

Εισαγωγή της τεχνολογίας Arduino
στην Πρωτοβάθμια Εκπαίδευση.
Τεχνικές Μαθήματος, απόκριση
μαθητών και συμπεράσματα.

Ράλλης Θεμιστοκλής

Σπάρτη 2017



Τ.Ε.Ι. ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΕΙΣΑΓΩΓΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ARDUINO ΣΤΗΝ
ΠΡΩΤΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ. ΤΕΧΝΙΚΕΣ ΜΑΘΗΜΑΤΟΣ,
ΑΠΟΚΡΙΣΗ ΜΑΘΗΤΩΝ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.

ΡΑΛΛΗΣ ΘΕΜΙΣΤΟΚΛΗΣ Α.Μ: 2007116

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΚΑΤΣΑΚΟΣ-ΜΑΥΡΟΜΙΧΑΛΗΣ ΝΙΚΟΛΑΟΣ

ΣΠΑΡΤΗ 2017

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 βάση 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 δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

Όνομα και Επώνυμο Συγγραφέα (Με Κεφαλαία):

.....

Υπογραφή (Ολογράφως, χωρίς μονογραφή):

.....

Ημερομηνία (Ημέρα – Μήνας – Έτος):

.....

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον επιβλέποντα καθηγητή μου, κ.Νικόλαο Κατσάκο – Μαυρομιχάλη, για τη συνεργασία και την καθοδήγησή του όσον αφορά στην ολοκλήρωση της πτυχιακής μου εργασίας. Θέλω να ευχαριστήσω την κ.Ιωάννα Πολυχρονάκου, ιδιοκτήτρια του Ιδιωτικού Εκπαιδευτηρίου Σπάρτης, για τη συνεργασία της και τη διάθεση του χώρου του σχολείου της για τα μαθήματα που έγιναν όπως επίσης και τους διευθυντές των υπόλοιπων σχολείων για τη συνεργασία τους και ιδιαίτερα τον κ.Κωνσταντίνο Αρβανίτη. Ευχαριστώ επίσης τη δασκάλα κ.Χρυσούλα Γρουμπού για τον χρόνο που αφιέρωσε όντας παρούσα στα μαθήματα και για τη βοήθειά της με τα παιδιά.

ΠΕΡΙΛΗΨΗ

Σκοπός της συγκεκριμένης εργασίας είναι να διερευνηθεί το κατά πόσο θα ήταν εφικτή η εισαγωγή μια τεχνολογίας και συγκεκριμένα της τεχνολογίας Arduino στην πρωτοβάθμια εκπαίδευση. Να μελετηθεί δηλαδή το πόσο εύκολα οι μαθητές των μεγάλων τάξεων του δημοτικού σχολείου, της Ε' και ΣΤ' τάξης, θα μπορούσαν να ανταποκριθούν σε μια τέτοια διδασκαλία με βάση όσα ήδη γνωρίζουν σχετικά με τους ηλεκτρονικούς υπολογιστές και όσα θα μάθαιναν στην πορεία. Επιμέρους στόχος είναι να διερευνηθεί το πόσο ενδιαφέρει τους ίδιους τους μαθητές να ασχοληθούν με τη συγκεκριμένη τεχνολογία.

Το δείγμα μας αποτέλεσαν μαθητές της Ε' και ΣΤ' τάξης των δημοτικών σχολείων της Σπάρτης και η έρευνα διενεργήθηκε τον Απρίλιο και τον Μάιο του 2017. Οι μαθητές που επιθυμούσαν να συμμετέχουν στην έρευνα παρακολούθησαν 8 μαθήματα σχετικά με την τεχνολογία Arduino και με τις προαπαιτούμενες γνώσεις που έπρεπε να είχαν για να κατανοήσουν καλύτερα. Στο τέλος κάθε μαθήματος ακολουθούσε η συμπλήρωση ενός ανώνυμου ερωτηματολογίου, το οποίο δημιουργήθηκε ειδικά για τους σκοπούς της παρούσας έρευνας.

Ύστερα από τη συλλογή των ερωτηματολογίων και την επεξεργασία τους παρατηρήθηκε ότι στην πλειοψηφία τους οι μαθητές ήταν θετικοί απέναντι σε όσα παρακολούθησαν και φάνηκε ότι θα τους ενδιέφερε να ασχοληθούν και πάλι μελλοντικά με κάτι παρόμοιο και να μάθουν ακόμα περισσότερα στοιχεία για την τεχνολογία Arduino και τις δυνατότητες που προσφέρει.

Συμπερασματικά, οι μαθητές και γενικότερα τα άτομα μικρής ηλικίας διάκεινται θετικά απέναντι στην απόκτηση καινούριων γνώσεων και ειδικά όταν αυτές αφορούν στον τομέα της τεχνολογίας και των υπολογιστών που διαρκώς εξελίσσονται και διευκολύνουν τη ζωή μας μέσα από τις δυνατότητες που μας δίνουν.

ΠΕΡΙΕΧΟΜΕΝΑ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ	4
ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΛΗΨΗ	8
ΠΕΡΙΕΧΟΜΕΝΑ.....	10
ΕΙΣΑΓΩΓΗ	13
1. ΚΕΦΑΛΑΙΟ 1 - ΘΕΩΡΙΑ.....	18
1.1 Εισαγωγή	18
1.2 Κυκλώματα	22
1.3 Προγραμματισμός	25
1.4 Τεχνολογία Arduino.....	34
1.4.1 Πίνακας μετρήσεων αντιστάσεων	43
1.4.2 Σχηματικό Διάγραμμα του Arduino Uno Rev3	44
2. ΚΕΦΑΛΑΙΟ 2 – ΠΡΑΚΤΙΚΗ ΑΣΚΗΣΗ	46
2.1 Εργασία 1 ^η – Πρώτες Συνδεσμολογίες	46
2.2 Εργασία 2 ^η – Φωτεινός Σηματοδότης	51
2.3 Εργασία 3 ^η – Ψηφιακό Θερμόμετρο	55
2.4. Εργασία 4 ^η – Ροοστάτης και Ηλεκτρικό Μοτέρ.....	60
3. ΚΕΦΑΛΑΙΟ 3 – ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ	64
3.1 Αποτελέσματα	64
3.1.1 Αποτελέσματα Κυκλώματα ανά ερώτημα	65
3.1.2 Αποτελέσματα Προγραμματισμός ανά ερώτημα	70
3.1.3 Αποτελέσματα Arduino ανά ερώτημα	75
3.1.4 Αποτελέσματα 2 ^{ης} Εργασίας ανά ερώτημα.....	80
3.1.5 Αποτελέσματα 3 ^{ης} Εργασίας ανά ερώτημα.....	85
3.1.6 Αποτελέσματα 4 ^{ης} Εργασίας ανά ερώτημα.....	90
3.2 Συμπεράσματα	95
4. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ	98
5. ΠΑΡΑΡΤΗΜΑ.....	99
5.1 Ερωτηματολόγιο Κυκλώματα, Προγραμματισμός και Arduino	99
5.2 Ερωτηματολόγιο Άσκηση 2 ^η , Άσκηση 3 ^η και Άσκηση 4 ^η	100
5.3 Μεθοδολογία Ανάλυσης Αποτελεσμάτων	101
5.3.1 Μεθοδολογία για Κυκλώματα, Προγραμματισμό και Arduino	101
5.3.2 Μεθοδολογία για Άσκηση 2 ^η , Άσκηση 3 ^η και Άσκηση 4 ^η	101

5.3.3 Μεθοδολογία Γενικού Μέσου Όρου.....	101
5.4 Υπεύθυνες Δηλώσεις – Ερωτηματολόγια - Παρουσιολόγια.....	101

ΕΙΣΑΓΩΓΗ

Αντικείμενο της συγκεκριμένης πτυχιακής εργασίας αποτελεί η εισαγωγή της τεχνολογίας Arduino στην πρωτοβάθμια εκπαίδευση. Σκοπός είναι να διερευνηθεί κατά πόσο μια τέτοια τεχνολογία θα ενδιέφερε τους μαθητές του δημοτικού σχολείου, ποια είναι η πιο κατάλληλη μέθοδος διδασκαλίας και πόσο εύκολο θα είναι να γίνει κάτι τέτοιο σαφές και κατανοητό από παιδιά ηλικίας 11 – 12 ετών.

Με τον όρο Arduino εννοούμε έναν μικροελεγκτή μονής πλακέτας, δηλαδή μια πολύ απλή πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Προγραμματίζεται με τη γλώσσα Wiring (που θεωρείται μια απλούστερη εκδοχή της C++) η οποία αποτελείται ουσιαστικά από τη γλώσσα C++ και ένα σύνολο βιβλιοθηκών που είναι επίσης υλοποιημένες στη C++. Το Arduino μπορεί να συνδεθεί με υπολογιστή μέσω προγραμμάτων Pure Data, Processing, κτλ. και να αλληλεπιδράσει με αυτές τις γλώσσες προγραμματισμού και μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαδραστικών αντικειμένων. Υπάρχουν διάφορες εκδόσεις του Arduino όπου μπορεί να αγοράσει κάποιος και τα διαγράμματα και οι πληροφορίες για το υλικό και τη συναρμολόγηση είναι ελεύθερα και διαθέσιμα. (<https://www.arduino.cc>)

Στις μέρες μας τα πάντα γύρω μας αλλάζουν με πολύ γρήγορους ρυθμούς. Αντιλαμβανόμαστε ότι αυτό έχει να κάνει με τη ραγδαία εξέλιξη της τεχνολογίας, τις νέες επινοήσεις και τις νέες εφευρέσεις. Όλο και περισσότερο παρατηρούμε την επιρροή που ασκούν οι νέες τεχνολογίες σε όλους τους τομείς της ζωής μας. Μεγάλη πρόκληση αποτελεί και η εισαγωγή και η διδασκαλία των νέων τεχνολογιών και στον χώρο της εκπαίδευσης από τα πρώτα κιόλας χρόνια της πορείας των μαθητών στο δημοτικό σχολείο.

Στη χώρα μας οι πρώτες προσπάθειες για ένταξη των τεχνολογιών στην Πρωτοβάθμια Εκπαίδευση ξεκίνησαν το σχολικό έτος 1996 – 97 με την ενσωμάτωση της Πληροφορικής μέσω του πιλοτικού προγράμματος Ολοήμερων Σχολείων. Το έτος 2002 – 03 το μάθημα είχε τον τίτλο «Πληροφορική» και διδασκόταν κατά τη διάρκεια του Ολοήμερου Δημοτικού Σχολείου στους μαθητές που παρέμεναν μετά το τέλος της πρωινής ζώνης. Έναν χρόνο αργότερα το μάθημα μετονομάζεται σε «Νέες Τεχνολογίες στην Εκπαίδευση», ώστε τελικά το 2010 παίρνει τον τίτλο «Τεχνολογίες της Πληροφορίας και των Επικοινωνιών» (Φ.12/879/88413 /Γ1- 28-07-2010, ΦΕΚ 1139/2010, τ.Β) και εισάγεται ως ξεχωριστό γνωστικό αντικείμενο στην πρωινή ζώνη (Φ.3/609/60745 /Γ1: «Ορισμός 800 12θέσεων Δημοτικών Σχολείων με Ενιαίο Αναμορφωμένο Εκπαιδευτικό Πρόγραμμα») και διδάσκεται από εκπαιδευτικούς Πληροφορικής. (Κόμης Β., 2004)

Μέχρι τώρα στους στόχους του Διαθεματικού Ενιαίου Πλαισίου Προγραμμάτων Σπουδών (ΔΕΠΣΣ) και του Αναλυτικού Προγράμματος Σπουδών (ΑΠΣ) είναι οι μαθητές να έρθουν σε επαφή με τον ηλεκτρονικό υπολογιστή και να αντιληφθούν ποιες είναι οι υπηρεσίες που μπορεί να τους προσφέρει. Να γνωρίσουν τον ηλεκτρονικό υπολογιστή ως μέσο για αναζήτηση πληροφοριών και επικοινωνία και ως εποπτικό μέσο διδασκαλίας. Στόχος είναι επίσης η ανάπτυξη ενός κώδικα δεοντολογίας, η κριτική στάση των μαθητών απέναντι στη χρήση του ηλεκτρονικού υπολογιστή και η ευαισθητοποίησή τους για τις επιπτώσεις που μπορεί να έχει σε όλους τους τομείς της ζωής μας. Με λίγα λόγια οι μαθητές αναμένεται τελειώνοντας το δημοτικό σχολείο να έχουν αποκτήσει τις ελάχιστες γνώσεις και δεξιότητες που απαιτούνται για να μπορούν να αξιοποιήσουν τον ηλεκτρονικό υπολογιστή. (Κόμης Β.,2004)

Η ένταξη των νέων τεχνολογιών στην Εκπαίδευση στηρίζεται σε τρία μοντέλα:

- 1)** Το τεχνοκεντρικό μοντέλο σύμφωνα με το οποίο η Πληροφορική αποτελεί αυτοτελές γνωστικό αντικείμενο και αποσκοπεί στην απόκτηση γνώσεων απαραίτητων για τη λειτουργία του ηλεκτρονικού υπολογιστή. (Κόμης & Μικρόπουλος, 2001)
- 2)** Το ολιστικό μοντέλο το οποίο εστιάζει στη διαθεματική προσέγγιση της μάθησης. Σύμφωνα με αυτό η Πληροφορική σαν γνωστικό αντικείμενο θα πρέπει να ενσωματωθεί στα επιμέρους γνωστικά αντικείμενα του προγράμματος σπουδών. (ΦΕΚ τ. Β΄ 1366, 1373, 1374, 1375, 1376/18-10-2001)
- 3)** Το πραγματολογικό μοντέλο το οποίο είναι συνδυασμός των δυο προηγούμενων μοντέλων. (Φ.12/879/88413 /Γ1-28-07-2010, ΦΕΚ 1139/2010, τ.Β) Σύμφωνα με αυτό πρώτα κρίνεται απαραίτητη η απόκτηση γενικών γνώσεων σχετικά με την Πληροφορική και στη συνέχεια μπορεί να γίνει δυνατή η ενσωμάτωσή της στα άλλα γνωστικά αντικείμενα.

Επικρατέστερο φαίνεται να είναι το πραγματολογικό μοντέλο καθώς αποτελεί μια μικτή – εφικτή προσέγγιση. (Κόμης Β.,2004)

Θεωρίες Μάθησης και Τεχνολογίες Πληροφορικής & Επικοινωνιών (ΤΠΕ)

Η αξιοποίηση των ΤΠΕ στην εκπαιδευτική διαδικασία εξαρτάται από τη φιλοσοφική προσέγγιση του εκπαιδευτικού και το εκπαιδευτικό πλαίσιο και συνδέεται με συγκεκριμένες θεωρίες μάθησης.

- **Συμπεριφοριστική προσέγγιση**

Με βάση αυτή τη θεωρία, εκπρόσωποι της οποίας είναι οι ψυχολόγοι Ρανλόν, Skinner (Ράπτης και Ράπτη, 2006) η μάθηση είναι μια διαδικασία απόκτησης όλο και περισσότερων γνώσεων. Στηρίζεται στην ενίσχυση και στην επανατροφοδότηση. Η θετική ενίσχυση οδηγεί στην αύξηση της αυτοπεποίθησης του μαθητή και στη συνέχιση της πορείας του. Κυρίαρχο άτομο στη διαδικασία της μάθησης είναι ο δάσκαλος, τις οδηγίες του οποίου οι μαθητές ακολουθούν αυστηρά. Σ' αυτήν τη θεωρία στηρίζεται η δημιουργία κλειστών εκπαιδευτικών λογισμικών που περιλαμβάνουν απαντήσεις σωστού – λάθους, αντιστοιχήσεις κτλ και χρησιμοποιούνται στα σχολεία συνήθως για την εξάσκηση δεξιοτήτων χαμηλού επιπέδου.

- **Γνωστική προσέγγιση**

Με βάση αυτή τη θεωρία, η μάθηση είναι μια διαδικασία δημιουργίας της γνώσης που επηρεάζεται από τις κοινωνικοπολιτισμικές συνθήκες και από την αλληλεπίδραση μεταξύ των ανθρώπων. Στα πλαίσια αυτής της θεωρίας ο ρόλος του δασκάλου είναι καθαρά καθοδηγητικός και διευκολυντικός. Οι ίδιοι οι μαθητές μέσα από τον αναστοχασμό και τη γνωστική σύγκρουση θα οδηγηθούν στην οικοδόμηση της γνώσης. Βασική αρχή της θεωρίας είναι η ανακαλυπτική μάθηση (Bruner, 1986) όπου ο ίδιος ο μαθητής σε συνεργασία με τον εκπαιδευτικό καλείται να ανακαλύψει τη γνώση. Σ' αυτήν τη θεωρία στηρίζεται η δημιουργία ανοικτών εκπαιδευτικών λογισμικών, όπου οι μαθητές έχουν τη δυνατότητα να δημιουργήσουν οι ίδιοι σχέδια εργασίας και δραστηριότητες με βάση τους στόχους που έχουν θέσει.

- **Κριτική προσέγγιση**

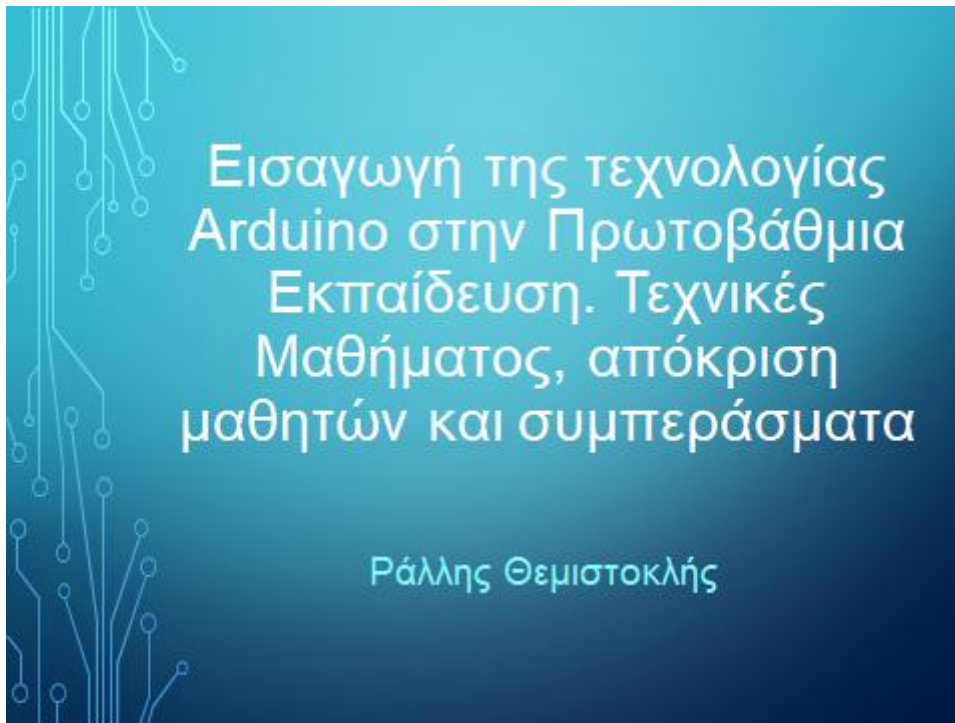
Σύμφωνα με αυτήν τη θεωρία, η μάθηση συνδέεται άμεσα με τον πραγματικό κόσμο και μάλιστα πρέπει να οδηγεί σε μετασχηματισμό και σε δημιουργία καλύτερων συνθηκών. Στα πλαίσια αυτά λοιπόν οι νέες τεχνολογίες γίνονται το μέσο για τη μεταφορά της γνώσης. Οι δυνατότητες που παρέχει η τεχνολογία αξιοποιούνται για την προώθηση της εκπαιδευτικής διαδικασίας.

Συνοψίζοντας, η προσφορά των ηλεκτρονικών υπολογιστών και των ΤΠΕ γενικότερα στην εκπαίδευση είναι πολύ σημαντική. Έρευνες της διεθνούς βιβλιογραφίας δείχνουν ότι γίνεται όλο και μεγαλύτερη προσπάθεια για πλήρη ενσωμάτωση των ηλεκτρονικών υπολογιστών στην εκπαιδευτική διαδικασία από τα πρώτα κίβλα βήματα των μαθητών στο δημοτικό σχολείο γιατί συμβάλλουν σημαντικά στην καλύτερη διεξαγωγή του μαθήματος, διευκολύνουν τον μαθητή να ανταποκριθεί καλύτερα στην εκπαιδευτική διαδικασία, προσφέρουν σημαντικές δυνατότητες στον εκπαιδευτικό για να βελτιώσει το έργο του και βοηθούν στην ένταξη των

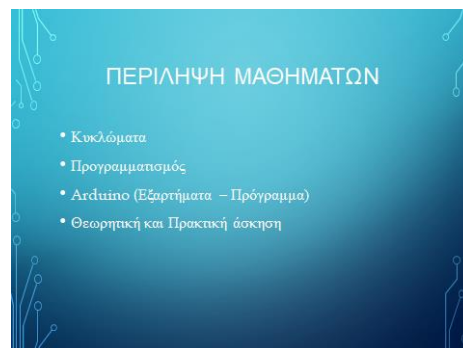
ατόμων στην κοινωνία, η οποία βασίζεται στη γνώση και στην πληροφορία. Άρα, μπορούμε να συμπεράνουμε ότι όσο περισσότερο καταφέρουμε να βάλουμε στη ζωή μας τον ηλεκτρονικό υπολογιστή και την τεχνολογία, τόσο καλύτερα αποτελέσματα θα έχουμε.

1. ΚΕΦΑΛΑΙΟ 1 - ΘΕΩΡΙΑ

1.1 Εισαγωγή

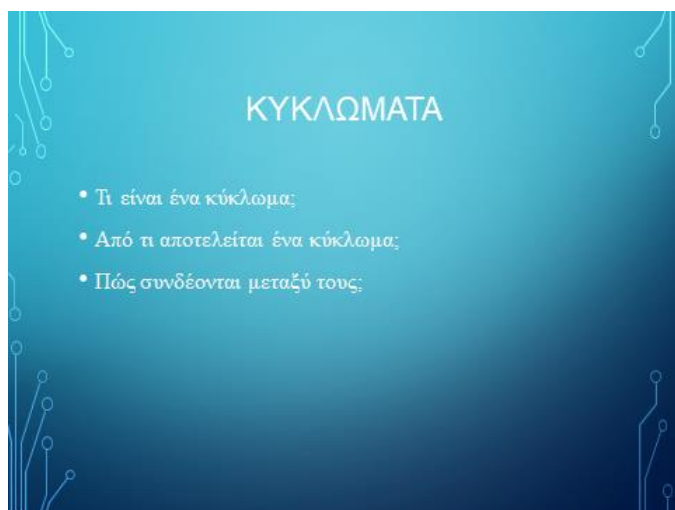


Όπως την πρώτη μέρα που τα παιδιά πηγαίνουν στο σχολείο, γνωρίζουν τον καινούριο τους δάσκαλο, ενημερώνονται για το περιεχόμενο των μαθημάτων που θα διδαχθούν και για την πορεία που θα ακολουθήσουν κατά τη διάρκεια της σχολικής χρονιάς έτσι και στο πρώτο μάθημα θα ακολουθηθεί η ίδια δομή προκειμένου να γίνει κατανοητό το περιεχόμενο των μαθημάτων που θα ακολουθήσουν. Η γνωριμία με τον δάσκαλο κρίνεται απαραίτητη ώστε οι μαθητές να νιώσουν λίγο πιο οικεία και να μπορέσει αργότερα να υπάρξει διάλογος και συζητήσεις σχετικά με τα μαθήματα που θα γίνουν. Οι μαθητές θεωρώντας τον δάσκαλο ένα δικό τους πρόσωπο, ένα άτομο στο οποίο θα μπορούν να απευθυνθούν και να ζητήσουν βοήθεια θα εκφράζουν χωρίς κανένα φόβο και δισταγμό τις απορίες τους. Μέσα λοιπόν, από τις ερωταποκρίσεις οι μαθητές θα οδηγηθούν πιο εύκολα στην κατάκτηση της γνώσης και θα ολοκληρωθεί με επιτυχία η εκπαίδευση. Ξεκινώντας το μάθημα και με αφορμή την πρώτη διαφάνεια έχουμε την ευκαιρία να συστηθούμε και να εξηγήσουμε βάση του τίτλου της εργασίας μας τι θα κάνουμε στα μαθήματα που θα ακολουθήσουν, να κατανοήσουν δηλαδή τα παιδιά ποιος είναι ο σκοπός της εκπαίδευσης. Οι ενότητες με τις οποίες θα ασχοληθούμε και αφορούν στην εργασία μας είναι: α) Τα κυκλώματα, β) Ο προγραμματισμός, γ) το



Arduino και δ) Η θεωρητική και πρακτική άσκηση που θα κάνουμε για να καταλάβουμε, να εξασκηθούμε και να εφαρμόσουμε όλα όσα θα μάθουμε.

Αφού τελείωσε η πρώτη γνωριμία και έγινε κατανοητός από όλους ο σκοπός της εργασίας ξεκινήσαμε να εξηγούμε ένα - ένα τα βήματα – μαθήματα. Το πρώτο μάθημά μας ήταν τα κυκλώματα. Το βασικό ήταν να εξηγήσουμε στα παιδιά γιατί

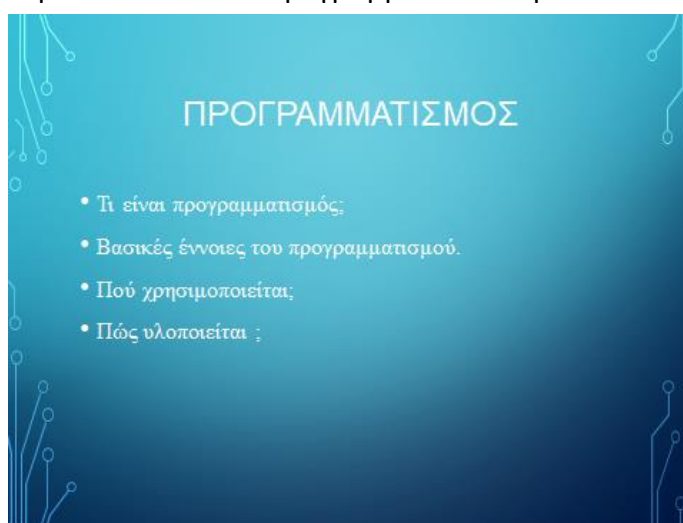


θα ασχοληθούμε με αυτήν την ενότητα και που θα μας βοηθήσει. Βασικός πυλώνας της εργασίας είναι η τεχνολογία Arduino. Μέσα από το Arduino μπορούμε να φτιάξουμε κυκλώματα και για αυτό είναι βασικό να γνωρίζουμε: α) Τι είναι ένα κύκλωμα, β) Από τι αποτελείται ένα κύκλωμα και γ) Πώς συνδέονται μεταξύ

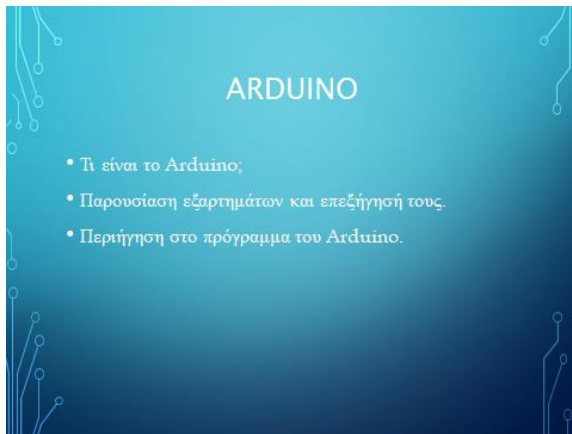
τους τα κυκλώματα.

Η Τεχνολογία Arduino εκτός από τη γνώση των κυκλωμάτων χρειάζεται και βασικές γνώσεις προγραμματισμού. Μερικά κυκλώματα θέλουν να προγραμματιστούν για

τη σωστή λειτουργία τους και για να κάνουν μια συγκεκριμένη εργασία. Στην πρακτική άσκηση που κάναμε χρειάστηκε να προγραμματίσουμε την πλακέτα Arduino προκειμένου να πετύχουμε το επιθυμητό αποτέλεσμα και να δούμε αυτό που είχαμε σχεδιάσει θεωρητικά. Στην ενότητα του Προγραμματισμού λοιπόν μάθαμε: α) Τι είναι ο Προγραμματισμός, β) Ποιες είναι οι βασικές έννοιες του προγραμματισμού, γ) Πού χρησιμοποιείται και δ) Πώς υλοποιείται.



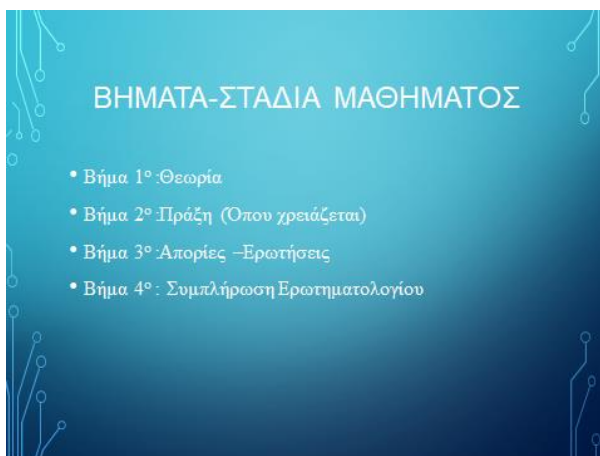
Μέχρι τώρα είχαμε αναφερθεί πολλές φορές στην έννοια του Arduino και στις βασικές προαπαιτούμενες γνώσεις



πάνω στα κυκλώματα και στον προγραμματισμό που πρέπει να έχουμε προκειμένου να προχωρήσουμε στην κατανόηση και κατά συνέπεια στην εκμάθηση της συγκεκριμένης τεχνολογίας αλλά δεν είχαμε εξηγήσει τι είναι το Arduino και από τι αποτελείται. Βασισμένοι λοιπόν και στις απορίες των παιδιών έως τώρα κάθε φορά που μας άκουγαν να

αναφέρουμε την έννοια αυτή, στη συγκεκριμένη ενότητα προσπαθήσαμε να τους εξηγήσουμε: α) Τι είναι το Arduino, β) Από ποια εξαρτήματα αποτελείται και κάναμε μια περιήγηση στο πρόγραμμα του Arduino. Με την αναφορά μας στην τεχνολογία Arduino ολοκληρώθηκε το θεωρητικό κομμάτι της εργασίας μας.

Στο σημείο αυτό καταστήσαμε ξεκάθαρα στους μαθητές τα βήματα – στάδια που θα ακολουθήσουμε μέχρι να ολοκληρώσουμε την έρευνά μας. Θεωρήσαμε σημαντικό οι μαθητές να γνωρίζουν από την πρώτη κιόλας επαφή τι ακριβώς θα μάθουν και ποια δομή θα έχει η έρευνα. Γνωρίζοντας όλα αυτά από την αρχή το να συνεχίσουν την παρακολούθηση των μαθημάτων θα ήταν ένας πολύ σημαντικός δείκτης του ενδιαφέροντός τους, όπως και έγινε. Τα βήματα που ακολουθήσαμε ήταν τα εξής:



α) Θεωρία, β) Πράξη, γ) Απορίες – Ερωτήσεις και δ) Συμπλήρωση Ερωτηματολογίου. Πρώτα από όλα λοιπόν αναφερθήκαμε σε όλη τη θεωρία και για την ακρίβεια στις βασικές γνώσεις που μπορούν να κατανοήσουν και να συγκρατήσουν και χρειάζεται να ξέρουν τα παιδιά αυτής της ηλικίας για να μπορέσουν να μάθουν την τεχνολογία Arduino. Κατόπιν, ακολούθησε πρακτική

άσκηση. Προγραμματίσαμε μαζί τα παιδιά την πλακέτα του Arduino αφού είχαμε φτιάξει όλοι μαζί διάφορα κυκλώματα και καταφέραμε να κάνουμε πράξη όλα όσα προηγουμένως είχαμε δει θεωρητικά. Στο τέλος κάθε συνάντησης οι μαθητές εξέφραζαν τις απορίες τους, τις οποίες και προσπαθούσαμε να εξηγήσουμε. Οι απορίες των μαθητών ήταν για μας οδηγός εφόσον μας βοηθούσαν να

καταλάβουμε τι είναι αυτό που δυσκολεύει περισσότερο τα παιδιά και κατ' επέκταση να οργανώσουμε καλύτερα τη διδασκαλία μας. Ύστερα οι μαθητές συμπλήρωναν ένα ερωτηματολόγιο που σκοπό είχε να διαπιστωθεί πόσο τους ενδιέφερε αυτό που παρακολούθησαν, πόσο κατανοητή ήταν η διδασκαλία, αν είχαν ακούσει ήδη για κάτι τέτοιο πρωτύτερα και αν θα ήθελαν να εμβαθύνουν περισσότερο.

1.2 Κυκλώματα

Αφού ολοκληρώσαμε το πρώτο εισαγωγικό μάθημα στο οποίο κατέστη σαφής η πορεία που θα ακολουθήσουμε, στο δεύτερο μάθημα σειρά είχε η αναλυτική ενασχόληση με τα κυκλώματα. Η ενότητα των κυκλωμάτων κρίνεται αναγκαία να αναφερθεί καθώς όπως προείπαμε αποτελούν βασική προαπαιτούμενη γνώση για την εκμάθηση της τεχνολογίας Arduino. Όσα άκουσαν οι μαθητές για τα κυκλώματα φάνηκε ότι δεν τους δυσκόλεψαν ιδιαίτερα καθώς γνώριζαν ήδη αρκετά πράγματα εφόσον η ενότητα αυτή αποτελεί αντικείμενο διδασκαλίας στο μάθημα της Φυσικής στην Ε' Τάξη.

Αρχικά λοιπόν εξηγήσαμε τι είναι το κύκλωμα και συγκεκριμένα το ηλεκτρικό κύκλωμα. Ηλεκτρικό κύκλωμα ονομάζεται μια ηλεκτρική αγωγή από ηλεκτρικό ρεύμα διαδρομή όπου τα στοιχεία που συμμετέχουν στο κύκλωμα ονομάζονται ηλεκτρικά στοιχεία. Ένα ολοκληρωμένο ηλεκτρικό κύκλωμα αποτελείται από ηλεκτρονικά εξαρτήματα, όπως αντιστάσεις, πυκνωτές, διόδους, τρανζίστορ κτλ., τα οποία είναι κατασκευασμένα πάνω στην ίδια πλακέτα(chip) και συνδέονται μεταξύ τους δημιουργώντας ένα πλήρες ηλεκτρικό κύκλωμα.

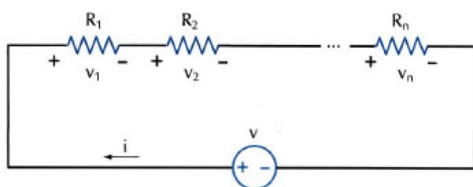
Αφού δώσαμε τον ορισμό του ηλεκτρικού κυκλώματος έπρεπε και να τον εξηγήσουμε. Μιλήσαμε για ηλεκτρικά στοιχεία. Υπάρχουν τρία βασικά είδη ηλεκτρικών στοιχείων. Το πρώτο είναι οι πηγές δηλαδή τα παραγωγικά ηλεκτρικά στοιχεία, το δεύτερο είναι οι αγωγοί δηλαδή οι δρόμοι από όπου περνάει το ρεύμα και το τρίτο είναι οι συσκευές δηλαδή τα καταναλωτικά ηλεκτρικά στοιχεία. Ρωτήσαμε τους μαθητές να μας δώσουν μερικά παραδείγματα για αυτά τα τρία βασικά ηλεκτρικά στοιχεία και οι απαντήσεις που έδωσαν ήταν πολύ εύστοχες. Στις πηγές ανέφεραν τις μπαταρίες και τις πρίζες όπου υπάρχει ρεύμα. Για τους αγωγούς είχαμε μία απάντηση και αυτή ήταν τα καλώδια που μεταφέρουν το ρεύμα. Και τέλος για τις συσκευές πήραμε ποικίλες απαντήσεις και καταλήξαμε στο συμπέρασμα ότι στην κατηγορία αυτή ανήκει οτιδήποτε χρησιμοποιεί και καταναλώνει ρεύμα.

Αφού τελειώσαμε με τα είδη των ηλεκτρικών στοιχείων ήρθε η ώρα να μιλήσουμε και για τα είδη των κυκλωμάτων. Εξηγήσαμε ότι τα κυκλώματα διαχωρίζονται σε ηλεκτρικά και ηλεκτρονικά ή αλλιώς σε αναλογικά και ψηφιακά. Υπενθυμίσαμε πάλι τι είναι τα ηλεκτρικά κυκλώματα και προσθέσαμε πάνω σε αυτό πως κάθε διάταξη που αποτελείται από αγωγίμους δρόμους όπου μπορεί να περάσει ηλεκτρικό ρεύμα είναι επίσης ένα ηλεκτρικό κύκλωμα. Τα ηλεκτρονικά κυκλώματα αποτελούνται από ημιαγωγούς ηλεκτρικού ρεύματος. Ημιαγωγοί είναι τα υλικά που ανάλογα τις συνθήκες, άλλοτε επιτρέπουν και άλλοτε όχι στο ηλεκτρικό ρεύμα να περάσει μέσα από αυτά. Είναι υλικά που μέσα από ελάχιστη επεξεργασία αλλάζουν ελαφρά συμπεριφορά και ηλεκτρικές ιδιότητες. Για παράδειγμα, όταν αυξάνεται η

θερμοκρασία του ημιαγωγού αυξάνεται και η αγωγιμότητά του. Τέτοια υλικά για παράδειγμα είναι το πυρίτιο.

Στη συνέχεια αναφέραμε ότι τα ηλεκτρικά κυκλώματα για τα οποία μιλάμε έχουν ένα επιπλέον χαρακτηριστικό. Αυτό είναι η συνδεσμολογία, δηλαδή τα διάφορα στοιχεία που συνδέονται για να δημιουργηθεί ένα κύκλωμα και ο τρόπος σύνδεσης τους. Αυτά τα στοιχεία είναι ο κόμβος, ο κλάδος και ο βρόχος. Εξηγήσαμε ότι ο κόμβος είναι το σημείο στο οποίο συνδέονται δύο ή περισσότερα στοιχεία. Ο κλάδος είναι μία διαδρομή η οποία ενώνει δύο κόμβους και δεν περνάει από κανέναν άλλο κόμβο που συνδέει περισσότερα από δύο στοιχεία. Και ο βρόχος είναι μια διαδρομή όπου ο αρχικός και ο τελικός της κόμβος ταυτίζονται, δηλαδή ξεκινά από έναν κόμβο και καταλήγει στον ίδιο ακριβώς από όπου και ξεκίνησε.

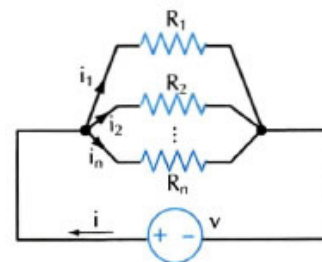
Μετά από τη σύνδεση των κυκλωμάτων έπρεπε να εξηγήσουμε και τη σύνδεση των στοιχείων (Αντιστάσεις). Υπάρχουν δύο απλές μορφές σύνδεσης. Είναι η σύνδεση σε



Εικόνα 1.2.1

σειρά (Εικόνα 1.2.1) και η παράλληλη σύνδεση (Εικόνα 1.2.2). Εξηγήσαμε ότι δύο ή περισσότερα στοιχεία είναι συνδεδεμένα σε σειρά όταν το τέλος του ενός στοιχείου είναι

συνδεδεμένο με την αρχή του άλλου στοιχείου ώστε να διαρρέονται από το ίδιο ρεύμα. Είπαμε επίσης ότι είναι συνδεδεμένα σε σειρά εφόσον βρίσκονται πάνω στον ίδιο κλάδο. Από την άλλη πλευρά όταν είναι συνδεδεμένα παράλληλα τότε δύο ή



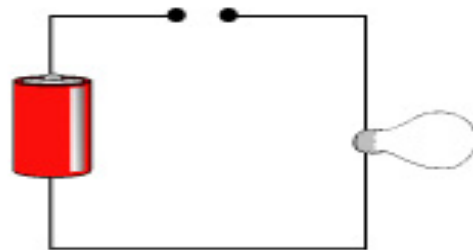
Εικόνα 1.2.2

περισσότερα στοιχεία έχουν κοινά άκρα και καταλήγουν στον ίδιο κόμβο με αποτέλεσμα να βρίσκονται όλα στην ίδια τάση.

Για κλείσουμε την ενότητα έπρεπε να εξηγήσουμε στα παιδιά πότε λέμε ότι ένα κύκλωμα είναι ανοιχτό (Εικόνα 1.2.3) και πότε κλειστό (Εικόνα 1.2.4). Αυτό για τα παιδιά ήταν πολύ εύκολο και όλα γνώριζαν την απάντηση. Κλειστό ηλεκτρικό



Εικόνα 1.2.3



Εικόνα 1.2.4

κύκλωμα έχουμε όταν τα ηλεκτρόνια ακολουθούν μια κλειστή διαδρομή και ανοικτό ηλεκτρικό κύκλωμα όταν τα ηλεκτρόνια δε μπορούν να κινηθούν μέσα σε αυτό.

Τελειώνοντας το μάθημα για τα κυκλώματα, τα παιδιά είχαν κάποιες απορίες σχετικά με τη διαδρομή του ρεύματος. Τότε έπρεπε να τους εξηγήσουμε τι είναι το εναλλασσόμενο και τι το συνεχές ρεύμα. Μάθαμε λοιπόν ότι το εναλλασσόμενο ρεύμα είναι όταν η κατεύθυνση και η έντασή του μεταβάλλονται, ενώ το συνεχές ρεύμα έχει σταθερή κατεύθυνση και σταθερή ροή. Η διαφορά τους είναι ότι στην περίπτωση που έχουμε εναλλασσόμενο ρεύμα μπορούμε να το τροποποιήσουμε ανάλογα με τις ανάγκες μας και να φτιάξουμε την κατεύθυνση και την τάση που θέλουμε ενώ στο συνεχές ξέρουμε ήδη αυτά τα στοιχεία και το προσαρμόζουμε όπως θέλουμε. Επίσης, τονίσαμε πώς μπορούμε να μετατρέψουμε το εναλλασσόμενο ρεύμα σε συνεχές με διάφορες διόδους και αντιστάσεις. Ουσιαστικά όλες οι συσκευές παίρνουν το εναλλασσόμενο ρεύμα και το μετατρέπουν σε συνεχές. Όταν κλήθηκαν οι μαθητές να αναφέρουν παραδείγματα αναφέρθηκε ως εναλλασσόμενο ρεύμα η πρίζα που έρχεται το ρεύμα και συνεχές η λάμπα από όπου περνάει το ρεύμα.

1.3 Προγραμματισμός

Ολοκληρώνοντας το πρώτο θεωρητικό μάθημα που ασχολείται με τα κυκλώματα έπρεπε να περάσουμε στο πιο δύσκολο θεωρητικό κομμάτι και σε κάτι πρωτόγνωρο για τα παιδιά, το οποίο δεν είναι άλλο από τον προγραμματισμό. Οι μαθητές ενθουσιάστηκαν στην ιδέα του προγραμματισμού καθώς όταν άκουσαν τη λέξη φαντάστηκαν διάφορα. Όμως, η διδασκαλία του κεφαλαίου αυτού στη συγκεκριμένη ηλικία μάλλον φαίνεται δύσκολη, ίσως και ανέφικτη.

Πρώτα από όλα έπρεπε να εξηγήσουμε στους μαθητές τον ορισμό του προγραμματισμού. Με βάση την ελληνική Wikipedia προγραμματισμός είναι: «Το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος για την πραγματοποίηση εργασιών ή για την επίλυση ενός δεδομένου προβλήματος. Ο προγραμματισμός περιλαμβάνει επίσης τον έλεγχο του προγράμματος για την επαλήθευση της ακρίβειάς του και την προπαρασκευή των οδηγιών με τις οποίες ένας υπολογιστής θα εκτελέσει τις εργασίες που καθορίζονται στις προδιαγραφές του προγράμματος». Ακούγοντας όλο αυτό οι μαθητές απόρησαν για το τι είναι τελικά ο προγραμματισμός. Έτσι εξηγήσαμε στους μαθητές ότι πιο απλά ο προγραμματισμός είναι η διαδικασία που ακολουθείς για να φτιάξεις το δικό σου πρόγραμμα, τη δική σου εφαρμογή ή ακόμα και τη δική σου ιστοσελίδα. Είναι μια σειρά από εντολές που τις γράφεις και εκτελούνται με τη σειρά, όπως είναι γραμμένες.

Για να μπορέσουν να καταλάβουν οι μαθητές πώς λειτουργεί ο προγραμματισμός έπρεπε να δώσουμε παραδείγματα. Ένα καλό παράδειγμα ήταν το ωρολόγιο πρόγραμμα του σχολείου. Το πρόγραμμα είναι συγκεκριμένο. Κάθε μαθητής ξέρει τι μάθημα έχει, ποια μέρα και τι ώρα το έχει. Έτσι είναι και ο προγραμματισμός. Φτιάχνεις το πρόγραμμα και καθορίζεις ακριβώς τι θα κάνει και τότε θα το κάνει ανάλογα με την εντολή που του έχεις δώσει. Δε θα κάνει κάτι διαφορετικό από αυτό που του έχεις πει.

Αφού οι μαθητές κατάλαβαν τι είναι ο προγραμματισμός έπρεπε να τους εξηγήσουμε τα υπολογιστικά προβλήματα. Υπολογιστικό πρόβλημα θεωρούμε κάθε ζήτημα που τίθεται προς επίλυση, κάθε κατάσταση που μας απασχολεί, κάθε ηλεκτρονική συμπεριφορά που επιθυμούμε να επιδειχθεί από τον υπολογιστή μας. Για αυτόν τον λόγο φτιάχνουμε και ένα πρόγραμμα. Για να μας επιλύσει το υπολογιστικό πρόβλημα. Επίσης ο προγραμματισμός περιλαμβάνει τον έλεγχο του προγράμματος για την επαλήθευση της ακρίβειας και της ορθότητάς του (αποσφαλμάτωση) και την προπαρασκευή των οδηγιών με τις οποίες ένας υπολογιστής θα εκτελέσει τις εντολές που καθορίζονται στις προδιαγραφές του προγράμματος. Δηλαδή όταν προγραμματίζουμε φροντίζουμε το πρόγραμμά μας να ελέγχει τα λάθη και να δίνει εντολές στον υπολογιστή στο πως και τότε θα εκτελεστούν οι εντολές του προγράμματος.

Αφού εξηγήσαμε αυτά ρωτήσαμε τους μαθητές να μας πουν που χρησιμοποιείται ο προγραμματισμός στις μέρες μας. Οι απαντήσεις που έδωσαν ήταν αναμενόμενες μιας και η τεχνολογία είναι πλέον στην καθημερινότητά τους. Όλοι είπαν για υπολογιστές, κινητά, τάμπλετ, ιστοσελίδες, εφαρμογές, τηλεοράσεις αλλά έδωσαν και μια μη αναμενόμενη απάντηση το αμάξι και το GPS (Global Positioning System).

ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΕΙΤΑΙ;

Ο προγραμματισμός στις μέρες μας είναι παντού. Κάθε ηλεκτρονική συσκευή χρησιμοποιεί ένα κομμάτι προγραμματισμού.

Παραδείγματα:

- Υπολογιστές
- Tablet
- Κινητά - Smartphones
- Εφαρμογές
- Τηλεοράσεις - Smart TV
- Ψηφιακά Ιατρικά Εργαλεία (Ψηφιακό Θερμόμετρο, Ψηφιακό πιεσόμετρο, κτλ)
- Τραπεζικά συστήματα
- Ιστοσελίδες
- Αυτοκίνητα



Εικόνα 1.3.1

Ο προγραμματισμός είναι μια μεγάλη ένότητα. Αρχικά αναφερθήκαμε στις γλώσσες προγραμματισμού. Υπάρχουν διάφορα είδη γλωσσών προγραμματισμού. Για την επίλυση ενός υπολογιστικού προβλήματος απαιτείται η ανάπτυξη προγράμματος. Για την ανάπτυξη ενός προγράμματος μπορεί να χρησιμοποιηθεί οποιαδήποτε γλώσσα προγραμματισμού. Όμως πρέπει να έχουμε υπόψη μας ότι κάθε γλώσσα προγραμματισμού σχεδιάζεται και αναπτύσσεται για συγκεκριμένο σκοπό, δίνοντας έμφαση σε ορισμένα χαρακτηριστικά σε βάρος βέβαια κάποιων άλλων. Αυτά τα χαρακτηριστικά αλλάζουν αρκετά συχνά όμως η γλώσσα εξελίσσεται και εξαρτάται σε μεγάλο βαθμό από το λειτουργικό σύστημα και από τον εξοπλισμό. Σχεδόν όλες οι γλώσσες προγραμματισμού έχουν κοινά χαρακτηριστικά και κατά κανόνα επεξεργάζονται τους ίδιους τύπους δεδομένων, υποστηρίζουν τις ίδιες βασικές δομές και έχουν παρόμοιες εντολές. Οι νεότερες εκδόσεις των γλωσσών προγραμματισμού συνήθως διαθέτουν πλουσιότερο ρεπερτόριο εντολών και άλλων δυνατοτήτων. Εξηγήσαμε στους μαθητές ότι δεν υπάρχει καλύτερη γλώσσα προγραμματισμού, απλά υπάρχει γλώσσα καταλληλότερη για την ανάπτυξη συγκεκριμένου τύπου εφαρμογών. Για παράδειγμα αν θες να φτιάξεις μια ιστοσελίδα θα χρησιμοποιήσεις τη γλώσσα προγραμματισμού HTML (HyperText Markup Language) και PHP (Hypertext Preprocessor). Αν θες να φτιάξεις ένα πρόγραμμα θα χρησιμοποιήσεις τις γλώσσες C,C++ και JAVA. Αν θες να φτιάξεις μια εφαρμογή θα χρησιμοποιήσεις το Android IDE που στην πραγματικότητα είναι μια διαφορετική έκδοση της γλώσσας προγραμματισμού JAVA. Σε αυτό το σημείο μπορούμε να αναφέρουμε πώς εδώ γίνεται η εφαρμογή αυτού που προείπαμε ότι κάθε γλώσσα προγραμματισμού σχεδιάζεται και αναπτύσσεται για συγκεκριμένο σκοπό αλλά όλες έχουν κοινά χαρακτηριστικά. Επομένως, η ανάπτυξη εφαρμογών σε Android συσκευές γίνεται από το Android IDE και έχει αναπτυχθεί από τη γλώσσα προγραμματισμού JAVA. Εμείς στα συγκεκριμένα μαθήματα και για τον προγραμματισμό του Arduino χρησιμοποιήσαμε τη γλώσσα προγραμματισμού Wiring. Η γλώσσα Wiring προέρχεται από τη γλώσσα C και C++.

Όπως είπαμε όμως οι γλώσσες προγραμματισμού έχουν κοινά χαρακτηριστικά μεταξύ τους και ακολουθούν γενικές αρχές των σύγχρονων γλωσσών προγραμματισμού. Αυτές οι αρχές είναι οι βασικές έννοιες του προγραμματισμού. Αναλύοντας σε βάθος αυτές τις αρχές μπορούμε να δούμε το πώς και από τι αποτελείται ένα πρόγραμμα, ξεκινώντας με το αλφάβητο μιας γλώσσας το οποίο αποτελείται από γράμματα (A-Z), αριθμούς (0-9) και τους ειδικούς χαρακτήρες (+, -, *, /, =) που χρησιμοποιούνται για προκαθορισμένες ενέργειες.

Συνεχίσαμε με τις σταθερές και μεταβλητές τιμές. Εξηγήσαμε ότι οι σταθερές είναι προκαθορισμένες τιμές που δε μεταβάλλονται κατά τη διάρκεια εκτέλεσης του προγράμματος. Οι σταθερές είναι αντίστοιχου τύπου δεδομένων, δηλαδή ακέραιες, πραγματικές, αλφαριθμητικές ή λογικές. Από την άλλη οι μεταβλητές παριστάνουν μία ποσότητα που η τιμή της μπορεί να μεταβάλλεται. Η τιμή της μεταβλητής μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος, ενώ αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της μεταβλητής.

Καθώς μιλήσαμε για τις σταθερές και τις μεταβλητές είπαμε και για τύπους δεδομένων που είναι επίσης μια βασική έννοια του προγραμματισμού. Οι υπολογιστές επεξεργάζονται δεδομένα διαφόρων τύπων, γι' αυτό είναι σημαντικό να κατανοήσουμε τους διαφορετικούς τύπους δεδομένων που χειρίζονται. Οι τύποι δεδομένων που υποστηρίζουν είναι οι αριθμητικοί (ακέραιους και πραγματικούς αριθμούς) και χαρακτήρες. Ο ακέραιος τύπος δεδομένων περιλαμβάνει τους ακέραιους που είναι γνωστοί από τα μαθηματικά. Ο πραγματικός τύπος δεδομένων περιλαμβάνει τους πραγματικούς αριθμούς που γνωρίζουμε επίσης από τα μαθηματικά. Ο χαρακτήρας τύπος δεδομένων αναφέρεται τόσο σε έναν χαρακτήρα όσο και σε μία σειρά χαρακτήρων. Τα δεδομένα αυτού του τύπου μπορούν να περιέχουν οποιονδήποτε χαρακτήρα παράγεται από το πληκτρολόγιο. Οι χαρακτήρες πρέπει υποχρεωτικά να βρίσκονται μέσα σε απλά εισαγωγικά, ' '. Τα δεδομένα αυτού του τύπου, επειδή περιέχουν τόσο αλφαριθμητικούς όσο και αριθμητικούς χαρακτήρες, ονομάζονται συχνά αλφαριθμητικά.

Μετά τους τύπους δεδομένων μιλήσαμε για τους τελεστές. Υπάρχουν οι αριθμητικοί τελεστές, οι τελεστές σύγκρισης και οι λογικοί τελεστές. Οι αριθμητικοί τελεστές καλύπτουν τις βασικές πράξεις: πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση. Οι τελεστές σύγκρισης συγκρίνουν τις αριθμητικές τιμές στο πρόγραμμά μας. Οι λογικοί τελεστές χρησιμοποιούνται για να δούμε αν ισχύει κάποια κατάσταση ώστε να εκτελεστεί ο ανάλογος αλγόριθμος ή το ανάλογο κομμάτι του προγράμματός μας.

Επειδή πολλές φορές έχουμε πολλές εντολές στο πρόγραμμά μας για να διευκολυνθούμε χρησιμοποιούμε μια άλλη αρχή που λέγεται Συνάρτηση. Οι συναρτήσεις ομαδοποιούν έναν αριθμό εντολών σχηματίζοντας μία μονάδα και δίνουν στην ομάδα εντολών ένα όνομα. Έπειτα, κάθε φορά που θα χρειαστεί ο

προγραμματιστής να κάνει χρήση αυτής της ομάδας εντολών, χρησιμοποιεί το όνομα αυτής της μονάδας. Η συνάρτηση χρησιμοποιείται για να κωδικοποιεί τα απλά προβλήματα.

Αμέσως μετά αναφερθήκαμε στις εκφράσεις. Εκφράσεις είναι οι τιμές που προκύπτουν από υπολογισμό. Μια έκφραση μπορεί να αποτελείται από μια μόνο μεταβλητή ή σταθερά μέχρι και μια πολύπλοκη μαθηματική παράσταση. Οι αριθμητικές εκφράσεις υλοποιούν απλές ή σύνθετες μαθηματικές πράξεις. Κάθε έκφραση παριστάνει μια συγκεκριμένη αριθμητική τιμή, η οποία βρίσκεται μετά την εκτέλεση των πράξεων. Γι' αυτό είναι απαραίτητο όλες οι μεταβλητές που εμφανίζονται σε μια έκφραση να έχουν οριστεί προηγουμένα, δηλαδή να έχουν κάποια τιμή. Οι Λογικές Εκφράσεις ονομάζονται έτσι γιατί η τιμή τους είναι πάντοτε μια εκ των δύο λογικών τιμών, δηλαδή είτε ΣΩΣΤΟ είτε ΛΑΘΟΣ (True or False).

Ακολουθως, μιλήσαμε για τις εντολές εισόδου/εξόδου. Σχεδόν όλα τα προγράμματα υπολογιστή δέχονται κάποια δεδομένα, τα επεξεργάζονται, υπολογίζουν τα αποτελέσματα και τέλος τα εμφανίζουν. Τα δεδομένα εισάγονται κατά τη διάρκεια της εκτέλεσης του προγράμματος από μία μονάδα εισόδου, για παράδειγμα το πληκτρολόγιο και τα αποτελέσματα γράφονται σε μία μονάδα εξόδου, για παράδειγμα την οθόνη.

Τέλος, αναφερθήκαμε στο πιο βασικό κομμάτι που είναι η δομή του Προγράμματος. Όπως κάθε εντολή ακολουθεί αυστηρούς συντακτικούς κανόνες, έτσι και ολόκληρο το πρόγραμμα έχει αυστηρούς κανόνες σχετικά με τον τρόπο που εκτελείται. Υπάρχει το τμήμα δήλωσης των σταθερών του προγράμματος, αν βέβαια το πρόγραμμά μας χρησιμοποιεί σταθερές. Υπάρχει επίσης το τμήμα δήλωσης μεταβλητών, όπου δηλώνονται υποχρεωτικά τα ονόματα όλων των μεταβλητών καθώς και ο τύπος τους. Ακολουθεί το κύριο μέρος του προγράμματος που περιλαμβάνει όλες τις εκτελέσιμες εντολές. Κάθε εντολή γράφεται σε ξεχωριστή γραμμή. Στο τέλος κάθε εντολής βάζουμε το αγγλικό ερωτηματικό (;) . Τονίσαμε ότι δεν αναγνωρίζεται το ελληνικό ερωτηματικό και αυτό θα έχει ως αποτέλεσμα να μη δουλέψει η εντολή.

Τελειώνοντας με το θεωρητικό κομμάτι και με τους ορισμούς, οι μαθητές όπως είναι και αναμενόμενο λόγω της ηλικίας τους είχαν πολλές απορίες και πολλά ερωτήματα. Για αυτό όπως γίνεται και στο σχολείο όπου ο δάσκαλος πρέπει να δώσει τους ορισμούς και μετά να τους εξηγήσει είτε προφορικά είτε με γραπτά παραδείγματα ανάλογα με το πως το επιτρέπει η κατάσταση έτσι και εμείς μετά περάσαμε στην εξήγηση όσων είπαμε με παραδείγματα. Σταδιακά λοιπόν αφήσαμε τους ορισμούς και τη θεωρία και περάσαμε στην πράξη όλων αυτών και σε μια απλούστερη και πιο προσιτή επεξήγηση για τα παιδιά. Όπως φαίνεται και στην εικόνα 1.3.2, δείξαμε ένα απλό πρόγραμμα φτιαγμένο

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void) {
5     int mathimata=3;
6     float number1,number2,number3,summary;
7     char name;
8
9     printf("Give a name\n");
10    scanf("%c",&name);
11
12    printf("Give 3 numbers\n");
13    scanf("%f",&number1);
14    scanf("%f",&number2);
15    scanf("%f",&number3);
16
17    if(number1>number2){
18        summary=number1-number2+number3;
19        printf("%f",summary);
20    }
21    else {
22        summary=number2-number1+number3;
23        printf("%f",summary);
24    }
25    printf("\n");
26    printf("Name : %c ",name);
27    printf("\n");
28    printf("Mathimata : %d ",mathimata);
29    printf("\n");
30    printf("Score : 1 = %f / 2 = %f / 3 = %f ",number1,number2,number3);
31    system ("pause");
32    return EXIT_SUCCESS;
33 }

```

Εικόνα 1.3.2

στη γλώσσα προγραμματισμού C. Πρώτα απ' όλα δείξαμε την αρίθμηση των γραμμών από τα αριστερά. Το πρόγραμμα διαβάζει τις εντολές με τη σειρά και γραμμή - γραμμή. Εξηγήσαμε ότι στις δύο πρώτες γραμμές εισάγουμε κάποιες βιβλιοθήκες όπως λέγονται που εισάγουν εντολές στο πρόγραμμά μας. Στην τρίτη γραμμή ξεκινάει το μέρος όπου αρχίζουμε να γράφουμε τις εντολές μας. Το `int main(void)` στην τέταρτη γραμμή δείχνει την αρχή του προγράμματος. Το σύμβολο “{” είναι η αρχή και το σύμβολο “}” στην τριακοστή τρίτη γραμμή δείχνει που τελειώνει το πρόγραμμα. Συνήθως τα πρώτα στοιχεία που γράφουμε είναι για να δηλώσουμε τι δεδομένα θα έχουμε και τους τύπους των δεδομένων, όπως είπαμε. Επομένως, στην πέμπτη γραμμή δηλώνουμε έναν ακέραιο (`int`) με την ονομασία `mathimata` και του δίνουμε την τιμή 3. Στην έκτη γραμμή δηλώνουμε πραγματικούς αριθμούς (`float`) χωρίς αρχικές τιμές με τις ονομασίες `number1`, `number2`, `number3` και `summary`. Στην έβδομη γραμμή δηλώνουμε έναν χαρακτήρα ή αλφαριθμητικό όπως είπαμε χωρίς κάποια τιμή με την ονομασία `name`. Στην όγδοη γραμμή αφήνουμε μια κενή γραμμή χωρίς αυτό να σημαίνει πως όταν τρέξει το πρόγραμμά μας θα αφήσει κάποια κενή γραμμή. Αυτό γίνεται με μια εντολή. Η κενή γραμμή απλά μπορεί να βοηθήσει τον προγραμματιστή ώστε να ξέρει πώς να ξεχωρίζει τα διάφορα κομμάτια προγραμματισμού. Στην ένατη γραμμή έχουμε την πρώτη εντολή

εξόδου. Η εντολή `printf` εκτυπώνει στην οθόνη ότι γράψουμε μέσα στα εισαγωγικά (αυτάκια), δηλαδή θα εκτυπώσει το μήνυμα «Give a name». Τα παιδιά παρατήρησαν πως είναι γραμμένο ακόμα το «\n» και το είχαν απορία γιατί δε θα εμφανιστεί. Εδώ είναι που είπαμε πριν για τις γραμμές και την εντολή. Το «\n» όταν τρέξει και διαβαστεί από το πρόγραμμα θα μας πάει στην επόμενη γραμμή. Στη δέκατη γραμμή έχουμε την εντολή εισόδου (`scanf`). Η εντολή αυτή δέχεται δεδομένα και τα αποθηκεύει εκεί που του λέμε. Έχουμε αφήσει την ενδέκατη γραμμή κενή και στη δωδέκατη γραμμή έχουμε πάλι την εντολή εξόδου (`printf`). Στις γραμμές δεκατρία, δεκατέσσερα και δεκαπέντε έχουμε πάλι εντολές εισόδου. Οι μαθητές σε αυτό το σημείο είχαν μια μεγάλη και εύστοχη απορία. Γιατί υπάρχει το «%d» στη μία εντολή εισόδου και «%f» στις άλλες εντολές εισόδου; Έτσι λοιπόν εξηγήσαμε πώς όταν μιλήσαμε για τους διαφορετικούς τύπους δεδομένων είχε σημασία γιατί πρέπει να πούμε στο πρόγραμμά μας τι τύπο δεδομένων θα αποθηκεύσει και που. Η εντολή εισόδου αναγνωρίζει τον τύπο δεδομένων ανάλογα με το γράμμα που έχει μέσα στα εισαγωγικά. Με το «%d» μιλάμε για ακέραιους, με το «%f» για τους πραγματικούς και με το «%c» για τους χαρακτήρες – αλφαριθμητικούς. Αν πάμε πίσω θα δούμε πως στην πρώτη εντολή εισόδου με το «%c» δίπλα το αποθηκεύουμε στην τιμή με το όνομα `name` που δηλώσαμε στην αρχή ως χαρακτήρα ενώ στις άλλες εντολές με το «%f» αποθηκεύουμε τους πραγματικούς αριθμούς. Συνεχίζουμε στη δέκατη έβδομη γραμμή όπου και συναντάμε δύο τελεστές. Τον λογικό τελεστή «if» που αν τον μεταφράσουμε σημαίνει «αν» και τον τελεστή σύγκρισης. Το «if» είναι λογικός τελεστής όπου ελέγχει μια κατάσταση και ανάλογα με το αποτέλεσμα εκτελεί τον ανάλογο κώδικα. Μέσα στις παρενθέσεις του «if» ελέγχεται η συνθήκη όπως λέγεται με τον τελεστή σύγκρισης όπου μας λέει ότι αν το «`number1`» είναι μεγαλύτερο από το «`number2`» τότε θα εκτελεστεί ο κώδικας μέσα στα ειδικά σύμβολα «{» και «}» μετά το «if». Έστω ότι ισχύει η συνθήκη εκτελείται στις επόμενες γραμμές ο ακόλουθος κώδικας. Στη δέκατη όγδοη γραμμή το «`summary`» γίνεται το αποτέλεσμα της πράξης που έχει μετά από το ίσον. Έχουμε απλές πράξεις αφαίρεσης και πρόσθεσης με τους αριθμητικούς τελεστές. Μετά έχουμε πάλι μια εντολή εξόδου όπου δείχνουμε το αποτέλεσμα στην οθόνη. Εδώ οι μαθητές παρατήρησαν ότι μέσα στα εισαγωγικά της εντολής «`printf`» υπάρχει το «%f» και αμέσως μετά το «`summary`». Αμέσως κατάλαβαν ότι πρόκειται για πραγματικό αριθμό όπως είπαμε προηγουμένως και αναλύσαμε τα ανάλογα σύμβολα και ότι λογικά θα εμφανίσει το αποτέλεσμα του «`summary`» μιας και είναι αμέσως μετά τα εισαγωγικά. Ρωτήσαμε τους μαθητές σε αυτό το σημείο τι θα γίνει αν δεν ισχύει η συνθήκη μέσα στο «if». Τα παιδιά απόρησαν μέχρι που είπαν ότι αν το «if» σημαίνει «αν» όπως είπαμε τότε θα υπάρχει και κάτι ακόμα και προσπάθησαν να μας εξηγήσουν με τη δική τους λογική πως αν δεν ισχύει ότι το «`number1`» είναι μεγαλύτερο από το «`number2`» τότε θα ισχύει το αντίθετο και το «`number1`» θα είναι μικρότερο από το «`number2`». Η σκέψη των παιδιών ήταν σωστή καθώς και το αποτέλεσμα με ένα μικρό λάθος.

Εξηγήσαμε πως δε θα είναι μόνο μικρότερο αλλά ίσως να είναι και ίσο διότι υπάρχει και αυτός ο τελεστής σύγκρισης. Συνεχίσαμε να δούμε αυτήν τη θεωρία και προχωρήσαμε στην εικοστή πρώτη γραμμή που υπάρχει το «else». Σαν μετάφραση το «else» σημαίνει «αλλιώς». Δηλαδή «αν» δεν ισχύει η συνθήκη πήγαινε στο «αλλιώς» κομμάτι του κώδικα. Επομένως θα εκτελέσει τον κώδικα που υπάρχει μέσα στο «else». Αμέσως μετά από τον έλεγχο για το ποιο κομμάτι κώδικα θα τρέξει συνεχίζουμε στο πρόγραμμα μας. Στην γραμμή είκοσι πέντε βλέπουμε την εντολή εξόδου «printf» και το μόνο που υπάρχει μέσα στα εισαγωγικά είναι το «\n». Αυτό το μόνο που θα κάνει είναι να αφήσει όπως προείπαμε μια κενή γραμμή. Μετά έχουμε εντολές εξόδου και αυτό γίνεται για να εκτυπώσουμε στην οθόνη και στον χρήστη τα αποτελέσματα. Στην γραμμή τριάντα ένα είναι μια εντολή συστήματος που απλά κάνει στο πρόγραμμα μας να παραμείνει ανοιχτό όταν τελειώσει και να μπορέσουμε να δούμε τα αποτελέσματα. Ένα βασικό που πρέπει να πούμε είναι ότι ο ακέραιος αριθμός που δηλώσαμε στην αρχή δεν είχε καμία μεταβολή, επομένως μπορούμε να πούμε πως αυτή είναι μία σταθερή τιμή. Συνοψίζοντας όλο το πρόγραμμα μπορούμε να πούμε τι κάνει ακριβώς. Είναι ένα πρόγραμμα που παίρνει μια μεταβλητή ενός αλφαριθμητικού και τριών πραγματικών αριθμών, κάνει μια πράξη μεταξύ των τριών πραγματικών αριθμών και μετά εμφανίζει τα αποτελέσματα αυτών.

Αυτός είναι ένα κώδικας γραμμένος σε γλώσσα προγραμματισμού C. Εμείς θα ασχοληθούμε με την γλώσσα προγραμματισμού Wiring όπως είπαμε και αφού καταλάβαμε πώς λειτουργεί ένα πρόγραμμα ήρθε η ώρα να δούμε τις εντολές που έχει η γλώσσα Wiring. Αυτός είναι ένα πίνακας με μερικές από τις πιο σημαντικές εντολές και παραμέτρους, όπου και δόθηκε στα παιδιά για να δούνε.

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false
OUTPUT	Σταθερά	int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	-	(pin, mode)	Καθορίζει αν το συγκεκριμένο ψηφιακό pin θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην

				παράμετρο mode (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	-	(pin, pinstatus)	Θέτει την κατάσταση pinstatus (HIGH ή LOW) στο συγκεκριμένο ψηφιακό pin.
digitalRead	Συνάρτηση	int	(pin)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού pin (0 για LOW και 1 για HIGH) εφόσον αυτό είναι pin εισόδου.
analogReference	Εντολή	-	(type)	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο type για να καθορίσει την τάση αναφοράς (Vref) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το pin AREF αντίστοιχα)
analogRead	Συνάρτηση	int	(pin)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο pin αναλογικής εισόδου στην κλίμακα 0 ως Vref
analogWrite	Εντολή	-	(pin, value)	Θέτει το συγκεκριμένο ψηφιακό pin σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος value καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με value 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
delay	Εντολή	-	(time)	Σταματά προσωρινά την ροή του προγράμματος για time ms. Η παράμετρος time είναι unsigned long (από 0 ως 2 ³²). Σημειώστε ότι παρά την

				προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
Serial.begin	Μέθοδος Κλάσης	-	(datarate)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος Κλάσης	-	(data)	Διοχετεύει τα δεδομένα data για αποστολή μέσω του σειριακού interface. Η παράμετρος data μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

1.4 Τεχνολογία Arduino

Ολοκληρώνοντας τα δύο πρώτα κεφάλαια της θεωρίας κυκλώματα και προγραμματισμό έφτασε η στιγμή που θα ασχοληθούμε με το αντικείμενο που μας ενδιαφέρει πραγματικά και είναι ο σκοπός όλων αυτών των μαθημάτων, την τεχνολογία Arduino. Το Arduino είναι ένας μικροελεγκτής, που αποτελείται κυρίως από ένα chip, κάποιες εισόδους και κάποιες εξόδους. Δίνει τη δυνατότητα σε κάθε προγραμματιστή να μπορεί να αλληλεπιδράσει με άλλες συσκευές, ακόμα και με το περιβάλλον. Αυτό σημαίνει ότι με ένα Arduino, ο προγραμματιστής είναι σε θέση να «διαβάσει» καταστάσεις του περιβάλλοντος π.χ. θερμοκρασία, υγρασία, φωτεινότητα κτλ αλλά και να προγραμματίσει το Arduino στις προκαθορισμένες εξόδους, μέσω επενεργητών όπως π.χ. κίνηση μοτέρ, “άναμμα” LED, ενεργοποίηση ενός ηχητικού buzzer κ.τ.λ. Εν συντομία λοιπόν, το Arduino μπορούμε να το δούμε σαν ένα σύστημα που δέχεται κάποιες εισόδους από το περιβάλλον ή από τον χρήστη, τις επεξεργάζεται και τέλος εκτελεί κάποια ενέργεια, ανάλογα με τα αποτελέσματα της παραπάνω επεξεργασίας. Το Arduino μπορεί να χρησιμοποιηθεί σχεδόν παντού αφού μπορούμε να ελέγξουμε συσκευές που δουλεύουν με 5V άμεσα, αλλά και μέχρι 220V (όπως οι συσκευές στο σπίτι μας) έμμεσα, μέσω κάποιων ειδικών εξαρτημάτων όπως ένας ρελέ ή ένα shield. Η ιδέα είναι πως με το Arduino ελέγχουμε έναν ρελέ και αυτό με τη σειρά του ελέγχει κάποια συσκευή που λειτουργεί με 220V. Ένας ρελέ όπως λέγεται είναι μια συσκευή όπου μπορούμε να συνδέσουμε ένα καλώδιο ρεύματος. Έτσι με μπορούμε να ελέγξουμε από την καφετιέρα του σπιτιού μέχρι τον



Εικόνα 1.4.1

θερμοσίφωνο και τα ρολά στα παράθυρα αν είναι ηλεκτρικά. Οτιδήποτε λειτουργεί με ρεύμα μπορούμε να το συνδέσουμε με ένα Arduino και υπάρχουν και αν λάβουμε και υπόψη ότι μπορούμε ένα Arduino να το συνδέσουμε και με το ιντερνέτ μπορούμε να ελέγξουμε τα πάντα στο σπίτι μας απομακρυσμένα. Για παράδειγμα θα μπορούσε να είναι συνδεδεμένο με το θερμοσίφωνο του σπιτιού και πριν γυρίσεις σπίτι να έχεις δώσει εντολή μέσω ιντερνέτ να ανοίξει το θερμοσίφωνο και να βρεις έτοιμο το ζεστό νερό ή να έχεις βάλει ψηφιακό θερμόμετρο και να βλέπεις την θερμοκρασία στο σπίτι και αυτό να είναι συνδεδεμένο με καλοριφέρ και air condition και να ανοίγει θέρμανση η ψύξη ανάλογα την εποχή και την θερμοκρασία. Υπάρχουν διάφορες πλακέτες Arduino και η κάθε μία έχει τις δικές τις

εισόδους και εξόδους. Εμείς θα ασχοληθούμε συγκεκριμένα με την πλακέτα Arduino UNO (Εικόνα 1.4.1).

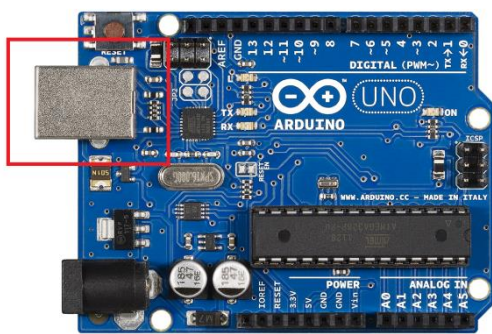
Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC (reduced instruction set computer) μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM (Static random-access memory) που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το χρόνο εκτέλεσης του προγράμματος (runtime). Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
- 1Kb μνήμης EEPROM (electrically erasable programmable read-only memory) η οποία μπορεί να χρησιμοποιηθεί για ακατέργαστη εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε μπορούμε να πούμε πως είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το υλικολογισμικό (firmware) του Arduino που είναι είδη εγκαταστημένο από τον προγραμματιστή του. Το υλικολογισμικό αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών μας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός προγραμματιστής υλικού (hardware programmer). Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση του χρόνου εκτέλεσης του προγράμματος μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).

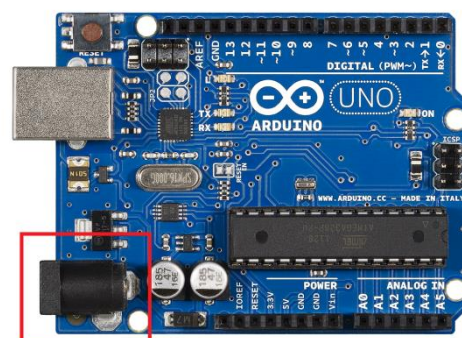
Πρέπει να προσθέσουμε πως η μνήμη Flash και η μνήμη EEPROM είναι σταθερές και ακόμα και μετά από διακοπή του ρεύματος οι πληροφορίες παραμένουν. Η μνήμη SRAM από την άλλη είναι ασταθής και οι πληροφορίες χάνονται όταν εναλλάσσεται το ρεύμα. Επειδή η SRAM δεν έχει μεγάλη μνήμη, αν τελειώσει, το πρόγραμμα μπορεί να αποτύχει με διάφορους τρόπους. Μπορεί να φαίνεται ότι το πρόγραμμα φορτώνει με επιτυχία, αλλά τελικά δεν τρέχει, ή τρέχει παράξενα. Για να ελεγχθεί εάν αυτό συμβαίνει, μπορούν να μειωθούν τα σχόλια ή οι σειρές ή άλλες δομές δεδομένων στο πρόγραμμά μας (χωρίς να αλλάξει ο κώδικας). Εάν

μετά τις αλλαγές στο πρόγραμμα μας λειτουργεί με επιτυχία, λογικά κατά πάσα πιθανότητα έχει εξαντληθεί η SRAM. Ένας τρόπος για να αντιμετωπιστεί αυτό το πρόβλημα είναι αν υπάρχουν πίνακες αναζήτησης ή άλλοι μεγάλοι πίνακες, τότε μπορεί να χρησιμοποιηθεί ο μικρότερος τύπος δεδομένων που είναι αναγκαίος για να αποθηκευτούν οι τιμές που χρειάζονται.

Η βασική πλακέτα Arduino UNO αποτελείται από μια διασύνδεση USB (Εικόνα 1.4.2) για σύνδεση με τον υπολογιστή, δίνοντας ρεύμα 5V στην πλακέτα και έτσι δίνει και τη δυνατότητα να την προγραμματίσουμε. Εκτός από τη διασύνδεση USB, στην πλακέτα υπάρχει και ένα μαύρο βύσμα (Εικόνα 1.4.3) το οποίο είναι η εξωτερική τροφοδοσία. Εκεί μπορούμε να συνδέσουμε έναν μετασχηματιστή 9V έως 12V ή μια μπαταρία ανάλογης τάσης. Με λίγα λόγια η πλακέτα Arduino UNO



Εικόνα 1.4.2

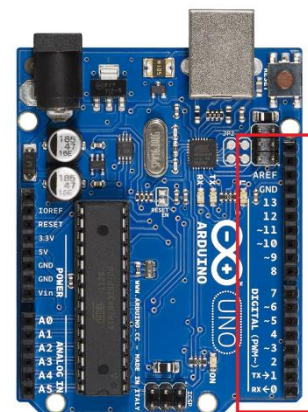


Εικόνα 1.4.3

μπορεί να χρησιμοποιηθεί και φορητά κάπου και δεν χρειάζεται να έχουμε μια πρίζα κοντά μας. Αυτό μπορεί να είναι χρήσιμο για ορισμένες εργασίες όπως για παράδειγμα την μέτρηση θερμοκρασίας σε ένα απόμερο μέρος.

Υπάρχουν 2 ειδών ακροδέκτες (pins), οι αναλογικοί και οι ψηφιακοί. Οι ψηφιακοί ακροδέκτες (Εικόνα 1.4.4) είναι στο πλήθος δεκατέσσερις. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA. Ως ψηφιακή έξοδος, ένα από αυτούς τους ακροδέκτες μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στον συγκεκριμένο ακροδέκτη. Μερικοί από αυτούς τους δεκατέσσερις ακροδέκτες, εκτός από τις ψηφιακές εισόδους/εξόδους έχουν και μια δεύτερη λειτουργία. Συγκεκριμένα:

- Με το pin0(RX) και το pin1(TX) συνήθως να χρησιμοποιούνται για επικοινωνία με TX/RX δηλαδή με τηλεκατευθύνσεις ή άλλες συσκευές επικοινωνίας. Χρησιμοποιούν σειριακή λειτουργία. Χρησιμοποιούνται για λήψη (RX)

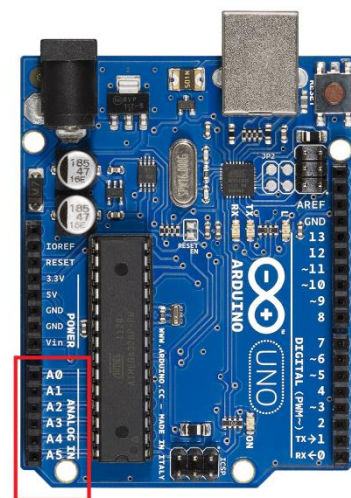


Εικόνα 1.4.4

και εκπομπή (TX) TTL σειριακών δεδομένων.

- Εξωτερικές Διακοπές: pin2 και pin3. Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης. Με την συνάρτηση `attachInterrupt()`. Ο σκανδαλισμός των διακοπών μπορεί να γίνεται στο λογικό 0 και 1.
- PWM (Pulse Width Modulation): pin3, pin5, pin6, pin9, pin10, και pin11 ακροδέκτες. Δίπλα από τους ακροδέκτες υπάρχει μαζί με τους αριθμούς το σύμβολο «~». Παρέχουν Έξοδο 8-bit (0 έως 256, καταστάσεις από 0 = σβηστό ως 255 = πλήρως αναμμένο) PWM με την συνάρτηση `analogWrite()`. Δηλαδή μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι. Το ίδιο σύστημα διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι για παράδειγμα μπορούμε ένα λαμπάκι LED να το συνδέσουμε σε κάποιο από αυτούς τους ακροδέκτες και να ελέγξουμε πλήρως την φωτεινότητά του αντί να έχουμε απλά την δυνατότητα να είναι αναμμένο ή σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Είναι σημαντικό να καταλάβετε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127 για παράδειγμα, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.
- SPI (Serial Peripheral Interface): pin10 (SS - Slave Select), pin11 (MOSI - Master Out Slave In), pin12 (MISO - Master In Slave Out), pin13 (SCK - Serial Clock). Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI, η οποία αν και παρέχεται από το hardware δεν είναι ακόμα διαθέσιμη στην γλώσσα προγραμματισμού του Arduino.
- LED (light-emitting diode): pin13. Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, το LED ανάβει ενώ όταν ο ακροδέκτης είναι LOW δεν ανάβει.

Τέλος υπάρχουν και οι αναλογικοί ακροδέκτες εισόδου. Είναι έξι ακροδέκτες στο πλήθος με ονομασίες από A0 έως A5 (Εικόνα 1.4.5). Κάνουν χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορείτε να τροφοδοτήσετε ένα από αυτά με μια τάση την οποία μπορείτε να κυμάνετε με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς V_{ref} η οποία, αν δεν κάνετε κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά σας μπορείτε να



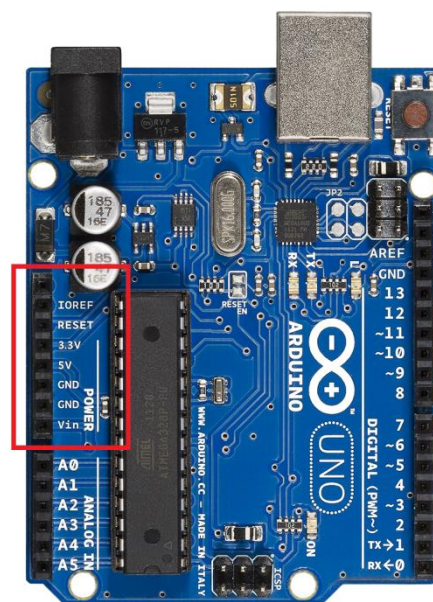
Εικόνα 1.4.5

«διαβάσετε» την τιμή του ακροδέκτη ως ένα ακέραιο αριθμό ανάλυσης 10sbit, από το 0 (όταν η τάση στον ακροδέκτη είναι 0V) μέχρι 1023 (όταν η τάση στον ακροδέκτη είναι 5V). Οι αναλογικοί ακροδέκτες δέχονται τις πληροφορίες από τις διάφορες συσκευές που μπορούν να συνδεθούν πάνω στο Arduino, και με μια κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακή και να στείλουμε εντολή στους ψηφιακούς ακροδέκτες. Οι λειτουργίες αυτών των ακροδεκτών είναι:

- TWI (Two-Wire Interface) Ακροδέκτες. Ο ακροδέκτης A4 ή αλλιώς SDA (Serial Data Line) και ο ακροδέκτης A5 ή αλλιώς SCL (Serial Clock Line). Αυτοί οι ακροδέκτες υποστηρίζουν την επικοινωνία TWI χρησιμοποιώντας την βιβλιοθήκη Wire.
- AREF : Αναφερόμενη τάση για αναλογικές εισόδους Χρησιμοποιείται με την συνάρτηση analogReference().
- Reset. Αν τεθεί σε κατάσταση LOW τότε κάνει επανεκκίνηση στον Μικροελεγκτή.

Δίπλα από τους ακροδέκτες αναλογικής εισόδου υπάρχει μια ακόμα συστοιχία από έξι ακόμα ακροδέκτες που λειτουργούν για το ρεύμα (Εικόνα 1.4.6). Η λειτουργίες του καθενός έχει ως εξής:

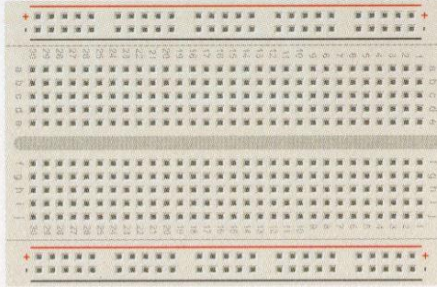
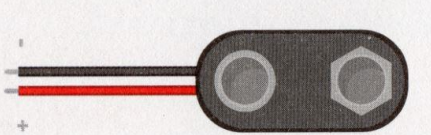
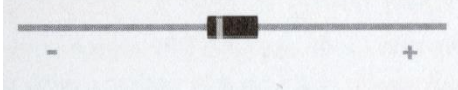
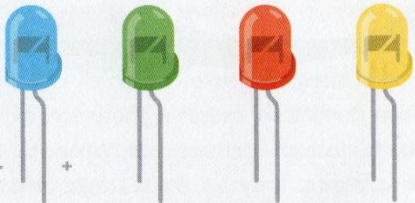
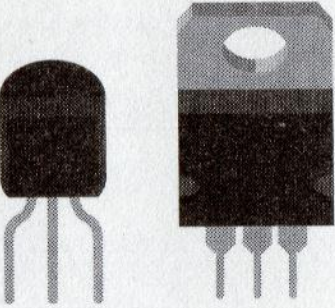
1. Ένδειξη **RESET**. Όταν γειωθεί σε οποιοδήποτε από τους δύο ακροδέκτες με την ένδειξη GND, έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
2. Ένδειξη **3.3V**. Μπορεί να τροφοδοτήσει τα εξαρτήματα του Arduino με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial Over USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
3. Ένδειξη **5V**. Μπορεί να τροφοδοτήσει τα εξαρτήματα του Arduino με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB, είτε από εξωτερική τροφοδοσία αφού πρώτα περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
4. Ένδειξη **GND**. Οι δύο ακροδέκτες με αυτή την ένδειξη είναι γειώσεις.



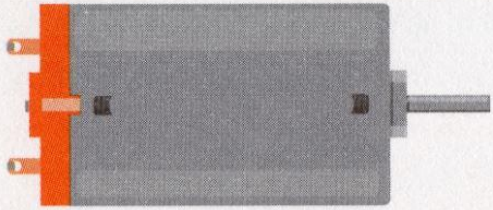

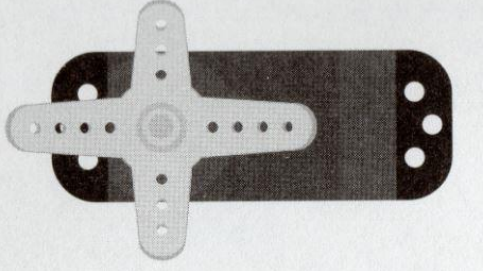
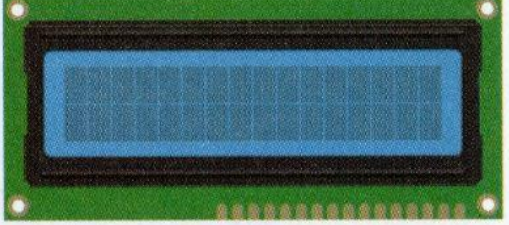

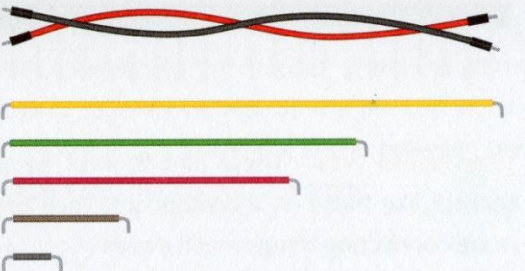
Εικόνα 1.4.6

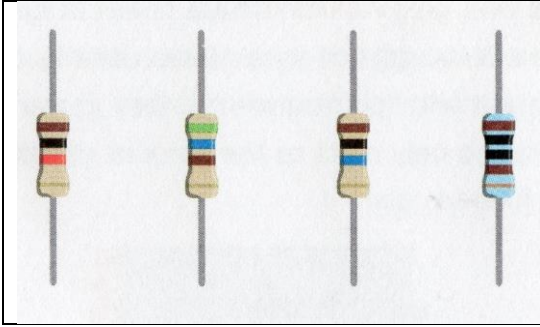
5. Ένδειξη **Vin**. Ο τελευταίος ακροδέκτης έχει διπλή λειτουργία. Σε συνδυασμό με τον ακροδέκτη γείωσης (GND) μπορεί να λειτουργήσει ως μέθοδο εξωτερικής τροφοδοσίας του Arduino στην περίπτωση που δεν χρησιμοποιήσουμε κάποια από τις άλλες τροφοδοσίες. Αν πάλι έχουμε συνδεδεμένη εξωτερική τροφοδοσία μπορούμε να χρησιμοποιήσουμε αυτόν τον ακροδέκτη για να τροφοδοτήσουμε εξαρτήματα με πλήρη τάση εξωτερικής τροφοδοσίας πριν περάσει από τον ρυθμιστή τάσης όπως γίνεται με τον ακροδέκτη των 5V.

Αφού τελειώσαμε με την ανάλυση της πλακέτας Arduino UNO ήρθε η ώρα να δούμε μερικά από τα εξαρτήματα που υπάρχουν και βοήθησε στην ανάπτυξη διαφόρων εργασιών.

	<p>Breadboard. Ένας πίνακας στον οποίο μπορείτε να φτιάξετε ηλεκτρονικά κυκλώματα. Αποτελείται από σειρές με τρύπες που σας επιτρέπουν να συνδέσετε τα καλώδια και τα εξαρτήματα μαζί.</p>
	<p>Κούμπωμα Μπαταρίας. Με αυτό συνδέουμε μια μπαταρία 9V για χρήση ρεύματος.</p>
	<p>Δίοδοι. Εξασφαλίζει ώστε το ρεύμα να πηγαίνει προς μια κατεύθυνση μόνο. Έχει σημασία από ποια μεριά θα το συνδέσουμε</p>
	<p>Λαμπάκια LED. Έχουν διάφορα χρώματα και επίσης λειτουργούν σαν δίοδους. Έχουν θετικό και αρνητικό και έχει σημασία πώς θα συνδεθούν. Λειτουργούν με τάση 3V.</p>
	<p>Τρανζίστορ. Μια συσκευή τριών "ποδιών" που λειτουργεί σαν ένας ηλεκτρικός διακόπτης. Μερικές λειτουργίες εκ των οποίων είναι η ενίσχυση, η σταθεροποίηση τάσης, η διαμόρφωση συχνότητας, η λειτουργία ως διακόπτης και ως μεταβλητή ωμική αντίσταση. Το τρανζίστορ μπορεί, ανάλογα με την τάση με την οποία πολώνεται, να ρυθμίζει την ροή του</p>

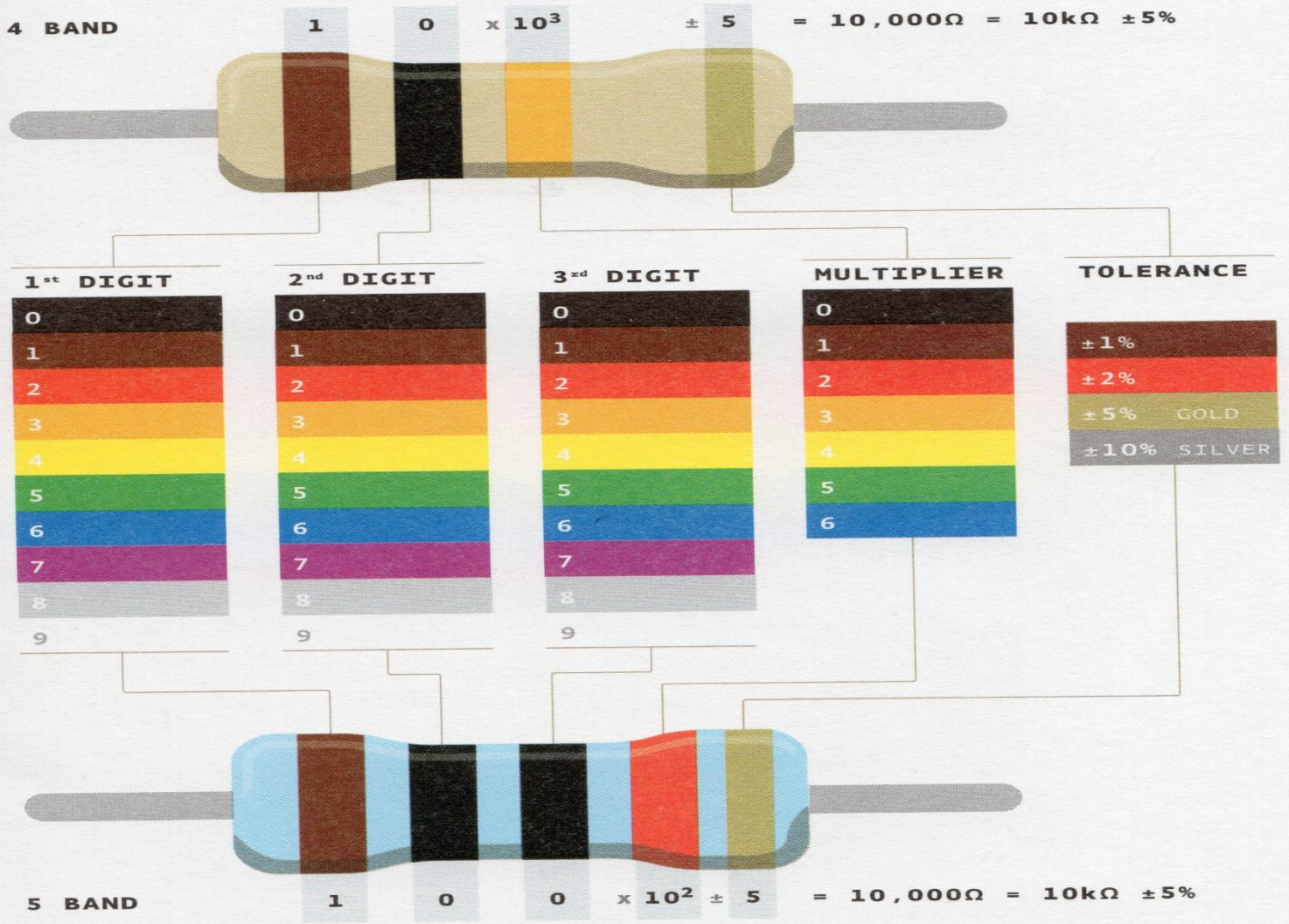
	<p>ηλεκτρικού ρεύματος που απορροφά από συνδεδεμένη πηγή τάσης.</p>
	<p>Πυκνωτές. Αυτά τα εξαρτήματα αποθηκεύουν και απελευθερώνουν ηλεκτρική ενέργεια σε ένα κύκλωμα. Όταν η τάση του κυκλώματος είναι υψηλότερη από αυτή που είναι αποθηκευμένη στους πυκνωτές επιτρέπει στο ρεύμα να ρέει δίνοντας στον πυκνωτή ένα φορτίο. Όταν η τάση του κυκλώματος είναι χαμηλότερη το αποθηκευμένο φορτίο απελευθερώνεται.</p>
	<p>Κουμπί Πίεσης. Με το συνεχόμενο πάτημά του αφήνει το ρεύμα να περάσει μέσα στο κύκλωμα.</p>
	<p>Αισθητήρας θερμοτήτας. Αλλάζει την τάση εξόδου ανάλογα με τη θερμοκρασία που έχει.</p>
	<p>Piezo. Χρησιμεύει για να ανιχνεύει δονήσεις και να παράγει ήχους.</p>
	<p>Ροοστάτης. Μεταβλητή αντίσταση με 3 ακίδες. Όσο το γυρνάς δίνει διαφορετική τάση ρεύματος σε αυτό που έχει συνδέσει.</p>

	<p>Μοτέρ. Είναι ένα ηλεκτρικό μοτέρ που μετατρέπει την ηλεκτρική ενέργεια σε κινητική.</p>
	<p>Αισθητήρας Κλίσης. Ένας τύπος διακόπτη που ανοίγει ή κλείνει ανάλογα με την κλίση του</p>
	<p>Servo motor. Ένας τύπος μοτέρ που γυρίζει περίπου 180°. Ανάλογα με το ρεύμα που δέχεται πάει στην ανάλογη θέση.</p>
	<p>Οθόνη LCD. Μια οθόνη ώστε να δείχνουμε εκεί τις πληροφορίες που θέλουμε</p>
	<p>Αντίσταση Φωτός. Μια μεταβλητή αντίσταση που αλλάζει την τάση ανάλογα με το πόσο φως πέφτει πάνω του.</p>
	<p>Καλώδια. Καλώδια για την σύνδεση μεταξύ Arduino και Breadboard και για την σύνδεση μεταξύ των εξαρτημάτων.</p>

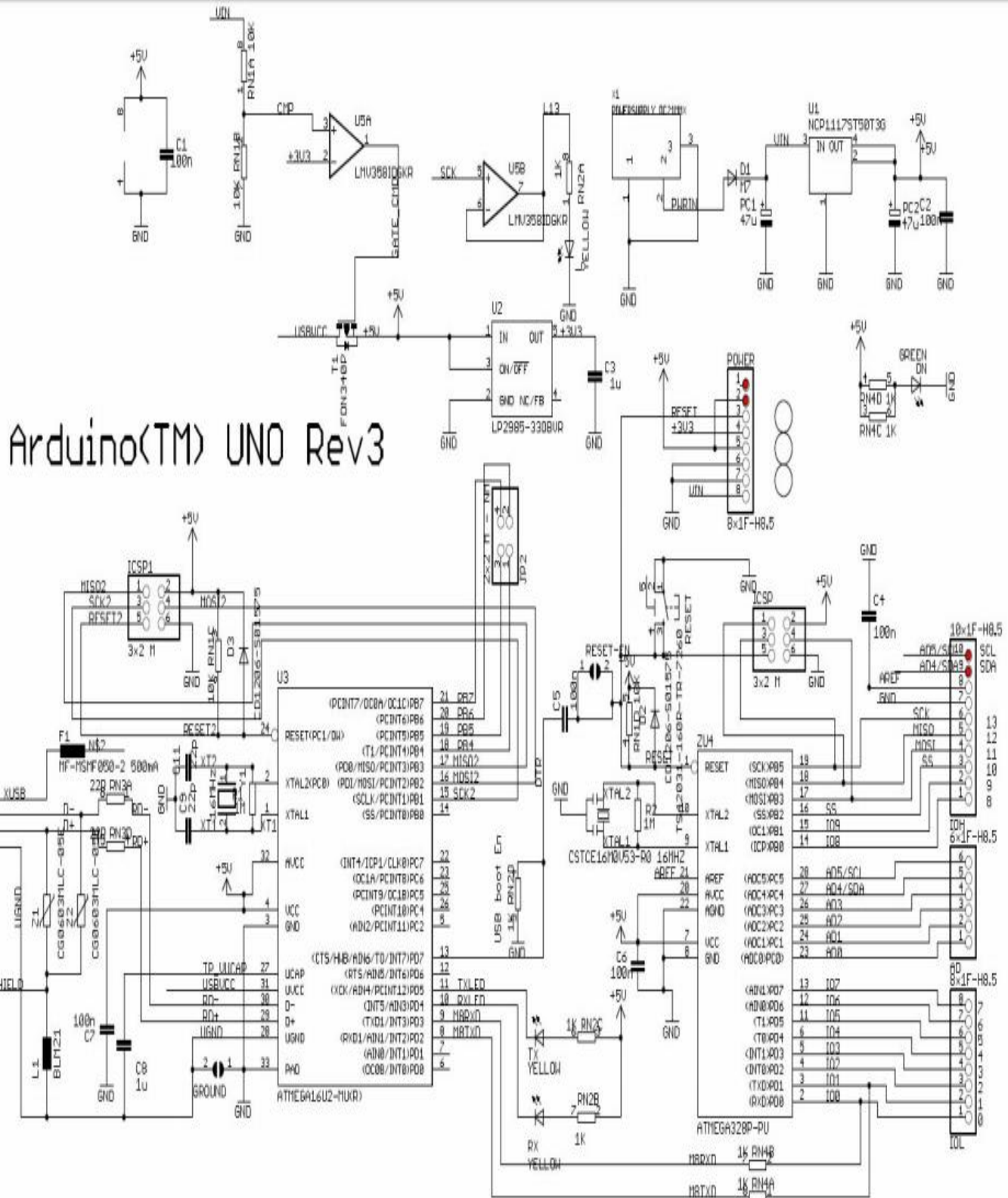


Αντιστάσεις. Αντιστέκεται στη ροή της ηλεκτρικής ενέργειας σε ένα κύκλωμα και ως αποτέλεσμα αλλάζει την τάση και το ρεύμα. Οι λωρίδες χρώματος δείχνουν την αξία της αντίστασης.

1.4.1 Πίνακας μετρήσεων αντιστάσεων



1.4.2 Σχηματικό Διάγραμμα του Arduino Uno Rev3



2. ΚΕΦΑΛΑΙΟ 2 – ΠΡΑΚΤΙΚΗ ΑΣΚΗΣΗ

2.1 Εργασία 1^η – Πρώτες Συνδεσμολογίες

Έχοντας τελειώσει με τα θεωρητικά μαθήματα πλέον και έχοντας αναλύσει τα κυκλώματα, τον προγραμματισμό και το Arduino ήρθε η στιγμή που θα θέσουμε σε εφαρμογή όσα μάθαμε. Οι μαθητές πλέον ήταν σε θέση να δουν και να δημιουργήσουν μόνοι τους μέσα από μια σειρά ασκήσεων – εργασιών τα πρώτα τους κυκλώματα και τους πρώτους κώδικες προγραμματισμού. Τα αναλύσαμε, τα εξηγήσαμε και είδαμε τις λειτουργίες τους και τις προοπτικές ανάπτυξής τους. Ύστερα από κάθε άσκηση ακολούθησαν ερωτηματολόγια για την έρευνά μας και συζήτηση για την άσκηση, ώστε να δούμε αν υπήρχε καμία ομοιότητα με ό,τι είχαν δει και που αλλού θα μπορούσαν να χρησιμοποιηθούν.

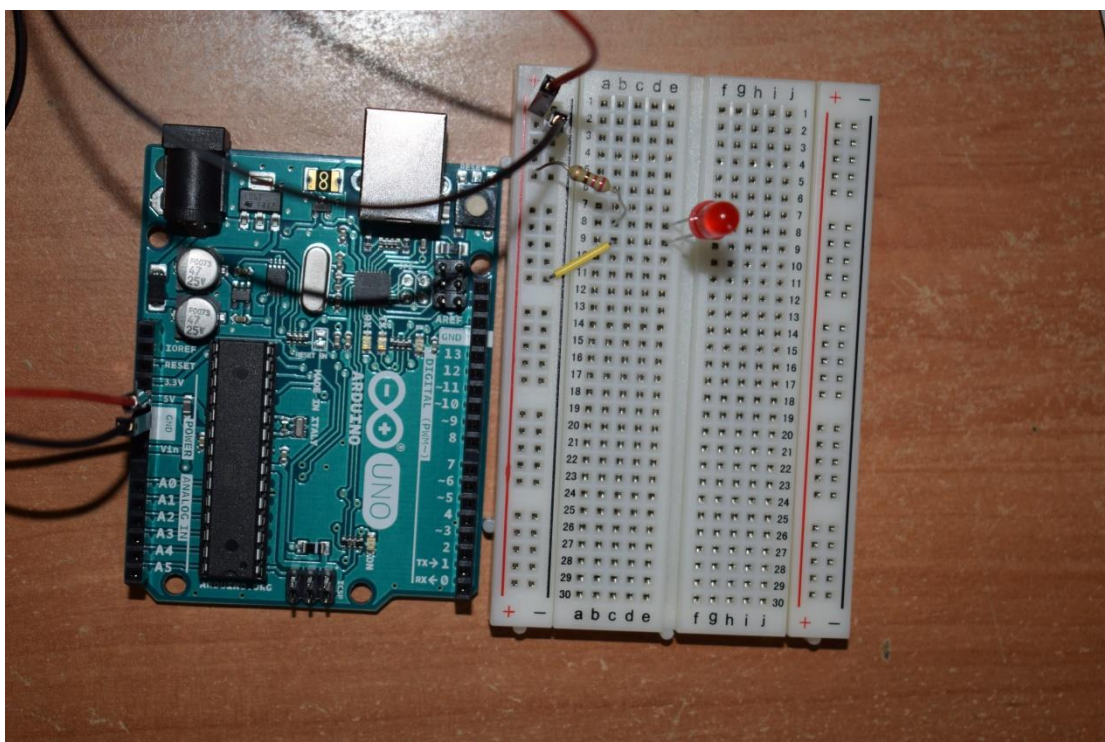
Στην πρώτη εργασία – άσκηση οι μαθητές είχαν την πρώτη τους επαφή με την πλακέτα Arduino και τα διάφορα εξαρτήματά του. Δεν υπήρξε κώδικας προγραμματισμού στην πρώτη εργασία γιατί σκοπός μας ήταν οι μαθητές να καταλάβουν πως γίνονται οι συνδέσεις στα κυκλώματα και πως συνδέονται τα διάφορα εξαρτήματα μεταξύ τους. Ακριβώς επειδή δεν υπήρχε κώδικας προγραμματισμού και ήταν η πρώτη επαφή των παιδιών με την τεχνολογία Arduino για αυτόν τον λόγο δε δόθηκε στους μαθητές ερωτηματολόγιο για τη συγκεκριμένη εργασία.

Στην πρώτη εργασία λοιπόν είδαμε πως λειτουργεί το Arduino συνδέοντας ένα λαμπάκι LED, μια αντίσταση και μερικά κουμπιά. Υπήρξαν τέσσερα βήματα όπου το καθένα ήταν συνέχεια του προηγούμενου. Αυτό έγινε για να καταλάβουν οι μαθητές πως επειδή φτιάξαμε ένα κύκλωμα δε σημαίνει πως δε μπορούμε να το επεξεργαστούμε και να το μετατρέψουμε σε ένα διαφορετικό κύκλωμα κάνοντας μια μετατροπή. Οι μαθητές έπρεπε να καταλάβουν πως ακόμα και αν κάτι δεν έχει συνδεθεί καλά ή δεν είναι στη σωστή θέση μπορούμε να τροποποιήσουμε το κύκλωμά μας και να το διορθώσουμε. Με λίγα λόγια μπορούμε το κύκλωμα που έχουμε φτιάξει ήδη, να το επεξεργαστούμε και να προσθέσουμε ή αφαιρέσουμε κάτι.

Στο πρώτο μέρος της εργασίας χρησιμοποιήσαμε εκτός από την πλακέτα Arduino, μια αντίσταση 220 OHM και ένα λαμπάκι LED που τα συνδέσαμε πάνω σε ένα Breadboard με μερικά καλώδια. Συνδέσαμε από την πλακέτα Arduino στο Breadboard με καλώδια, από το 5,5V στο θετικό (+) και από το GND στο αρνητικό (-). Έπειτα συνδέσαμε την αντίσταση πάνω στο Breadboard στο θετικό (+) και στη θέση B8. Στο σημείο αυτό εξηγήσαμε στους μαθητές ότι έπρεπε να ήμασταν ιδιαίτερα προσεκτικοί γιατί οι αντιστάσεις έχουν συγκεκριμένη φορά που συνδέονται. Η χρυσή γραμμή πάει στο τέλος της σύνδεσης. Η αντίσταση λειτουργεί σαν δίοδος και αν δεν τοποθετηθεί σωστά δε θα περάσει το ρεύμα από αυτό το

σημείο του κυκλώματός μας. Στη συνέχεια συνδέσαμε το λαμπάκι LED στη θέση E8 και στη θέση E9. Το λαμπάκι λειτουργεί επίσης σαν μια δίοδος όπως και η αντίσταση και έχει θετικό και αρνητικό. Έχει επίσης την ιδιότητα να μπορεί να λειτουργήσει σαν αντίσταση της τάξης των 3V αφού αυτήν την τάση αντέχει. Το θετικό είναι το μακρύ σιδεράκι και το αρνητικό το κοντό. Και τέλος για να ολοκληρώσουμε το κύκλωμα μας συνδέσαμε ένα καλώδιο από το B9 στο αρνητικό (-).

Το τελικό αποτέλεσμα φαίνεται στην εικόνα 2.1.1. Μια απλή σύνδεση με ένα λαμπάκι LED που όταν συνδέουμε την τροφοδοσία ρεύματος ανάβει. Οι μαθητές απόρησαν γιατί βάλουμε μια αντίσταση στο κύκλωμά μας. Όπως είπαμε στη θεωρία



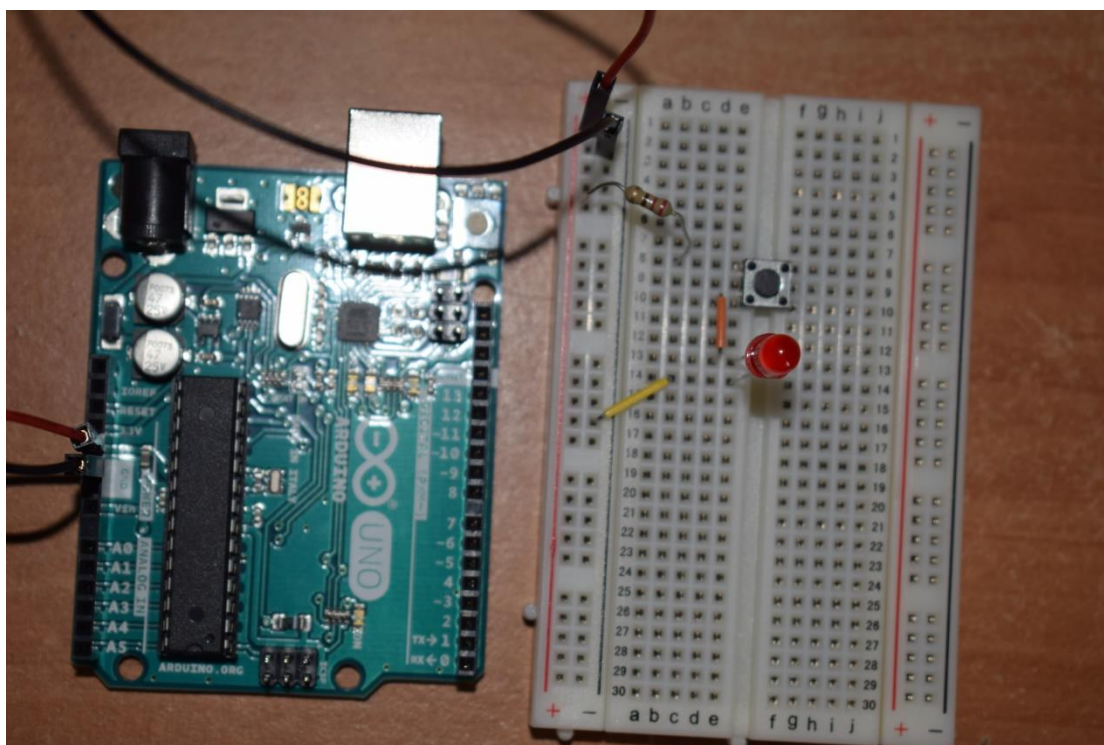
Εικόνα 2.1.1

η πλακέτα Arduino δίνει ρεύμα 5V όπως και στη σύνδεση που έγινε από την πλακέτα Arduino στο Breadboard. Το λαμπάκι LED πάλι λειτουργεί με ρεύμα της τάξης των 3V. Αν λοιπόν δε βάλουμε μια αντίσταση ώστε το ρεύμα να είναι ίσο ή λιγότερο από 3V το λαμπάκι μας θα καεί. Τα παιδιά σε αυτό το σημείο μας ρώτησαν πως αφού το λαμπάκι λειτουργεί και σαν αντίσταση αν βάζαμε 2 λαμπάκια σε σύνδεση σε σειρά όπως είπαμε στα κυκλώματα αν θα χρειαζόταν να βάλουμε την αντίσταση και αν τελικά θα άναβαν τα λαμπάκια. Εδώ καταλάβαμε πως τα παιδιά είχαν αρχίσει ήδη να καταλαβαίνουν πως λειτουργούν τα κυκλώματα και πως ήταν πολύ καλοί παρατηρητές. Η απάντηση που δώσαμε ήταν πως αν είχαμε 2 λαμπάκια συνδεδεμένα σε σειρά και θα άναβαν και δε θα χρειαζόταν να έχουμε την αντίσταση στο κύκλωμά μας. Η μόνη διαφορά είναι πως μπορεί να μην είχαν την ίδια φωτεινότητα αλλά χαμηλότερη.

Συνεχίσαμε στο δεύτερο μέρος της εργασίας. Για να αλλάξουμε κάτι στο κύκλωμα μας πρέπει πρώτα να σταματήσουμε την τροφοδοσία ρεύματος. Αυτό αν και ακούγεται λογικό κρίθηκε σκόπιμο να αναφερθεί στα παιδιά γιατί λόγω του ενθουσιασμού που είχαν να προχωρήσουν στο επόμενο μέρος της εργασίας δεν το σκέφτηκαν. Έτσι στο δεύτερο μέρος χρησιμοποιήσαμε το προηγούμενο κύκλωμα και προσθέσαμε ένα κουμπί. Δεν είχαμε καμιά μεγάλη αλλαγή στο κύκλωμά μας. Αφήσαμε τις συνδέσεις μεταξύ της πλακέτας Arduino και του Breadboard όπως και την αντίσταση όπως ήταν ήδη. Μετά συνδέσαμε το κουμπί μας πάνω στο Breadboard στις θέσεις E8, E10, F8 και F10. Εκεί βρισκόταν και το λαμπάκι μας προηγουμένως επομένως το βγάλαμε και τώρα έπρεπε να το συνδέσουμε σε καινούρια θέση. Συνδέσαμε το λαμπάκι LED στις θέσεις E13 το θετικό και E14 το αρνητικό. Ύστερα συνδέσαμε με ένα καλώδιο το κουμπί με το λαμπάκι LED. Το καλώδιο συνδέεται στη θέση D10 που είναι δίπλα στο κουμπί και στην θέση D13 δίπλα από το λαμπάκι LED. Αν παρατηρήσουμε καλύτερα εκεί που τελειώνει το ένα ξεκινάει το άλλο. Και τέλος συνδέσαμε ένα καλώδιο από το B14 στο αρνητικό (-).

Το τελικό αποτέλεσμα φαίνεται στην εικόνα 2.1.2. Συνδέσαμε την πλακέτα Arduino στην τροφοδοσία και παρατηρήσαμε τι θα συμβεί. Το λαμπάκι μας δεν άναβε πλέον όμως με το πάτημα του κουμπιού άναβε και παρέμενε αναμμένο όσο πατούσαμε το κουμπί. Το κουμπί είναι κουμπί πίεσης και δεν αφήνει το ρεύμα να περάσει αν δεν παραμείνει πατημένο.

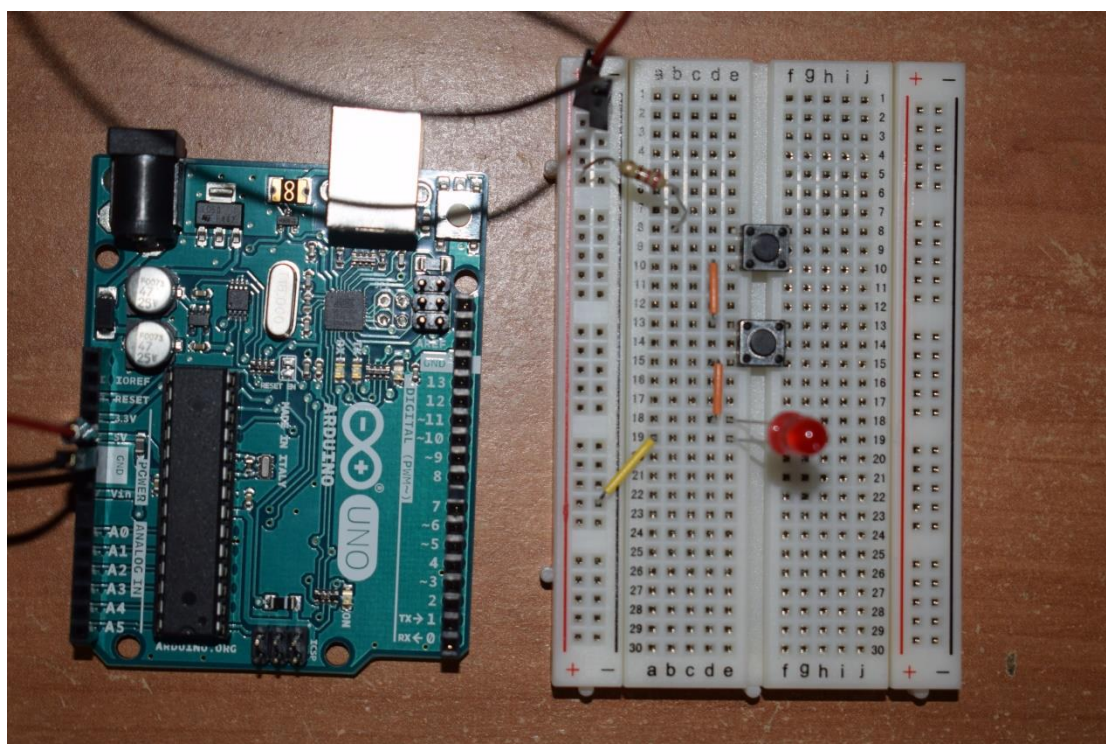
Πάμε τώρα στο τρίτο μέρος της εργασίας. Εδώ προσθέσαμε ένα δεύτερο κουμπί στο κύκλωμά μας για να δούμε τι διαφορά θα έχει στη λειτουργία του κυκλώματος.



Εικόνα 2.1.2

Αφήσαμε πάλι τη σύνδεση πλακέτας Arduino και Breadboard ως έχει. Αφήσαμε επίσης την αντίσταση και το πρώτο κουμπί ως έχουν. Συνδέσαμε το δεύτερο κουμπί στις θέσεις E13, E15, F13 και F15. Εκεί που ήταν το λαμπάκι δηλαδή τοποθετήσαμε το δεύτερο κουμπί μας και πλέον το καλώδιο που συνέδεε το πρώτο κουμπί με το λαμπάκι LED τώρα συνδέει τα 2 κουμπιά. Συνδέσαμε το λαμπάκι στις θέσεις E18, E19 και με ένα καλώδιο συνδέσαμε από το δεύτερο κουμπί στο λαμπάκι μας στις θέσεις D15, D18. Και πάλι για να ολοκληρωθεί το κύκλωμά μας συνδέσαμε ένα καλώδιο από το A19 στο αρνητικό.

Το τελικό αποτέλεσμα φαίνεται στην εικόνα 2.1.3. Συνδέσαμε την πλακέτα Arduino



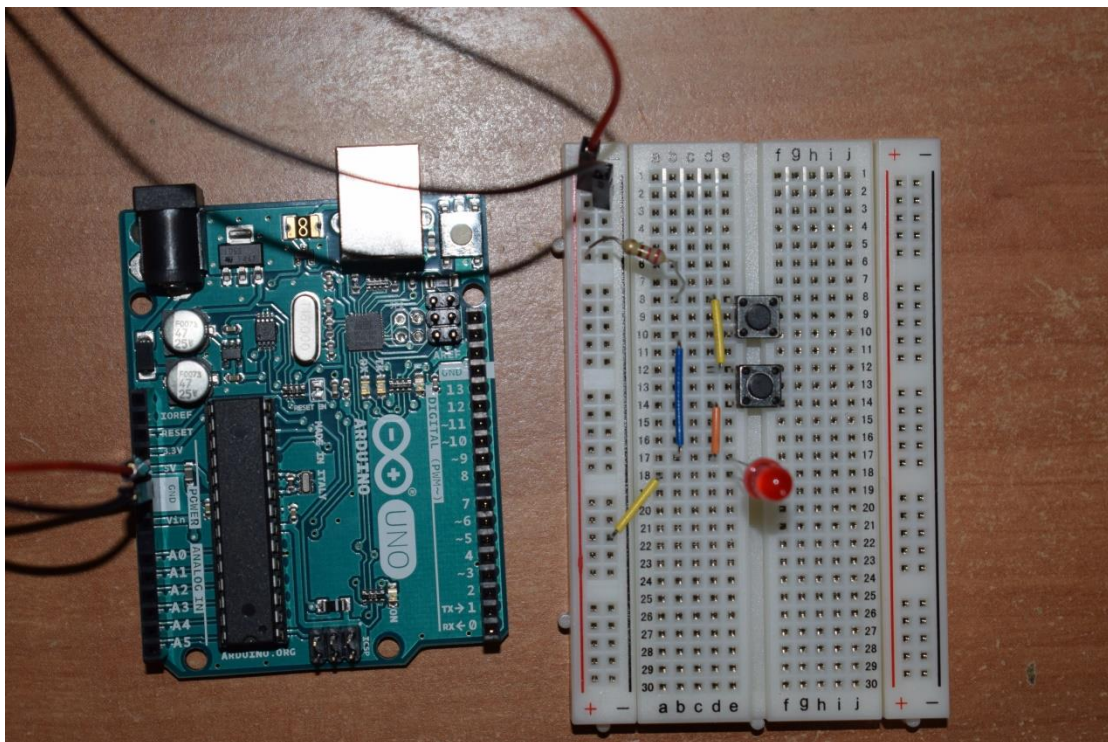
Εικόνα 2.1.3

στην τροφοδοσία και παρατηρήσαμε τα καινούρια αποτελέσματα. Παρατηρήσαμε ότι το λαμπάκι δεν άναβε πάλι και ήταν αναμενόμενο όπως και στο προηγούμενο μέρος της εργασίας αφού δεν είχε αλλάξει και πολύ. Πατώντας όμως το ένα από τα δύο κουμπιά όποιο και να είναι αυτό είδαμε το ίδιο αποτέλεσμα, δηλαδή το λαμπάκι παρέμεινε κλειστό. Όταν όμως, πατήσαμε και τα δύο κουμπιά μαζί είδαμε το λαμπάκι μας να ανάβει. Οι μαθητές αμέσως κατάλαβαν πώς αυτή είναι μια σύνδεση σε σειρά και για να ανάψει το λαμπάκι πρέπει να περάσει και από τα δύο κουμπιά το ρεύμα άρα πρέπει να είναι πατημένα και τα δύο ταυτόχρονα.

Φτάσαμε έτσι στο τέταρτο και τελευταίο μέρος της εργασίας μας. Σε αυτό το μέρος δεν θα προσθέσαμε κάτι περεταίρω στο κύκλωμά μας, απλά αλλάξαμε τη συνδεσμολογία για να δημιουργήσουμε ένα διαφορετικό κύκλωμα. Αφήσαμε τη σύνδεση μεταξύ πλακέτας Arduino και Breadboard, αφήσαμε και τα κουμπιά στις

ίδιες θέσεις και μεταφέραμε το λαμπάκι μας στις θέσεις E18, E19. Αλλάξαμε την καλωδίωση στο κύκλωμά μας, συνδέοντας τα καλώδια στις θέσεις B10 με B17, D8 με D12 και D14 με D17 και ολοκληρώσαμε το κύκλωμά μας συνδέοντας ένα καλώδιο από το A18 στο αρνητικό.

Το τελικό αποτέλεσμα φαίνεται στην εικόνα 2.1.4. Συνδέσαμε την τροφοδοσία ρεύματος και είδαμε τα αποτελέσματα. Το λαμπάκι όπως ήταν αναμενόμενο δεν άναψε και συνεχίσαμε να πατήσουμε ένα κουμπί. Πατήσαμε το πρώτο και άναψε το λαμπάκι μας. Πατήσαμε το δεύτερο και πάλι άναψε το λαμπάκι μας. Πατήσαμε και τα δύο μαζί και πάλι άναψε. Οι μαθητές αναγνώρισαν και πάλι τη σύνδεση. Είναι μια παράλληλη σύνδεση και όποιο κουμπί και να πατηθεί θα δώσει ρεύμα στο λαμπάκι μας.



Εικόνα 2.1.4

2.2 Εργασία 2^η – Φωτεινός Σηματοδότης

Έχοντας τελειώσει με την πρώτη εργασία και αφού οι μαθητές απέκτησαν μια πρώτη εμπειρία και επαφή με την πλακέτα Arduino και με τα διάφορα εξαρτήματά της, προχωρήσαμε στη δεύτερη εργασία όπου εισήχθη και ο προγραμματισμός. Στη δεύτερη εργασία είδαμε πως λειτουργεί η πλακέτα του Arduino συνδέοντας μερικά λαμπάκια LED, μερικές αντιστάσεις και ένα κουμπί. Σκοπός μας ήταν να προγραμματίσουμε το Arduino και να καθορίσουμε πότε θα ανάβει το κάθε λαμπάκι και για πόση ώρα θα παραμένει αναμμένο. Με το πάτημα λοιπόν του κουμπιού καθορίζαμε πότε θα ανάψει το λαμπάκι και για πόση ώρα θα παραμείνει αναμμένο.

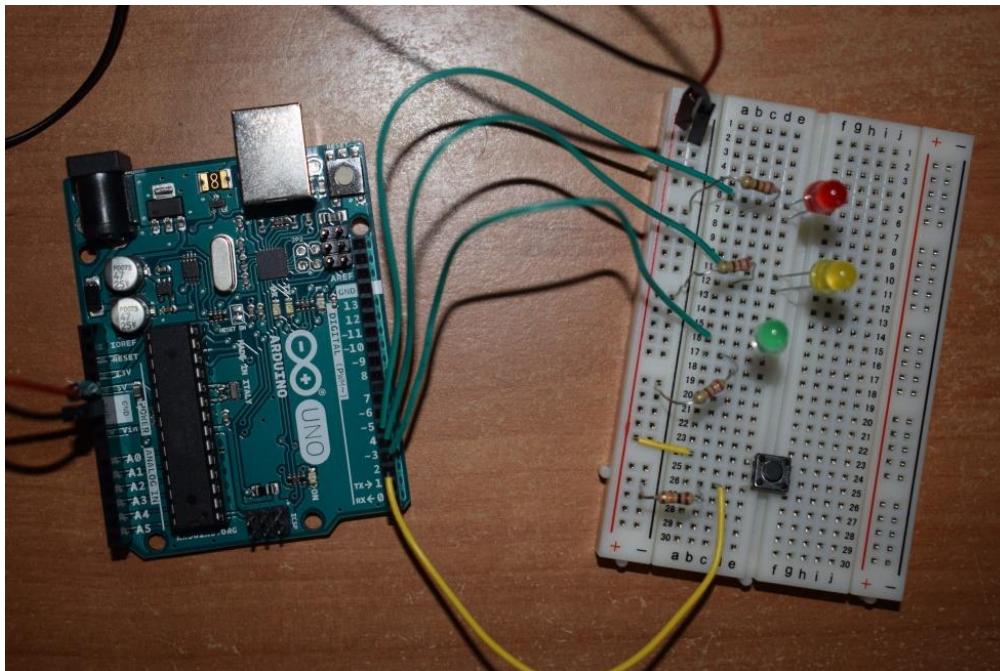
Τα εξαρτήματα που χρειαστήκαμε για αυτήν την εργασία ήταν τρεις αντιστάσεις των 220Ω και μια αντίσταση των 10KΩ, τρία λαμπάκια LED με διαφορετικά χρώματα και συγκεκριμένα ένα κόκκινο, ένα κίτρινο και ένα πράσινο και ένα κουμπί.

Αρχικά, ξεκινήσαμε με τις συνδέσεις μας στο Breadboard. Πρώτα από όλα συνδέσαμε τις αντιστάσεις των 220Ω. Συνδέσαμε την πρώτη αντίσταση στο B7 το θετικό (+) και μετά στο αρνητικό (-), τη δεύτερη από το B12 το θετικό στο αρνητικό (-) και την τρίτη από το B17 το θετικό στο αρνητικό (-). Έπειτα έπρεπε να συνδέσουμε τα λαμπάκια LED. Αρχικά, συνδέσαμε το κόκκινο λαμπάκι στο E6 το θετικό και στο E7 το αρνητικό. Ακολούθως συνδέσαμε το κίτρινο λαμπάκι στο E11 το θετικό και στο E12 το αρνητικό και τέλος το πράσινο λαμπάκι στο E16 το θετικό και στο E17 στο αρνητικό. Μετά συνδέσαμε το κουμπί στις θέσεις E24, E26, F24 και F26. Στη συνέχεια συνδέσαμε την αντίσταση των 10KΩ στη θέση B26 στο θετικό και απευθείας στο αρνητικό. Τελειώνοντας, συνδέσαμε ένα καλώδιο από το θετικό στο A24. Αυτές ήταν οι συνδέσεις μας στο Breadboard.

Μετά από όλα αυτά έπρεπε να συνδέσουμε την πλακέτα Arduino με το Breadboard. Χρησιμοποιήσαμε καλώδια και συνδέσαμε με την εξής σειρά:

- Από το 5V στο θετικό (+)
- Από το GND στο αρνητικό (-)
- Από το 2 στο C26
- Από το 3 στο A16
- Από το 4 στο A11
- Από το 5 στο A6

Στην εικόνα 2.2.1 φαίνεται το τελικό αποτέλεσμα.



Εικόνα 2.2.1

Αφού τελειώσαμε με τις συνδέσεις, παρατηρήσαμε πως είναι όλα συνδεδεμένα πάνω στους ψηφιακούς ακροδέκτες της πλακέτας Arduino και αρχίσαμε να προγραμματίζουμε την πλακέτα μας μέσω της εφαρμογής Arduino IDE.

1.	<code>int switchstate = 0;</code>	Δήλωση ακέραιης μεταβλητής με την ονομασία <code>switchstate</code> και δίνουμε την τιμή 0.
2.	<code>void setup() {</code>	Η αρχή του προγράμματος για τη δήλωση των ενεργειών των ακροδεκτών. Τρέχει μια φορά στο πρόγραμμά μας
3.	<code>pinMode(2, INPUT);</code>	Ορισμός του ακροδέκτη 2 ως εισαγωγή πληροφοριών.
4.	<code>pinMode(3, OUTPUT);</code>	Ορισμός του ακροδέκτη 3 ως έξοδο πληροφοριών.
5.	<code>pinMode(4, OUTPUT);</code>	Ορισμός του ακροδέκτη 4 ως έξοδο πληροφοριών.
6.	<code>pinMode(5, OUTPUT);</code>	Ορισμός του ακροδέκτη 5 ως έξοδο πληροφοριών.
7.	<code>}</code>	Τέλος προγράμματος δήλωσης ενεργειών.
8.	<code>void loop() {</code>	Αρχή προγράμματος εκτέλεσης κώδικα. Τρέχει συνεχώς σαν βρόχος και κάνει το ίδιο ξανά και ξανά.
9.	<code>switchstate = digitalRead(2);</code>	Η μεταβλητή μας <code>switchstate</code> διαβάζει και παίρνει την τιμή από τον ακροδέκτη 2.
10.	<code>if (switchstate == LOW) {</code>	Έναρξη πρώτης δομής ελέγχου της μεταβλητής <code>switchstate</code> . Αν ισχύει η συνθήκη εκτελείται ο κώδικας.
11.	<code>digitalWrite(3, HIGH);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.

12.	<code>digitalWrite(4, LOW);</code>	Γράφει στον ακροδέκτη 4 την τιμή που του λέμε.
13.	<code>digitalWrite(5, LOW);</code>	Γράφει στον ακροδέκτη 5 την τιμή που του λέμε.
14.	<code>delay(1000);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή.
15.	<code>digitalWrite(3, LOW);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
16.	<code>digitalWrite(4, LOW);</code>	Γράφει στον ακροδέκτη 4 την τιμή που του λέμε.
17.	<code>digitalWrite(5, HIGH);</code>	Γράφει στον ακροδέκτη 5 την τιμή που του λέμε.
18.	<code>delay(1000);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή.
19.	<code>digitalWrite(3, LOW);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
20.	<code>digitalWrite(4, HIGH);</code>	Γράφει στον ακροδέκτη 4 την τιμή που του λέμε.
21.	<code>digitalWrite(5, LOW);</code>	Γράφει στον ακροδέκτη 5 την τιμή που του λέμε.
22.	<code>delay(1000);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή.
23.	<code>}</code>	Τέλος πρώτης δομής ελέγχου της μεταβλητής switchstate.
24.	<code>else {</code>	Αρχή δεύτερης δομής ελέγχου της μεταβλητής switchstate. Αν ισχύει η συνθήκη εκτελείται ο κώδικας.
25.	<code>delay(1000);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή.
26.	<code>digitalWrite(3, HIGH);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
27.	<code>digitalWrite(4, LOW);</code>	Γράφει στον ακροδέκτη 4 την τιμή που του λέμε.
28.	<code>digitalWrite(5, LOW);</code>	Γράφει στον ακροδέκτη 5 την τιμή που του λέμε.
29.	<code>delay(2500);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή
30.	<code>digitalWrite(3, LOW);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
31.	<code>digitalWrite(4, LOW);</code>	Γράφει στον ακροδέκτη 4 την τιμή που του λέμε.
32.	<code>digitalWrite(5, HIGH);</code>	Γράφει στον ακροδέκτη 5 την τιμή που του λέμε.
33.	<code>delay(1000);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή.
34.	<code>}</code>	Τέλος δεύτερης δομής ελέγχου της μεταβλητής switchstate.
35.	<code>}</code>	Τέλος προγράμματος εκτέλεσης κώδικα. Το τέλος του βρόγχου και ξεκινάει πάλι από την αρχή.

Τελειώνοντας με το θεωρητικό μέρος περάσαμε στην πράξη. Όταν συνδέσαμε την τροφοδοσία και φορτώσαμε τον κώδικά μας στην πλακέτα Arduino είδαμε πως πρώτα άναψε το κόκκινο λαμπάκι για ένα δευτερόλεπτο ενώ τα άλλα δύο ήταν σβηστά. Αμέσως μόλις πέρασε το ένα δευτερόλεπτο έσβησε το κόκκινο λαμπάκι και άναψε το πράσινο. Αφού πέρασε ένα δευτερόλεπτο ακόμα έσβησε και το πράσινο λαμπάκι και άναψε το κίτρινο. Αυτή η διαδικασία συνεχιζόταν μέχρι να πατήσουμε το κουμπί. Μόλις πατήσαμε το κουμπί το κύκλωμά μας ξεκίνησε να συμπεριφέρεται διαφορετικά. Πλέον το κόκκινο λαμπάκι παρέμενε ανοικτό για περισσότερο από ένα

δευτερόλεπτο. Αμέσως μόλις έσβηνε άναβε το πράσινο για ένα ακριβώς δευτερόλεπτο και μετά πάλι το κόκκινο. Παρατηρήσαμε ότι το κίτρινο λαμπάκι δεν άναβε καθόλου και το κόκκινο παρέμενε αναμμένο περισσότερη ώρα.

Οι μαθητές εντυπωσιάστηκαν και μόλις τους ρωτήσαμε αν έχουν δει κάτι παρόμοιο στην καθημερινότητά τους μας έδωσαν διάφορες απαντήσεις. Μερικές από αυτές ήταν τα φανάρια στους δρόμους και το φλας των αυτοκινήτων. Αυτή η απάντηση μας ξάφνιασε. Έτσι ζητήσαμε από τον μαθητή που μας την έδωσε να μας εξηγήσει τη σκέψη του. Εκείνος μας είπε ότι κάθε φορά ανάβει είτε το δεξί είτε το αριστερό φλας ανάλογα με το κουμπί που θα πατήσουμε ενώ όταν πατήσουμε το κουμπί του alarm ανάβουν και τα δυο ταυτόχρονα. Πράγματι η λογική για τον κώδικα προγραμματισμού ήταν σωστή. Όλοι οι μαθητές κατάφεραν να συνδέσουν αυτά που είδαν με την καθημερινότητά τους και εμείς πετύχαμε τον στόχο που είχαμε θέσει εξαρχής.

2.3 Εργασία 3^η – Ψηφιακό Θερμόμετρο

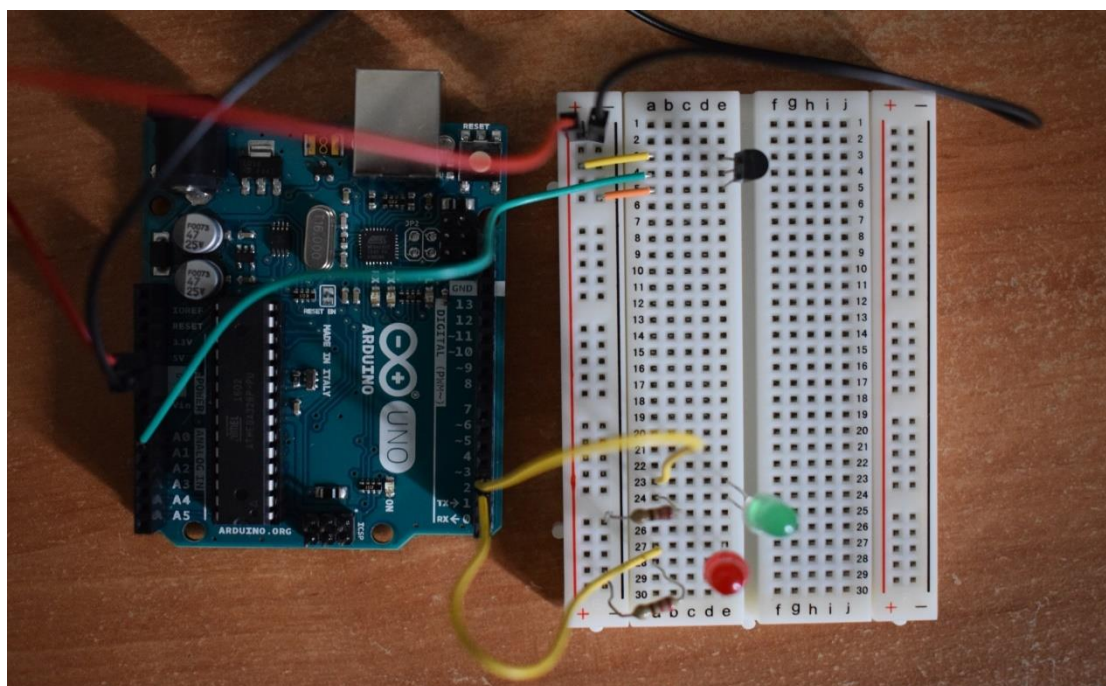
Στην τρίτη εργασία συνδέσαμε ένα ψηφιακό θερμόμετρο και μερικά λαμπάκια. Σκοπός μας ήταν να προγραμματίσουμε την πλακέτα Arduino έτσι ώστε να «διαβάζει» τη θερμοκρασία που δείχνει κάθε φορά το θερμόμετρο και ανάλογα με το αποτέλεσμα να ανάβει ένα λαμπάκι.

Τα εξαρτήματα που χρησιμοποιήσαμε ήταν δυο αντιστάσεις των 220Ω, δυο λαμπάκια LED (ένα κόκκινο και ένα πράσινο) και ένα ψηφιακό θερμόμετρο.

Ξεκινήσαμε τη συνδεσμολογία στο Breadboard πρώτα με τις δυο αντιστάσεις. Την πρώτη αντίσταση τη συνδέσαμε στο A24 το θετικό και μετά στο αρνητικό και τη δεύτερη στο A28 το θετικό και μετά στο αρνητικό. Ακολούθως συνδέσαμε τα λαμπάκια LED. Το ένα λαμπάκι στις θέσεις E23 το θετικό και E24 το αρνητικό και το άλλο λαμπάκι στις θέσεις E27 το θετικό και E28 το αρνητικό. Συνδέσαμε ακόμα το ψηφιακό θερμόμετρο στις θέσεις E2, E4 και E5. Συνδέσαμε επίσης και ένα καλώδιο από το θετικό (+) στο A3 και από το αρνητικό (-) στο A5. Ύστερα ακολούθησαν οι συνδέσεις από την πλακέτα Arduino στο Breadboard με τον εξής τρόπο:

- Από το 5V στο θετικό
- Από το GND στο αρνητικό
- Από το A0 στο A4
- Από το 2 στο A27
- Από το 3 στο A23

Το τελικό αποτέλεσμα φαίνεται στην εικόνα 2.3.1.



Εικόνα 2.3.1

Αφού κάναμε τις συνδέσεις ήρθε η ώρα να δούμε των κώδικα προγραμματισμού.

1.	<code>const int sensorPin = A0;</code>	Δήλωση σταθερής ακέραιης τιμής με την ονομασία <code>sensorPin</code> και παίρνει τιμές από τον ακροδέκτη <code>A0</code> .
2.	<code>const float baselineTemp = 25.0;</code>	Δήλωση σταθερής πραγματικής τιμής με όνομα <code>baselineTemp</code> και τιμή <code>25,0</code> .
3.	<code>void setup() {</code>	Η αρχή του προγράμματος για τη δήλωση των ενεργειών των ακροδεκτών. Τρέχει μια φορά στο πρόγραμμά μας.
4.	<code>Serial.begin(9600);</code>	Αρχικοποίηση της σειριακής επικοινωνίας με ρυθμό <code>9600 bits</code> ανά δευτερόλεπτο.
5.	<code>for (int pinNumber = 2; pinNumber < 4; pinNumber++) {</code>	Έναρξη δομής επαναληπτικού ελέγχου συνθήκης της ακέραιας μεταβλητής <code>pinNumber</code> και δήλωσή της. Θα εκτελείται ο κώδικας για όσο ισχύει η συνθήκη.
6.	<code>pinMode(pinNumber, OUTPUT);</code>	Ορισμός του ακροδέκτη που έχει αποθηκευτεί στη μεταβλητή <code>pinNumber</code> ως εξαγωγή πληροφοριών.
7.	<code>digitalWrite(pinNumber, LOW);</code>	Γράφει στον ακροδέκτη που έχει αποθηκευτεί στη μεταβλητή <code>pinNumber</code> την τιμή που του λέμε.
8.	<code>}</code>	Τέλος δομής επαναληπτικού ελέγχου.
9.	<code>}</code>	Τέλος προγράμματος δήλωσης ενεργειών.
10.	<code>void loop() {</code>	Αρχή προγράμματος εκτέλεσης κώδικα. Τρέχει συνεχώς σαν βρόχος και κάνει το ίδιο ξανά και ξανά.
11.	<code>int sensorVal = analogRead(sensorPin);</code>	Διάβασμα της αναλογικής τιμής της τάσης στο <code>sensorPin</code> και απόδοση της τιμής στη μεταβλητή <code>sensorVal</code> .
12.	<code>Serial.print("sensor Value: ");</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει ό,τι υπάρχει μέσα στα εισαγωγικά.
13.	<code>Serial.print(sensorVal);</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει την τιμή της μεταβλητής <code>sensorVal</code>
14.	<code>float voltage = (sensorVal / 1024.0) * 5.0;</code>	Δήλωση πραγματικής μεταβλητής με την ονομασία <code>voltage</code> .
15.	<code>Serial.print(", Volts: ");</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει ό,τι υπάρχει μέσα στα εισαγωγικά.
16.	<code>Serial.print(voltage);</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει την τιμή της μεταβλητής <code>voltage</code> .
17.	<code>Serial.print(", degrees C: ");</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει ό,τι υπάρχει μέσα στα εισαγωγικά.

18.	<code>float temperature = (voltage - .5) * 100;</code>	Δήλωση πραγματικής μεταβλητής με την ονομασία <code>temperature</code> .
19.	<code>Serial.println(temperature);</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει την τιμή της μεταβλητής <code>temperature</code> .
20.	<code>if (temperature < baselineTemp + 2) {</code>	Έναρξη πρώτης δομής ελέγχου της μεταβλητής <code>temperature</code> . Αν ισχύει η συνθήκη εκτελείται ο κώδικας.
21.	<code>digitalWrite(2, LOW);</code>	Γράφει στον ακροδέκτη 2 την τιμή που του λέμε.
22.	<code>digitalWrite(3, LOW);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
23.	<code>}</code>	Τέλος πρώτης δομής ελέγχου της μεταβλητής <code>temperature</code> .
24.	<code>else if (temperature >= baselineTemp + 2 && temperature < baselineTemp + 4) {</code>	Έναρξη δεύτερης δομής ελέγχου της μεταβλητής <code>temperature</code> . Αν ισχύει η συνθήκη εκτελείται ο κώδικας.
25.	<code>digitalWrite(2, HIGH);</code>	Γράφει στον ακροδέκτη 2 την τιμή που του λέμε.
26.	<code>digitalWrite(3, LOW);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
27.	<code>}</code>	Τέλος δεύτερης δομής ελέγχου της μεταβλητής <code>temperature</code> .
28.	<code>else if (temperature >= baselineTemp + 4 && temperature < baselineTemp + 6) {</code>	Έναρξη τρίτης δομής ελέγχου της μεταβλητής <code>temperature</code> . Αν ισχύει η συνθήκη εκτελείται ο κώδικας.
29.	<code>digitalWrite(2, LOW);</code>	Γράφει στον ακροδέκτη 2 την τιμή που του λέμε.
30.	<code>digitalWrite(3, HIGH);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
31.	<code>}</code>	Τέλος τρίτης δομής ελέγχου της μεταβλητής <code>temperature</code> .
32.	<code>else if (temperature >= baselineTemp + 6) {</code>	Έναρξη τέταρτης δομής ελέγχου της μεταβλητής <code>temperature</code> . Αν ισχύει η συνθήκη εκτελείται ο κώδικας.
33.	<code>digitalWrite(2, HIGH);</code>	Γράφει στον ακροδέκτη 2 την τιμή που του λέμε.
34.	<code>digitalWrite(3, HIGH);</code>	Γράφει στον ακροδέκτη 3 την τιμή που του λέμε.
35.	<code>}</code>	Τέλος τέταρτης δομής ελέγχου της μεταβλητής <code>temperature</code> .
36.	<code>delay(1);</code>	Εντολή για την καθυστέρηση του προγράμματος σε ms μέχρι την επόμενη εντολή.

37.	}	Τέλος προγράμματος εκτέλεσης κώδικα. Το τέλος του βρόγχου και ξεκινάει πάλι από την αρχή.
-----	---	---

Τελειώνοντας με το θεωρητικό μέρος και πριν περάσουμε στην πράξη μερικοί από τους μαθητές είχαν μια εύστοχη ερώτηση. Αφού το ψηφιακό θερμόμετρο είχαμε πει πως περνάει από μέσα του ρεύμα πώς θα μας δείξει τη θερμοκρασία; Εδώ έπρεπε να εξηγήσουμε πως μετατρέπουμε μέσα από τον κώδικα το ρεύμα σε θερμοκρασία. Πράγματι το ψηφιακό θερμόμετρο παίρνει τιμές μέχρι το 1024 αλλά με απλές μαθηματικές πράξεις δείξαμε την πραγματική τάση που είναι από 0V μέχρι 5V. Στη γραμμή 14 του κώδικά μας έχουμε μια πραγματική μεταβλητή που την ονομάσαμε voltage. Μέσα εκεί αποθηκεύσαμε την πραγματική τιμή σε Volt με μια απλή πράξη. Παίρνοντας την τιμή που βγάζει ο αισθητήρας του ψηφιακού θερμόμετρου διαιρέσαμε με το 1024 και μετά το αποτέλεσμα το πολλαπλασιάσαμε με το 5. Με αυτήν την απλή μαθηματική πράξη δείξαμε τα πραγματικά Volt που περνάνε μέσα από το ψηφιακό θερμόμετρο. Τότε ήρθε η στιγμή να δούμε και τη θερμοκρασία. Αν ψάξουμε λίγο το ίντερνετ θα βρούμε πληροφορίες από μηχανικούς και ηλεκτρολόγους που μας δίνουν κάποιες εξηγήσεις για τους αισθητήρες και το εύρος των Volt τους. Στη συγκεκριμένη περίπτωση στο ψηφιακό θερμόμετρο κάθε 10millivolts ισοδυναμούν σε ένα βαθμό κελσίου (1° C). Δηλαδή για κάθε 0,01V έχουμε ένα βαθμό κελσίου. Τώρα μέσα από μαθηματικές πράξεις μετατρέψαμε τα Volt σε θερμοκρασία. Αυτό το κάνουμε στη δέκατη όγδοη γραμμή του κώδικά μας. Δηλώσαμε μια πραγματική μεταβλητή με όνομα temperature. Η πράξη που γίνεται είναι που παίρνει το αποτέλεσμα από τη γραμμή 14 του κώδικα, δηλαδή την πραγματική τιμή σε Volt και αφαιρεί 0,5 μονάδες και το αποτέλεσμα το πολλαπλασιάζει με το 100. Μέσα από μαθηματικές πράξεις δείξαμε τη θερμοκρασία. Αφού εξηγήσαμε πως γίνεται η μετατροπή από Volt σε θερμοκρασία ήρθε η ώρα που συνδέσαμε την τροφοδοσία και φορτώσαμε τον κώδικά μας στην πλακέτα Arduino. Ανοίξαμε πρώτα μέσα από το πρόγραμμα του Arduino IDE την «Παρακολούθηση σειριακής» για να βλέπουμε τις πληροφορίες στην οθόνη μας. Αμέσως μόλις άνοιξε το παράθυρο στον υπολογιστή μας άρχισαν να φαίνονται οι πληροφορίες από το θερμόμετρο. Στην πλακέτα μας δε θα ανάψει κάποιο λαμπάκι αν δε φτάσει τη συγκεκριμένη θερμοκρασία. Στην οθόνη φαίνονται όλες οι θερμοκρασίες. Όταν φτάσει σε μια συγκεκριμένη θερμοκρασία και στο πρόγραμμά μας μπει μέσα στον βρόγχο για τον έλεγχο και ισχύει η συνθήκη θα ανάψει το εκάστοτε λαμπάκι LED. Τα παιδιά πειραματίστηκαν κρατώντας το ψηφιακό θερμόμετρο και προσπαθούσαν να του ανεβάσουν τη θερμοκρασία για να δουν τι θα γίνει. Όσο ανέβαινε έβλεπαν στην οθόνη τη θερμοκρασία να αλλάζει. Σιγά σιγά ανάβανε και τα λαμπάκια. Πρώτα άναψε το ένα. Μετά έσβησε το πρώτο και άναψε το δεύτερο και μόλις ανέβηκε και άλλο η θερμοκρασία ανάψανε και τα δύο. Μετά από αυτό οι μαθητές ήθελαν να ρίξουν τη θερμοκρασία έτσι ακούμπησαν ένα

παγωμένο μπουκάλι με νερό πάνω στο ψηφιακό θερμομέτρο. Σιγά σιγά τα λαμπάκια LED έσβησαν και στην οθόνη με τις πληροφορίες έπεφτε η θερμοκρασία όπως ήταν αναμενόμενο. Όταν τελειώσαμε με την εργασία οι μαθητές ρωτήθηκαν αν έχουν δει και που έχουν δει κάτι παρόμοιο. Οι μαθητές έδωσαν πολλά παραδείγματα. Είπαν:

- Το ψηφιακό θερμομέτρο για να μετρήσουμε τη θερμοκρασία του σώματος για να δούμε αν έχουμε πυρετό.
- Το θερμοσίφωνο. Το ανάβουμε και ζεσταίνει το νερό. Όταν λειτουργεί είναι αναμμένο ένα λαμπάκι και μόλις φτάσει στη θερμοκρασία που πρέπει το λαμπάκι σβήνει.
- Το ψυγείο. Λειτουργεί για να παγώνει ό,τι έχει μέσα και μόλις φτάσει στη θερμοκρασία που πρέπει σταματάει. Αυτό συμβαίνει γιατί έχει ένα ψηφιακό θερμομέτρο.
- Η κουζίνα. Όταν την ανάβουμε βάζουμε μια θερμοκρασία και ανάβει ένα λαμπάκι. Μόλις το λαμπάκι σβήσει μας λέει ότι ο φούρνος έφτασε στη θερμοκρασία που ζητήσαμε.
- Το air condition. Ανάλογα τη θερμοκρασία που θέλουμε, είτε με ζεστό είτε με κρύο αέρα, το ρυθμίζουμε και αυτό λειτουργεί μέχρι να φτάσει στη θερμοκρασία που έχουμε ζητήσει.
- Το καλοριφέρ. Όπως και το air condition έτσι και αυτό το ρυθμίζουμε και μόλις φτάσει στη θερμοκρασία που έχει ρυθμιστεί, σβήνει το καλοριφέρ. Επίσης έχει και ένα λαμπάκι για να βλέπουμε αν λειτουργεί.
- Ο βραστήρας του νερού. Μόλις τον ανοίξουμε να ζεστάνει το νερό ανάβει ένα λαμπάκι και μόλις φτάσει σε κάποια θερμοκρασία κλείνει.

Μέσα σε όλες τις απαντήσεις τρεις ήταν οι πιο αξιοσημείωτες και οι πιο απρόσμενες που ξεχώρισαν μέσα σε όλες καθώς και οι πιο ασυνήθιστες που θα μπορούσε να ακούσει κάποιος από άτομα τέτοιας ηλικίας.

- Στο αμάξι ο δείκτης της θερμοκρασία του νερού. Στα καινούρια αμάξια είναι ηλεκτρονικός ο δείκτης και δείχνει τη θερμοκρασία του νερού και αν η θερμοκρασία ανέβει πολύ ανάβει ένα λαμπάκι.
- Στο αμάξι το θερμομέτρο που λέει τη θερμοκρασία.
- Το πλυντήριο. Στο πλυντήριο ρυθμίζουμε τη θερμοκρασία που θα πλυθούν τα ρούχα και συγκεκριμένα έχει τις θερμοκρασίες 30° C, 60° C, 90° C και πάει λέγοντας. Το πλυντήριο πρώτα περιμένει το νερό να φτάσει στη θερμοκρασία που το έχουμε ρυθμίσει χωρίς να το αφήνει να το ξεπεράσει και μετά ξεκινάει.

2.4. Εργασία 4^η – Ροοστάτης και Ηλεκτρικό Μοτέρ

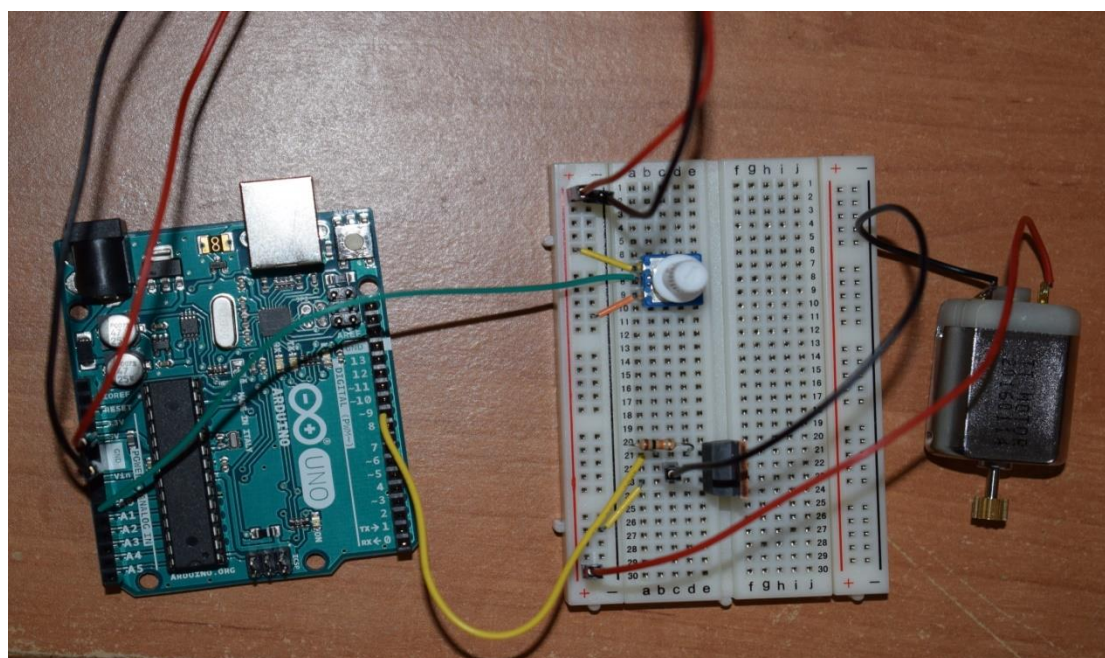
Στην τέταρτη εργασία συνδέσαμε έναν ροοστάτη και ένα ηλεκτρικό μοτέρ. Σκοπός μας ήταν να προγραμματίσουμε την πλακέτα Arduino έτσι ώστε να παίρνει από τον ροοστάτη την ανάλογη τιμή και μετά να δίνει εντολή στο ηλεκτρικό μοτέρ και όσο γυρνάμε τον ροοστάτη τόσο περισσότερο να γυρίζει το ηλεκτρικό μοτέρ.. Δηλαδή σκοπός ήταν να ελέγξουμε μέσα από τον ροοστάτη την ταχύτητα του μοτέρ.

Τα εξαρτήματα που χρησιμοποιήσαμε ήταν μία αντίσταση των 10KΩ, ένας ροοστάτης, ένα τρανζίστορ και ένα ηλεκτρικό μοτέρ.

Ξεκινήσαμε τη συνδεσμολογία στο Breadboard πρώτα με την αντίσταση. Τη συνδέσαμε στο B21 το θετικό και D21 το αρνητικό. Συνδέσαμε τα καλώδια από το θετικό (+) στο A7, από το A9 αρνητικό (-) και από το A23 στο αρνητικό (-). Έπειτα συνδέσαμε τον ροοστάτη στις θέσεις A7, A9 και E8. Μετά το τρανζίστορ στις θέσεις E21, E22 και E23. Συνδέσαμε το ηλεκτρικό μοτέρ στις θέσεις A22 το μαύρο καλώδιο που είναι το αρνητικό (-) και το κόκκινο καλώδιο απευθείας στο θετικό (+). Ύστερα ακολούθησαν οι συνδέσεις από την πλακέτα Arduino στο Breadboard με τον εξής τρόπο:

- Από το 5V στο θετικό (+).
- Από το GND στο αρνητικό (-).
- Από το A0 στο A8
- Από το 9 στο A21

Το τελικό αποτέλεσμα φαίνεται στην εικόνα 2.4.1.



Εικόνα 2.4.1

Αφού κάναμε τις συνδέσεις ήρθε η ώρα να δούμε των κώδικα προγραμματισμού.

1.	<code>const int mofPin = 9;</code>	Δήλωση σταθερής ακέραιης τιμής με την ονομασία <code>mofPin</code> και δίνουμε την τιμή 9.
2.	<code>const int potPin = A0;</code>	Δήλωση σταθερής ακέραιης τιμής με την ονομασία <code>potPin</code> και παίρνει τιμές από τον ακροδέκτη A0.
3.	<code>int motorSpeed = 0;</code>	Δήλωση ακέραιης τιμής με την ονομασία <code>motorSpeed</code> και δίνουμε την τιμή 0.
4.	<code>void setup() {</code>	Η αρχή του προγράμματος για τη δήλωση των ενεργειών των ακροδεκτών. Τρέχει μια φορά στο πρόγραμμά μας.
5.	<code>pinMode(mofPin, OUTPUT);</code>	Ορισμός του σταθερού ακέραιου <code>mofPin</code> ως έξοδο πληροφοριών.
6.	<code>pinMode(potPin, INPUT);</code>	Ορισμός του σταθερού ακέραιου <code>potPin</code> ως είσοδο πληροφοριών.
7.	<code>Serial.begin(9600);</code>	Αρχικοποίηση της σειριακής επικοινωνίας με ρυθμό 9600 bits ανά δευτερόλεπτο.
8.	<code>}</code>	Τέλος προγράμματος δήλωσης ενεργειών.
9.	<code>void loop() {</code>	Αρχή προγράμματος εκτέλεσης κώδικα. Τρέχει συνεχώς σαν βρόχος και κάνει το ίδιο ξανά και ξανά.
10.	<code>motorSpeed = analogRead(potPin) / 4;</code>	Διάβασμα της αναλογικής τιμής της τάσης στο <code>potPin</code> και απόδοση της τιμής δια τέσσερα στη μεταβλητή <code>motorSpeed</code> .
11.	<code>Serial.println(motorSpeed);</code>	Επικοινωνία της σειριακής και εμφάνιση πληροφοριών στην οθόνη μας. Εμφανίζει την τιμή της μεταβλητής <code>motorSpeed</code> .
12.	<code>analogWrite(mofPin, motorSpeed);</code>	Στέλνει αναλογικές τιμές στον ακροδέκτη που είναι αποθηκευμένος στη μεταβλητή <code>mofPin</code> και δίνει την τιμή της μεταβλητής <code>motorSpeed</code> .
13.	<code>}</code>	Τέλος προγράμματος εκτέλεσης κώδικα. Το τέλος του βρόγχου και ξεκινάει πάλι από την αρχή.

Τελειώνοντας με το θεωρητικό μέρος και πριν περάσουμε στην πράξη έπρεπε να εξηγήσουμε κάτι. Επειδή η εντολή `analogRead` στη δέκατη γραμμή του κώδικά μας διαβάζει τιμές από 0 έως 1024 πρέπει να γίνει μια μετατροπή γιατί η εντολή

analogWrite στη δωδέκατη γραμμή του κώδικα μας δίνει τιμές από 0 έως 255. Για αυτό στη δέκατη γραμμή του κώδικα όταν διαβάζει από την εντολή analogRead και πριν αποθηκευτεί το διαιρούμε με το τέσσερα και αποθηκεύεται στο εύρος των τιμών που χρειάζεται για να χρησιμοποιηθεί στην εντολή analogWrite. Αφού τελειώσαμε με τον κώδικα προγραμματισμού χρειάστηκε να εξηγήσουμε κάτι ακόμα. Το τρανζίστορ που βρίσκεται στην εργασία αυτή. Το τρανζίστορ έχει τρία πόδια, την πύλη (gate), την υποδοχή (drain) και την πηγή (source). Στην υποδοχή συνδέεται το ηλεκτρικό μοτέρ, στην πηγή συνδέουμε τη γείωση (GND) και στην πύλη συνδέουμε από τον ακροδέκτη 9 του Arduino. Το τρανζίστορ χρησιμοποιείται για να ρυθμίσει την τάση, ανάλογα με την τάση που δέχεται από τον ροοστάτη. Το τρανζίστορ που έχει συνδεθεί στο ηλεκτρικό μοτέρ θα ρυθμίσει την «ταχύτητα» του ηλεκτρικού μοτέρ ανάλογα με τις τιμές που θα πάρει. Το τρανζίστορ έχει συνδεθεί σε ακροδέκτη με PWM (Pulse Width Modulation), έτσι ανάλογα με τον «παλμό» που δίνει, αφήνει την ανάλογη τάση ρεύματος να περάσει. Το τρανζίστορ αποθηκεύει ρεύμα και ανάλογα με το αν η τάση είναι μικρότερη από ότι πρέπει δίνει ρεύμα και αν είναι μεγαλύτερη από ότι χρειάζεται δεν αφήνει το ρεύμα να περάσει. Έτσι μέσα από το ηλεκτρικό μοτέρ θα περνάει όσο ρεύμα του λέμε μέσω του ροοστάτη. Όταν ο ροοστάτης είναι κλειστός ή αλλιώς στο 0 το ηλεκτρικό μοτέρ είναι κλειστό και περνάει 0V και όταν είναι τελείως ανοιχτό ή στο 255 το ηλεκτρικό μοτέρ δουλεύει σε πλήρη ισχύ στα 5V. Υπάρχουν και οι ενδιάμεσες τιμές όμως και εδώ παίζει ρόλο το τρανζίστορ. Αν για παράδειγμα ο ροοστάτης είναι στη μέση ή στο 128 τότε αφήνει να περάσει η μισή τάση. Μπορούμε να πούμε δηλαδή 2,5V και δουλεύει το ηλεκτρικό μοτέρ με μισή ισχύ. Αφού εξηγήσαμε και αυτό περάσαμε στην πράξη συνδέοντας την τροφοδοσία και φορτώσαμε τον κώδικά μας στην πλακέτα Arduino. Ανοίξαμε πάλι πρώτα μέσα από το πρόγραμμα του Arduino IDE την «Παρακολούθηση σειριακής» για να βλέπουμε τις πληροφορίες στην οθόνη μας. Αμέσως μόλις άνοιξε το παράθυρο στον υπολογιστή μας άρχισαν να φαίνονται οι πληροφορίες από τον ροοστάτη. Ο ροοστάτης ήταν κλειστός και έτσι η ένδειξη ήταν μηδέν και το ηλεκτρικό μοτέρ δεν είχε κίνηση. Καθώς όμως γυρνάγαμε τον ροοστάτη βλέπαμε τις πληροφορίες να αλλάζουν και να ανεβαίνει η τιμή και να βλέπουμε το ηλεκτρικό μοτέρ να τίθεται σε λειτουργία και να αρχίζει να κουνιέται. Όσο πιο πολύ το γυρνάγαμε τόσο πιο μεγάλη τιμή είχαμε στην οθόνη μας και τόσο πιο γρήγορα κουνιόταν το ηλεκτρικό μοτέρ.

Όταν τελειώσαμε με την εργασία οι μαθητές ρωτήθηκαν αν έχουν δει και που έχουν δει κάτι παρόμοιο. Οι μαθητές είπαν πως έχουν δει κάτι παρόμοιο σε παιχνίδια και συγκεκριμένα τηλεκατευθυνόμενα αμάξια και ελικόπτερα. Το έχουν δει επίσης σε drone. Όσο πατάγανε το κουμπί στο τηλεχειριστήριο τόσο περισσότερο δούλευε το ηλεκτρικό μοτέρ μέσα στο παιχνίδι. Μια επίσης μη αναμενόμενη απάντηση που έδωσαν ήταν ο ηλεκτρικός αναδευτήρας για τον καφέ. Έχει ένα κουμπί και όσο το πατάμε πάει και πιο γρήγορα ο αναδευτήρας.

3. ΚΕΦΑΛΑΙΟ 3 – ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

3.1 Αποτελέσματα

Αφού τα μαθήματα τελείωσαν, συλλέξαμε τα ερωτηματολόγια και τα μελετήσαμε για να εξάγουμε τα αποτελέσματα και τα συμπεράσματά μας. Αξίζει να αναφερθεί ότι ο αριθμός των μαθητών που παρακολουθούσαν το κάθε μάθημα δεν ήταν συγκεκριμένος. Αυτό σημαίνει ότι άλλα μαθήματα τα παρακολούθησαν περισσότεροι και άλλα λιγότεροι μαθητές σε καμιά περίπτωση όμως το δείγμα μας δεν ήταν μικρότερο των 100 ατόμων. Επίσης δεδομένου ότι το δείγμα μας αποτελείτο από παιδιά μικρής ηλικίας, το μάθημα και η αξιολόγησή του δε γινόταν πάντα με την πρέπουσα σοβαρότητα. Έτσι αρκετές φορές λάβαμε ερωτηματολόγια λάθος ή ελλιπώς συμπληρωμένα ή ακόμα και κενά. Στον παρακάτω πίνακα μπορούμε να δούμε αναλυτικά τις παρουσίες των μαθητών και το σύνολο των έγκυρων και άκυρων ερωτηματολογίων.

Μαθήματα	Παρουσίες Μαθητών	Έγκυρα	Άκυρα
Εισαγωγή	111	-	-
Κυκλώματα	105	101	4
Προγραμματισμός	111	102	9
Arduino	108	102	6
Εργασία 1 ^η	104	-	-
Εργασία 2 ^η	109	102	7
Εργασία 3 ^η	108	102	6
Εργασία 4 ^η	106	103	3

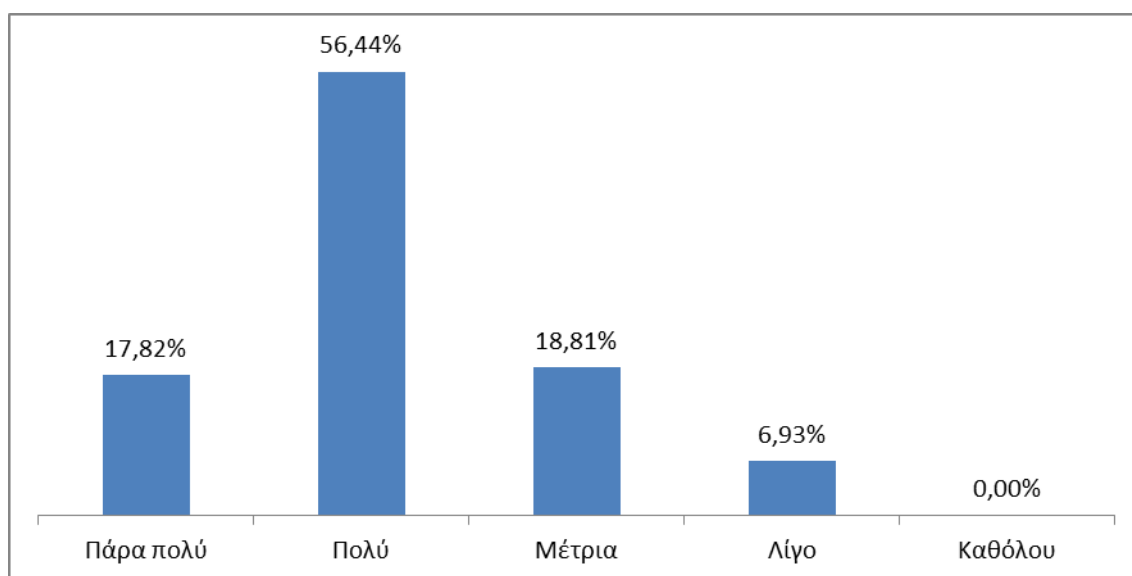
3.1.1 Αποτελέσματα Κυκλώματα ανά ερώτημα

Γράφημα 1: Ποσοστό έγκυρων και άκυρων ερωτηματολογίων.



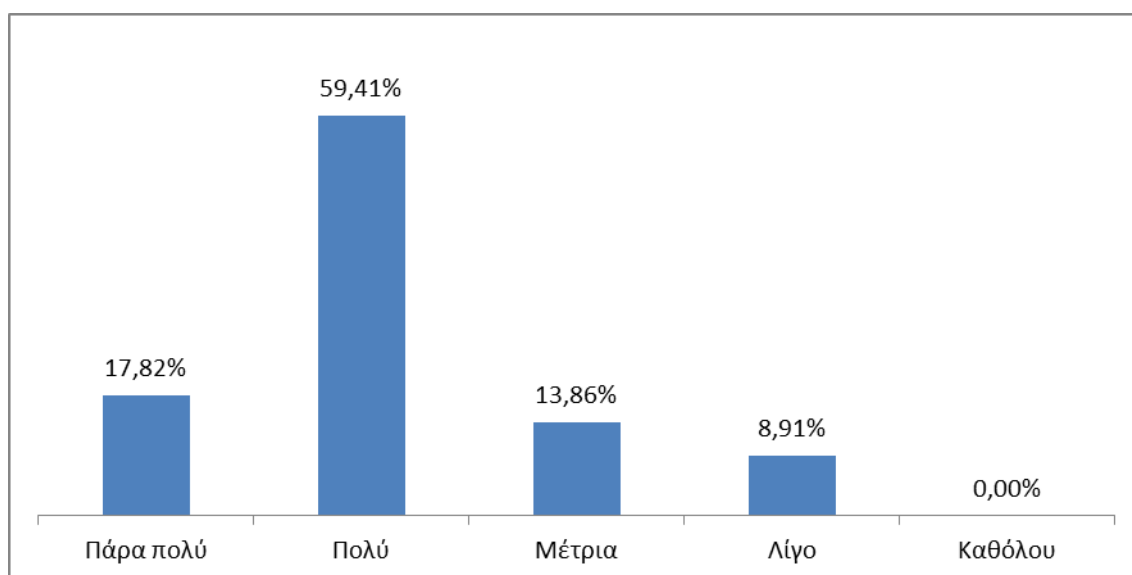
Στο σύνολο τα ερωτηματολόγια μας ήταν 105. Τα έγκυρα ερωτηματολόγια ήταν 101 και τα άκυρα 4.

Γράφημα 2: Ήταν καλά οργανωμένη η παρουσίαση του μαθήματος;



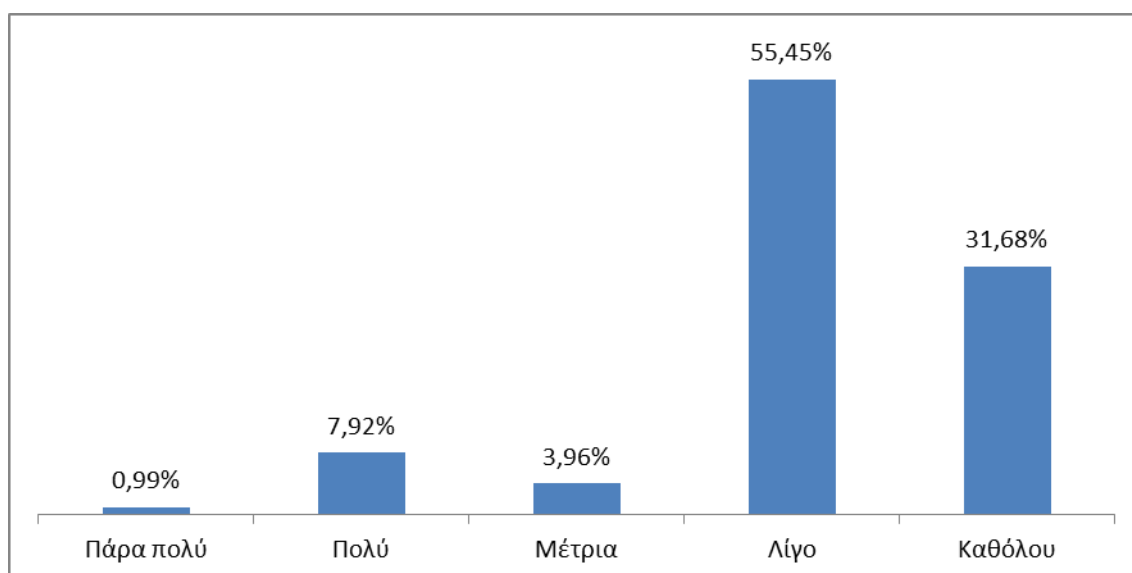
Από το γράφημα φαίνεται ότι το 74,26% του δείγματός μας θεώρησε την παρουσίαση του μαθήματος από πολύ έως και πάρα πολύ καλά οργανωμένη ενώ μόνο το 6,93% θεώρησε ότι υπάρχουν περιθώρια βελτίωσης.

Γράφημα 3: Ήταν σαφής η διδασκαλία του μαθήματος;



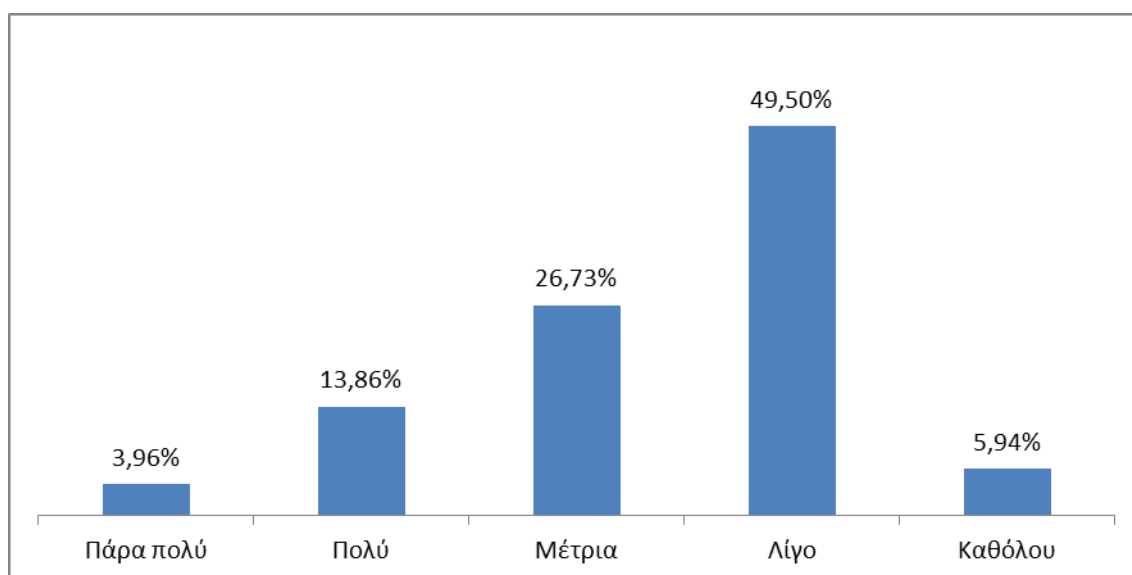
Στο σύνολο το δείγμα μας φαίνεται ικανοποιημένο από τη διδασκαλία που πραγματοποιήθηκε καθώς μόνο το 8,91% απάντησε λίγο και το 13,86% απάντησε μέτρια. Η πλειοψηφία του δείγματος θεώρησε πολύ σαφής τη διδασκαλία του μαθήματος.

Γράφημα 4: Πιστεύετε ότι χρειάζονται περισσότερα παραδείγματα για να κατανοήσετε αυτά που διδαχθήκατε;



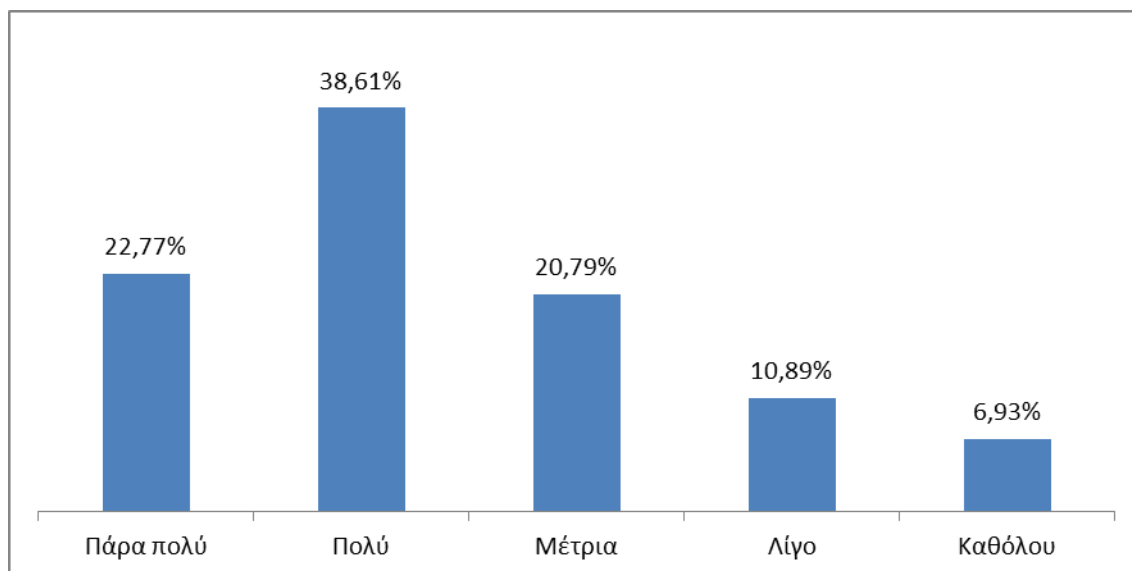
Η πλειοψηφία του δείγματος (87,13%) φαίνεται να κατανόησε πάρα πολύ καλά όλα όσα διδάχθηκε χωρίς να χρειάζεται περισσότερα παραδείγματα για περαιτέρω κατανόηση του θέματος.

Γράφημα 5: Το μάθημα σας προκάλεσε απορίες;



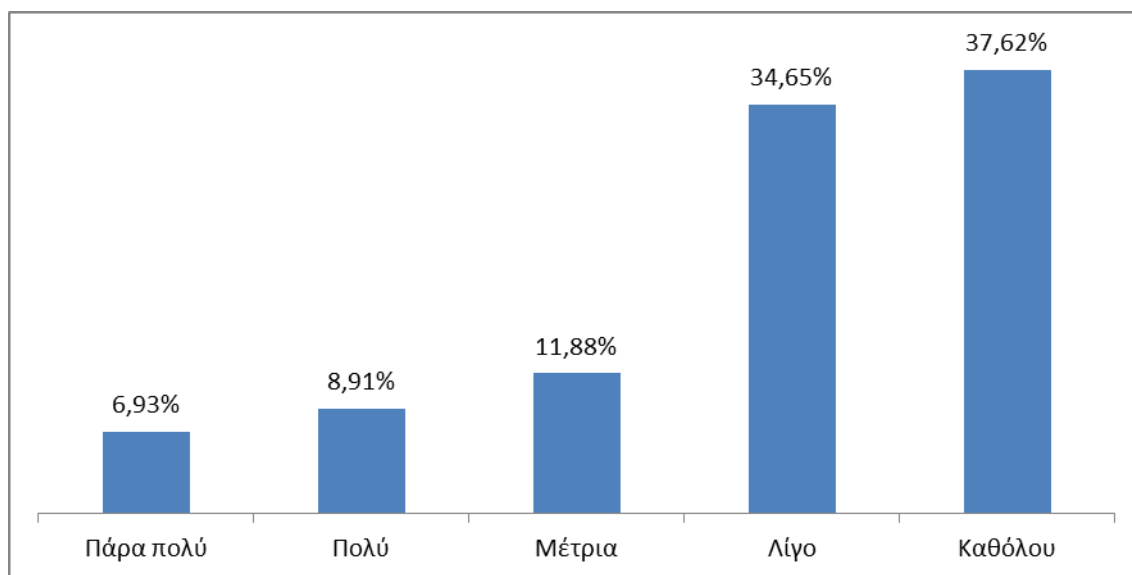
Από το γράφημα φαίνεται ότι το 17,82% του δείγματος είχε πολλές απορίες μετά το τέλος του μαθήματος, το 26,73% κατανόησε μερικώς ενώ το 55,44% κατανόησε πάρα πολύ καλά το μάθημα και είχε ελάχιστες απορίες.

Γράφημα 6: Θεωρείτε ότι μπορείτε να ανταποκριθείτε στο μάθημα με βάση αυτά που ξέρατε μέχρι τώρα;



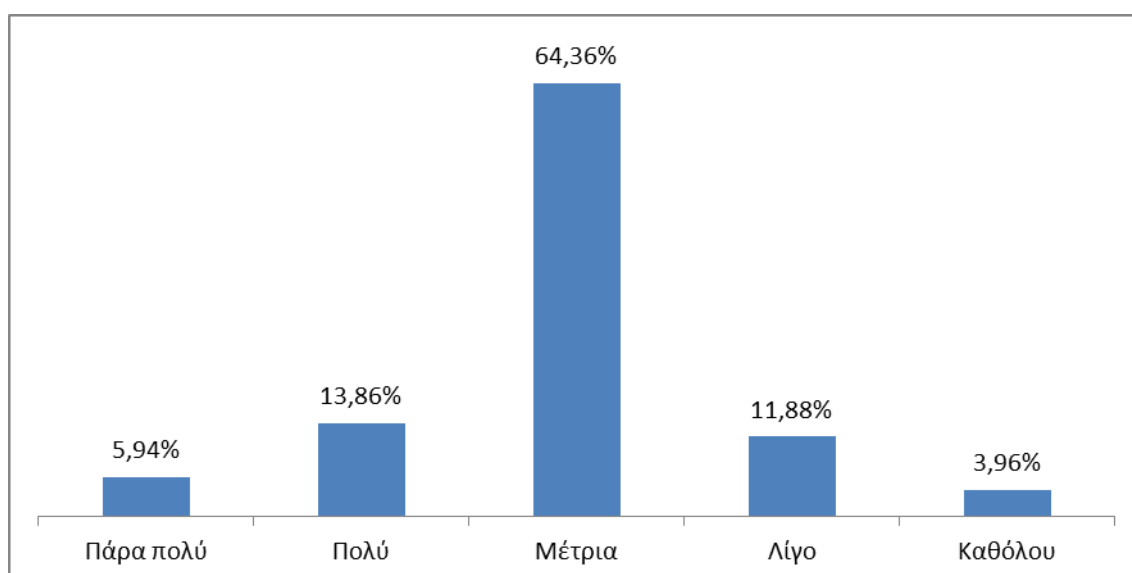
Η πλειοψηφία του δείγματος (82,17%) απάντησε ότι οι γνώσεις που είχε έως τώρα ήταν αρκετές για να μπορέσει να ανταποκριθεί στο μάθημα από μέτρια έως και πάρα πολύ καλά.

Γράφημα 7: Πιστεύετε ότι χρειάζονται περισσότερες ώρες διδασκαλίας για το συγκεκριμένο μάθημα;



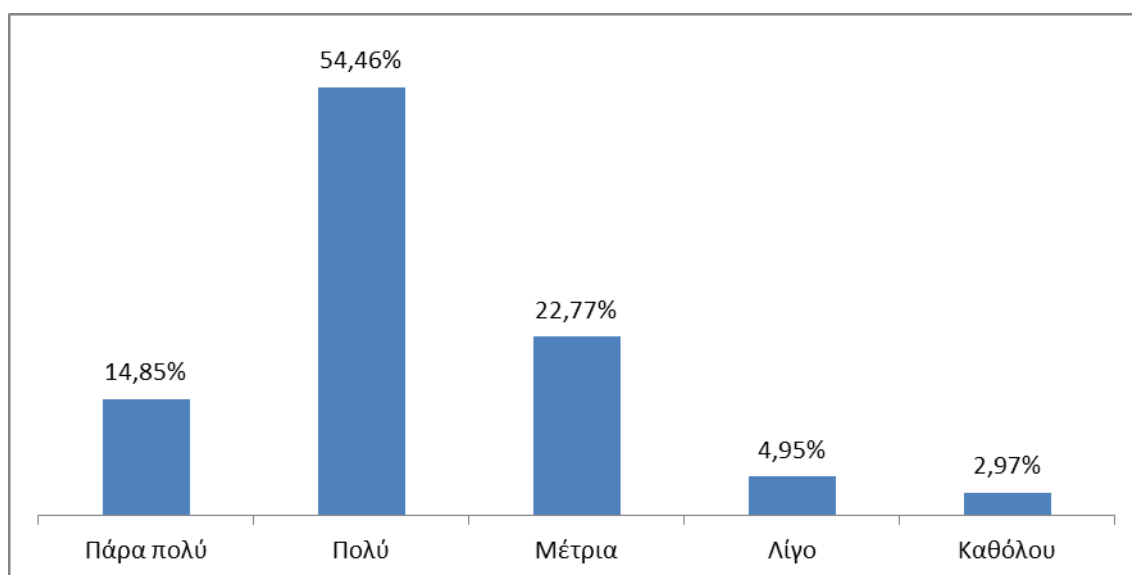
Οι περισσότεροι μαθητές θεώρησαν ότι οι ώρες που αφιερώθηκαν στη διδασκαλία του συγκεκριμένου μαθήματος ήταν αρκετές για την κατανόησή του. Μόνο το 27,72% του δείγματος έκρινε ότι ίσως να χρειάζονταν ορισμένες επιπλέον ώρες διδασκαλίας.

Γράφημα 8: Θα σας ενδιέφερε να παρακολουθήσετε παρόμοιο μάθημα στο μέλλον;



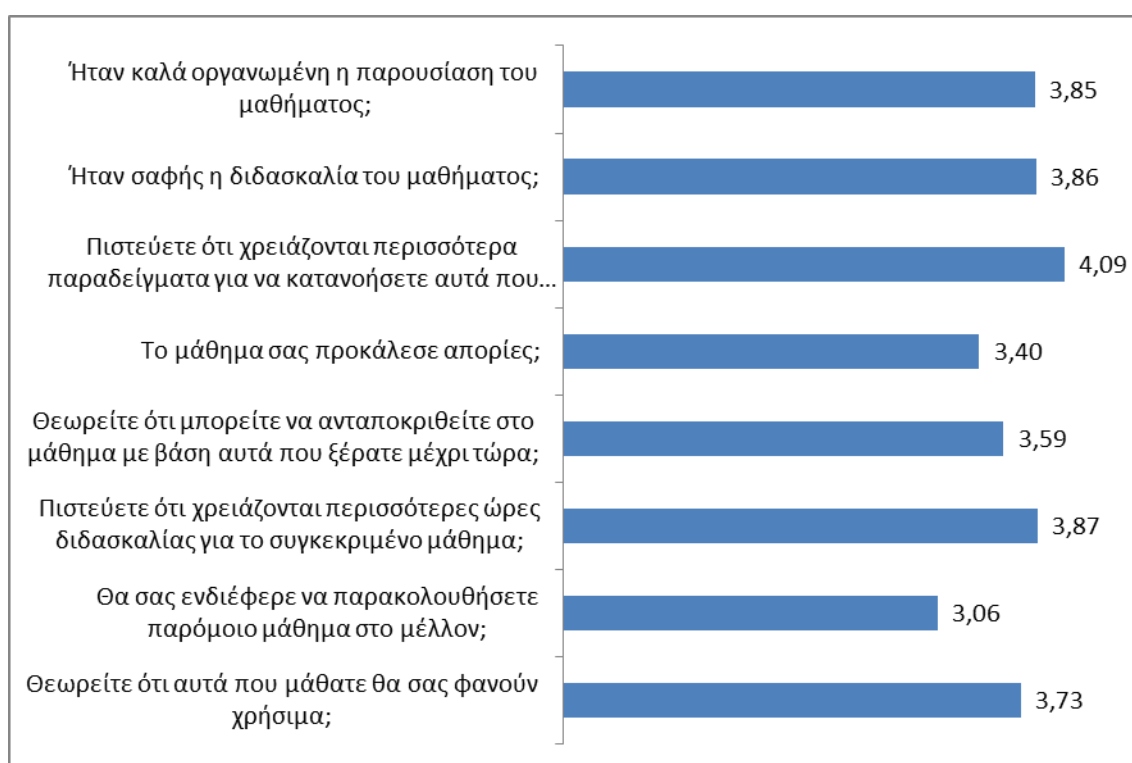
Οι περισσότεροι μαθητές φάνηκαν θετικοί στην ενδεχόμενη παρακολούθηση ενός παρόμοιου μαθήματος μελλοντικά.

Γράφημα 9: Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;



Μόνο το 7,92% του δείγματος έκρινε ότι όσα διδάχθηκε δε θα του φανούν χρήσιμα μελλοντικά ενώ οι υπόλοιποι μαθητές θεώρησαν αρκετά έως και πάρα πολύ χρήσιμα όλα όσα έμαθαν.

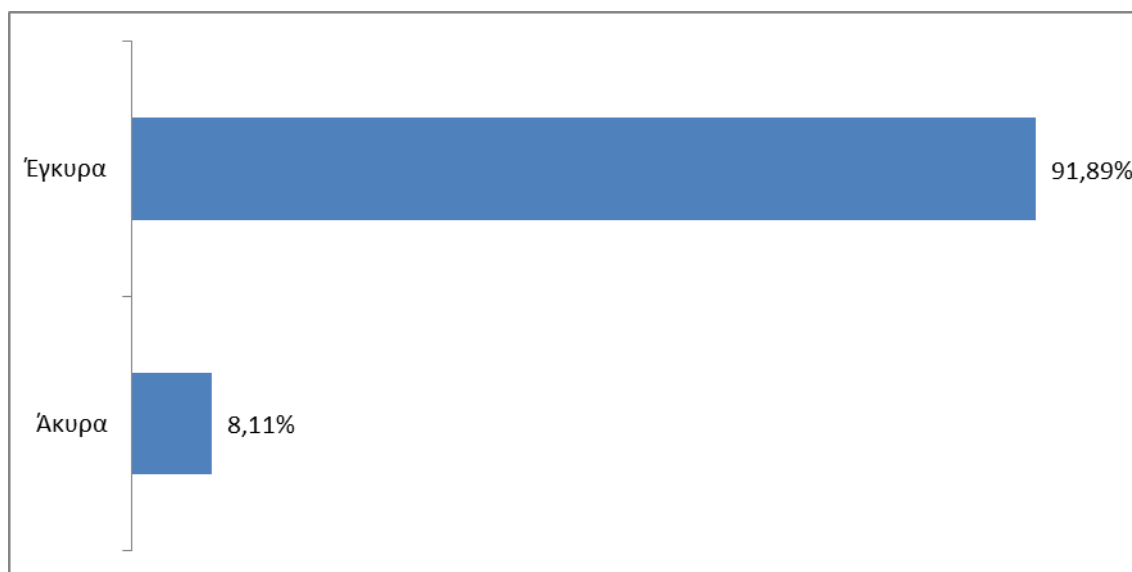
Γράφημα 10: Μέσος όρος κάθε ερώτησης στα Κυκλώματα



Η γενική βαθμολογία για το συγκεκριμένο μάθημα είναι 3,68 στα 5 (βλ Παράρτημα 5.3.1).

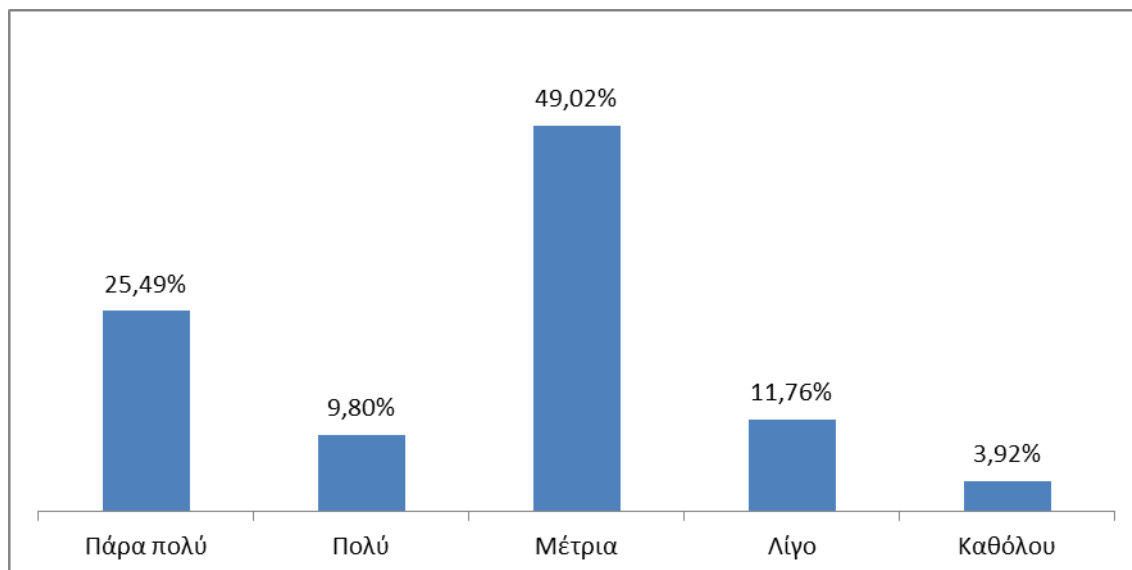
3.1.2 Αποτελέσματα Προγραμματισμός ανά ερώτημα

Γράφημα 1: Ποσοστό έγκυρων και άκυρων ερωτηματολογίων.



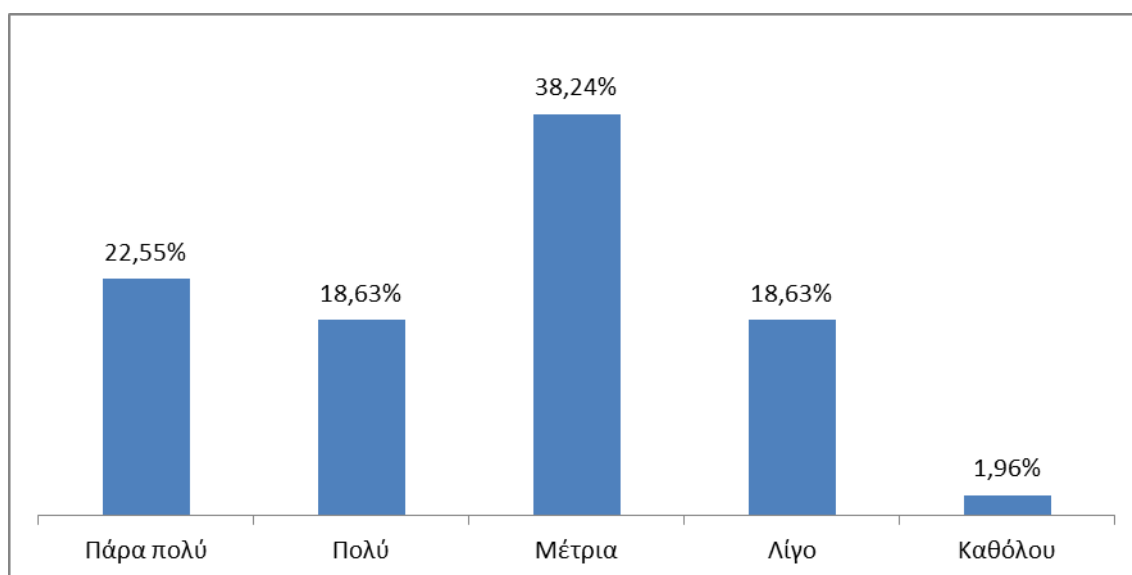
Στο σύνολο τα ερωτηματολόγια μας ήταν 111. Τα έγκυρα ερωτηματολόγια ήταν 102 και τα άκυρα 9.

Γράφημα 2: Ήταν καλά οργανωμένη η παρουσίαση του μαθήματος;



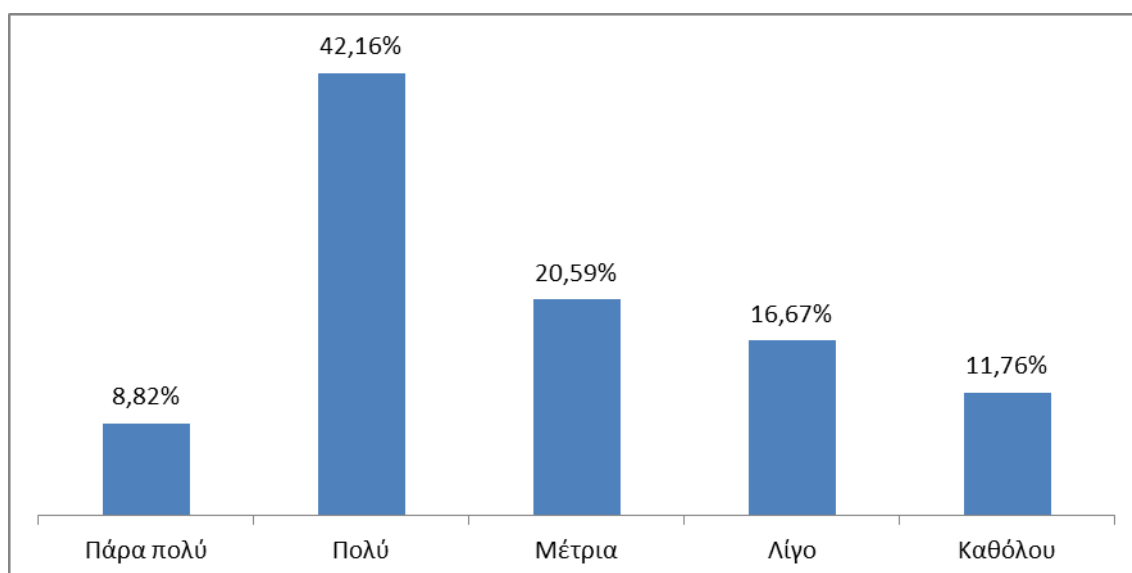
Στην πλειοψηφία του το δείγμα μας (84,31%) αποφάνθηκε ότι η παρουσίαση του μαθήματος ήταν από μέτρια έως και πάρα πολύ καλά οργανωμένη.

Γράφημα 3: Ήταν σαφής η διδασκαλία του μαθήματος;



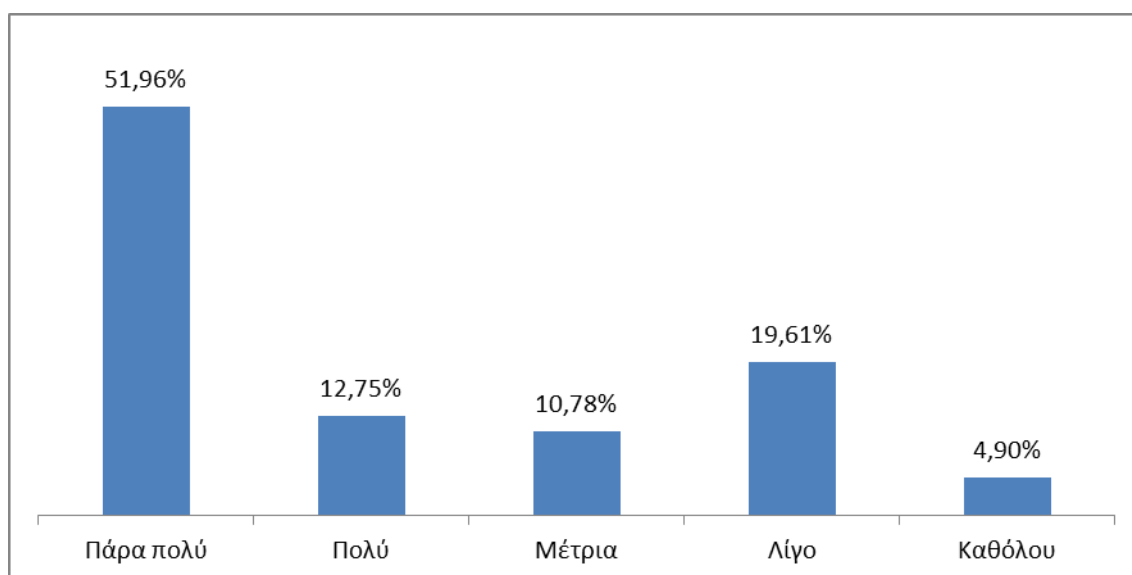
Μόνο το 20,59% του δείγματός μας δεν έμεινε ικανοποιημένο από τη διδασκαλία του μαθήματος ενώ το 38,24% θεώρησε αρκετά σαφή τη διδασκαλία και το 41,18% έμεινε απόλυτα ικανοποιημένο από όσα διδάχθηκε.

Γράφημα 4: Πιστεύετε ότι χρειάζονται περισσότερα παραδείγματα για να κατανοήσετε αυτά που διδαχθήκατε;



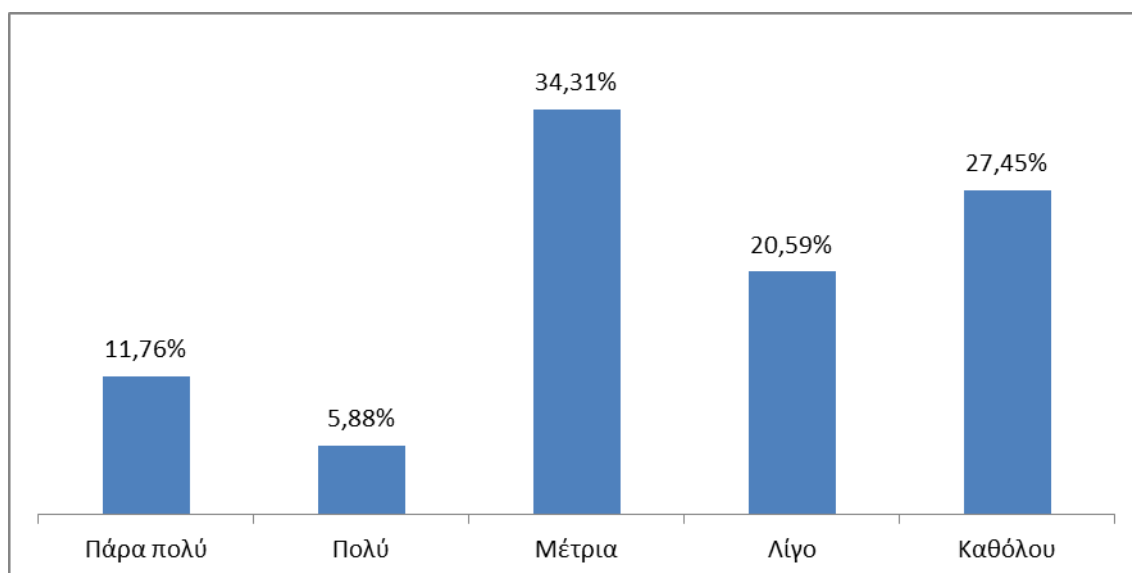
Σύμφωνα με το γράφημα οι περισσότεροι μαθητές φαίνεται ότι θα επιθυμούσαν να ακούσουν περισσότερα παραδείγματα για να κατανοήσουν καλύτερα τη διδασκαλία του συγκεκριμένου μαθήματος και μόνο ένα μικρό ποσοστό των μαθητών (11,76%) θεώρησε αρκετά τα παραδείγματα που δόθηκαν.

Γράφημα 5: Το μάθημα σας προκάλεσε απορίες;



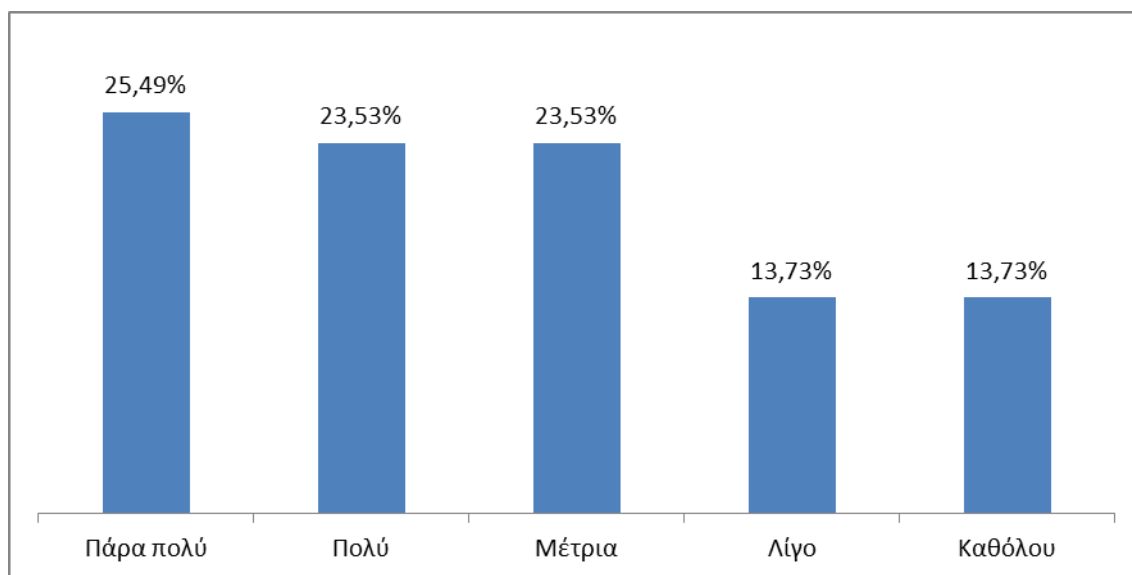
Περισσότερο από το 50% του δείγματός μας αποφάνθηκε ότι μετά το τέλος του μαθήματος είχε πάρα πολλές απορίες ενώ μόνο το 4,90% δεν είχε καθόλου απορίες.

Γράφημα 6: Θεωρείτε ότι μπορείτε να ανταποκριθείτε στο μάθημα με βάση αυτά που ξέρατε μέχρι τώρα;



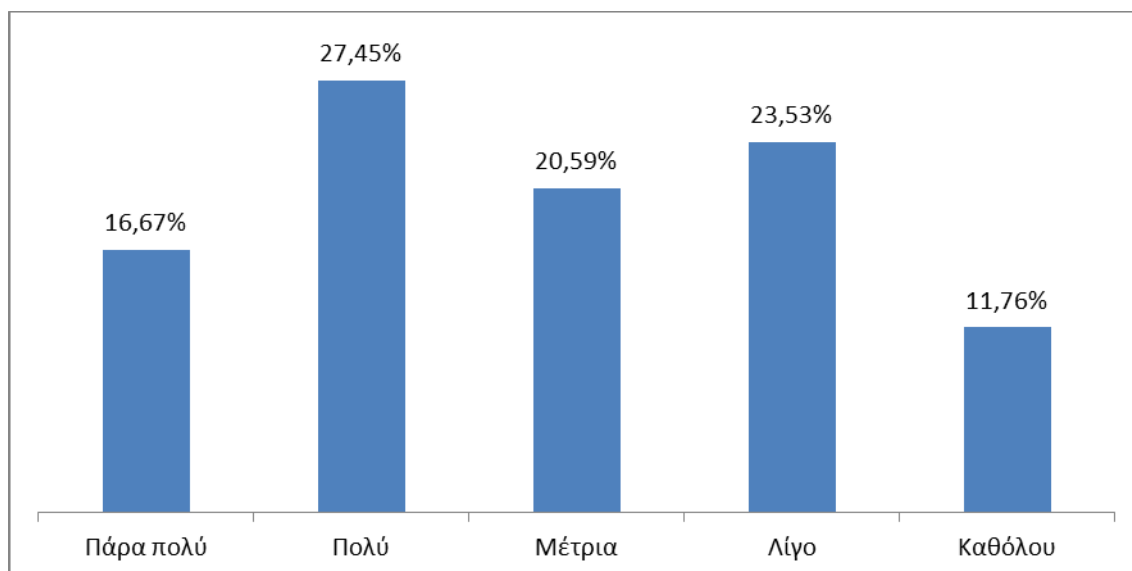
Το γράφημα δείχνει ότι οι περισσότεροι μαθητές είχαν περιορισμένες έως τώρα γνώσεις για τον προγραμματισμό και επομένως δε μπορούσαν να ανταποκριθούν εύκολα στο μάθημα ενώ μόνο ένα μικρό ποσοστό (17,64%) γνώριζε πολύ έως και πάρα πολύ καλά το θέμα.

Γράφημα 7: Πιστεύετε ότι χρειάζονται περισσότερες ώρες διδασκαλίας για το συγκεκριμένο μάθημα;



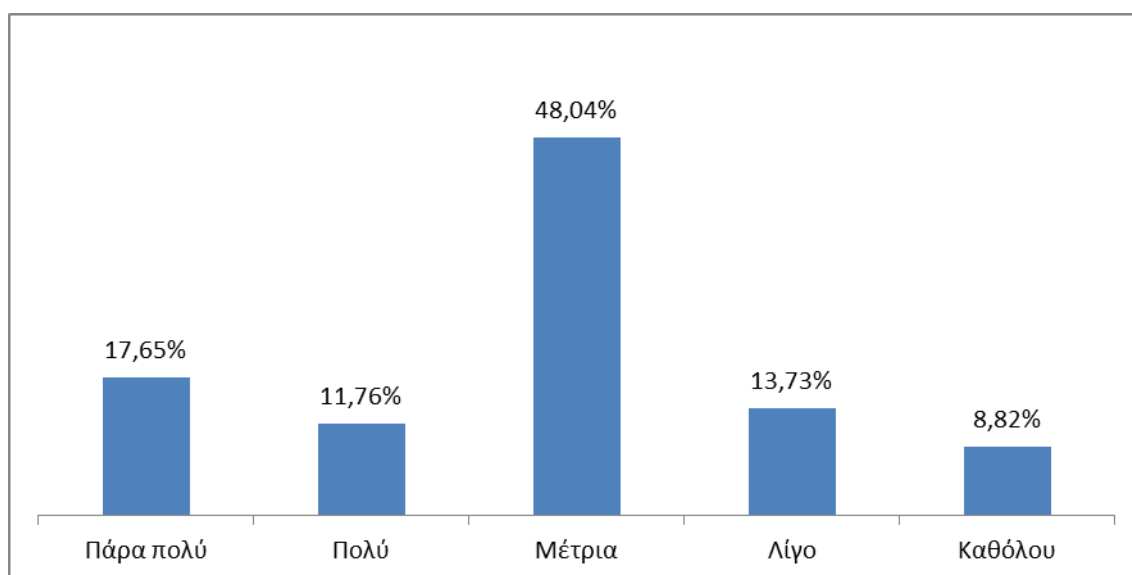
Οι περισσότεροι μαθητές απάντησαν ότι χρειάζονταν πολύ περισσότερες ώρες διδασκαλίας για να μπορέσουν να κατανοήσουν καλύτερα το μάθημα ενώ μόνο ένα 13,73% του δείγματος ήταν απόλυτα καλυμμένο και ένα άλλο 13,73% απάντησε ότι χρειάζονταν λίγες μόνο ακόμα ώρες.

Γράφημα 8: Θα σας ενδιέφερε να παρακολουθήσετε παρόμοιο μάθημα στο μέλλον;



Το 35,29% του δείγματος απάντησε ότι θα ενδιαφερόταν πολύ λίγο ή και καθόλου για να παρακολουθήσει κάποιο παρόμοιο μάθημα μελλοντικά, το 44,12% θα επιθυμούσε να ασχοληθεί ξανά με το συγκεκριμένο θέμα ενώ το 20,59% ήταν αναποφάσιστο για το αν θα ήθελε να ξανά ασχοληθεί με κάτι παρόμοιο.

Γράφημα 9: Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;



Η πλειοψηφία του δείγματος (48,04%) θεώρησε αρκετά σημαντικά όλα όσα διδάχθηκε για το μέλλον.

Γράφημα 10: Μέσος όρος κάθε ερώτησης στον Προγραμματισμό



Η γενική βαθμολογία για το συγκεκριμένο μάθημα είναι 2,91 στα 5 (βλ Παράρτημα 5.3.1).

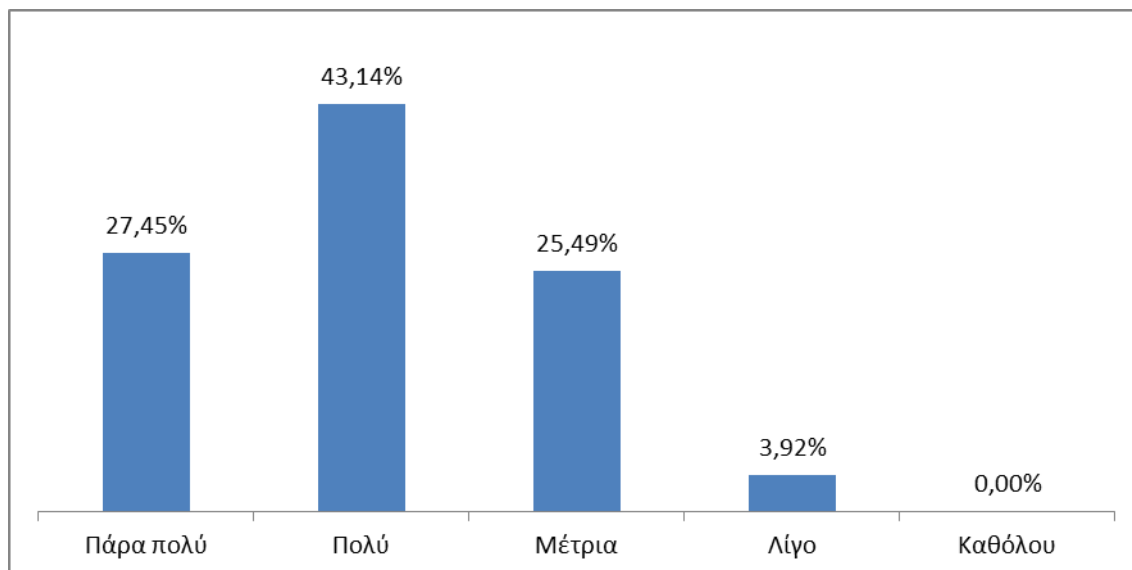
3.1.3 Αποτελέσματα Arduino ανά ερώτημα

Γράφημα 1: Ποσοστό έγκυρων και άκυρων ερωτηματολογίων.



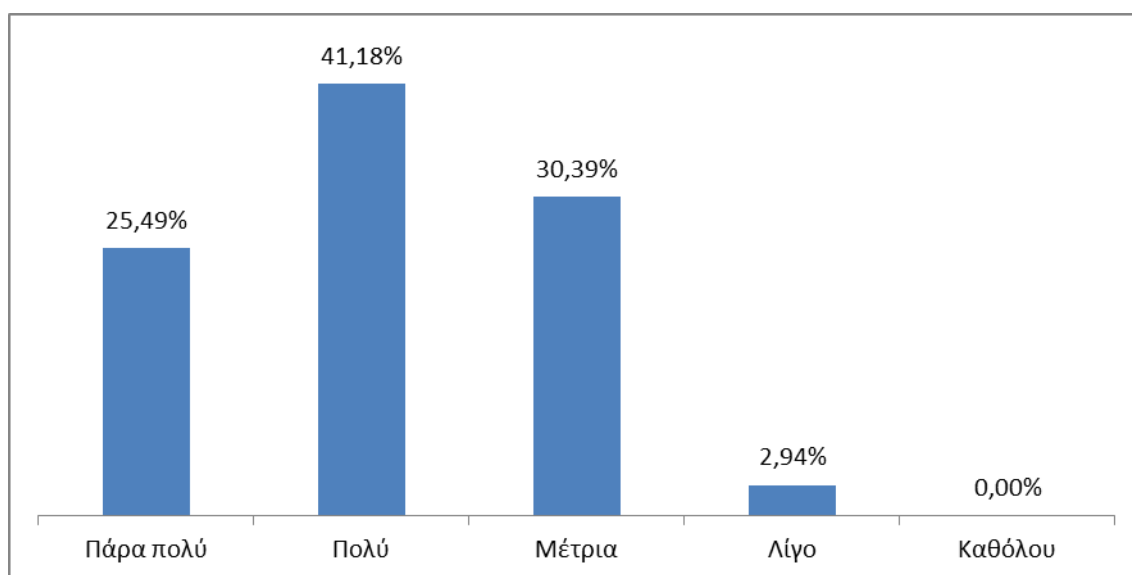
Στο σύνολο τα ερωτηματολόγια μας ήταν 108. Τα έγκυρα ερωτηματολόγια ήταν 102 και τα άκυρα 6.

Γράφημα 2: Ήταν καλά οργανωμένη η παρουσίαση του μαθήματος;



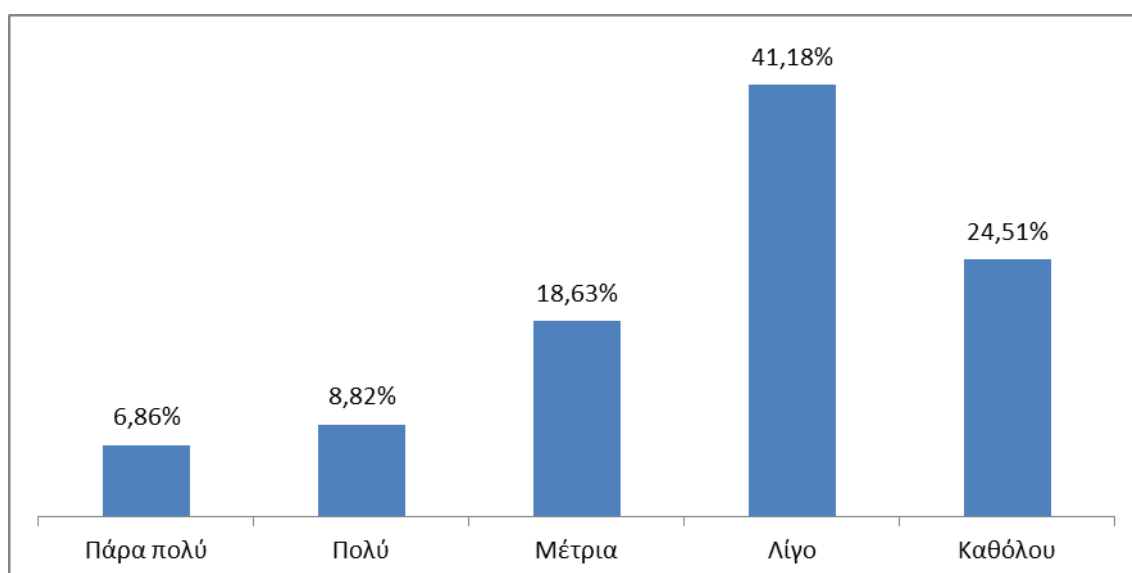
Μόνο το 3,92% φάνηκε δυσαρεστημένο από την παρουσίαση του μαθήματος.

Γράφημα 3: Ήταν σαφής η διδασκαλία του μαθήματος;



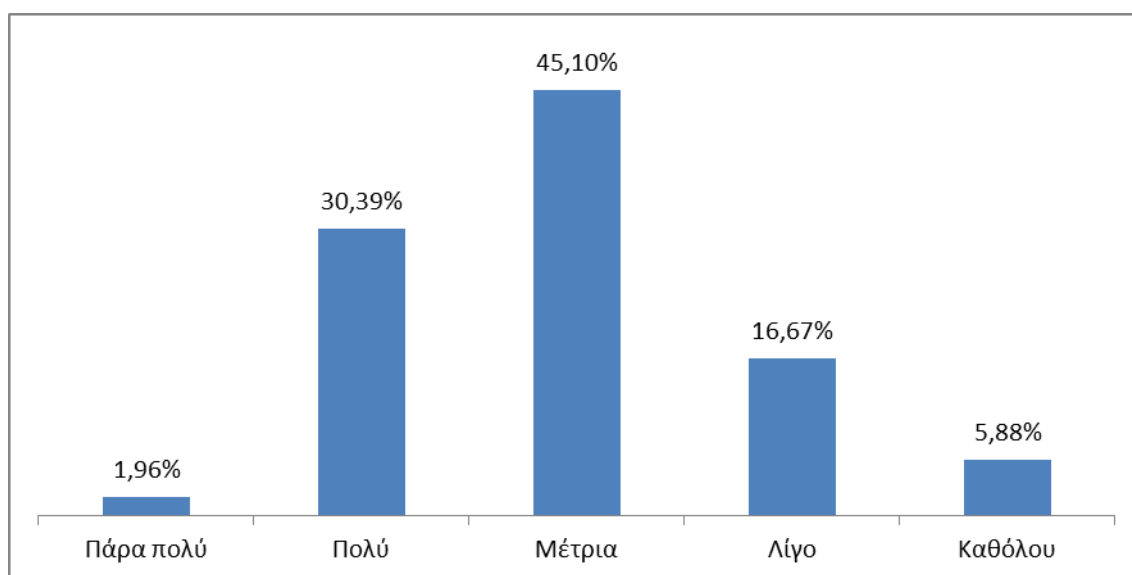
Μόνο το 2,94% δε θεώρησε σαφή τη διδασκαλία του μαθήματος.

Γράφημα 4: Πιστεύετε ότι χρειάζονται περισσότερα παραδείγματα για να κατανοήσετε αυτά που διδαχθήκατε;



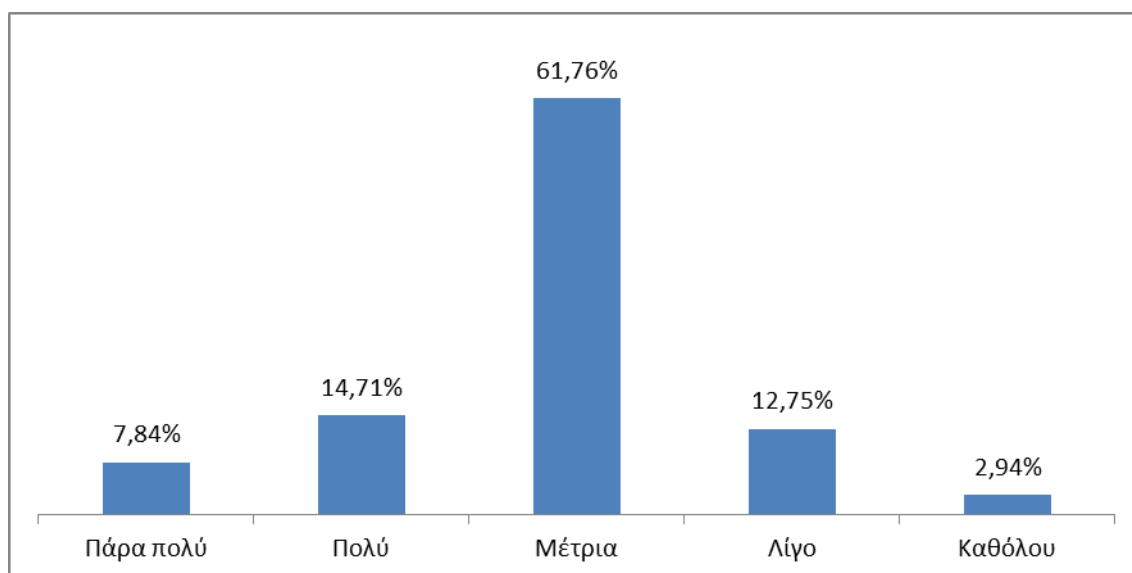
Οι μαθητές φαίνεται να κατανόησαν εύκολα όσα διδάχθηκαν καθώς μόνο ένα 15,68% απάντησε ότι χρειαζόταν πολλά ακόμα παραδείγματα για να κατανοήσει επαρκώς τη διδασκαλία.

Γράφημα 5: Το μάθημα σας προκάλεσε απορίες;



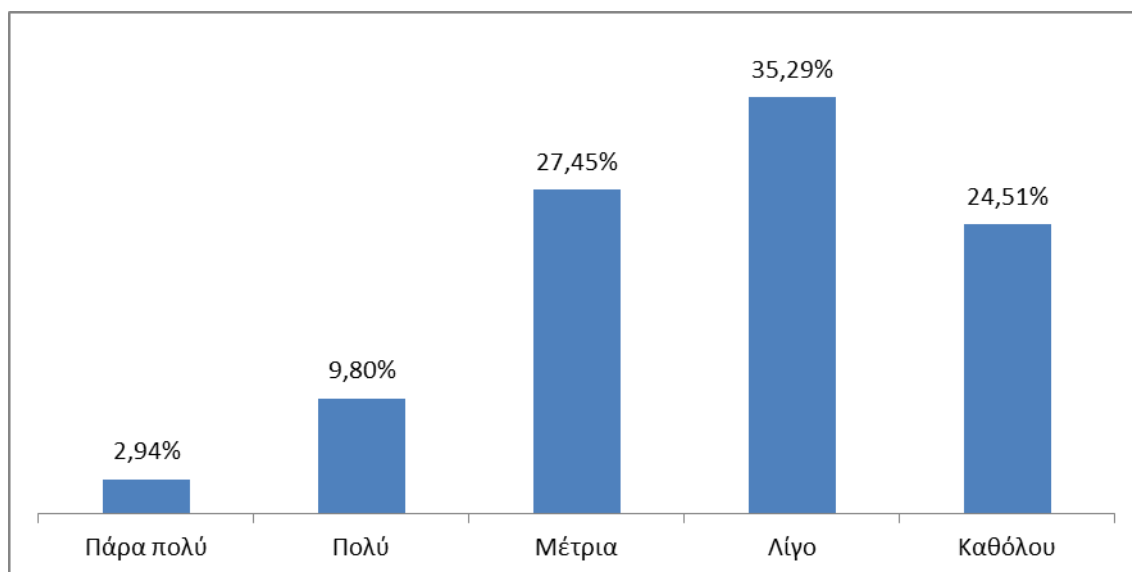
Σύμφωνα με το γράφημα το μάθημα προκάλεσε απορίες στους μαθητές από μέτρια έως και πολύ.

Γράφημα 6: Θεωρείτε ότι μπορείτε να ανταποκριθείτε στο μάθημα με βάση αυτά που ξέρατε μέχρι τώρα;



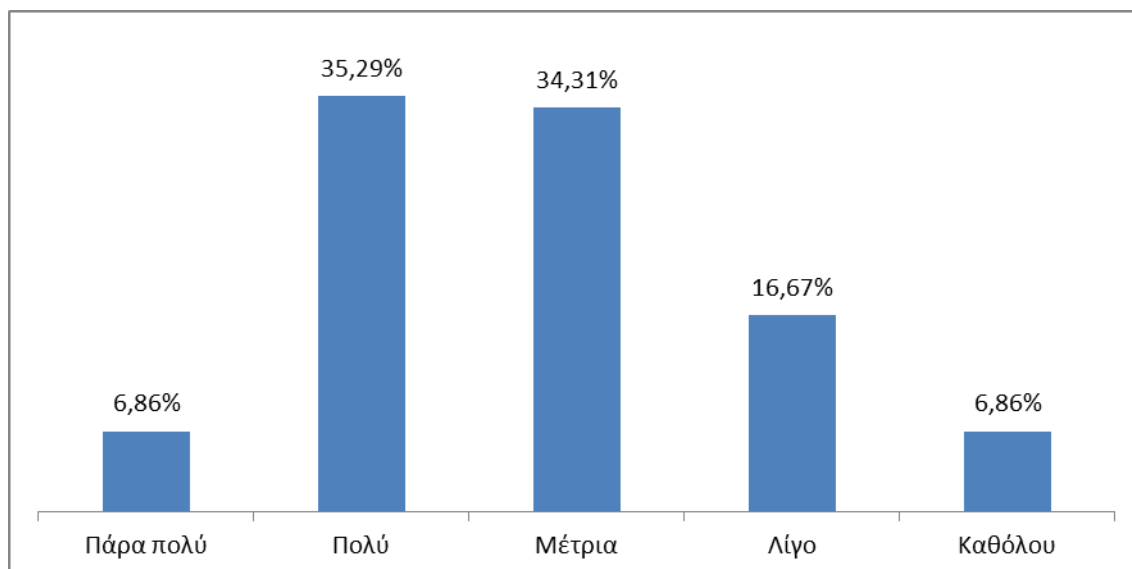
Το 61,76% του δείγματος απάντησε ότι είχε κάποιες γνώσεις σχετικές με το θέμα αλλά δεν ήταν αρκετές για να κατανοήσει πλήρως το μάθημα.

Γράφημα 7: Πιστεύετε ότι χρειάζονται περισσότερες ώρες διδασκαλίας για το συγκεκριμένο μάθημα;



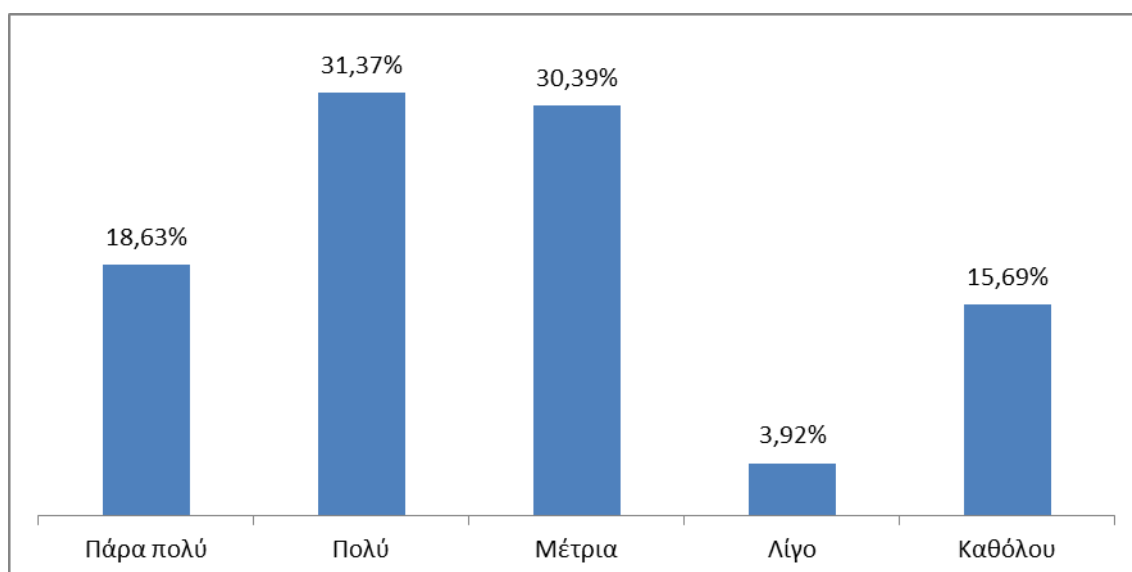
Το μάθημα φαίνεται ότι δε δυσκόλεψε τους μαθητές καθώς πολύ μικρό ποσοστό απάντησε ότι χρειαζόταν επιπλέον ώρες διδασκαλίας.

Γράφημα 8: Θα σας ενδιέφερε να παρακολουθήσετε παρόμοιο μάθημα στο μέλλον;



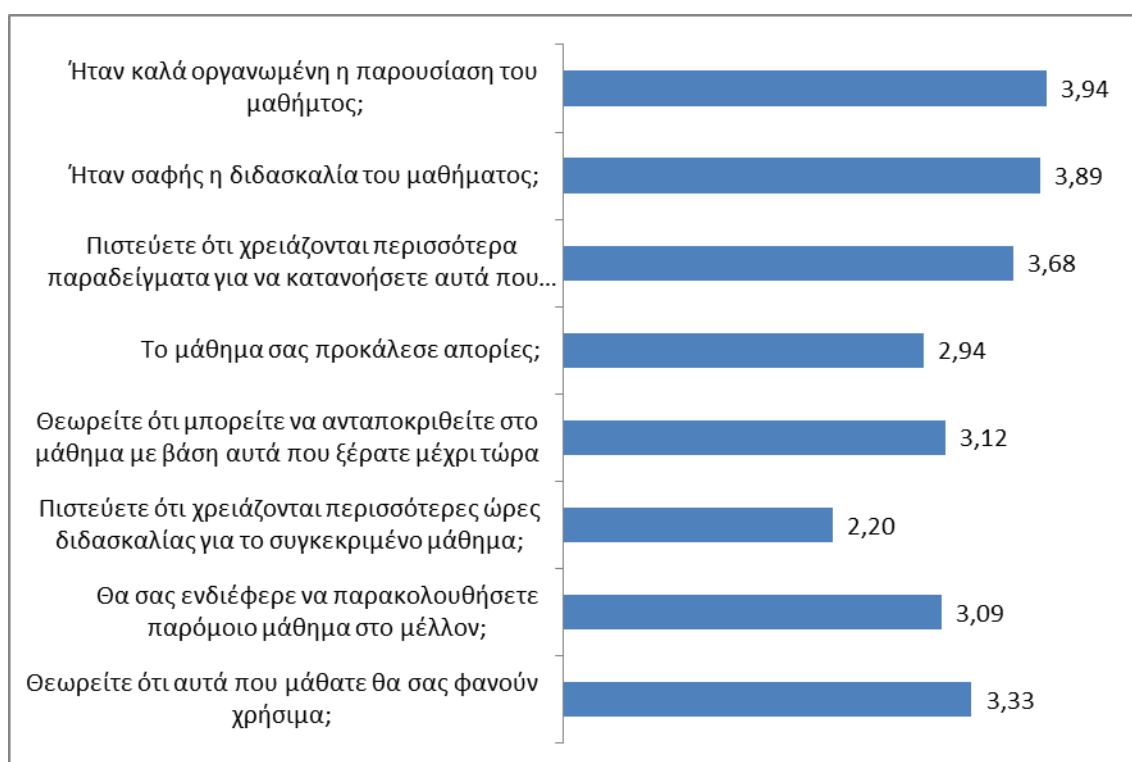
Το μεγαλύτερο ποσοστό των μαθητών έδειξε ενδιαφέρον για την ενδεχόμενη μελλοντική παρακολούθηση κάποιου παρόμοιου μαθήματος.

Γράφημα 9: Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;



Πολύ μικρό ποσοστό των μαθητών (19,61%) θεώρησε άχρηστα όσα διδάχθηκε.

Γράφημα 10: Μέσος όρος κάθε ερώτησης στο Arduino



Η γενική βαθμολογία για το συγκεκριμένο μάθημα είναι 3,27 στα 5 (βλ Παράρτημα 5.3.1).

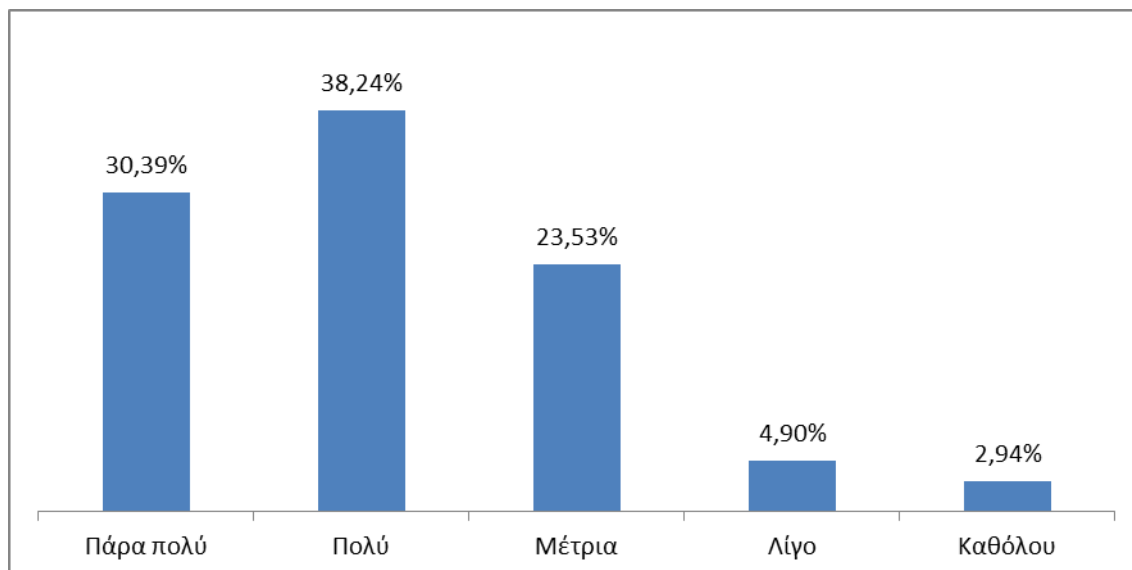
3.1.4 Αποτελέσματα 2^{ης} Εργασίας ανά ερώτημα

Γράφημα 1: Ποσοστό έγκυρων και άκυρων ερωτηματολογίων.



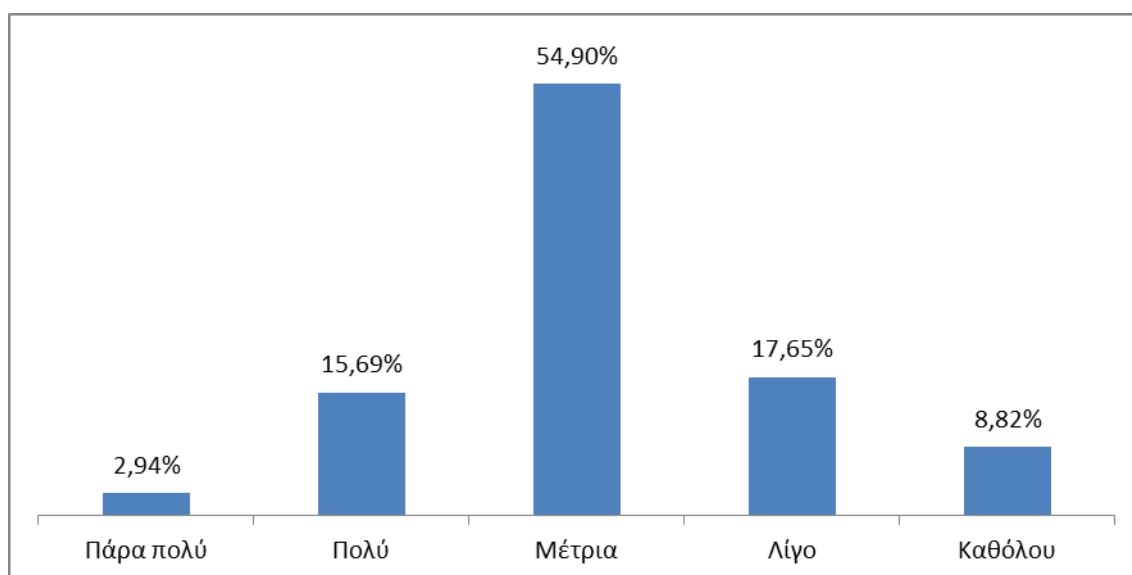
Στο σύνολο τα ερωτηματολόγια μας ήταν 109. Τα έγκυρα ερωτηματολόγια ήταν 102 και τα άκυρα 7.

Γράφημα 2: Ήταν σαφής η διδασκαλία του μαθήματος;



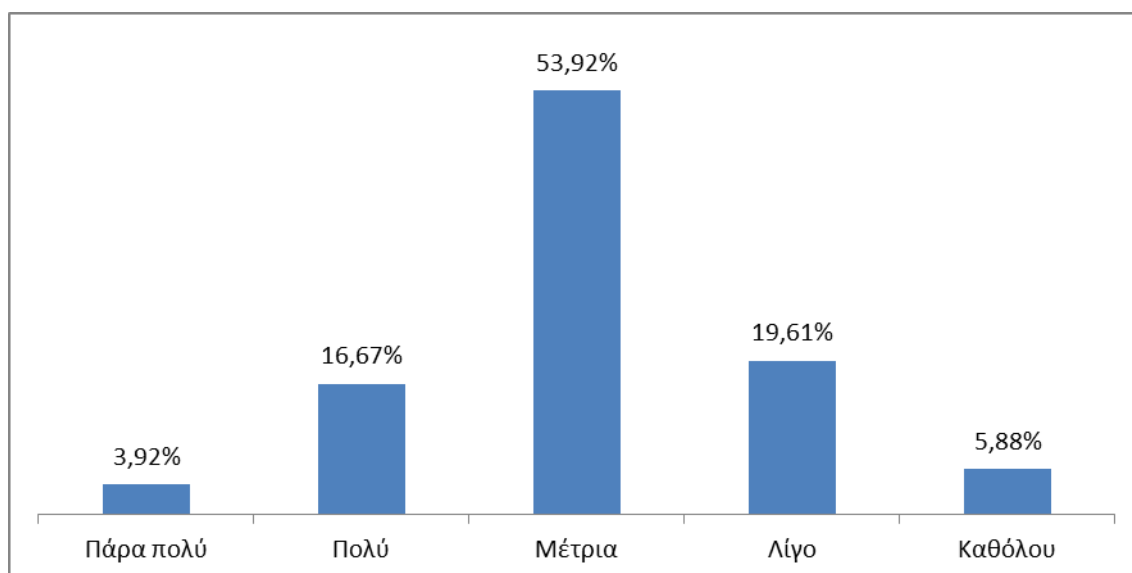
Η πλειοψηφία των μαθητών ήταν πολύ ικανοποιημένη από τη διδασκαλία του μαθήματος.

Γράφημα 3: Έχετε δει γύρω σας κάτι που να λειτουργεί με παρόμοιο τρόπο;



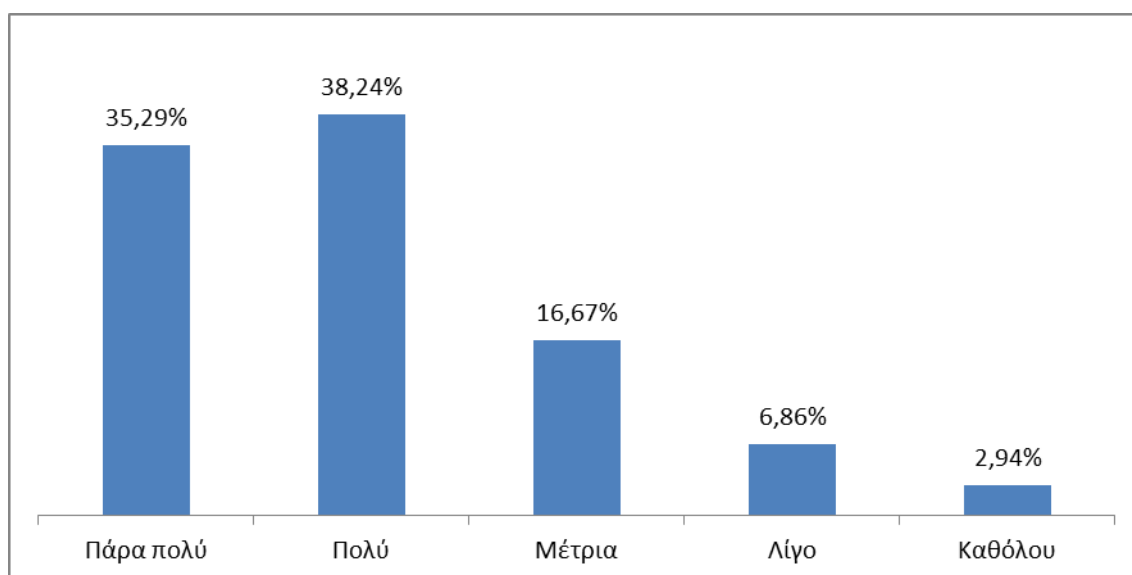
Ένα μεγαλύτερο ποσοστό από το 50% μπορούσε να συσχετίσει ό,τι είδε με πράγματα της καθημερινότητάς του.

Γράφημα 4: Με βάση αυτό που είδατε μπορείτε να σκεφτείτε κάτι παρόμοιο που θα μπορούσατε να φτιάξετε;



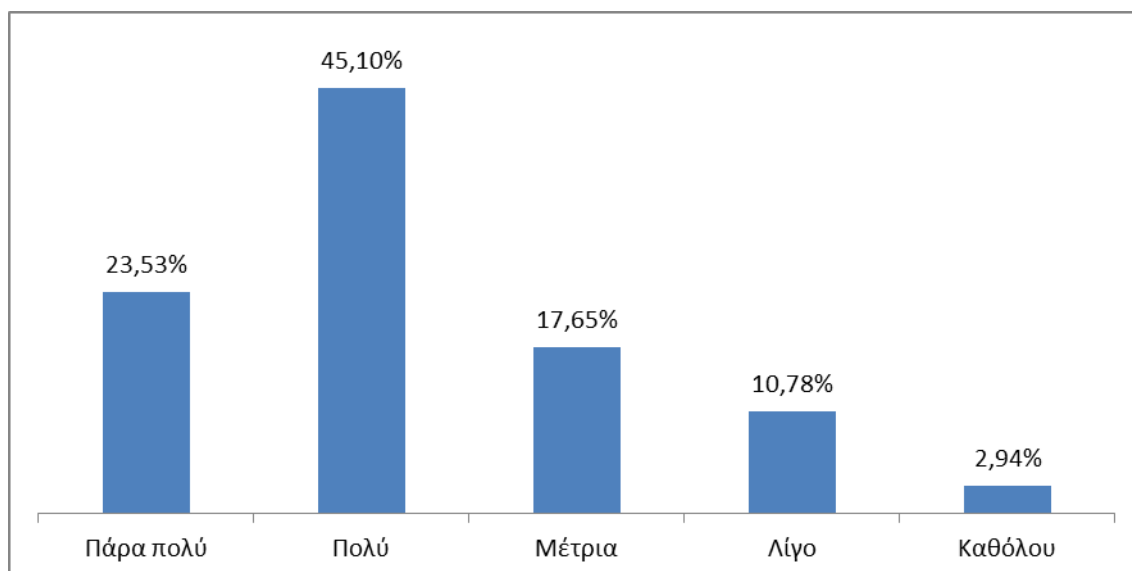
Οι περισσότεροι μαθητές φαίνεται να είχαν ιδέες για πράγματα που θα μπορούσαν να φτιάξουν και να λειτουργούν με παρόμοιο τρόπο με αυτό που είδαν.

Γράφημα 5: Καταλάβατε τη σύνδεση και τη λειτουργία των εξαρτημάτων;



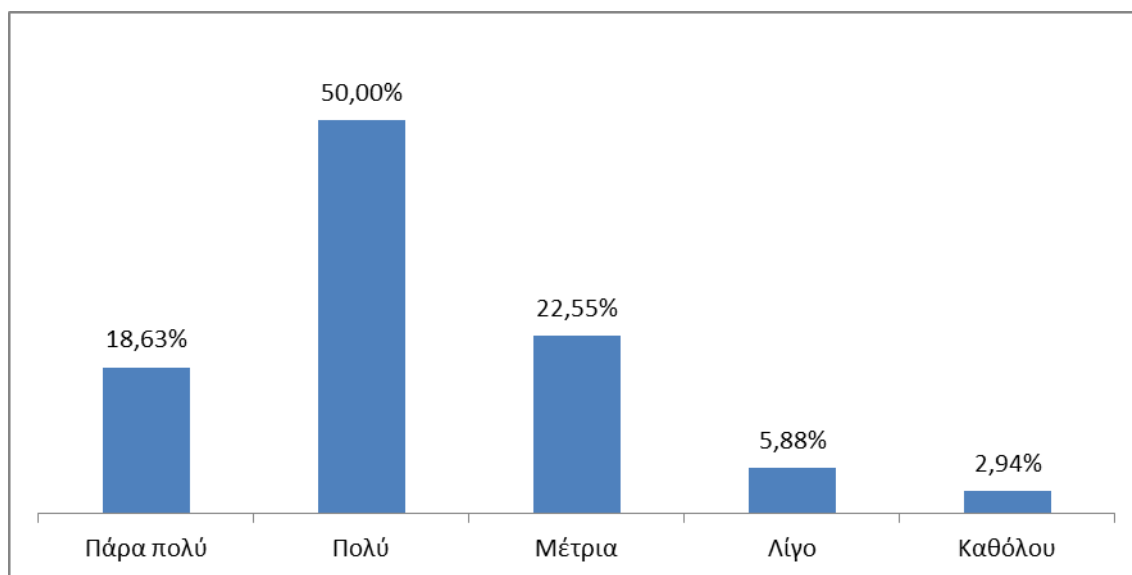
Ένα πολύ μικρό ποσοστό των μαθητών (9,8%) φαίνεται να δυσκολεύτηκε να κατανοήσει τη σύνδεση και τη λειτουργία των εξαρτημάτων που χρησιμοποιήθηκαν.

Γράφημα 6: Καταλάβατε πώς λειτουργεί ο κώδικας προγραμματισμού σε αυτήν την άσκηση;



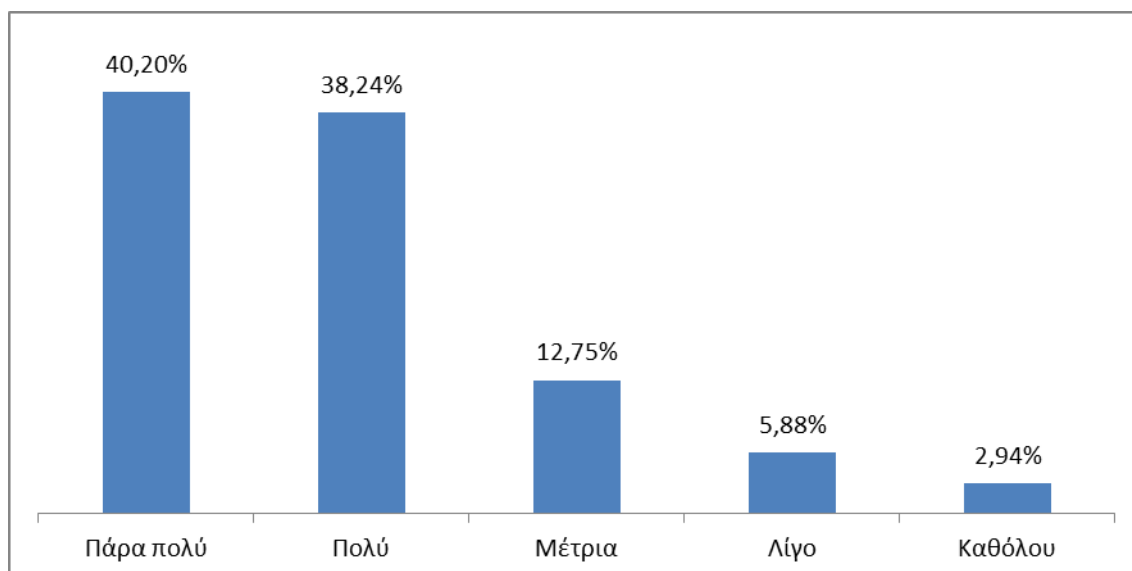
Μόνο το 13,72% δυσκολεύτηκε να κατανοήσει τον κώδικα προγραμματισμού της εργασίας ενώ το 68,63% κατάλαβε πάρα πολύ καλά.

Γράφημα 7: Πόσο εύκολο σας φαίνεται να φτιάξετε μόνοι σας κάτι ανάλογο ή να προσθέσετε κάτι πάνω σε αυτά;



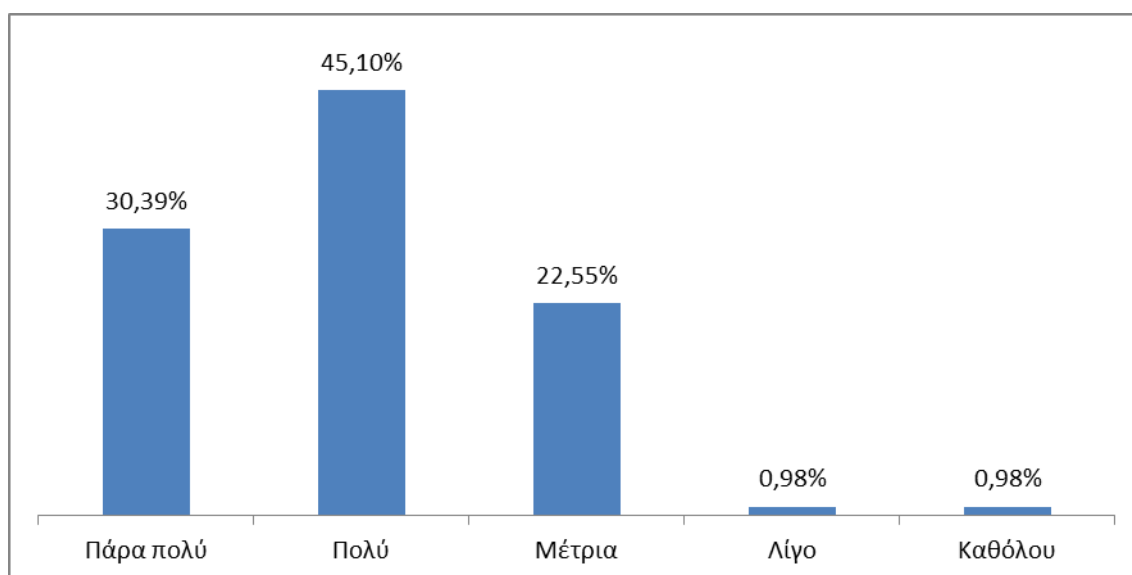
Μόνο το 8,82% απάντησε ότι δε θα του φαινόταν εύκολο να φτιάξει κάτι ανάλογο ή να εξελίξει περισσότερο τα ήδη υπάρχοντα.

Γράφημα 8: Θα σας ενδιέφερε να ασχοληθείτε εσείς οι ίδιοι και να φτιάξετε κάτι δικό σας;



Το μεγαλύτερο ποσοστό των μαθητών (78,44%) έδειξε ενδιαφέρον να ασχοληθεί με το συγκεκριμένο θέμα και να φτιάξει κάτι δικό του.

Γράφημα 9: Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;



Μόνο το 1,96% δε θεώρησε χρήσιμα όσα διδάχθηκε ενώ το μεγαλύτερο ποσοστό των μαθητών έδειξε πολύ μεγάλο ενδιαφέρον για όσα έμαθε.

Γράφημα 10: Μέσος όρος κάθε ερώτησης στη 2^η Εργασία



Η γενική βαθμολογία για το συγκεκριμένο μάθημα είναι 3,66 στα 5 (βλ Παράρτημα 5.3.2).

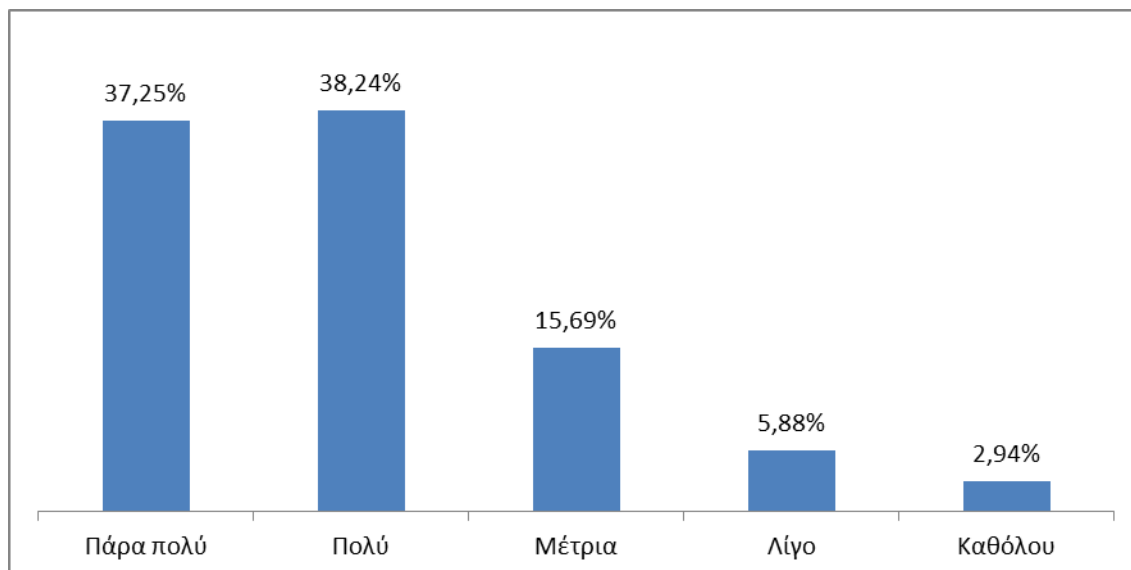
3.1.5 Αποτελέσματα 3^{ης} Εργασίας ανά ερώτημα

Γράφημα 1: Ποσοστό έγκυρων και άκυρων ερωτηματολογίων.



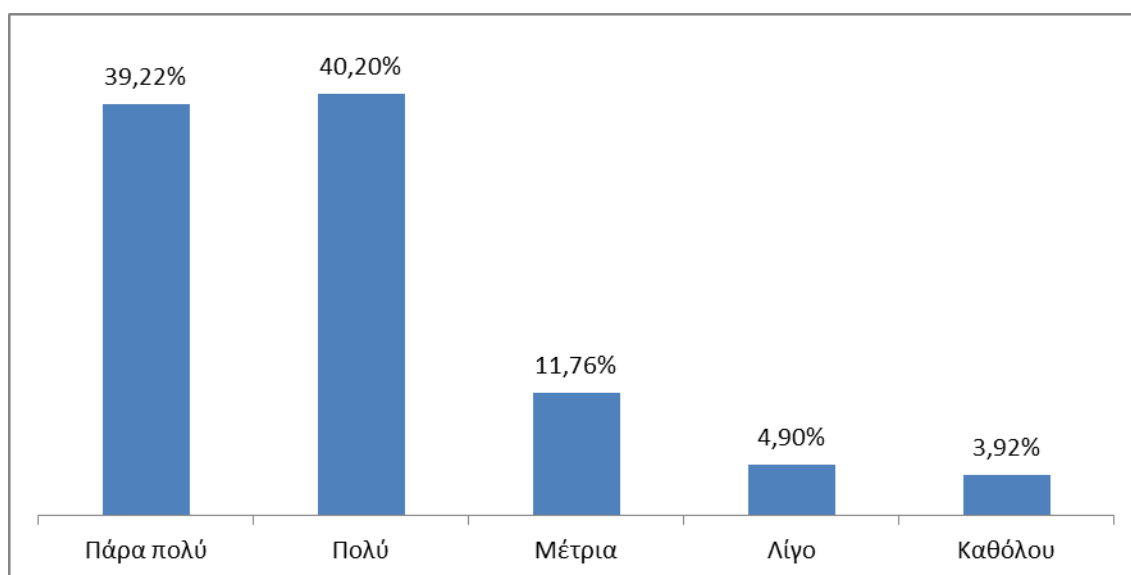
Στο σύνολο τα ερωτηματολόγια μας ήταν 108. Τα έγκυρα ερωτηματολόγια ήταν 102 και τα άκυρα 6.

Γράφημα 2: Ήταν σαφής η διδασκαλία του μαθήματος;



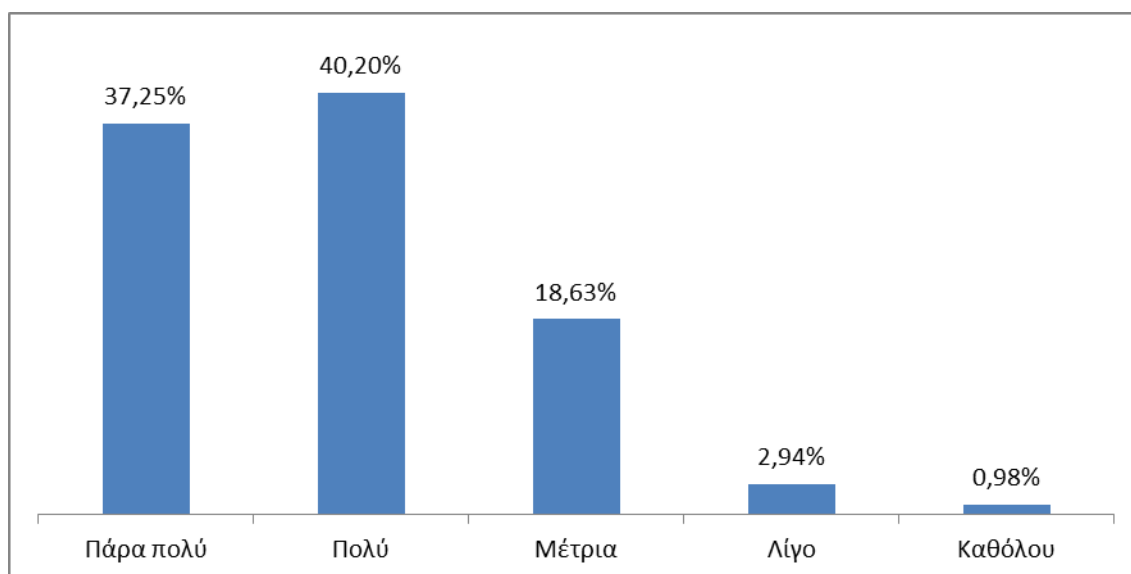
Οι περισσότεροι μαθητές σύμφωνα με το γράφημα φάνηκαν ικανοποιημένοι από τη διδασκαλία του μαθήματος.

Γράφημα 3: Έχετε δει γύρω σας κάτι που να λειτουργεί με παρόμοιο τρόπο;



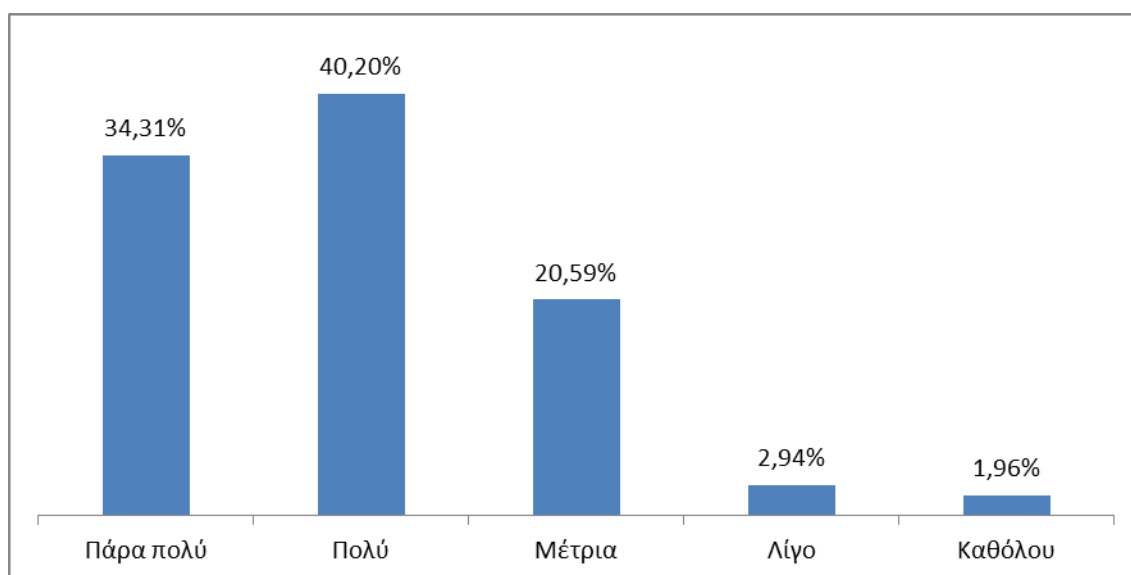
Στο 79,42% του δείγματος φάνηκε πολύ γνώριμο ότι είδε στη συγκεκριμένη εργασία.

Γράφημα 4: Με βάση αυτό που είδατε μπορείτε να σκεφτείτε κάτι παρόμοιο που θα μπορούσατε να φτιάξετε;



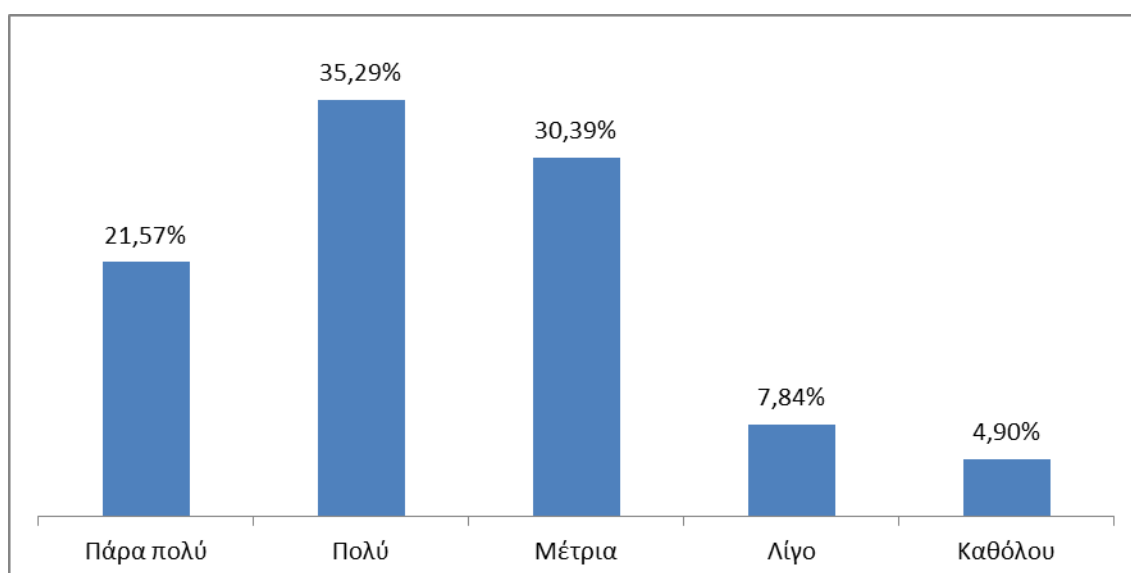
Το 3,92% του δείγματος δήλωσε ότι θα δυσκολευόταν να σκεφτεί κάτι που θα μπορούσε να φτιάξει και να λειτουργεί με παρόμοιο τρόπο.

Γράφημα 5: Καταλάβατε τη σύνδεση και τη λειτουργία των εξαρτημάτων;



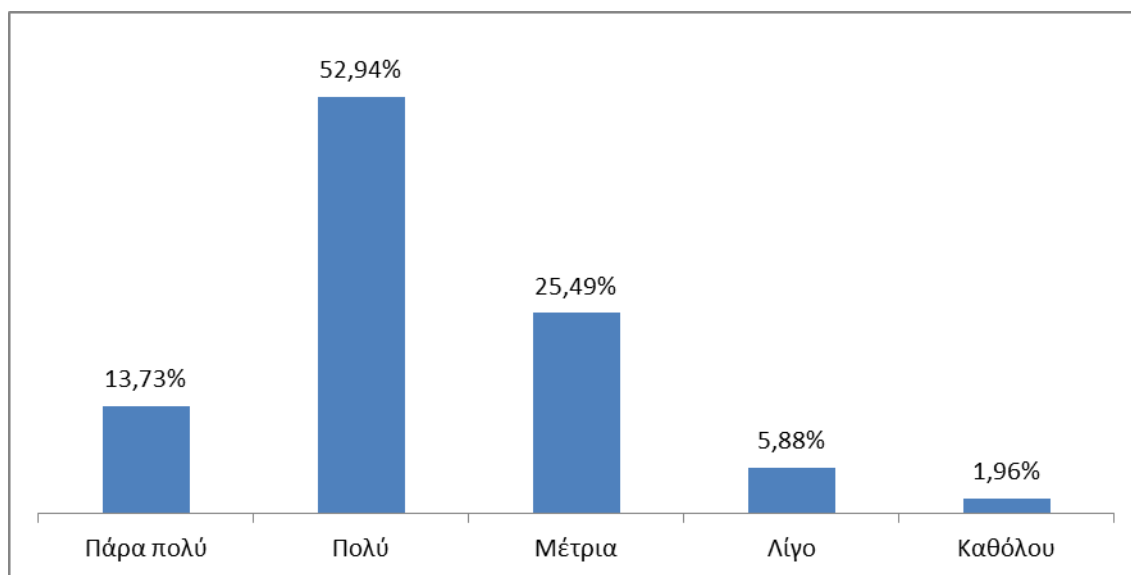
Ελάχιστοι μαθητές έδειξαν αδυναμία στην κατανόηση της σύνδεσης και της λειτουργίας των εξαρτημάτων της εργασίας.

Γράφημα 6: Καταλάβατε πώς λειτουργεί ο κώδικας προγραμματισμού σε αυτήν την άσκηση;



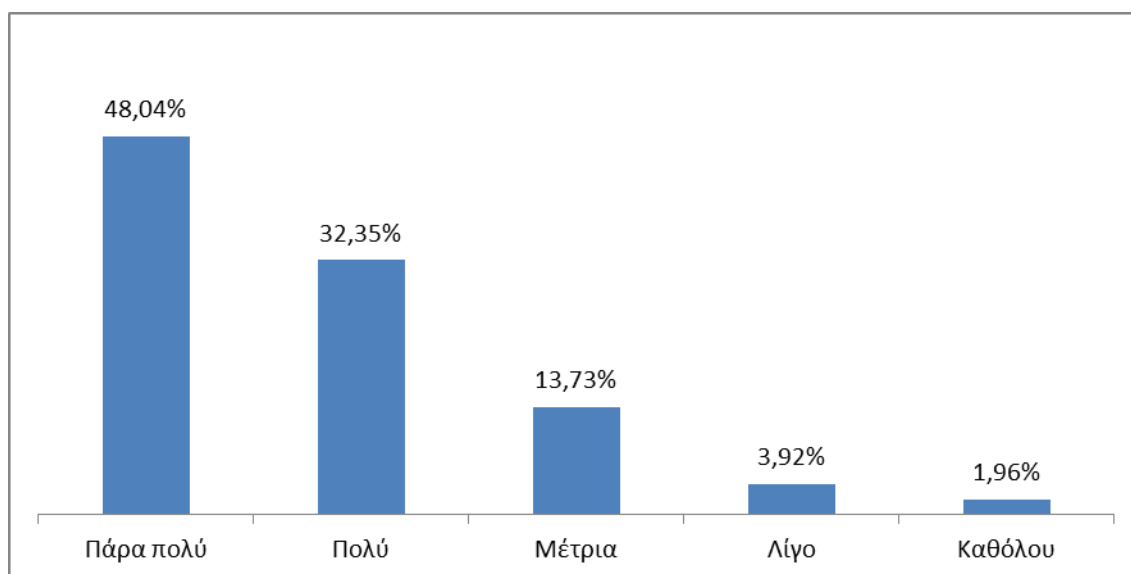
Το 56,86% του δείγματος κατανόησε πολύ έως και πάρα πολύ καλά τον κώδικα προγραμματισμού, το 30,39% κατανόησε αρκετά καλά και μόνο το 12,74% δυσκολεύτηκε να κατανοήσει.

Γράφημα 7: Πόσο εύκολο σας φαίνεται να φτιάξετε μόνοι σας κάτι ανάλογο ή να προσθέσετε κάτι πάνω σε αυτά;



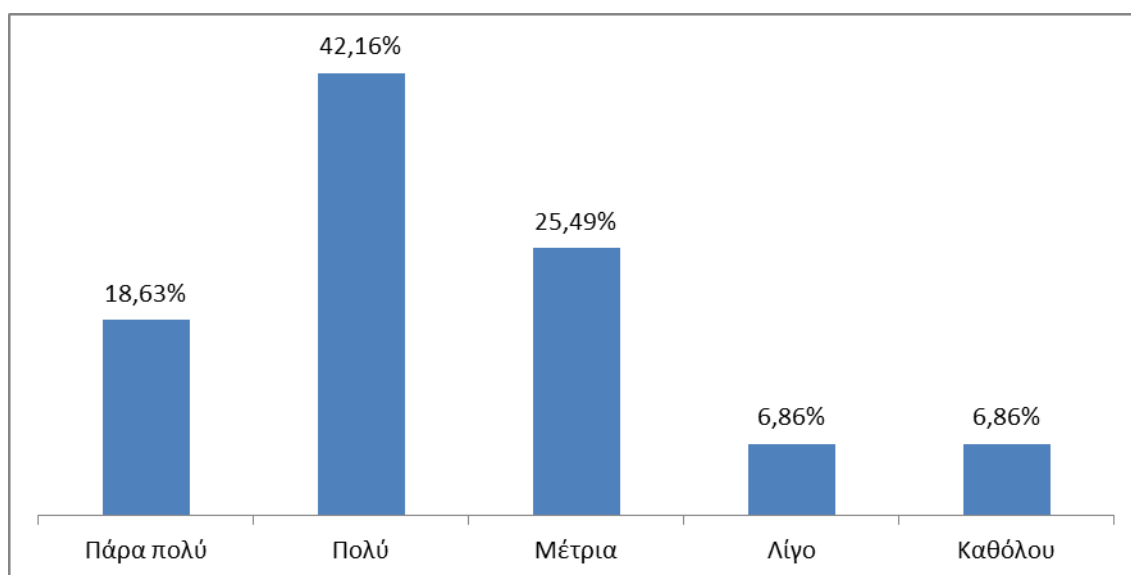
Για τους περισσότερους μαθητές δεν ήταν δύσκολο να φτιάξουν οι ίδιοι κάτι ανάλογο.

Γράφημα 8: Θα σας ενδιέφερε να ασχοληθείτε εσείς οι ίδιοι και να φτιάξετε κάτι δικό σας;



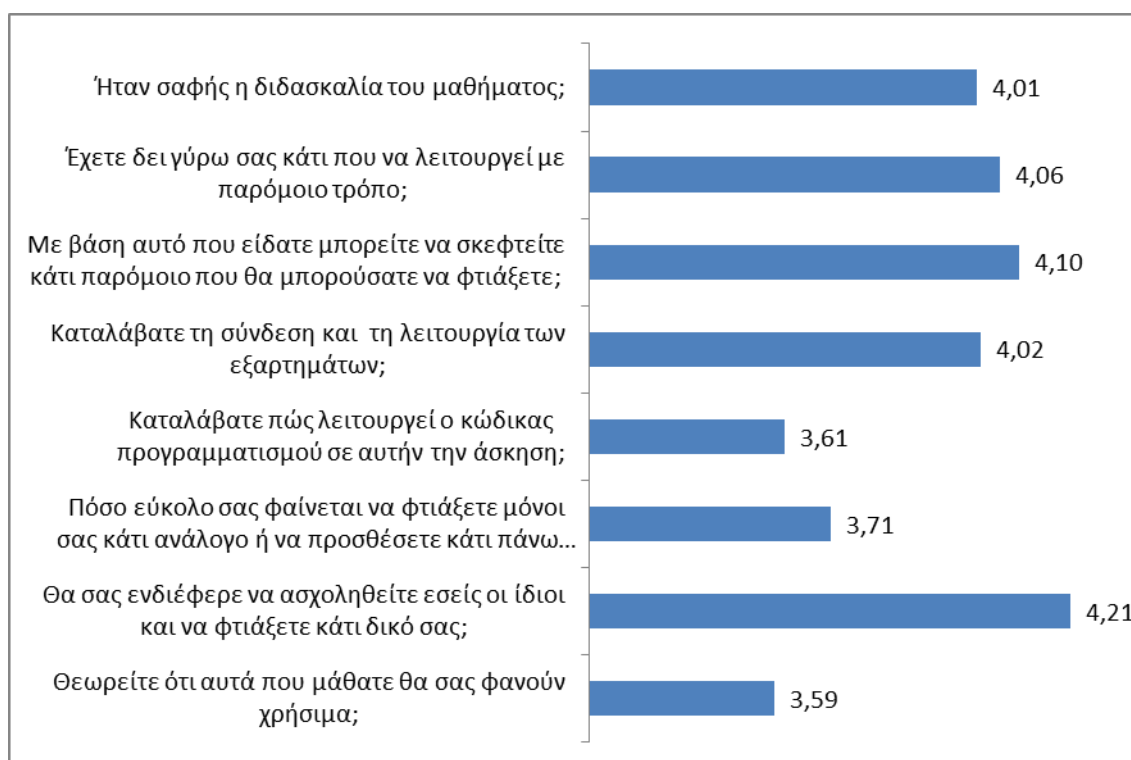
Το 48,04% των μαθητών έδειξε πάρα πολύ μεγάλο ενδιαφέρον να ασχοληθεί με κάτι παρόμοιο και να πραγματοποιήσει κάτι δικό του.

Γράφημα 9: Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;



Μόνο ένα μικρό ποσοστό των μαθητών (13,72%) φάνηκε να μην ενδιαφέρεται για αυτά που έμαθε και να αμφισβήτησε τη μελλοντική τους αξία.

Γράφημα 10: Μέσος όρος κάθε ερώτησης στην 3^η Εργασία



Η γενική βαθμολογία για το συγκεκριμένο μάθημα είναι 3,91 στα 5 (βλ Παράρτημα 5.3.2).

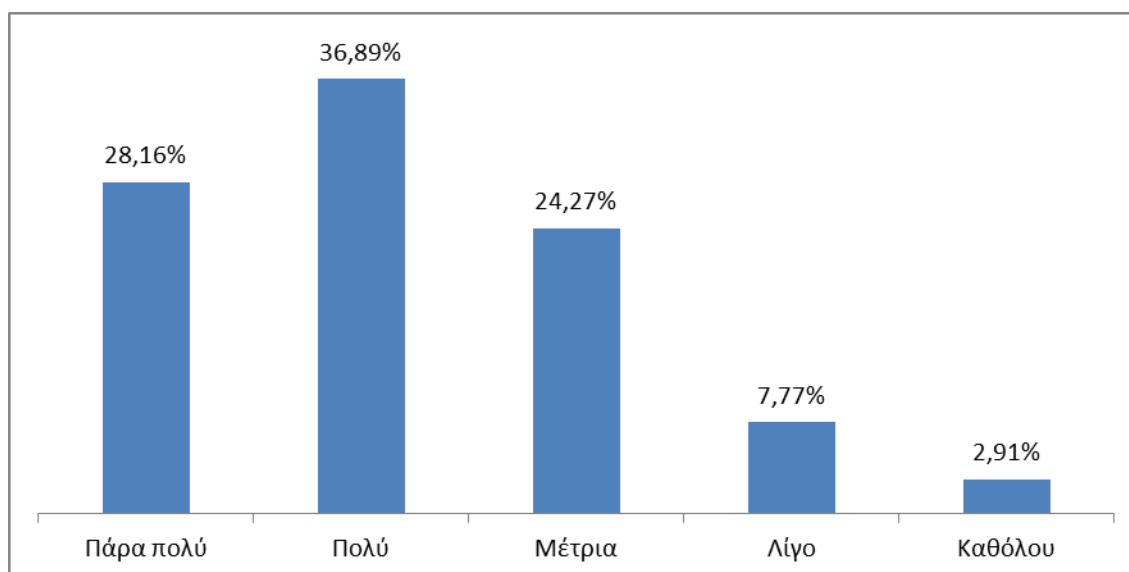
3.1.6 Αποτελέσματα 4^{ης} Εργασίας ανά ερώτημα

Γράφημα 1: Ποσοστό έγκυρων και άκυρων ερωτηματολογίων.



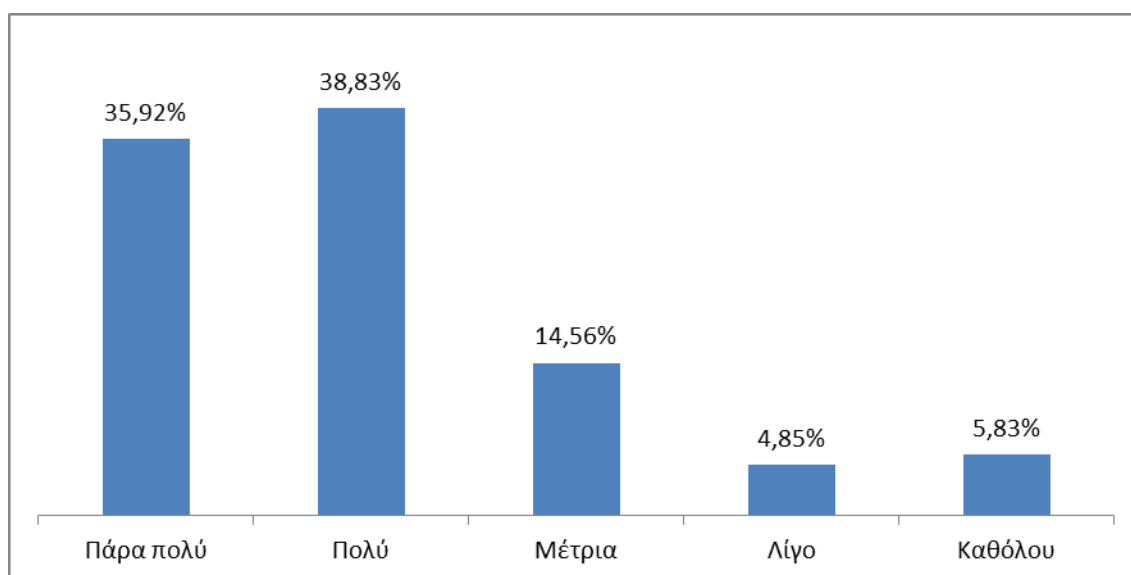
Στο σύνολο τα ερωτηματολόγια μας ήταν 106. Τα έγκυρα ερωτηματολόγια ήταν 103 και τα άκυρα 3.

Γράφημα 2: Ήταν σαφής η διδασκαλία του μαθήματος;



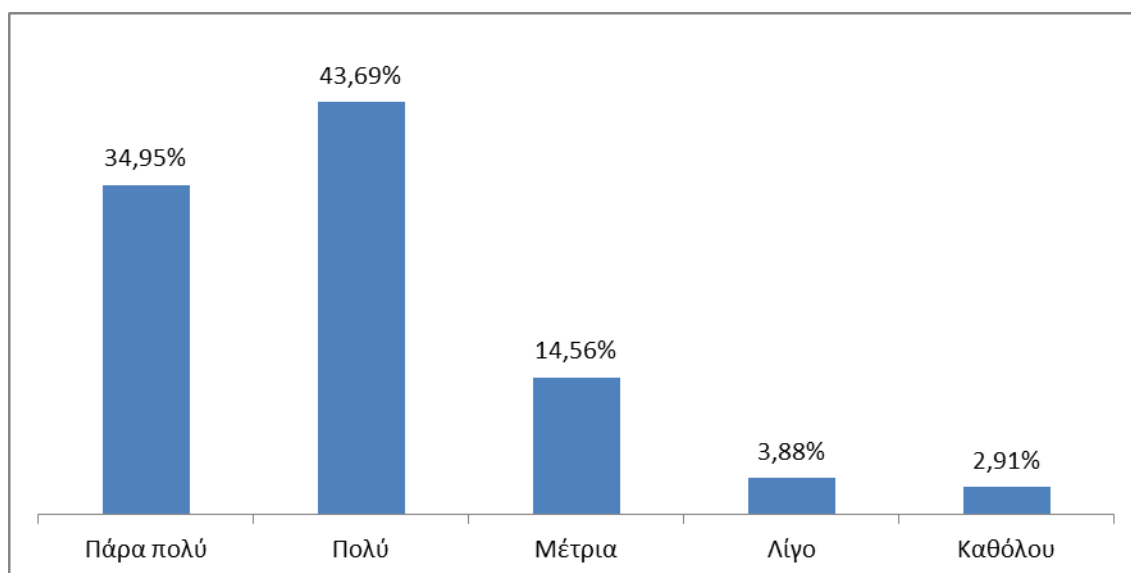
Για το 89,32% των μαθητών η διδασκαλία του μαθήματος ήταν από αρκετά έως και πάρα πολύ σαφής.

Γράφημα 3: Έχετε δει γύρω σας κάτι που να λειτουργεί με παρόμοιο τρόπο;



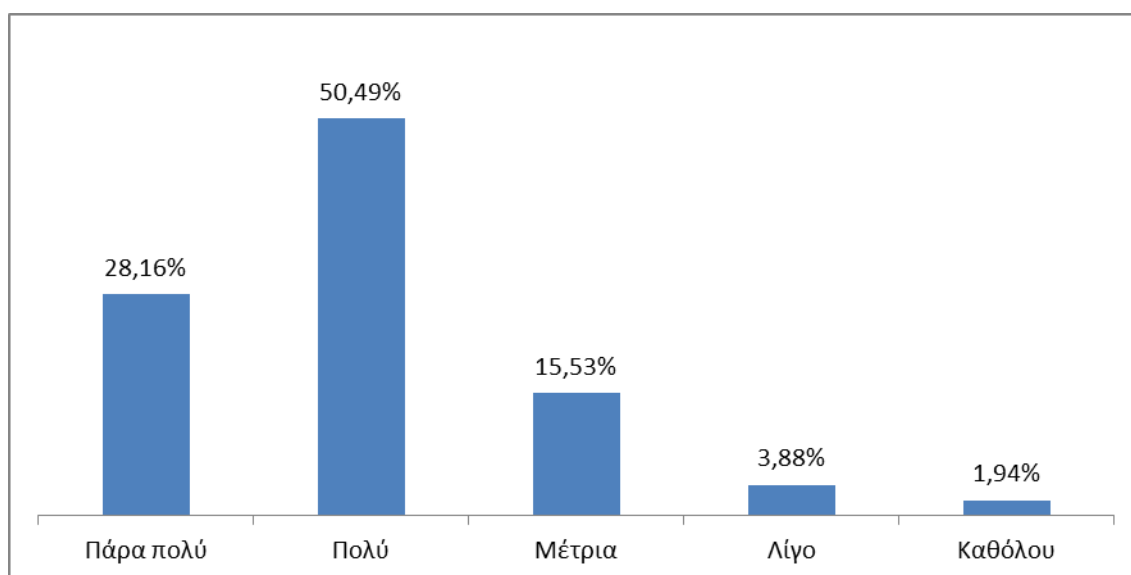
Μόνο το 10,68% δήλωσε ότι δεν έχει δει καθόλου ή πολύ ελάχιστα κάτι που να λειτουργεί με παρόμοιο τρόπο.

Γράφημα 4: Με βάση αυτό που είδατε μπορείτε να σκεφτείτε κάτι παρόμοιο που θα μπορούσατε να φτιάξετε;



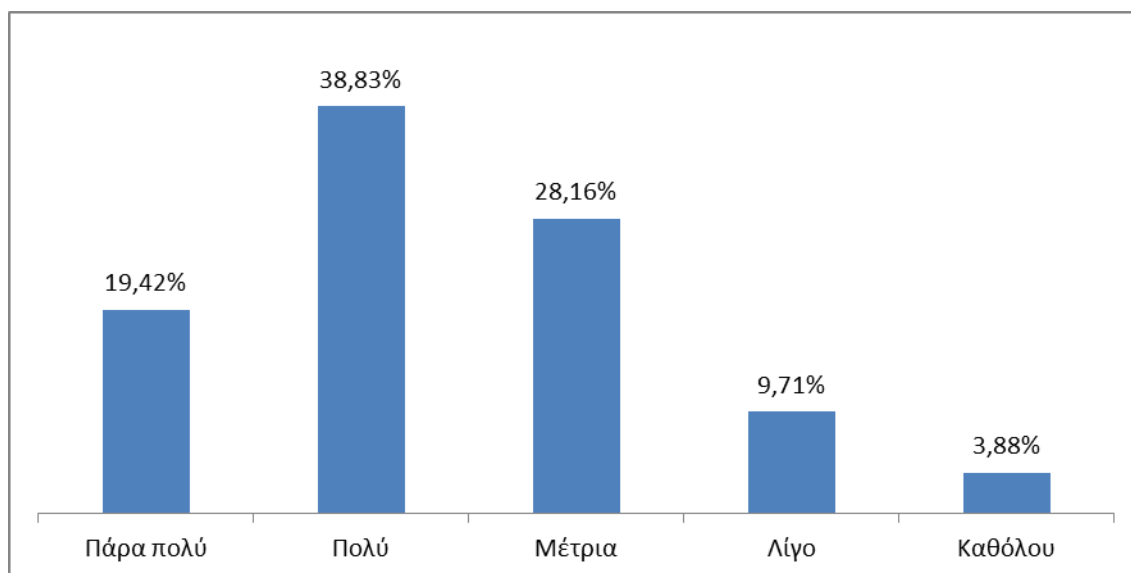
Μόνο το 6,79% δήλωσε ότι δε θα μπορούσε εύκολα ή και καθόλου να σκεφτεί κάτι παρόμοιο που θα μπορούσε να φτιάξει.

Γράφημα 5: Καταλάβατε τη σύνδεση και τη λειτουργία των εξαρτημάτων;



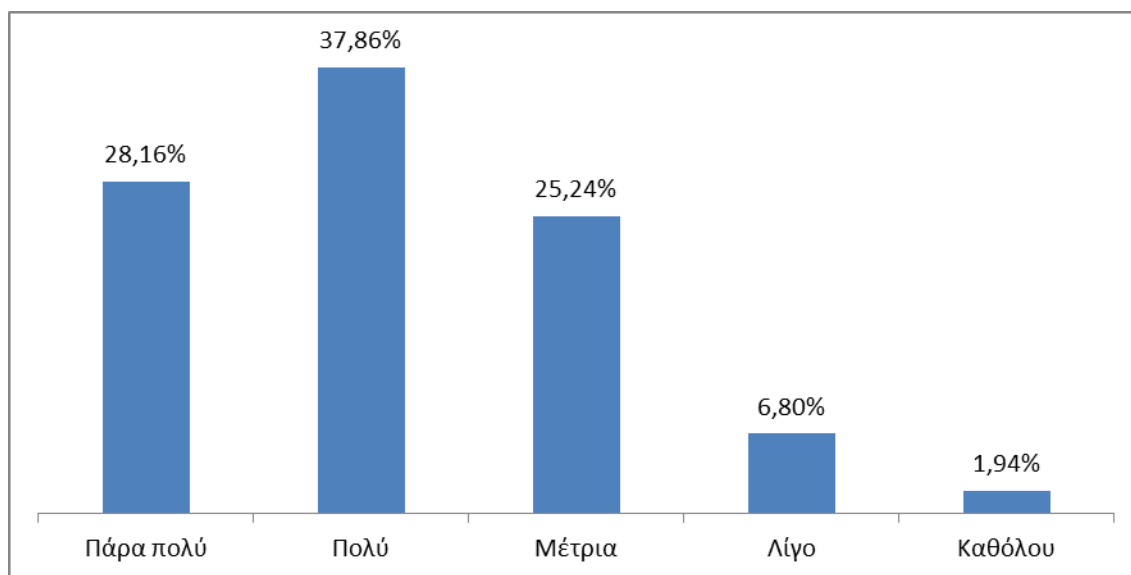
Η πλειοψηφία των μαθητών έδειξε να καταλάβει πολύ καλά τη σύνδεση και τη λειτουργία των εξαρτημάτων της εργασίας.

Γράφημα 6: Καταλάβατε πώς λειτουργεί ο κώδικας προγραμματισμού σε αυτήν την άσκηση;



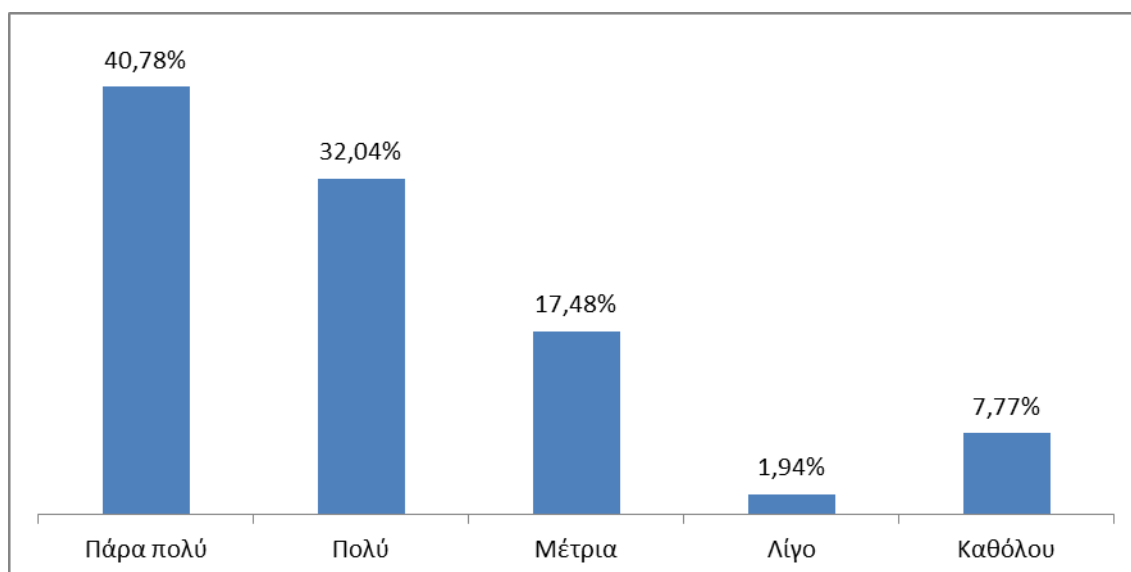
Το 58,25% των μαθητών θεώρησε εύκολο να κατανοήσει τον κώδικα, ενώ το 28,15% είχε μερικές δυσκολίες. Το 13,59% του δείγματος είχε μεγάλη δυσκολία στην κατανόηση.

Γράφημα 7: Πόσο εύκολο σας φαίνεται να φτιάξετε μόνοι σας κάτι ανάλογο ή να προσθέσετε κάτι πάνω σε αυτά;



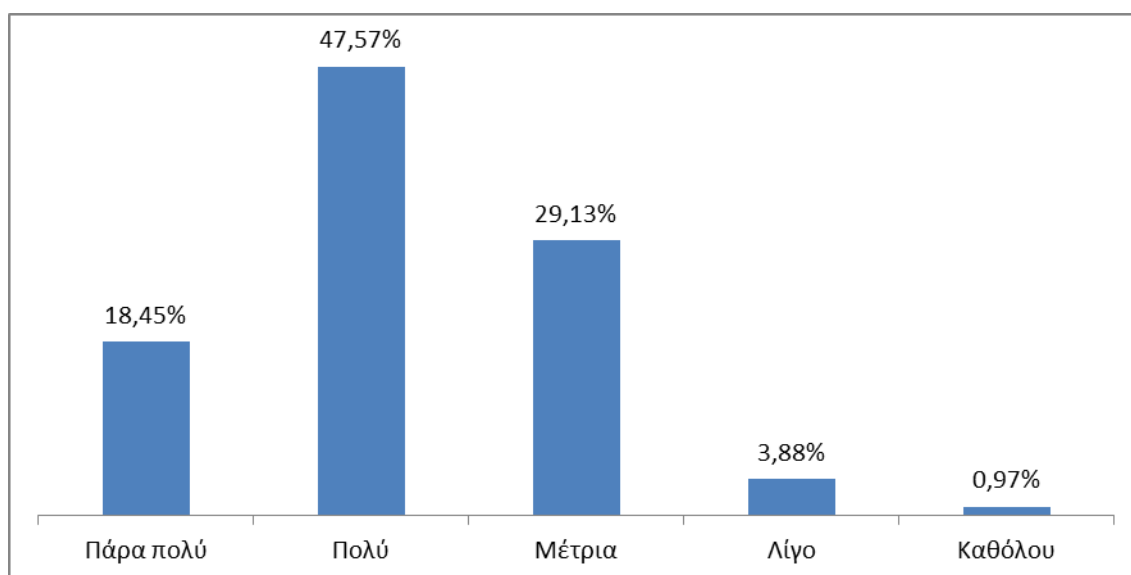
Το 25,24% του δείγματος κατάλαβε αρκετά καλά τον κώδικα προγραμματισμού, το 66,05 πολύ και πάρα πολύ καλά και μόνο το 8,74 δυσκολεύτηκε να τον καταλάβει ή δεν τον κατάλαβε καθόλου.

Γράφημα 8: Θα σας ενδιέφερε να ασχοληθείτε εσείς οι ίδιοι και να φτιάξετε κάτι δικό σας;



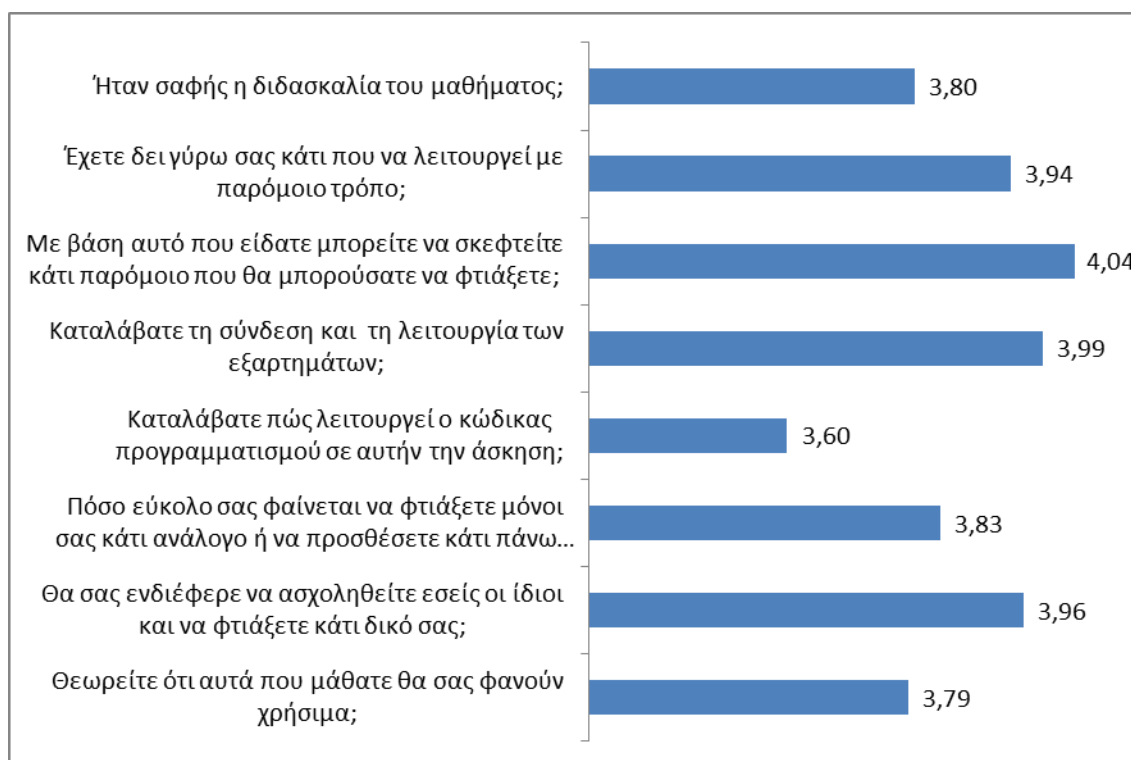
Το 72,82% του δείγματος έδειξε πολύ μεγάλο ενδιαφέρον να ασχοληθεί και να φτιάξει κάτι παρόμοιο.

Γράφημα 9: Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;



Μόνο το 4,85% των μαθητών θεώρησε ότι όσα διδάχθηκε έχουν ελάχιστη χρησιμότητα για το μέλλον.

Γράφημα 10: Μέσος όρος κάθε ερώτησης στην 4^η Εργασία



Η γενική βαθμολογία για το συγκεκριμένο μάθημα είναι 3,87 στα 5 (βλ Παράρτημα 5.3.2).

3.2 Συμπεράσματα

Η έρευνα έδειξε ότι η εισαγωγή της τεχνολογίας Arduino στην πρωτοβάθμια εκπαίδευση θα μπορούσε να πραγματοποιηθεί αλλά σίγουρα θα υπήρχαν κομμάτια που θα δυσκόλευαν τους μαθητές καθώς θα επρόκειτο για κάτι καινούριο με το οποίο δε θα είχαν ξανά ασχοληθεί κατά το παρελθόν. Αξιοσημείωτο ήταν βέβαια το γεγονός ότι τα παιδιά έδειξαν μεγάλο ενδιαφέρον να μάθουν καλύτερα αυτά που διδάχθηκαν μέσα από τα μαθήματα αλλά και ακόμα πιο αξιοσημείωτο ότι ζήτησαν την παράταση των μαθημάτων και τη δημιουργία περισσότερων κυκλωμάτων και εργασιών με βάση την τεχνολογία Arduino.

Οι μαθητές μπόρεσαν να ανταποκριθούν στο μάθημα των κυκλωμάτων με μεγάλη ευκολία, μιας και υπάρχει παρόμοιο μάθημα στο βιβλίο της Φυσικής της Ε΄ τάξης του δημοτικού. Οι μαθητές είχαν δει παρόμοια πράγματα και δε χρειάστηκε μεγάλη προσπάθεια για να καταλάβουν πως λειτουργούν και πως συμπεριφέρονται τα κυκλώματα.

Το μεγαλύτερο πρόβλημα το συναντήσαμε στο μάθημα του προγραμματισμού. Ο προγραμματισμός ήταν κάτι πρωτόγνωρο για τα παιδιά και δύσκολο. Οι μαθητές μπορούσαν μέχρι ενός σημείου να καταλάβουν το πώς λειτουργεί ο προγραμματισμός μέσα από παραδείγματα της καθημερινής τους ζωής. Το δύσκολο κομμάτι ήταν ο κώδικας προγραμματισμού που έπρεπε να δείξουμε και να εξηγήσουμε στα παιδιά. Οι μαθητές μπόρεσαν να καταλάβουν την έννοια του προγραμματισμού και πώς χρησιμοποιείται και λειτουργεί και που μπορεί να χρησιμοποιηθεί γενικότερα αλλά τους ήταν πολύ δύσκολο να πάρουν τον κώδικα προγραμματισμού και να τον χρησιμοποιήσουν. Επιπλέον πρόβλημα αποτέλεσε το γεγονός ότι ο κώδικας προγραμματισμού είναι στα αγγλικά και μερικοί μαθητές δεν είχαν τις απαιτούμενες γνώσεις αγγλικών. Ακόμα, δυσκολεύτηκαν αρκετά να καταλάβουν τις διαφορές σε μεταβλητές και σταθερές αλλά και να αναγνωρίσουν τους διάφορους τύπους μεταβλητών. Όλα αυτά όμως, μέχρι που παρακολούθησαν ένα πρόγραμμα στη γλώσσα προγραμματισμού C και πάνω σε αυτό εξηγήθηκε κάθε λεπτομέρεια. Συζητήθηκε όλο το μάθημα αλλά και γραμμή - γραμμή ο κώδικας προγραμματισμού ώστε να κατανοήσουν πλήρως όσα έμαθαν. Γενικότερα οι μαθητές είχαν μεγάλη δυσκολία σε αυτό το μάθημα και αυτό συνέβη γιατί έβλεπαν εντολές, μεταβλητές, τύπους μεταβλητών και διάφορα άλλα σε περίπλοκη μορφή. Από τα μαθήματα μπορούμε να βγάλουμε το συμπέρασμα πώς αν οι μαθητές είχαν έναν ψευδοκώδικα και συγκεκριμένα στα ελληνικά με απλές και ευκολονόητες εντολές η μάθηση του προγραμματισμού θα ήταν πιο εύκολη.

Στο μάθημα της τεχνολογίας Arduino οι μαθητές ήταν επιφυλακτικοί γιατί δεν ήξεραν τι θα αντιμετωπίσουν. Όταν ξεκίνησε το μάθημα και εξηγήσαμε στους μαθητές ότι μια πλακέτα Arduino είναι σαν ένας μικρός υπολογιστής και συγκεκριμένα σαν μια μητρική πλακέτα άρχισαν να ενθουσιάζονται και να

ανυπομονούν για το τι θα δουν. Καθώς εξηγούσαμε τι είναι η τεχνολογία Arduino και ποια είναι η χρησιμότητά της, οι μαθητές είχαν πολλές ερωτήσεις για το που μπορεί να χρησιμοποιηθεί και με ποιον τρόπο. Το γεγονός ότι είχαν άμεση επαφή με τα εξαρτήματα της πλακέτας Arduino κατά την εξέλιξη του μαθήματος έκανε το ενδιαφέρον τους ακόμα μεγαλύτερο και ανυπομονούσαν για τη συναρμολόγηση του πρώτου τους κυκλώματος.

Μετά το τέλος των θεωρητικών μαθημάτων παρατηρήθηκε μεγάλη ανυπομονησία να αρχίσουν να επεξεργάζονται τα εξαρτήματα του Arduino. Δεν ήταν δύσκολη η συναρμολόγηση των κυκλωμάτων όπου έκαναν και ο κώδικας δεν είχε πολλές σύνθετες πράξεις για να τους δυσκολέψει. Τα μαθήματα εξελίχθηκαν σταδιακά ώστε σιγά σιγά να κατανοούν τι κάνουν, να αποκτούν εμπειρία και να αρχίσουν να σκέφτονται τι άλλο θα μπορούσαν να κάνουν με αυτήν την τεχνολογία και πως.

Ενθουσιασμένοι που κατάφεραν να κατασκευάσουν μόνοι τους κυκλώματα και να γράψουν κώδικα προγραμματισμού, ζήτησαν να συνεχιστούν τα μαθήματα και να γίνουν περισσότερες εργασίες. Στις μέρες μας, οι μαθητές και γενικότερα τα παιδιά αυτής της ηλικίας, ενθουσιάζονται όταν ασχολούνται με την τεχνολογία και πόσο μάλλον όταν καταφέρνουν να φτιάξουν ολόκληρα κυκλώματα και να τα προγραμματίσουν. Οι μαθητές είχαν πολλές ιδέες σχετικά με το τι θα μπορούσαν να φτιάξουν και αυτό συμβαίνει γιατί στη φαντασία των παιδιών δεν υπάρχουν όρια. Αφήνουν τη φαντασία τους ελεύθερη και προσπαθούν να καταφέρουν τα πάντα. Όμως αυτό κάποιες φορές μπορεί να έχει επιπτώσεις στα παιδιά γιατί όταν ένα παιδί απογοητευτεί ή κάποιος του πει ότι δε μπορεί να γίνει αυτό που επιθυμεί, τότε το παιδί σταματά να σκέφτεται και να έχει ιδέες.

Εν κατακλείδι, μπορούμε να πούμε πως η εισαγωγή της τεχνολογίας Arduino στην πρωτοβάθμια εκπαίδευση στους μαθητές της Ε' και ΣΤ' τάξης του δημοτικού είναι εφικτή αλλά για να μπορεί να γίνει πιο εύκολα κατανοητή από τα παιδιά θα ήταν προτιμότερο να υπάρξει ένας πιο εύκολος τρόπος προγραμματισμού, όπως για παράδειγμα να δημιουργηθεί ένας ψευδοκώδικας.

Αφού συλλέξαμε τις απαντήσεις που έδωσαν τα παιδιά στα ερωτηματολόγια μας βγάλαμε τον μέσο όρο της βαθμολογίας κάθε μαθήματος και τον γενικό μέσο όρο βαθμολογίας και καταλήξαμε στο παρακάτω γράφημα.

Γράφημα 1: Μέσος όρος βαθμολογίας κάθε μαθήματος και Συνολικός μέσος όρος κατανόησης μαθημάτων



Ο μέσος όρος κατανόησης όλων των μαθημάτων και εργασιών ανέρχεται στο 3,55/5 που είναι ένα ικανοποιητικό αποτέλεσμα (βλ Παράρτημα 5.3.3).

4. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ

1. Αποστολάκης Ε., Παναγοπούλου Ε., Σάββας Σ., Τσαγλιώτης Ν., Πανταζής Γ., Σωτηρίου Σ., Τόλιας Β., Τσαγκογέωργα Α., Καλκάνης Γ., «Φυσικά» Ε΄ Δημοτικού – Ερευνώ και Ανακαλύπτω, ΟΕΔΒ, Αθήνα
2. Αποστολάκης Ε., Παναγοπούλου Ε., Σάββας Σ., Τσαγλιώτης Ν., Πανταζής Γ., Σωτηρίου Σ., Τόλιας Β., Τσαγκογέωργα Α., Καλκάνης Γ., «Φυσικά» ΣΤ΄ Δημοτικού – Ερευνώ και Ανακαλύπτω, ΟΕΔΒ, Αθήνα
3. Κόμης, Β. (2004). Εισαγωγή στις εκπαιδευτικές εφαρμογές των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών. Αθήνα: Εκδόσεις Ν. Τεχνολογιών.
4. Κόμης Β. & Μικρόπουλος Α. (2001). Πληροφορική στην Εκπαίδευση, Πάτρα: Ε.Α.Π
5. Ράπτης, Α. & Ράπτη, Α.(2006). Μάθηση και Διδασκαλία στην Εποχή της Πληροφορίας. Τόμος Α΄ Ολική Προσέγγιση. Αθήνα: Ατραπός.
6. Bruner, J. (1986). Actual minds, possible words. Cambridge, Massachusetts: Harvard University Press.
7. Arduino – διαδικτυακή σελίδα <https://www.arduino.cc/>
8. Βικιπαίδεια – διαδικτυακή σελίδα <https://el.wikipedia.org/>

5. ΠΑΡΑΡΤΗΜΑ

5.1 Ερωτηματολόγιο Κυκλώματα, Προγραμματισμός και Arduino

Τίτλος Μαθήματος

Ερωτηματολόγιο Αξιολόγησης Μαθήματος

Συμπληρώστε τις ακόλουθες ερωτήσεις βάζοντας ✓ στο αντίστοιχο κουτάκι.

<p>1) Ήταν καλά οργανωμένη η παρουσίαση του μαθήματος;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>2) Ήταν σαφής η διδασκαλία του μαθήματος;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>
<p>3) Πιστεύετε ότι χρειάζονται περισσότερα παραδείγματα για να κατανοήσετε αυτά που διδαχθήκατε;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>4) Το μάθημα σας προκάλεσε απορίες;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>
<p>5) Θεωρείτε ότι μπορείτε να ανταποκριθείτε στο μάθημα με βάση αυτά που ξέρατε μέχρι τώρα;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>6) Πιστεύετε ότι χρειάζονται περισσότερες ώρες διδασκαλίας για το συγκεκριμένο μάθημα;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>
<p>7) Θα σας ενδιέφερε να παρακολουθήσετε παρόμοιο μάθημα στο μέλλον;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>8) Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>

5.2 Ερωτηματολόγιο Άσκηση 2^η, Άσκηση 3^η και Άσκηση 4^η

Τίτλος Μαθήματος

Ερωτηματολόγιο Αξιολόγησης Μαθήματος

Συμπληρώστε τις ακόλουθες ερωτήσεις βάζοντας ✓ στο αντίστοιχο κουτάκι.

<p>1) Ήταν σαφής η διδασκαλία του μαθήματος;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>2) Έχετε δει γύρω σας κάτι που να λειτουργεί με παρόμοιο τρόπο;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>
<p>3) Με βάση αυτό που είδατε μπορείτε να σκεφτείτε κάτι παρόμοιο που θα μπορούσατε να φτιάξετε;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>4) Καταλάβατε τη σύνδεση και τη λειτουργία των εξαρτημάτων;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>
<p>5) Καταλάβατε πώς λειτουργεί ο κώδικας προγραμματισμού σε αυτήν την άσκηση;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>6) Πόσο εύκολο σας φαίνεται να φτιάξετε μόνοι σας κάτι ανάλογο ή να προσθέσετε κάτι πάνω σε αυτά;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>
<p>7) Θα σας ενδιέφερε να ασχοληθείτε εσείς οι ίδιοι και να φτιάξετε κάτι δικό σας;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>	<p>8) Θεωρείτε ότι αυτά που μάθατε θα σας φανούν χρήσιμα;</p> <p><input type="radio"/> Πάρα πολύ</p> <p><input type="radio"/> Πολύ</p> <p><input type="radio"/> Μέτρια</p> <p><input type="radio"/> Λίγο</p> <p><input type="radio"/> Καθόλου</p>

5.3 Μεθοδολογία Ανάλυσης Αποτελεσμάτων

5.3.1 Μεθοδολογία για Κυκλώματα, Προγραμματισμό και Arduino

Αφού βγάλαμε τα ποσοστά για κάθε ερώτηση ξεχωριστά, μετά δώσαμε διαφορετικό συντελεστή βαρύτητας σε κάθε απάντηση. Είχαμε 5 απαντήσεις και δώσαμε συντελεστή βαρύτητας από το 1 μέχρι το 5 όπου 1 = Καθόλου, 2 = Λίγο, 3 = Μέτρια, 4 = Πολύ και 5 = Πάρα πολύ. Προσοχή στις ερωτήσεις 3,4 και 6, γιατί το νόημα των απαντήσεων είναι αντίστροφο. Πήραμε τα ποσοστά σαν αριθμούς και πολλαπλασιάσαμε με τον ανάλογο συντελεστή βαρύτητας την κάθε απάντηση. Μετά πήραμε και τα δεύτερα αποτελέσματα και βγάλαμε τον μέσο όρο δηλαδή τη Γενική Βαθμολογία κάθε μαθήματος.

5.3.2 Μεθοδολογία για Άσκηση 2^η, Άσκηση 3^η και Άσκηση 4^η

Χρησιμοποιήσαμε την ίδια μεθοδολογία με τις προηγούμενες ασκήσεις, με τη μόνη διαφορά ότι δεν υπήρχαν ερωτήσεις με αντίστροφο νόημα.

5.3.3 Μεθοδολογία Γενικού Μέσου Όρου

Πήραμε τις γενικές Βαθμολογίες όλων των μαθημάτων και βγάλαμε τον Μέσο Όρο της συνολικής κατανόησης των μαθημάτων και των εργασιών που πραγματοποιήθηκαν.

5.4 Υπεύθυνες Δηλώσεις – Ερωτηματολόγια - Παρουσιολόγια

Στη γραμματεία είναι κατατεθειμένο Παράρτημα (σε ξεχωριστό δέσιμο) με όλες τις υπεύθυνες δηλώσεις των διευθυντών των σχολείων, τα απαντημένα ερωτηματολόγια των μαθητών καθώς και καταστάσεις με τις παρουσίες των μαθητών υπογεγραμμένες από τους αρμόδιους. Συγκεκριμένα υπάρχουν δύο ντοσιέ με τα ακόλουθα:

- Ντοσιέ 1
 - 1) Υπεύθυνες Δηλώσεις Ενημερωτικών Σχολείων
 - 2) Υπεύθυνες Δηλώσεις Παρουσιολογίων
 - 3) Έγκυρα Ερωτηματολόγια Κυκλώματα
 - 4) Έγκυρα Ερωτηματολόγια Προγραμματισμός
 - 5) Έγκυρα Ερωτηματολόγια Arduino
- Ντοσιέ 2
 - 6) Έγκυρα Ερωτηματολόγια Άσκηση 2^η
 - 7) Έγκυρα Ερωτηματολόγια Άσκηση 3^η
 - 8) Έγκυρα Ερωτηματολόγια Άσκηση 4^η
 - 9) Άκυρα Ερωτηματολόγια

