτά κουμπιών της γραμμής εργαλείων ή από μερικά αναπτυσσόμενα μενού. Ο χρήστης δεν χρειάζεται να κληγγάει με τα αρχεία "makefiles" ή τα επιχειρήματα της γραμμής εντολών, τα οποία μπορούν να δημιουργήσουν σημαντικά εμπόδια για τον αρχάριο.

Η περιοχή περιλαμβάνει μια σειριακή οθόνη, επιτρέποντας στον χρήστη να στείλει δεδομένα και να λαμβάνει δεδομένα από το διοικητικό συμβούλιο, διευκολύνοντας το σφάλμα χωρίς να χρειάζεται πρόσθετο λογισμικό. Στην πραγματικότητα, όλα τα προγράμματα που απαιτούνται για την ανάπτυξη του Arduino περιλαμβάνονται σε ένα αρχείο που έχει ληφθεί από την ιστοσελίδα του Arduino (εκτός από τους χρήστες του

Linux, οι οποίοι πρέπει να εγκαταστήσουν κάποια πακέτα με το εργαλείο διαχείρισης πακέτων της διανομής τους).

Το ίδιο το GUI βασίζεται στο αναπτυξιακό περιβάλλον επεξεργασίας, ενώ τα σκίτσα συντάσσονται από το `avr-gcc` και φορτώνονται με το `uispr`. Ο πηγαίος κώδικας διανέμεται υπό τη Δημόσια Άδεια GNU (GPL). Η βασική βιβλιοθήκη Arduino αποτελείται από λειτουργίες AVR C / C++ που συντάσσονται μαζί με το σκίτσο του χρήστη. Το συνδυασμένο δυαδικό αρχείο μπορεί στη συνέχεια να μεταφορτωθεί στο διοικητικό συμβούλιο Arduino.

Χρησιμοποιώντας ένα API συμβατό με την καλωδίωση, ο πυρήνας Arduino ενσωματώνει μέρη χαμηλού επιπέδου του προγραμματισμού των μικροελεγκτών (π.χ. χειρισμός καταγραφών), επιτρέποντας στους χρήστες να επικεντρωθούν στο συγκεκριμένο έργο τους (Lita, Cioc, Visan, 2006).

Ειδικότερα, αυτό εξοικονομεί για τους χρήστες την ανάγκη να διαβάσουν το φύλλο δεδομένων 300 σελίδων για τον μικροελεγκτή, τη μόνη αξιόπιστη πηγή πληροφοριών για τη λειτουργικότητα χαμηλού επιπέδου. Οι χρήστες εξακολουθούν να προγραμματίζουν στο πρότυπο C / C++, ωστόσο, έτσι ώστε οι γνώσεις προγραμματισμού που αποκτούν μπορούν να μεταφερθούν σε πολλές άλλες καταστάσεις.

Στην πραγματικότητα, ο πλήρης πηγαίος κώδικας στον πυρήνα Arduino (με άδεια χρήσης στο LGPL) περιλαμβάνεται στη διανομή, έτσι ώστε οι χρήστες να μπορούν να μάθουν πώς λειτουργεί και να το τροποποιούν. Λόγω της περιορισμένης χωρητικότητας του μικροελεγκτή, κάποιος κώδικας χωρίζεται σε ξεχωριστές βιβλιοθήκες, στις οποίες μπορούν να συμπεριληφθούν ειδικά θέματα όταν απαιτείται για ένα συγκεκριμένο σκίτσο.

Ο καθένας μπορεί να γράψει μια πρόσθετη βιβλιοθήκη, η οποία μπορεί να εγκατασταθεί απλά μετακινώντας την στον σωστό κατάλογο. Το περιβάλλον και ο πυρήνας είναι στενά ενσωματωμένα, έτσι ώστε ένα μόνο πάτημα ενός κουμπιού να συνθέτει ένα σκίτσο και ένα άλλο να το μεταφορτώνει στο διοικητικό συμβούλιο. Το περιβάλλον εκτελεί κάποια βασική προεπεξεργασία στον κώδικα του χρήστη, αποκρύπτοντας ορισμένες ασήμαντες συντακτικές οχλήσεις.

Ο πυρήνας περιλαμβάνει μια λίστα λέξεων-κλειδιών που πρέπει να επισημαίνονται στο περιβάλλον. Οι διαθέσιμες βιβλιοθήκες παρατίθενται σε ένα μενού μέσα στο περιβάλλον, από το οποίο μπορούν να επιλεγούν για να συμπεριληφθούν σε ένα συγκεκριμένο σκίτσο. Το περιβάλλον έρχεται με παραδείγματα σχημάτων για βασικές

εργασίες, επιτρέποντας στους χρήστες να δοκιμάσουν κάποια πράγματα χωρίς να γράψουν κώδικα αλλά και να παρέχουν απλές και σαφείς αναφορές στη γλώσσα και τις λειτουργίες του Arduino. Το υλικό και το λογισμικό, επίσης, έχουν σχεδιαστεί για να συνεργάζονται. Το λογισμικό πρέπει να υποστηρίζει μόνο τις λίγες πιθανές διαμορφώσεις υλικού και να λειτουργεί μόνο με το bootloader που είναι προ εγκατεστημένο στις κάρτες, περιορίζοντας τον αριθμό των απαιτούμενων επιλογών διαμορφώσεων (Warren, Adams, & Molle, 2011).

4.2.3. Βασική δομή και λειτουργίες

Όπως παρουσιάζεται στην εικόνα 10 πιο κάτω, το Arduino Ethernet αποτελεί μια δομή μικροελεγκτή ο οποίος διαθέτει κύριο μικροελεγκτή AT mega328. Διαθέτει δέκα τέσσερις συνολικά εισόδους και εξόδους ψηφιακής μορφής, έξι εισόδους αναλογικής μορφής, έναν ταλαντωτή 16 MHz, ένα βύσμα κατηγορίας RJ45 και ένα κουμπί για να επιτυγχάνεται επαναφορά (reset).

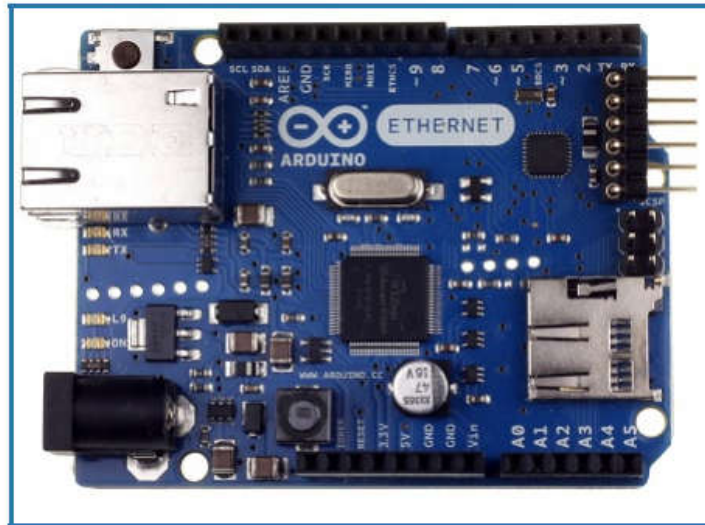
Το σύνολο των θυρών 10,11,12, 13 αποτελούν τις συνδέσεις με τα Ethernet και απαγορεύεται να γίνεται χρήση τους. Με αυτό τον τρόπο επιτυγχάνεται μείωση του αριθμού για τις διαθέσιμες θύρες σε 10, εκ των οποίων οι 4 βρίσκονται σε διαθεσιμότητα και έτσι δημιουργείται PWM. Τα Ethernet διαφέρουν από το σύνολο των άλλων κατηγοριών Arduino καθώς δεν αποτελούνται από απλά USB αντιθέτως, έχουν διεπαφές wiznet Ethernet.

Υπάρχουν αναγνώστες κάρτας microSD, οι οποίοι είναι σε θέση να χρησιμοποιηθούν με στόχο την αποθήκευση των δεδομένων, καθώς γίνεται χρήση μέσα από την βιβλιοθήκη SD. Η 10 θύρα αποτελεί την διεπαφή Wiznet SS, διότι οι κάρτες SD χρησιμοποιούν την θύρα 4. Συνολικά, ένα εύρος έξι ακροδεκτών εισχωρούν εντός των τμημάτων των προγραμματισμών μέχρι να υπάρξει συμβατότητα με τα USB. Ακόμη, διενεργείται αναστοιχειοθέτηση (reset), το οποίο δεν παρεμποδίζει την φόρτωση των δεδομένων χωρίς να πατηθεί το κουμπί για αυτή την λειτουργία στο επάνω τμήμα των Arduino Ethernets.

Σε περίπτωση που η σύνδεση πραγματοποιείται μέσω USB επάνω στα Arduino Ethernets τότε οι μικροελεγκτές τροφοδοτούνται μέσω των USB. Εντός των Arduino Ethernets υπάρχουν τυποποιημένα pinouts 1.0. τα οποία αποτελούν τις θύρες SDA και SCL, οι οποίες είναι κοντά από την θύρα AREF. Το σύνολο αυτών των δύο θυρών έχουν τοποθετηθεί κοντά στο κουμπί για την λειτουργία επαναφοράς (reset).

Υπάρχει η επιλογή να χρησιμοποιηθεί το i2C ή κάποιο τμήμα από το TWI με στόχο να επιτευχθεί συμβατότητα για όλους τους τύπους Arduino. Τα IOREF λειτουργούν

προστατευτικά για τα Arduino Ethernet από τις τάσεις τις οποίες διαθέτει για παροχή του τροφοδοτικού. Τέλος, πλησίον της θύρας IOREF υπάρχουν συνδεδεμένες θύρες, οι οποίες υπάρχουν με στόχο μελλοντική κάλυψη ανάγκης για χρήση (Margolis, 2016).



Εικόνα 4. 2 Arduino Ethernet

4.2.3.1 Μέθοδοι Τροφοδοσίας

Τα Arduino Ethernet είναι σε θέση να παρέχουν τροφοδοσίες εντός των εξωτερικών παροχών ηλεκτρικής ενέργειας ή επίσης να τροφοδοτηθούν μέσα από τις σειριακές συνδέσεις των καλωδίων USB, όπως παρουσιάζεται πιο πάνω, στην εικόνα 10. Οι εξωτερικές τροφοδοσίες είναι σε θέση να προέλθουν από την μία μέσω των προσαρμογέων AC-to-DC είτε μέσα από μπαταρίες. Οι μπαταρίες έχουν την δυνατότητα παροχής συνδέσεων στα Arduino Ethernet μέσα από τις θύρες Gnd και Vin. Παρέχεται η επιλογή λειτουργίας αποκλειστικά με μια μοναδική εξωτερική πηγή παροχής με ένταση από 6 μέχρι 20 V.

Τα Arduino σε περίπτωση που τροφοδοτηθούν με 7V και κάτω, το σύνολο των θυρών με τα 5V θα δύναται να αποδώσει λιγότερο από 5 V και έτσι θα δημιουργείται αστάθεια. Σε περίπτωση που γίνεται χρήση περισσότερων από 12 V υπάρχει περίπτωση οι ρυθμιστές τάσεων να υπερθερμανθούν και να ζημιωθούν οι πλακέτες. Οι συνιστάμενες παροχές τιμών στο σύστημα τροφοδοσίας κυμαίνονται από 7 μέχρι 12 V.

Το σύνολο των θυρών που ελέγχουν την τάση ορίζονται ως VIN, οι οποίες ελέγχουν τις τάσεις των εισερχόμενων στοιχείων στα Arduino Ethernets, σε περιπτώσεις που γίνεται χρήση των εξωτερικών πηγών ενέργειας. Η 5V, η οποία αποτελεί την θύρα εξαγωγής των ρυθμιζόμενων τάσεων των 5 V και είναι σε θέση να

δέχεται τροφοδοσία μέσω ρεύματος το οποίο μπορεί να έχει συνεχή μορφή από 7 έως 12 V, ή να παρέχεται μέσω συνδέσεων USB με σταθερή παροχή των 5 V.

Οι τάσεις εντός των θυρών 5V ή 3,3 V δημιουργούν παράκαμψη για τους ρυθμιστές και προκαλούνται βλάβες στα Arduino. Οι μεγαλύτερες καταναλώσεις ηλεκτρικού ρεύματος αγγίζουν τα 50 ma. Η θύρα GND αποτελεί την θύρα γείωσης. Ακόμη, η IOREF θύρα, είναι εκείνη που δημιουργεί παροχές τάσεων στα Arduino ώστε να μπορούν να λειτουργήσουν οι μικροελεγκτές. Έχει την δυνατότητα να προχωρά σε αναγνώριση των τάσεων και σε επιλογή των κατάλληλων πηγών ενέργειας.

Από την άλλη, οι μονάδες PoE έχουν δημιουργηθεί για να εξάγουν τις ενεργειακές δυνάμεις μέσω ενός καλωδίου συνεστραμμένου ζεύγους και τύπου 5 Ethernet. (Schultz and van Vugt, 2015).

4.2.3.2 Προγραμματισμός

Για να πραγματοποιηθεί προγραμματισμός στο σύστημα Arduino Ethernet δύναται να εφαρμοστούν δύο είδη, είτε μέσα από τον προγραμματισμό σε έξι θύρες, είτε μέσα από τον εξωτερικό ISP προγραμματιστή. Το σύνολο των 6 θυρών των προγραμματισμών διαθέτουν συμβατότητα με το σύνολο των καλωδίων FTDI USB και με τα USB. Υπάρχει ο χαρακτηρισμός για τις λειτουργίες των αυτόματων αναστοιχειοθετήσεων (reset), οι οποίες επιτρέπουν την φόρτωση των δεδομένων με την απουσία των πιέσεων στα κουμπιά που είναι καθορισμένα για να διεξάγουν αυτή την ενέργεια.

Σε περίπτωση που υπάρξει σύνδεση με τους προσαρμογείς FTDI μορφής USB, τα Arduino Ethernets, παρέχουν τροφοδοσία μέσω των USB. Υπάρχει ακόμη η δυνατότητα προγραμματισμού του Arduino Ethernet με εξωτερικούς προγραμματιστές, για παράδειγμα με AVRISP mkII ή με USBTinyISP. Είναι ωστόσο απαραίτητο το ‘‘κάψιμο’’ σε ένα συγκεκριμένο τμήμα του προγράμματος. Έτσι, θα υπάρξει τμηματική διαγραφή των boot loaders, ενώ το σύστημα Arduino Ethernet θα είναι σε θέση να συνεχίσει την λειτουργία του (Schwartz, 2014).

4.2.3.3 Βασικές μνήμες

Τα συστήματα Arduino είναι βασισμένα στο ATmega328, ειδικότερα έναν 8bit RISC μικροελεγκτή, που είναι σε ετοιμότητα συγχρονισμού σε 16MHz. Πιο συγκεκριμένα, το ATmega328 αποτελείται από τις ενσωματωμένες μνήμες.

Πρώτον, υπάρχει η μνήμη 2kB SRAM, η οποία χαρακτηρίζεται ως η ωφέλιμη μνήμη, καθώς είναι σε θέση να κάνει χρήση των προγραμμάτων με στόχο την αποθήκευση των μεταβλητών, των πινάκων κ.α. όταν το πρόγραμμα βρίσκεται σε

λειτουργία. Σε περίπτωση που υπάρξει διακοπή στο παρεχόμενο ρεύμα ή εάν πατηθούν τα κουμπιά που δημιουργούν αναστοιχειοθέτηση, η μνήμη δεν είναι σε θέση να διατηρήσει τις πληροφορίες και τα στοιχεία της.

Η μνήμη με το 1kB EEPROM, που έχει την δυνατότητα εγγραφής ή ανάγνωσης πληροφοριών με την απουσία datatype, ανά byte από το αντίστοιχο πρόγραμμα όταν βρίσκεται σε λειτουργία. Αντίθετα προς την SRAM, η μνήμη αυτή μπορεί να διατηρήσει το σύνολο των δεδομένων της ακόμα και αν σταματήσει η τροφοδοσία ή γίνει reset. Ακόμη, η μνήμη των 32 kB Flash, εκ των οποίων τα 2kB αποδίδονται στην χρήση των Firmware των προγραμμάτων Arduino, τα οποία έχουν ήδη εγκατασταθεί από τον κατασκευαστή.

Τα Firmware αυτά, μέσα από τις ορολογίες των Arduino χαρακτηρίζονται και bootloader, και αποτελούν απαραίτητο στοιχείο για να πραγματοποιηθεί η εγκατάσταση του προγράμματος των μικροελεγκτών μέσα από τις θύρες USB. Τα 30 KBb από τη μνήμη που έχουν απομείνει μπορούν να χρησιμοποιηθούν για να αποθηκευτούν τα προγράμματα, μετά από την ολοκλήρωση της μεταγλώττισης εντός του υπολογιστή.

Η Flash μνήμη, σε αντιστοιχία με την EEPROM είναι σε θέση να διατηρήσουν τα δεδομένα τους, ακόμη και αν σταματήσει η τροφοδοσία τους ή πραγματοποιηθεί reset. Ωστόσο, η συγκεκριμένη μνήμη με βάση τις τυπικές της δυνατότητες δεν αποδίδει όταν χρησιμοποιείται σε runtime εντός του προγράμματος, εξαιτίας της περιορισμένης και αναλογικής μνήμης, η οποία είναι διαθέσιμη σε εκείνο (2Kb SRAM + 1Kb EEPROM), ο σχεδιασμός μιας βιβλιοθήκης η οποία παρέχει την δυνατότητα χρήσης της συνολικής χωρητικότητας η οποία δεν χρησιμοποιείται (30Kb εκτός από το μέγεθος του προγράμματος που έχει μεταγλωττιστεί) (Schultz and van Vugt, 2015).

4.2.3.4 Παροχές επικοινωνίας

Τα συστήματα Arduino είναι σε θέση να πραγματοποιήσουν επικοινωνία με τους υπολογιστές, ή με σύνολο άλλων όμοιων πλακετών ή άλλων μικροελεγκτών. Υπάρχει παροχή της δυνατότητας για σειριακές επικοινωνιακές δομές μέσα από την USB θύρα, όπως παρουσιάζεται στην εικόνα 7.7.

Το σύνολο των λογισμικών τα οποία παρέχονται δίνουν την δυνατότητα αποστολής εντολών και πληροφοριών σε γενικές γραμμές τόσο από όσο και προς τις πλακέτες και να γίνει φόρτωση των προγραμμάτων στους bootloaders, οι οποίοι πρόκειται να το γίνεται με αυτόματο τρόπο από την πλακέτα.

Για να χρησιμοποιήσει ο διαχειριστής το Arduino μέσα από τον υπολογιστή του χρειάζεται να γίνεται χρήση του Arduino IDE, το οποίο έχει στηριχθεί στη δομή των Java και ειδικότερα μπορεί να παρέχει: Περιορισμένο πλαίσιο πρακτικής για να συγγραφούν τα προγράμματα που ορίζονται ως sketch στη γλώσσα επικοινωνίας των Arduino Ethernets. Ακόμη, αποδίδει μερικό αριθμό έτοιμων βιβλιοθηκών, τους compilers για να μεταγλωττιστούν τα Sketch και ένα serial monitor το οποίο εκτελεί παρακολούθηση των επικοινωνιών στις συριακές θύρες (USB). Τα Arduino Ethernets, έχουν την δυνατότητα επικοινωνίας με τους υπολογιστές, με άλλους Arduino ή με διαφορετικό μικροελεγκτή.

Ακόμη, το ATmega328, είναι σε θέση υποστήριξης των επικοινωνιακών απαιτήσεων TWI και SPI. Τα λογισμικά των Arduino Ethernets αποτελούνται από βιβλιοθήκες με στόχο να απλοποιηθεί η χρήση του TWI. Ενώ για την βελτιστοποίηση της επικοινωνίας μέσα από SPI γίνεται χρήση βιβλιοθηκών SPI. Τα Arduino Ethernets έχουν την δυνατότητα σύνδεσης με καλώδια δικτύων μέσα από τις θύρες Ethernet. Όταν συνδεθεί με τα δίκτυα, γίνεται παροχή των διευθύνσεων IP και άλλη μια MAC. Το σύνολο των βιβλιοθηκών Ethernet μπορούν να υποστηρίξουν καθολικά τα Arduino Ethernets. Οι αναγνώσεις των καρτών micro SD πραγματοποιούνται μέσα από τις βιβλιοθήκες SD (Schwartz, 2014).

4.2.3.5 Η δομή ενός προγράμματος και η γλώσσα

Το πρόγραμμα το οποίο είναι γραμμένο στα IDE των Arduino ορίζονται ως sketch και παρέχεται με αυτόματο τρόπο το όνομα αναλόγως με την ημέρα που μπήκε σε λειτουργία. Το αρχείο αυτό, μπορεί να επεκταθεί σε αρχείο INO. Το σχόλιο για μία γραμμή μπορεί να γραφτεί με διπλή κάθετο ή πολλές από αυτές '//'. (//σχόλιο). Τα σχόλια για πιο πολλές γραμμές αποτυπώνονται μέσα σε '/*' και '*/'. (/* σχόλιο */).

Τα IDE είναι σε θέση να αναγνωρίσουν το σύνολο των σχολίων και να προχωρούν σε χρωματισμό γκριζών, για να γίνει εμφανής το ότι δεν βρίσκονται εν ενεργεία για τα προγράμματα ή εντός των debugging. Το πρόγραμμα των Arduino, μπορεί να διαιρεθεί σε τρία είδη, στη δομή (structure), στις τιμές (values) και στις συναρτήσεις (functions).

Η γλώσσα η οποία χρησιμοποιείται από το Arduino Ethernet, είναι στηριζόμενη στην γλώσσα Wiring, δηλαδή μια διαφορετική εκδοχή των C/ C++ η οποία απευθύνεται σε μικροελεγκτή αρχιτεκτονικής τύπου AVR σε αντιστοιχία με τον AT mega. Είναι σε θέση να ανταποκριθεί για το σύνολο των βασικών δομών των C και για περιορισμένα γνωρίσματα των C++.

Για compiler γίνεται χρήση του AVR gcc και με κεντρική βιβλιοθήκη την AVR libc. Εξαιτίας των καταγωγών αυτής από την C, η γλώσσα του Arduino Ethernet κάνει χρήση των όμοιων βασικών εντολών και συναρτήσεων με παρόμοιες συντάξεις και ίδιους τύπους για τα στοιχεία αλλά και το σύνολο όμοιων τελεστών σε αντιστοιχία με την C.

Ωστόσο, υπάρχει ένα σύνολο συγκεκριμένων εντολών, συναρτήσεων και σταθερών οι οποίες λειτουργούν βοηθητικά για να γίνει διαχειριστεί κανείς το ειδικό λογισμικό των Arduino Ethernets (Lita, Cioc, Visan, 2006).

4.2.3.6 Ασπίδες του Arduino- Arduino Shields

Για την προστασία του το Arduino διαθέτει "ασπίδες", οι οποίες είναι πλακέτες που είναι σε θέση να παρέχουν σύνδεση με τις κορυφές των μικροελεγκτών Arduino, και έτσι να επεκτείνεται η δυνατότητα του.

Η ποικιλία των shields βασίζονται σε όμοια δεδομένα, ήτοι στηρίζονται στον άμεσο τρόπο για να τοποθετηθούν αλλά και στο περιορισμένο χρηματικό ποσό που χρειάζεται για να παραχθούν. Ένας αριθμός από αυτά παρουσιάζεται ακολούθως:

Arduino Wifi Shield παρέχει την ασύρματη σύνδεση του Arduino στο διαδίκτυο.



Εικόνα 4. 3 Arduino Wifi Shield

Arduino Ethernet Shield παρέχει ενσύρματη σύνδεση του Arduino στο διαδίκτυο με χρήση του καλωδίου RJ45.



Εικόνα 4. 4 Arduino Ethernet Shield

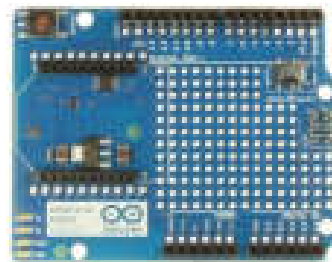
Μέσω των Wireless SD Shields παρέχεται η δυνατότητα εντός της πλακέτας του Arduino να δημιουργείται επικοινωνία με ασύρματο τρόπο για τις ασύρματες μονάδες. Οι μονάδες είναι σε θέση πραγματοποίησης επικοινωνίας μέχρι και σε 100 πόδια.

απόσταση εντός των εσωτερικών χωρών ενώ για εξωτερικό χώρο ισχύει το όριο των 300 ποδιών. Οι μονάδες αποτελούνται από θύρες υποδοχών SD.



Εικόνα 4. 5 Arduino Wifi Shield

Το Wireless Proto Shield παρέχει την δυνατότητα στο Arduino για ασύρματη επικοινωνία προς ασύρματες μονάδες. Οι μονάδες δύνανται να επικοινωνήσουν μέχρι και σε απόσταση 100 ποδιών μακριά εντός περιοχών ενώ σε 300 ποδιών για εξωτερικό περιβάλλον. Ωστόσο, δεν εμπεριέχει θύρες υποδοχών SD.



Εικόνα 4. 6 Arduino Proto Shield

Τα Arduino Motor Shields δύνανται να οδηγήσουν δύο DC κινητήρες από τις ίδιες συσκευές, καλύπτοντας τον έλεγχο στη ταχύτητα και τις κατευθύνσεις για την καθεμία ανεξάρτητα (Lita, Cìoc, Visan, 2006).



Εικόνα 4. 7 Arduino Motor Shields

4.2.3.7 Ακροδέκτες

Όλοι οι μικροελεγκτές Arduino διαθέτουν τόσο σημεία εισόδου όσο και σημεία εξόδου για να μπορούν να αλληλοεπιδρούν με το πλαίσιο και με το σύνολο των εξαρτημάτων τους. Όλοι οι ακροδέκτες (pin) χαρακτηρίζονται τόσο ως “εισόδου” όσο και “εξόδου”.

Τα Arduino Uno διαθέτουν είκοσι ακροδέκτες εκ των οποίων οι 14 χαρακτηρίζονται με ψηφιακά γνωρίσματα και οι έξι με αναλογικά. Ακολούθως, γίνεται παρουσίαση αυτών και παρέχεται ανάλυση των χαρακτηριστικών τους. Πρώτον, ο ακροδέκτης 0 και 1, λειτουργεί με τη μορφή RX και TX των σειριακών θυρών, σε περιπτώσεις όπου τα προγράμματα ενεργοποιούν τις σειριακές θύρες.

Κατ’ αυτόν τον τρόπο σε περιπτώσεις όπου τα προγράμματα αποστέλλουν πληροφορίες στις σειριακές θύρες γίνεται προώθηση αυτών και οι θύρες USB μέσα από τον εκλεκτή Serial-Over-USB, αλλά και από τους ακροδέκτες 0 γίνεται η ανάγνωση των δεδομένων από τις άλλες συσκευές. Κάτι τέτοιο υποδεικνύει πως εντός του προγράμματος γίνεται ενεργοποίηση των σειριακών interface, πραγματοποιείται απώλεια για δύο ψηφιακές εισόδους ή εξόδους της πλατφόρμας. Από την άλλη, ο ακροδέκτης 2 και 3 λειτουργεί και με εξωτερικά χαρακτηριστικά -interrupts (interrupt 0 και 1 αντίστοιχα).

Η ρύθμισή τους πραγματοποιείται εντός του προγράμματος με στόχο η λειτουργία τους να γίνεται κατ’ αποκλειστικότητα μέσω ψηφιακών εισόδων, που σε περιπτώσεις ύπαρξης αλλαγών, η τυπική εξέλιξη και πορεία του προγράμματος παύει σε άμεσο χρόνο και γίνεται εκτέλεση των συγκεκριμένων συναρτήσεων. Το σύνολο των εξωτερικών interrupts έχουν μεγάλη χρησιμότητα για την εφαρμογή που σχετίζεται με τους ακριβείς συγχρονισμούς.

Ακόμη, ο ακροδέκτης 3, 5, 6, 9, 10 και 11 είναι σε θέση για εκτέλεση λειτουργίας και με υπό ψευδο-αναλογική μορφή εξόδου εντός των συστημάτων PWM (Pulse Width Modulation) (Banzi, n.d.).

Ακόμη, το Arduino μέσω των ANALOG IN προσθέτει μια επιπλέον σειρά που αποτελείται από έξι pins, τα οποία έχουν αριθμηθεί από το 0 μέχρι το 5. Η γενικότερη προδιάθεση για να γίνεται αναφορά στο συγκεκριμένο δύναται να καθοριστεί μέσω των εντολών στο 1.1 V ανάμεσα δηλαδή στα 2 και τα 5V, με στόχο την τροφοδοσία από έξω των τάσεων για τα pins με τις σημάνσεις AREF, τα οποία εντάσσονται στις απέναντι πλευρές των πλακετών.

Συμπερασματικά, σε περίπτωση που γίνει τροφοδότηση του ακροδέκτη AREF με 3,3 V και ακολούθως γίνει ανάγνωση από άλλον ακροδέκτη υπό αναλογική είσοδο για τον οποίο γίνεται εφαρμογή τάσεων 1,65V, το σύστημα Arduino χρειάζεται να επανέλθει στο ποσό των 512 (Stewart, 2016).

5. Ανάλυση

5.1 Μειονεκτήματα

Η δομή – μέγεθος του Arduino είναι και το κύριο μειονέκτημα του. Κατά τη διάρκεια της κατασκευής ενός έργου πρέπει η κατασκευή να όσο το δυνατόν μικρότερο. Αλλά με τις μεγάλες δομές του Arduino δέν είναι εφικτό. Εάν κάποιος εργάζεται σε ένα μικρό μικροελεγκτή όπως το ATmega8 μπορεί εύκολα να κάνετε το PCB σας όσο το δυνατόν μικρότερο.

5.2 Πλεονεκτήματα

5.2.1 Έτοιμο για χρήση

Το μεγαλύτερο πλεονέκτημα του Arduino είναι η έτοιμη για χρήση δομή. Καθώς το Arduino έρχεται σε μια πλήρη συσκευασία που περιλαμβάνει τον ρυθμιστή 5V, έναν πυκνωτή, έναν ταλαντωτή, έναν μικροελεγκτή, σειριακή διεπαφή επικοινωνίας, LED και κεφαλίδες για τις συνδέσεις.

Δεν χρειάζεται να σκεφτείτε συνδέσεις προγραμματιστών για προγραμματισμό ή οποιαδήποτε άλλη διεπαφή. Απλά συνδέστε το σε θύρα USB του υπολογιστή σας και αυτό είναι. Η επαναστατική αυτή ιδέα πρόκειται να αλλάξει τον κόσμο μετά από λίγα μόνο λόγια κωδικοποίησης (Stewart, 2016).

5.2.2 Βιβλιοθήκες

Ένα άλλο μεγάλο πλεονέκτημα του Arduino είναι η βιβλιοθήκη των παραδειγμάτων που υπάρχουν μέσα στο λογισμικό του Arduino. Αυτό το πλεονέκτημα χρησιμοποιώντας ένα παράδειγμα μέτρησης τάσης. Για παράδειγμα, εάν π διαχειριστής θέλει να μετρήσει την τάση χρησιμοποιώντας τον μικροελεγκτή ATmega8 και θέλει να εμφανίσει την έξοδο στην οθόνη του υπολογιστή, τότε πρέπει να περάσει ολόκληρη τη διαδικασία.

Η διαδικασία θα ξεκινήσει από την εκμάθηση των ADC του μικροελεγκτή για τη μέτρηση, πέρασε από την εκμάθηση της σειριακής επικοινωνίας για την εμφάνιση και θα τελειώσει σε USB - Serial μετατροπείς. Εάν θέλετε να ελέγξετε ολόκληρη αυτή τη διαδικασία, κάντε κλικ στον παρακάτω σύνδεσμο (Bolor, Shah, Shah, & Schwartz, 2016).

5.2.3 Εύχρηστες Λειτουργίες

Κατά τη διάρκεια της κωδικοποίησης του Arduino, ορισμένες λειτουργίες που κάνουν τη ζωή τόσο εύκολη. Ένα άλλο πλεονέκτημα του Arduino είναι η αυτόματη δυνατότητα μετατροπής μονάδων. Επιπλέον, κατά τη διάρκεια του εντοπισμού σφαλμάτων δεν πραγματοποιούνται μετατροπές μονάδων. Πρέπει να

χρησιμοποιούνται όλες τις δυνάμεις για τα κύρια μέρη των έργων σας. Δεν χρειάζεται να δίνεται σημασία για τα δευτερεύοντα προβλήματα (Boxall, 2013).

5.2.4 Μεγάλη κοινότητα

Υπάρχουν πολλά φόρουμ στο διαδίκτυο στα οποία οι άνθρωποι μιλάνε για το Arduino. Μηχανικοί, χομπίστες και επαγγελματίες κάνουν τα έργα τους μέσω του Arduino. Έχοντας εύκολα την δυνατότητα βρείτε βοήθεια για τα πάντα. Επιπλέον, ο ιστότοπος του Arduino εξηγεί καθεμία από τις λειτουργίες του Arduino.

Έτσι, θα πρέπει να ολοκληρώσουμε τα πλεονεκτήματα του Arduino λέγοντας ότι κατά την εργασία σε διαφορετικά έργα πρέπει απλώς να ανησυχείτε για την καινοτόμο ιδέα, και τα υπόλοιπα θα χειριστεί ο ίδιος ο Arduino (Stewart, 2016).

5.3 Προοπτική Εξέλιξης

Το όλο θέμα της "πλατφόρμας Arduino" είναι να επιτρέψει την εύκολη και γρήγορη δημιουργία πρωτοτύπων. Η δυνατότητα να συνδέεται μια οθόνη LCD και να προβάλλονται μηνύματα σε αυτό μέσα σε λίγα λεπτά, αντί για ώρες, είναι απλά εκπληκτικά ισχυρή και βολική .

Για τα πρωτότυπα, η πλατφόρμα Arduino σας προσφέρει πολλές προ-καλωδιώσεις και δωρεάν βιβλιοθήκες κώδικα που θα επιτρέψουν να επικεντρωθείτε στη δοκιμή της ιδέας, να δημιουργεί κυκλώματα υποστήριξης ή να γραφτούν αρκετές σειρές κώδικα χαμηλού επιπέδου(Boxall, 2013)..

6. Συμπεράσματα

Το Arduino είναι κάτι περισσότερο από υλικό και λογισμικό. Από την αρχή έγινε προσπάθεια να ενθαρρυνθεί μια κοινότητα να διαμορφωθεί γύρω από το έργο, δώθηκαν συμβουλές στα ενδιαφερόμενα μέρη και έγιναν εκπαιδεύσεις σε εργαστήρια για να παρουσιαστεί στους ανθρώπους ο τρόπος που μπορούν να το χρησιμοποιήσουν.

Τα σχολεία όπως το ITP στο Πανεπιστήμιο της Νέας Υόρκης, το Goldsmiths στο Λονδίνο, η Fabbrica στην Ιταλία και το Universidad Politécnic de Catalunya στην Ισπανία είδαν τις δυνατότητες του Arduino και δεσμεύτηκαν να τα αγοράσουν, παρέχοντας ασφάλεια και κίνητρα για εκτεταμένη παραγωγή. Από τότε, το Arduino χρησιμοποιήθηκε σε πανεπιστήμια σε όλο τον κόσμο, όπως το MIT Media Lab, το Parsons School of Design, το Πανεπιστήμιο της Καλιφόρνιας Irvine, το Πανεπιστήμιο Malmö στη Σουηδία, το Πανεπιστήμιο Keio στην Ιαπωνία, το Εθνικό Πανεπιστήμιο Τεχνών της Ταϊβάν στην Ταϊβάν και περισσότερα .

Φαίνεται ότι το Arduino είναι ανοιχτό και αρκετά εύκολο για να το ξεκινήσει ο καθένας, χωρίς να χρειάζεται να διδαχθεί. Το Arduino έχει επίσης αποδειχθεί δημοφιλές στους καλλιτέχνες που επιδιώκουν να ενσωματώσουν την ηλεκτρονική και τη διαδραστικότητα στα έργα τους. Έχει διεξαχθεί ένα εργαστήριο για καλλιτέχνες τον Οκτώβριο του 2005 στη Μαδρίτη της Ισπανίας. Από τότε έγινε διδασκαλία στον Καναδά, στο Ηνωμένο Βασίλειο, στις ΗΠΑ, στη Γερμανία, στην Ελβετία, στις Κάτω Χώρες, στη Νορβηγία, στην Ιταλία, στην Αυστρία, στη Σουηδία, στη Σλοβενία κλπ.

Πραγματοποιήθηκαν εργαστήρια σε μέρη όπως η Ταϊβάν, Το Μεξικό, τη Χιλή, την Κολομβία και την Τουρκία. Τα έργα τέχνης που χρησιμοποιούν το Arduino εμφανίστηκαν στο Κέντρο Pompidou στο Παρίσι της Γαλλίας στο Φεστιβάλ Ars Electronica στο Linz της Αυστρίας στο Salone del Mobile στο Μιλάνο της Ιταλίας.

Το κύριο εργαλείο διάδοσης για το Arduino είναι ένας δικτυακός τόπος, συνεργαζόμενος με δεκαπέντε άτομα από διάφορες χώρες. Ένα wiki με δυνατότητα επεξεργασίας στο κοινό παρέχει ένα χώρο στους χρήστες του Arduino να γράφουν tutorials, να παρέχουν παράδειγμα κώδικα, να σχεδιάζουν κυκλώματα μεταφόρτωσης και να μοιράζονται αυτά που έχουν κάνει με τον Arduino. Στο φόρουμ, οι χρήστες μπορούν να ζητήσουν βοήθεια με διάφορα θέματα: να ξεκινήσουν τη λειτουργία της πλατφόρμας, να εντοπίσουν τον κώδικα τους, να προσδιορίσουν ποια εξαρτήματα θα χρησιμοποιήσουν για μια συγκεκριμένη εργασία κ.λπ.

Το έργο δεν περιορίζεται στα αγγλικά. Δώδεκα συντάκτες εργάστηκαν για τη δημιουργία μιας ισπανικής έκδοσης του ιστοτόπου Arduino και ένας έχει γράψει έναν

μακρύ οδηγό για τους δασκάλους που επιθυμούν να χρησιμοποιήσουν τον Arduino στις τάξεις τους. Υπάρχουν ισπανικά, γαλλικά και ιταλικά συμβούλια στο φόρουμ και άνθρωποι οι οποίοι έχουν γράψει επίσης σεμινάρια σε γερμανικά, ιαπωνικά, κινέζικα και σε πολλές άλλες γλώσσες. Ίσως οι πιο ικανοποιητικές είναι οι πολλές συμβολές που έχει δώσει το κοινό στο έργο.

Αναφερθήκαμε ήδη στην τεκμηριωμένη αποτύπωση σχετικά με το wiki Arduino και τα πολλά εργαστήρια που διοργανώνονται από άτομα που συνεργάζονται για το έργο. Άλλες συνεισφορές περιλαμβάνουν βιβλιοθήκες κώδικα για συγκεκριμένο υλικό, προσαρμοσμένες ασπίδες για διάφορα κυκλώματα, μπαλώματα στο περιβάλλον Arduino, ακόμη και εικονίδια και σχέδια χρωμάτων. Παρόλο που αυτή η διεθνής συνεργασία είναι κοινή στον κόσμο ανοιχτού κώδικα, η διπλή προσέγγιση του Arduino από φτηνά κατασκευασμένα διοικητικά συμβούλια και ανοιχτά σχέδια του επέτρεψαν να ευδοκιμήσει στον τομέα του υλικού. Συμπεράσματα Παρουσιάσαμε το Arduino, μια ανοιχτή πλατφόρμα για τη δημιουργία πρωτότυπων ηλεκτρονικών.

Ελπίζουμε ότι το Arduino δίνει τη δυνατότητα της εφαρμογής του μοντέλου ανοιχτού κώδικα να εφαρμοστεί ως ένα εργαλείο που θα χρησιμοποιηθεί από ένα ευρύ φάσμα ανθρώπων που δεν περιορίζεται σε ερευνητικό εργαστήριο ή σε όσους μπορούν να αντέξουν οικονομικά ακριβό εξοπλισμό (Boxall, 2013).

7. Βιβλιογραφία

1. Banzi, M. *Getting started with Arduino*.
2. Bolor, A., Shah, S., Shah, U., & Schwartz, M. (2016). *Arduino*. Birmingham: Packt Publishing.
3. Boxall, J. (2013). *Arduino workshop*. San Francisco: No Starch Press.
4. Hintz, K. , Tabak, D. 1992, *Microcontrollers : Architecture, Implementation & Programming*, McGraw Hill.
5. Lita, I. Cioc, D. Visan. (2006). A New Approach of Automobile Localization System Using GPS and GSM/GPRS Transmission, ISSE 2006 29th International Spring Seminar on Electronics Technology : nano technologies for electronics packaging : May 10-14, 2006, International Meeting Centre, St. Marienthal, Germany : conference proceedings 115–119.
6. Margolis, M. (2016). *Arduino cookbook*. [Place of publication not identified]: O'Reilly Media.
7. McRoberts, M. (2013). *Beginning Arduino*. Berkeley, CA: Apress.
8. Peatman, J.B. (2003). *Embedded Design with the PIC 18F452 Microcontroller*, Prentice Hall.
9. Schultz, B., & van Vugt, F. (2015). Tap Arduino: An Arduino microcontroller for low-latency auditory feedback in sensorimotor synchronization experiments. *Behavior Research Methods*, 48(4), 1591-1607. <http://dx.doi.org/10.3758/s13428-015-0671-3>
10. Schwartz, M. (2014). *Arduino home automation projects*. Birmingham, UK: Packt Pub.
11. Stewart, B. (2016). *Adventures in Arduino*. New York: Wiley.
12. Warren, J., Adams, J., & Molle, H. (2011). *Arduino Robotics*. Berkley: Apress.
13. www.ellak.gr