

ΔΙ.ΠΑ.Ε

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΩΝΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΘΕΣΣΑΛΟΝΙΚΗ

ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ:

«Ανάπτυξη συστήματος αυτοματισμού για διαχωρισμό
χρωματιστών αντικειμένων »



ΤΟΥ: **ΡΟΥΣΟΠΟΥΛΟΥ ΧΡΗΣΤΟΥ**

ΕΠΙΒΛ/ΚΑΘΗΓΗΤΗΣ: **ΓΙΑΚΟΥΜΗΣ ΑΓΓΕΛΟΣ**

ΘΕΣΣΑΛΟΝΙΚΗ 2021

ΕΥΧΑΡΙΣΤΙΕΣ

Θεωρούμε υποχρέωσή της να ευχαριστήσουμε τον επιβλέποντα καθηγητή Άγγελο Γιακουμή για την πολύτιμη καθοδήγησή του. Επιπλέον, θέλουμε να ευχαριστήσουμε θερμά όλους της καθηγητές για όλα όσα διδαχθήκαμε αυτά τα υπέροχα χρόνια αλλά και το υπόλοιπο προσωπικό που συμβάλει για την λειτουργία του τμήματος. Επιπρόσθετα, οφείλουμε να αφιερώσουμε την πτυχιακή της εργασία της γονείς της που της συμπαραστάθηκαν όλα τα χρόνια της φοίτησης της στο ΑΤΕΙ Θεσσαλονίκης.

ΠΕΡΙΛΗΨΗ-ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Η εργασία έχει θέμα την καινοτομία αυτόματου συστήματος διαλογής χρωμάτων προϊόντος. Η ταξινόμηση των προϊόντων είναι μια πολύ δύσκολη βιομηχανική διαδικασία. Η συνεχής μη αυτόματη ταξινόμηση δημιουργεί προβλήματα συνοχής. Αυτή η ετικέτα περιγράφει ένα πρωτότυπο εργασίας που έχει σχεδιαστεί για την αυτόματη ταξινόμηση αντικειμένων με βάση το χρώμα. Ο αισθητήρας TCS230 χρησιμοποιήθηκε για να ανιχνεύσει το χρώμα του προϊόντος και ο μικροελεγκτής Arduino nano χρησιμοποιήθηκε για τον έλεγχο της συνολικής διαδικασίας. Η αναγνώριση του χρώματος βασίζεται στην ανάλυση συχνότητας της εξόδου του αισθητήρα TCS230. Χρησιμοποιήθηκαν δύο σερβοκινητήρες που θα μεταφέρουν τα χρωματιστά αντικείμενα. Το πρώτο σέρβο προορίζεται για την τοποθέτηση του προϊόντος που πρόκειται να αναλυθεί από τον αισθητήρα χρώματος και το δεύτερο σέρβο προορίζεται για τη μετακίνηση της τσουλήθρας με χωριστά διαμερίσματα για τον διαχωρισμό των προϊόντων. Τα πειραματικά αποτελέσματα υπόσχονται ότι το πρωτότυπο θα εκπληρώσει της ανάγκες για υψηλότερη παραγωγή και ακριβή ποιότητα στον τομέα της αυτοματοποίησης

The project is about the innovation of an automatic product color sorting system. Product classification is a very difficult industrial process. Continuous manual sorting creates consistency problems. This label describes a working prototype designed to automatically sort objects by color. The TCS230 sensor was used to detect product color and the Arduino nano microcontroller was used to control the overall process. Color recognition is based on the frequency analysis of the TCS230 sensor output. Two servomotors were used to carry the colored objects. The first servo is intended for positioning the product to be analyzed by the color sensor and the second servo is for moving the slide with separate compartments to separate the products. Experimental results promise that the prototype will meet the needs for higher production and precise quality in the field of automation.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ΑΥΤΟΜΑΤΙΣΜΟΣ, ΣΥΣΤΗΜΑ, ΚΑΙΝΟΤΟΜΙΑ, ΑΠΟΔΟΤΙΚΟΤΗΤΑ, ΔΙΑΛΟΓΗ

Περιεχόμενα

Κεφάλαιο 1^ο : Εισαγωγικές Έννοιες

- 1.1 Εισαγωγή.....
- 1.2 Ιστορική αναδρομή.....
- 1.3 Γενικά στοιχεία αυτοματισμού.....
- 1.4 Πλεονεκτήματα της χρήσης αυτοματισμών.....
- 1.5 Γενικά στοιχεία πλακετών.....
- 1.6 Γενικά στοιχεία μικροελεγκτών.....
- 1.7 Γενικές πληροφορίες για το Arduino.....
- 1.8 Οι μικροελεγκτές Arduino: Uno & Motor Shield.....
- 1.9 Arduino nano.....

Κεφάλαιο 2^ο : Η Γλώσσα C++

- 2.1 Η Ιστορία της γλώσσας C++.....
- 2.2 Γενικά στοιχεία της γλώσσας C++.....

Κεφάλαιο 3^ο : Project Διαλογής Χρώματος

- 3.1 Ηλεκτρονικοί διαλογείς χρώματος γενικά
- 3.2 Λειτουργία.....
- 3.3 Κώδικας προγράμματος.....
- 3.4 Πως λειτουργεί ο αισθητήρας χρώματος.....

Κεφάλαιο 1^ο

Εισαγωγικές Έννοιες

1.1 Εισαγωγή

Με την εξέλιξη της τεχνολογίας στο τομέα των μικροηλεκτρονικών συσκευών δημιουργήθηκαν πολλά εργαλεία και μέθοδοι που προσφέρουν στους μηχανικούς μια τεράστια ποικιλία αντικειμένων ικανών να πραγματοποιήσουν οποιαδήποτε εντολή τους τεθεί με τη κατάλληλη καθοδήγηση από τον προγραμματιστή τους. Πρωταγωνιστικό ρόλο, στην αύξηση παραγωγής μικροηλεκτρονικών συσκευών, κατέχουν οι πλακέτες τυπωμένου κυκλώματος. Η ανάπτυξη του μικροεπεξεργαστή προήλθε από την ανάγκη για την ανάπτυξη της τεχνολογίας που σχετίζονται με την κατασκευή κυκλωμάτων. Η έμπνευση και η δημιουργία μερικών από τις μεγαλύτερες καινοτομίες στα υπολογιστικά συστήματα ευθύνεται στους μικροεπεξεργαστές. Η αρχική ιδέα και η πρώτη κατασκευή ενός επεξεργαστή, πραγματοποιήθηκε στις αρχές του 1970. Η κατασκευή του πρώτου επεξεργαστή, με τη βοήθεια ενός ολοκληρωμένου κυκλώματος μεγάλης ολοκλήρωσης, άλλαξε ριζικά τον τρόπο κατασκευής των κεντρικών μονάδων επεξεργασίας. Ο μικροεπεξεργαστής είναι μια γρήγορη συσκευή, όχι όσο ένας υπολογιστής, έτσι ώστε κάθε εντολή που εκτελείται από αυτόν να γίνεται με πολύ γρήγορη ταχύτητα. Μια πλακέτα τυπωμένου κυκλώματος μπορεί να είναι μονής ή διπλής όψεως ή να αποτελείται από πολλαπλά επίπεδα. Οι πλακέτες αυτές χρησιμοποιούνται για την υποστήριξη και τη διασύνδεση ηλεκτρονικών στοιχείων μέσω αγωγίων μονοπατιών. Τα μονοπάτια αυτά τυπώνονται πάνω στη πλακέτα. Μια από τις δημοφιλέστερες και μεγαλύτερες οικογένειες πλακετών είναι οι μικροελεγκτές (microcontrollers). Ο μικροελεγκτής είναι ένας επεξεργαστής, που μπορεί να λειτουργήσει με ελάχιστα εξαρτήματα λόγω των πολλών υποσυστημάτων που διαθέτει. Μια από τις μεγαλύτερες και δημοφιλέστερες οικογένειες μικροελεγκτών είναι η Arduino. Ο Arduino αποτελεί ένα μικροελεγκτή μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους και εξόδους, η οποία μπορεί να προγραμματιστεί σε γλώσσα Wiring(C++). Κατάλληλοι μικροελεγκτές Arduino για την πραγματοποίηση του project είναι ο Arduino Uno και ο Motor shield και ο Nano. Ο Arduino Uno αποτελεί ένα μικροελεγκτή με δεκατέσσερις ψηφιακές εξόδους και έξι αναλογικές εξόδους. Προγραμματίζεται με το Arduino IDE με τη βοήθεια ενός καλωδίου USB. Ο Arduino Motor Shield αποτελεί έναν διπλό οδηγό πλήρους γέφυρας, που έχει σχεδιαστεί για την οδήγηση επαγωγικών φορτίων όπως οι ηλεκτρονόμοι, οι ηλεκτρομαγνητικές βαλβίδες, DC και βηματικούς κινητήρες. Συνδυασμένος ο μικροελεγκτής με το Motor Shield δημιουργούν τον κινητήριο μοχλό του αυτοκινούμενου οχήματος. Όλα τα παραπάνω φυσικά αναλύονται σε μεγαλύτερο βαθμό στα επόμενα κεφάλαια. Το αυτοκινούμενο όχημα αποτελεί σημαντική εύρεση, καθώς σκοπός είναι να αναπτυχθεί ένα αντικείμενο με τεχνητή νοημοσύνη που θα καθοδηγείται από τους ανθρώπους για την σωστή εκτέλεση του ζητούμενου προγράμματος. Η συγκεκριμένη τεχνητή νοημοσύνη θα πρέπει να αναπτυχθεί σε τέτοιο σημείο, ώστε να μπορεί να αντιληφθεί το περιβάλλοντα χώρο και τις συνθήκες που επικρατούν αυτόματα χωρίς την ανθρώπινη παρέμβαση.

1.2 Ιστορική Αναδρομή

Ο αυτοματισμός είναι μία πολύ παλιά ιστορία. Και σε μεγάλο βαθμό είναι Ελληνική. Η λέξη «αυτόματο» είναι Ελληνική και τη συναντάμε κατ' αρχάς στα Ομηρικά έπη. Στην αρχαιότητα οι Έλληνες, αρχικά φαντάζονταν, οραματίζονταν αυτόματα συστήματα και στη συνέχεια οι Έλληνες μηχανικοί της αρχαιότητας μελετούσαν, σχεδίαζαν και κατασκεύαζαν αυτόματα και επιπλέον έγραφαν γι' αυτά. Ο αυτόματος έλεγχος, και ειδικά η εφαρμογή της ανάδρασης, ήταν θεμελιώδους σημασίας για την ανάπτυξη του αυτοματισμού. Αρχικά, η ανάγκη για ακριβή μέτρηση του χρόνου έδωσε την ώθηση για την μελέτη αυτόματων συστημάτων. Γύρω στο 270 π.Χ. στην Αλεξάνδρεια της Αιγύπτου ο Έλληνας Κτεσίβιος εφεύρε ένα είδος ρυθμιστή στάθμης για τη χρήση του σε ένα ρολόι νερού. Ο ρυθμιστής αυτός διατηρούσε σταθερή τη στάθμη του νερού μιας δεξαμενής το οποίο είχε ως αποτέλεσμα τη σταθερή ροή νερού προς μία δεύτερη δεξαμενή, η στάθμη της οποίας εξαρτιόταν από το χρόνο που είχε περάσει από την έναρξη της διαδικασίας. Χρησιμοποιήθηκαν από Έλληνες και Άραβες για να ελέγχουν συσκευές όπως κλεψύδρες, λάμπες λαδιού και διανομείς κρασιού, όπως και το επίπεδο του νερού στις δεξαμενές. Τα πιο γνωστά Ελληνικά ονόματα, που ασχολήθηκαν με τέτοια συστήματα, είναι του Κτεσίβιου, του Φίλωνα (3ος αιώνας π.Χ.) και του Χίρωνα (πρώτος αιώνας μ.Χ.), οι οποίοι δραστηριοποιήθηκαν στην ανατολική Μεσόγειο(Αλεξάνδρεια). Ακόμη, πολύ Άραβες μηχανικοί σχεδίασαν συστήματα που βασιζόταν στην αρχή της ανατροφοδότησης. Δυστυχώς, όμως, οι έρευνες τους διακόπηκαν το 1258 όταν οι Μογγόλοι κατέλαβαν τη Βαγδάτη. Η εφεύρεση της ατμομηχανής από τον J Watt το 1769, σηματοδοτεί την έναρξη της Βιομηχανικής Επανάστασης στην Ευρώπη. Η ραγδαία εξέλιξη της τεχνολογίας έκανε δύσκολο το χειροκίνητο χειρισμό των προηγμένων, για την εποχή, μηχανημάτων. Αυτό οδήγησε στην εμφάνιση ποικίλων συσκευών ελέγχου κάποιες από τις οποίες αφορούσαν ρυθμιστές στάθμης, ρυθμιστές θερμοκρασίας, πίεσης και ταχύτητας. Το 1788 ο Watt χρησιμοποίησε τη φυγόκεντρο συσκευή αυτομάτου ελέγχου flyball (centrifugal governor) για τη ρύθμιση της ταχύτητας της περιστρεφόμενης ατμομηχανής. Καθώς τα συστήματα ελέγχου άρχισαν να γίνονται ιδιαίτερα περίπλοκα, η μέθοδος σχεδίασης δοκιμής και λάθους συνδυασμένη με τη διαίσθηση των μηχανικών άρχισε να αποδεικνύεται ανεπαρκής. Στα μέσα του 19ου αιώνα χρησιμοποιήθηκαν για πρώτη φορά τα μαθηματικά για τον έλεγχο της ευστάθειας των συστημάτων ανατροφοδότησης. Από τις διαφορικές εξισώσεις έως το κριτήριο ευστάθειας του Routh και τις συναρτήσεις μεταφοράς, αυτή η μαθηματική μοντελοποίηση προωθήθηκε μέχρι το 1900 οπότε και κάνει την εμφάνισή της η θεωρία συστημάτων ελέγχου. Το 1840, ο Βρετανός αστρονόμος G.B.Airy ανέπτυξε μία συσκευή που έδινε τη δυνατότητα για εκτενέστερη παρατήρηση ενός άστρου με το τηλεσκόπιο. Η συσκευή διέθετε ένα σύστημα ελέγχου ταχύτητας το οποίο περιστρέφε το τηλεσκόπιο ώστε να αντιστέται στην αλλαγή θέσης λόγω της περιστροφής της γης. Ο Airy ήταν ο πρώτος που χρησιμοποίησε διαφορικές εξισώσεις για την μελέτη της ευστάθειας του συστήματος κλειστού βρόγχου. Λίγο αργότερα, το 1868, ο J.C. Maxwell στο έγγραφό του "On Governors", ήταν σε θέση να εξηγήσει τις αστάθειες που παρουσιάστηκαν στο σύστημα αυτομάτου ελέγχου flyball, χρησιμοποιώντας διαφορικές εξισώσεις για να περιγράψει το σύστημα ελέγχου. 2 Μέχρι τον 20ο αιώνα

η ανάλυση των συστημάτων ελέγχου γινόταν στο πεδίο του χρόνου. Μόνο στη δεκαετία του 20 και του 30 το πεδίο της συχνότητας με τα μαθηματικά του Laplace, Fourier και Cauchy άρχισε να παίζει σημαντικό ρόλο. Το κύριο πρόβλημα στα συστήματα μαζικής επικοινωνίας ήταν η ανάγκη για περιοδική ενίσχυση των σημάτων φωνής σε μακρές τηλεφωνικές γραμμές χωρίς την ταυτόχρονη ενίσχυση και του θορύβου. Σε αυτή την περιοχή ήταν που ο H. S. Black έδειξε τη χρησιμότητα της αρνητικής ανατροφοδότησης το 1927 με τους H. Nyquist και H. W. Bode να εισάγουν νέες τεχνικές σχεδίασης την επόμενη δεκαετία. Κατά τη διάρκεια των παγκόσμιων πολέμων, η ανάπτυξη του ελέγχου με ανατροφοδότηση έγινε ένα θέμα επιβίωσης. Ήταν το κλειδί στην ανάπτυξη συστημάτων ελέγχου των πλοίων, συστημάτων πλοήγησης καθώς και ακριβείας σκόπευσης των όπλων. Επίσης, η εξάλειψη του θορύβου έγινε και πάλι σημαντικό ζήτημα κατά τη διάρκεια μελέτης των προβλημάτων της επεξεργασίας των πληροφοριών η οποία συνδέθηκε με την εφεύρεση του «ραντάρ». Η δουλειά του MIT Radiation Lab οδήγησε στην ανάπτυξη προηγμένων τεχνικών σχεδίασης. Μετά τον δεύτερο παγκόσμιο πόλεμο αρχίζει η ηλεκτρονική εποχή. Ήδη από τις αρχές του 20ου αιώνα έχουμε τις πρώτες ηλεκτρονικές συσκευές, το ραδιόφωνο και, αργότερα, την τηλεόραση, τους ασύρματους και τα ραντάρ. Το κύριο εξάρτημα αυτών των συσκευών ήταν η ηλεκτρονική λυχνία. Η ανακάλυψη του τρανζίστορ το 1950 ήταν η αρχή της ηλεκτρονικής επανάστασης των ημιαγωγών. Το θαυματουργό αυτό στοιχείο αντικατέστησε την ακριβή, ογκώδη και ενεργειακά βόρα ηλεκτρονική λυχνία και έκανε τις ηλεκτρονικές συσκευές μικρότερες, εύκολες στην κατασκευή και απείρως πιο φθηνές. Το 1945 κατασκευάστηκε ο πρώτος ηλεκτρονικός υπολογιστής, ο ENIAC, ο οποίος χρησιμοποιούσε λυχνίες. Ο ENIAC δεν θύμιζε σε τίποτα τους σημερινούς υπολογιστές, ήταν ένα ολόκληρο εργοστάσιο το οποίο έλυνε μαθηματικές εξισώσεις. Μετά το 1950 και με τη χρήση των τρανζίστορ έχουμε τους πρώτους πραγματικούς υπολογιστές, οι οποίοι χρησιμοποιούνται κυρίως στο θέμα της μηχανογράφησης, δηλαδή στην αποθήκευση και διαχείριση μεγάλων αρχείων δεδομένων. Προς το τέλος της δεκαετίας του '50, Η σχεδίαση συστημάτων ελέγχου στράφηκε και πάλι προς το πεδίο του χρόνου. Μπορεί αρχικά να φαίνεται παράξενο αλλά τελικά έχει νόημα με την εισαγωγή σύνθετων, μη γραμμικών πολλών μεταβλητών συστημάτων που έχουν σχέση με εφαρμογές της αεροδιαστημικής. Από τη δεκαετία του '60 ήδη οι μηχανικοί άρχισαν να σκέφτονται τρόπους για να αξιοποιήσουν τις καταπληκτικές δυνατότητες των υπολογιστών στη βιομηχανία. Από τις πρώτες εφαρμογές των υπολογιστών στη βιομηχανία ήταν οι αυτόματες εργαλειομηχανές (τόρνοι, φρέζες κ.λπ.), οι οποίες μέχρι τότε χρησιμοποιούσαν κυρίως μηχανολογικούς και λιγότερο ηλεκτρολογικούς αυτοματισμούς. Η επιτυχημένη αυτή εφαρμογή οδήγησε τους μηχανικούς να αρχίσουν να σκέφτονται την αντικατάσταση όλων των αυτοματισμών ενός εργοστασίου από ένα υπολογιστή. Σήμερα οι ψηφιακοί υπολογιστές έχουν ολοκληρωτικά επικρατήσει των αναλογικών. Εκατομμύρια απ' αυτούς είναι εγκατεστημένοι στη βιομηχανία όπου ελέγχουν διεργασίες παρακολουθώντας και ελέγχοντας πλήθος μεταβλητών. Αρχικά τα συστήματα αυτόματου ελέγχου εισήχθησαν στην βιομηχανία λόγω της ανάγκης μείωσης του κόστους παραγωγής. Η ανάγκη περιορισμού των ελαττωματικών προϊόντων που παράγονται σε μια γραμμή παραγωγής καθώς και η ανάγκη παραγωγής προϊόντων υψηλής ποιότητας – εντός αυστηρών προδιαγραφών οδήγησαν στην αυξανόμενη εισαγωγή αυτοματισμών στην βιομηχανία. Φυσική συνέπεια της τάσης αυτής είναι ο

άνθρωπος να αποσύρεται σταδιακά από τη χειρονακτική εργασία, που εκτελείται πλέον από «έξυπνες» προγραμματιζόμενες μηχανές. Μέχρι την δεκαετία του 1960, τα συστήματα υλοποίησης των αλγορίθμων ελέγχου ήταν μηχανικά. Αποτελούνται από ηλεκτρομηχανικούς διακόπτες, τους ηλεκτρονόμους ή αλλιώς 3 ρελέ (relays) που ανοίγουν ή κλείνουν μία επαφή ανάλογα με την ροή ηλεκτρικού ρεύματος. Η χρήση των ρελέ παρουσίαζε πολλές δυσκολίες όπως ο δύσκολος προγραμματισμός τους (κάθε ρελέ έπρεπε να καλωδιωθεί ξεχωριστά), η περίπλοκη κατασκευή και η περιορισμένη διάρκεια ζωής.

Η βιομηχανία μέχρι και τη δεκαετία του 80 μπορούμε να πούμε ότι χρησιμοποιούσε ελάχιστα τα ηλεκτρονικά. Το 90% και πλέον των αυτοματισμών καταλάμβαναν οι αυτοματισμοί με ηλεκτρονόμους. Τα ηλεκτρονικά χρησιμοποιούνταν κυρίως για κάποιες "ευφρείς" εργασίες, και οι πλακέτες αυτές τοποθετούνταν μέσα στους πίνακες των ηλεκτρονόμων. Κι ενώ η τεχνολογία προχωρά, φθάνουμε στη δεκαετία του 90 όπου τεχνολογικά έγινε μεγάλο άλμα (συσκευές μικρότερες, φθηνότερες, με σημαντικά αυξημένες δυνατότητες συγκριτικά με αυτές της προηγούμενης δεκαετίας) αλλά παράλληλα αυξήθηκε δυσανάλογα το κόστος εκπόνησης των προγραμμάτων και της θέσης σε λειτουργία των εγκαταστάσεων.

Οι κατασκευαστές ρίχνουν πλέον σημαντικό βάρος στο λογισμικό όπου παρέχονται έτοιμες λύσεις για τομείς του αυτοματισμού με τη βοήθεια βιβλιοθηκών, εκμεταλλεύονται την πρόοδο των ηλεκτρονικών υπολογιστών και χρησιμοποιούν την εξέλιξη στο λειτουργικό τους σύστημα για να μειώσουν τους χρόνους στον προγραμματισμό των. Εμφανίζονται νέες γλώσσες προγραμματισμού για τεχνολόγους σε γραφική μορφή (block διαγράμματα συναρτήσεων), όπου ο χρήστης μέσω βιβλιοθηκών κι έχοντας γνώση μόνο της παραγωγικής διαδικασίας "συνθέτει" τον 4 αυτοματισμό του. Τα υπόλοιπα γίνονται στο παρασκήνιο για λογαριασμό του. Υποστηρίζεται τέλος και η εξέλιξη στις γλώσσες προγραμματισμού των ηλεκτρονικών υπολογιστών (Pascal, C++) για χρήστες που είναι εξοικειωμένοι σε τέτοια περιβάλλοντα. Τέλος, ιδιαίτερη έμφαση δίνεται πλέον στη δικτύωση – ασύρματη ή ενσύρματη για τον προγραμματισμό και την επιτήρηση εξ αποστάσεως μέσω ειδικών συσκευών επικοινωνίας και λογισμικού για ηλεκτρονικό υπολογιστή (SCADA) καθώς και στις επικοινωνίες Internet.

1.3 Γενικά στοιχεία αυτοματισμών

Πριν να ξεκινήσουμε να δίνουμε τον ορισμό του Αυτοματισμού και των παραμέτρων του, ας δώσουμε μια αίσθηση της σημασίας του και της παρουσίας του στη ζωή μας. Ο αυτοματισμός είναι παντού. Σχεδόν τα πάντα είναι αυτοματισμός. Το κλειδί που γυρίζει στην πόρτα, το ποδήλατό μας, ο ανελκυστήρας στις πολυκατοικίες, το αυτοκίνητο, το αεροπλάνο, ένας παιδικός βώλος που χτυπά και μετατοπίζει έναν άλλο, το ίδιο μας το σώμα.

Σήμερα, στην παρούσα κατάσταση έντονου ανταγωνισμού, η αποδοτικότητα της παραγωγής θεωρείται γενικά ως το κλειδί της επιτυχίας. Η αποδοτικότητα της παραγωγής περιλαμβάνει την ταχύτητα με την οποία ο εξοπλισμός παραγωγής και η γραμμή παραγωγής μπορούν να μειώσουν το υλικό και το κόστος εργασίας του

προϊόντος, βελτιώνοντας την ποιότητα και μειώνοντας τις απορρίψεις, ελαχιστοποιώντας τον χρόνο διακοπής του εξοπλισμού παραγωγής και τον χαμηλού κόστους εξοπλισμό παραγωγής. Λαμβάνοντας υπόψη αυτό το θέμα, αναπτύσσεται το έργο το οποίο είναι πολύ χρήσιμο για τις βιομηχανίες. Κύριοι στόχοι της μελέτης είναι η μελέτη διαφόρων διαδικασιών διαλογής, ο σχεδιασμός και η κατασκευή αυτόματου συστήματος διαλογής χρωμάτων προϊόντος και τοποθέτηση στο κατάλληλο κουτί . Τα βασικά πλεονεκτήματα του συστήματος είναι ο μικρότερος χρόνος που απαιτείται για την ταξινόμηση του προϊόντος, καθώς όλο το σύστημα εκτελείται με μηχανή, υπάρχει λιγότερη πιθανότητα λάθους, λιγότερη δύναμη του ανθρώπου. Εάν η βιομηχανία μπορεί να παράγει το προϊόν εντός του απαιτούμενου εύρους, τότε η ζήτηση του προϊόντος θα αυξηθεί. Έτσι, η εταιρεία θα επωφεληθεί.

Ο αυτοματισμός είναι η διαδικασία μιας μηχανής κατά την οποία ένα μέρος της εκτελεί μια κίνηση, η οποία θα κάνει ένα άλλο μέρος της να κάνει μια δική του κίνηση, η οποία θα κάνει πάλι ένα άλλο μέρος να κάνει με τη σειρά του μια δική του κίνηση. Έτσι ένα σύνολο μερών λειτουργεί αποτελώντας μία μηχανή ή ένα υπερσύνολο συνδεδεμένων μηχανών. Η χρησιμότητα του Αυτοματισμού είναι καθοριστική για τον πολιτισμό μας. Ο κλάδος αυτός συντελεί στη μεγιστοποίηση της απόδοσης των μηχανών, όταν συνδέονται και συνεργάζονται, ώστε να παραγάγουν ή ώστε να παραγάγουν πιο γρήγορα, πιο δυνατά, πιο αποτελεσματικά. Για να το αντιληφθούμε αυτό θα πρέπει να γνωρίζουμε πως ο Αυτοματισμός επιτρέπει να δουλέψουν για μας οι μηχανές εκεί που δεν υπάρχει πολύς χώρος, εκεί που δεν υπάρχει φως να δούμε, αέρας να αναπνεύσουμε. Στους βυθούς των ωκεανών, στους κρατήρες των ηφαιστείων, στα βάθη των ορυχείων, στο διάστημα χάρη στον Αυτοματισμό οι μηχανές κάνουν την δουλειά τους.

Οι μηχανές μπορούν να εκτελούν εξαιρετικά επαναλαμβανόμενες εργασίες καλύτερα από τους ανθρώπους. Η κόπωση των εργαζομένων στις γραμμές συναρμολόγησης μπορεί να οδηγήσει σε μειωμένη απόδοση και να προκαλέσει προβλήματα στην διατήρηση της ποιότητας του προϊόντος. Ένας υπάλληλος που εκτελεί καθήκοντα επανεξέτασης ξανά και ξανά μπορεί τελικά να μην αναγνωρίσει το χρώμα του προϊόντος. Η αυτοματοποίηση πολλών από τις εργασίες στις βιομηχανίες μπορεί να συμβάλει στη βελτίωση της αποτελεσματικότητας του συστήματος παραγωγής. Ο σκοπός αυτού του μοντέλου είναι να σχεδιάσει και να εφαρμόσει ένα σύστημα το οποίο διαχωρίζει αυτόματα τα προϊόντα με βάση το χρώμα τους. Αυτή η μηχανή αποτελείται από τρία μέρη: μεταφορικό σερβοκινητήρα, αισθητήρα χρώματος και μοτέρ συνεχούς ρεύματος. Η έξοδος και η είσοδος αυτών των τμημάτων διασυνδέθηκε με Arduino nano. Για να μειωθούν οι ανθρώπινες προσπάθειες σε μηχανικούς ελιγμούς, αναπτύσσονται διάφοροι τύποι μηχανών διαλογής. Οι θεματικές μηχανές είναι πολύ δαπανηρές λόγω της πολυπλοκότητας της διαδικασίας κατασκευής. Μία κοινή απαίτηση στον τομέα της εκτίμησης χρωμάτων είναι αυτή της ανίχνευσης και της αναγνώρισης χρώματος.

1.4 Πλεονεκτήματα της χρήσης αυτοματισμών

Βασικά πλεονεκτήματα της χρήσης αυτοματισμών είναι:

1. Μείωση του λειτουργικού κόστους και αύξηση της αξιοπιστίας του συστήματος, τα οποία οφείλονται στους εξής παράγοντες:

- Απαιτείται ελάχιστο ανθρώπινο δυναμικό, καθώς οι αυτοματισμοί αναλαμβάνουν σημαντικό τμήμα της εργασίας το οποίο φέρουν εις πέρας με μεγαλύτερη αποτελεσματικότητα. Παράλληλα, με αυτό τον τρόπο, μειώνονται τα ανθρώπινα σφάλματα σε λειτουργίες ρουτίνας τα οποία είναι και τα πλέον συνήθη.

- Εντοπίζονται και αντιμετωπίζονται ταχύτερα και αποτελεσματικότερα τα διάφορα σφάλματα αλλά και οι καταστάσεις συναγερμού (alarms). Προλαμβάνονται έτσι καταστάσεις, που θα μπορούσαν να οδηγήσουν σε μη αναστρέψιμες ή εκτεταμένες βλάβες και επιπλέον οι όποιες αστοχίες αντιμετωπίζονται στον ελάχιστο δυνατό χρόνο.

- Είναι δυνατή πλέον, η εγκατάσταση και λειτουργία προηγμένων συστημάτων ελέγχου, συντονισμού και διαχείρισης του δικτύου σε όλα τα επίπεδα εφαρμόζοντας σύγχρονες τεχνικές όπως νευρωνικά δίκτυα, έμπειρα συστήματα (expert systems) ή ασαφή λογική (fuzzy logic).

2. Μείωση του κόστους συντήρησης, το οποίο αποτελεί έναν πάγιο στόχο εκείνων που σχεδιάζουν και υλοποιούν τέτοια συστήματα. Αυτό επιτυγχάνεται με :

- Μείωση των αστοχιών και του χρόνου αποκατάστασής τους. Παλαιότερα, και το πλέον απλό σύστημα αυτοματισμού και διαχείρισης μιας εγκατάστασης, απαιτούσε αρκετά χιλιόμετρα καλωδίωσης αλλά και πολύπλοκο εξοπλισμό, γεγονός που δημιουργούσε πολλές εστίες πιθανών σφαλμάτων και επιπλέον, όταν αυτά συνέβαιναν, ο εντοπισμός τους αλλά και η διόρθωσή τους αποτελούσαν σύνθετη και επίπονη διαδικασία. Η εισαγωγή ταχύτερων δικτύων υπολογιστών έλυσε σε σημαντικό βαθμό όλα αυτά τα προβλήματα, αφού πλέον η βασική απαίτηση σε καλωδίωση είναι ένα απλό καλώδιο του Profibus, Fieldbus ή του Modbus στις συνηθέστερες των περιπτώσεων. Επιπλέον, οι αναρίθμητες διαφορετικές συσκευές αντικαταστάθηκαν από όμοιες κάρτες στις οποίες το μόνο που διαφοροποιείται είναι το λογισμικό και έτσι χρειαζόμαστε ελάχιστα εξαρτήματα για τις ανάγκες της συντήρησης.

- Μείωση του κόστους τακτικής συντήρησης του κύριου εξοπλισμού μίας εγκατάστασης, αφού μπορούμε να έχουμε μια πραγματική εικόνα του τρόπου λειτουργίας αλλά και των καταπονήσεών του. Παραδείγματος χάριν, μέχρι τώρα η συντήρηση ενός μετασχηματιστή γινόταν με βάση στατιστικούς πίνακες. Αντίθετα τώρα, με μικρό κόστος, μπορούμε να έχουμε στην διάθεση μας ένα μεγάλο αριθμό πληροφοριών για την πραγματική του κατάσταση, όπως τα φορτία του, η ισχύς φόρτισής του, η θερμοκρασία του, ένα πλήρες ιστορικό ρευμάτων βραχυκύκλωσης διαμέσου αυτού και άλλες τέτοιες πληροφορίες, οι οποίες μπορούν να μας καθορίσουν με μεγαλύτερη ακρίβεια το πότε πρέπει να γίνουν εργασίες συντήρησης. Είναι σαφές ότι τέτοιου είδους πληροφορίες δε μπορούν να αποκτηθούν με τους συμβατικούς 5 (αναλογικούς) τρόπους και το κυριότερο να τύχουν επεξεργασίας προκειμένου να βγάλουμε κάποιο συμπέρασμα.

- Μείωση του κόστους συντήρησης και λειτουργίας των συστημάτων ελέγχου και προστασίας. Η εισαγωγή των μικροεπεξεργαστών, έδωσε τη δυνατότητα πραγματοποίησης εξαιρετικά σύνθετων λειτουργιών ελέγχου και προστασίας, οι οποίες με χρήση απλών αναλογικών ηλεκτρονικών και διακοπών θα ήταν όχι απλώς δύσκολες αλλά και σε πολλές περιπτώσεις αδύνατες. Οι σύγχρονες εγκαταστάσεις δεν είναι μόνο πιο πολύπλοκες, αλλά έχουν και μεγαλύτερες διαστάσεις, οπότε η χρήση σύγχρονων πληροφοριακών συστημάτων είναι μονόδρομος.

3. Μείωση του κόστους του εγκατεστημένου υλικού, η οποία μπορεί να προέλθει από τα εξής σημεία:

- Μείωση του κόστους καλωδίωσης για τα συστήματα ελέγχου και προστασίας. Κάθε μεγάλη εγκατάσταση έλεγχου και προστασίας, απαιτεί αρκετά χιλιόμετρα καλωδίων που πέρα από το υψηλό κόστος προκαλεί και άλλα προβλήματα και περιορισμούς, όπως αυτεπαγωγές, απώλειες σήματος και ηλεκτρομαγνητικές παρεμβολές.

- Μείωση του κόστους εγκατάστασης εξοπλισμού σχετιζομένου με συγκεκριμένες λειτουργίες. Με τα συμβατικά μέσα, κάθε νέα λειτουργία απαιτούσε την αγορά και εγκατάσταση νέου εξοπλισμού καθώς και καλωδίωσης. Κάτι τέτοιο δεν είναι απαραίτητο με τα νέα συστήματα SCADA, αφού το μόνο που πιθανόν να απαιτείται επιπλέον, είναι καινούργιοι αισθητήρες, οι οποίοι προσαρμόζονται στον υπάρχοντα εξοπλισμό με κατάλληλη επέμβαση στο λογισμικό. Επιπλέον, η εγκατάσταση του νέου εξοπλισμού, στις περισσότερες περιπτώσεις, δεν έχει σημαντικές απαιτήσεις ως προς την κτιριακή υποδομή, αφού τέτοια συστήματα καταλαμβάνουν πολύ μικρό χώρο.

4. Καλύτερη και αποτελεσματικότερη διαχείριση της παραγωγής και κατανάλωσης ενέργειας. Το πρόβλημα της βέλτιστης χρησιμοποίησης της ηλεκτρικής αλλά και των λοιπών μορφών ενέργειας, είναι ένα θέμα που απασχολεί σήμερα όλα τα αναπτυγμένα κράτη, όχι μόνο επειδή αυξάνεται η τιμή της kWh, αλλά και επειδή απαιτούνται όλο και μεγαλύτερα ποσά ενέργειας, ενώ παράλληλα, τα ενεργειακά αποθέματα εξαντλούνται. Προκειμένου οι βιομηχανίες να αντιμετωπίσουν το φλέγον αυτό ζήτημα κινούνται σε δύο βασικούς άξονες, ένα μακροπρόθεσμο και ένα βραχυπρόθεσμο. Μακροπρόθεσμο, οι βιομηχανίες θέλουν να γνωρίζουν με τη μέγιστη δυνατή ακρίβεια, τη ζήτηση φορτίου ανά ώρα, ημέρα, μήνα και έτος, ώστε να ελαχιστοποιούν την κατανάλωσή τους, με κριτήριο την εξοικονόμηση ενέργειας και επιπλέον να σχεδιάζουν τη στρατηγική τους για τις μελλοντικές χρονικές περιόδους. Βραχυπρόθεσμο, απαιτείται η -κατά το δυνατόν- ομαλοποίηση της ροής της ενέργειας, προκειμένου να αποφεύγουμε εξαιρετικά οδυνηρές καταστάσεις, όπως οι υπερβάσεις του φορτίου αλλά και του συντελεστή ισχύος.

1.5: Γενικά στοιχεία πλακετών

Στην απλούστερη μορφή της, μια πλακέτα τυπωμένου κυκλώματος (Printed Circuit Board, PCB) είναι κατασκευασμένη από μια λεπτή πλακέτα από μονωτικό υλικό, η οποία υποστηρίζει τα εξαρτήματα του κυκλώματος και τις αγωγίμες διαδρομές. Οι πλακέτες αυτές κατασκευάζονται κατά κύριο λόγο από χαλκό, στην μία ή και στις δυο πλευρές της που συνδέονται μεταξύ τους τα εξαρτήματα. Για την πραγματοποίηση μιας επαφής κόλλησης ανάμεσα στο εξάρτημα και την αγωγή διαδρομή, οι ακροδέκτες των εξαρτημάτων συγκολλούνται πάνω στις πίστες (pads), οι οποίες αποτελούν

τιμήματα της αγωγίμης διαδρομής. Η ένωση των ακροδεκτών με τις πίστες μπορεί να επιτευχθεί με δυο τρόπους:

- 1) τοποθετείται το εξάρτημα στην αντίστοιχη πίστα και γίνεται απευθείας η συγκόλληση, ή
- 2) οι πίστες διαθέτουν τρύπες που διαπερνούν την πλακέτα και με αυτόν τον τρόπο επιτυγχάνεται η στήριξη των εξαρτημάτων. Η πρώτη διαδικασία ονομάζεται surface mount technology, ενώ αντίστοιχα η δεύτερη ονομάζεται through hole technology.

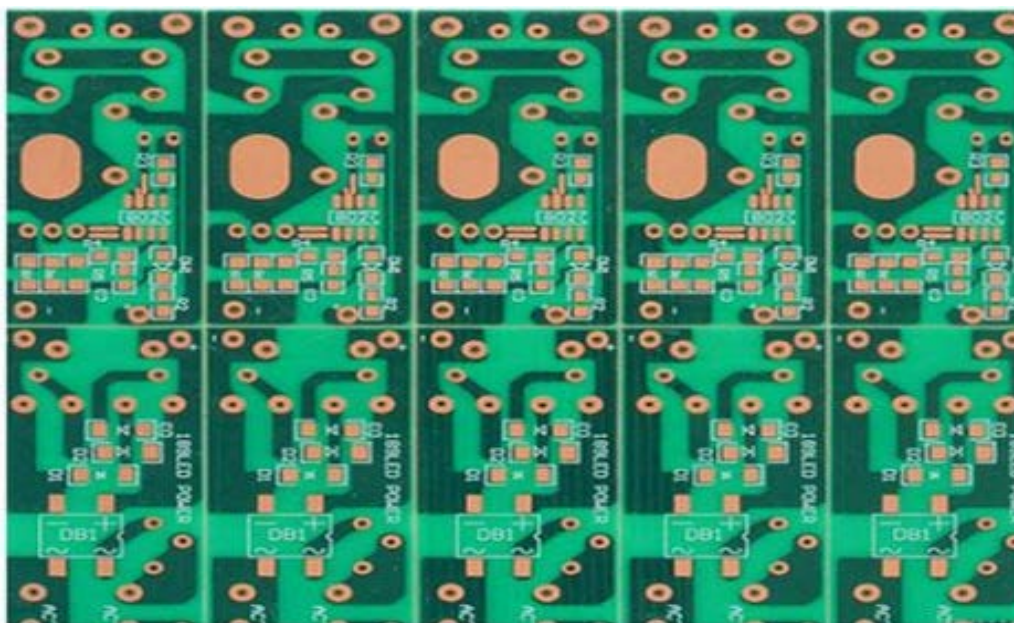
Γενικότερα, στο εμπόριο υπάρχουν τρεις τύποι πλακέτας τυπωμένου κυκλώματος. Οι τρεις αυτοί τύποι είναι οι εξής:

A) Τυπωμένη πλακέτα μονής όψεως (single sided PCB)

B) Τυπωμένη πλακέτα διπλής όψεως (double sided PCB)

Γ) Τυπωμένη πλακέτα πολλαπλών στρωμάτων (multilayer PCB) Αναλυτικά, η κάθε μια από αυτές είναι οι εξής :

A) Τυπωμένη πλακέτα μονής όψεως Οι πλακέτες μονής όψεως διαθέτουν στη μια πλευρά όλες τις αγωγίμες διαδρομές, ενώ στην άλλη πλευρά τα ηλεκτρονικά εξαρτήματα. Πολλές φορές, υπάρχει πιθανότητα τα ηλεκτρονικά εξαρτήματα να βρίσκονται στην ίδια πλευρά με τις αγωγίμες διαδρομές.



Εικόνα 1: Τυπωμένη πλακέτα μονής όψεως.

B) Τυπωμένη πλακέτα διπλής όψεως Οι πλακέτες διπλής όψεως χρησιμοποιούν αγωγίμες διαδρομές σε κάθε πλευρά της πλακέτας. Στην περίπτωση αυτή, τα ηλεκτρονικά εξαρτήματα μπορούν να τοποθετηθούν σε μια από τις δυο πλευρές ή ακόμα και στις δυο πλευρές. Οι διασυνδέσεις μεταξύ των αγωγίμων διαδρομών στα δυο επίπεδα σε συνδυασμό με τα εξαρτήματα εξασφαλίζουν μεγαλύτερη πυκνότητα εξαρτημάτων σε σύγκριση με τις πλακέτες μονής όψεως. Οι διασυνδέσεις των αγωγίμων διαδρομών στα δυο επίπεδα επιτυγχάνεται συνήθως με εξαρτήματα,

καλώδια, pins και επιμεταλλωμένες τρύπες, οι οποίες χρησιμοποιούνται αποκλειστικά από τις αγώγιμες διαδρομές.



Εικόνα 2: Πλακέτα διπλής όψεως.

Γ) Τυπωμένη πλακέτα πολλαπλών στρωμάτων. Οι πλακέτες πολλαπλών στρωμάτων διαθέτουν πολλαπλά επίπεδα με αγώγιμες διαδρομές, δυο από τις αγώγιμες διαδρομές βρίσκονται στην επιφάνεια της πλακέτας αυτής. Στο εσωτερικό της, η πλακέτα διαθέτει τα υπόλοιπα επίπεδα με έναν αριθμό αγώγιμων διαδρομών τα οποία συνδυάζονται με μονωτικά επίπεδα. Ένα πέρασμα μεταξύ των επιπέδων της πλακέτας απλά διαπερνά, ενώ οι εσωτερικές αγώγιμες διαδρομές συνδέονται με ένα κρυφό πέρασμα.



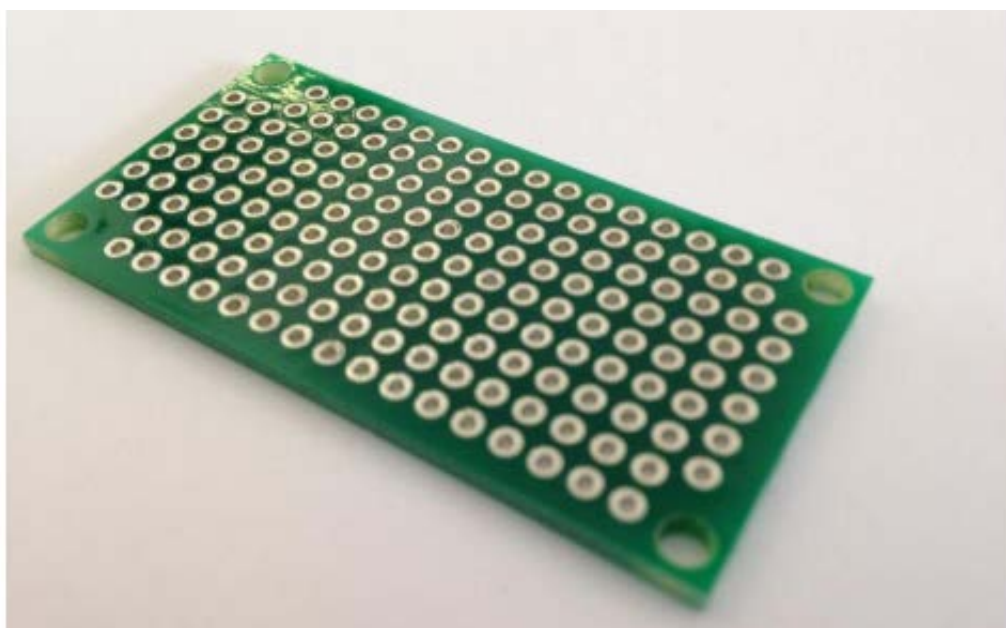
Εικόνα 3: Τυπωμένη πλακέτα πολλαπλών στρωμάτων.

Οι πλακέτες τυπωμένου κυκλώματος μπορούν να εντοπιστούν με διάφορες παραλλαγές ως προς την σχεδίαση τους. Παρακάτω, παρατίθενται οι τρεις πιο γνωστές παραλλαγές:

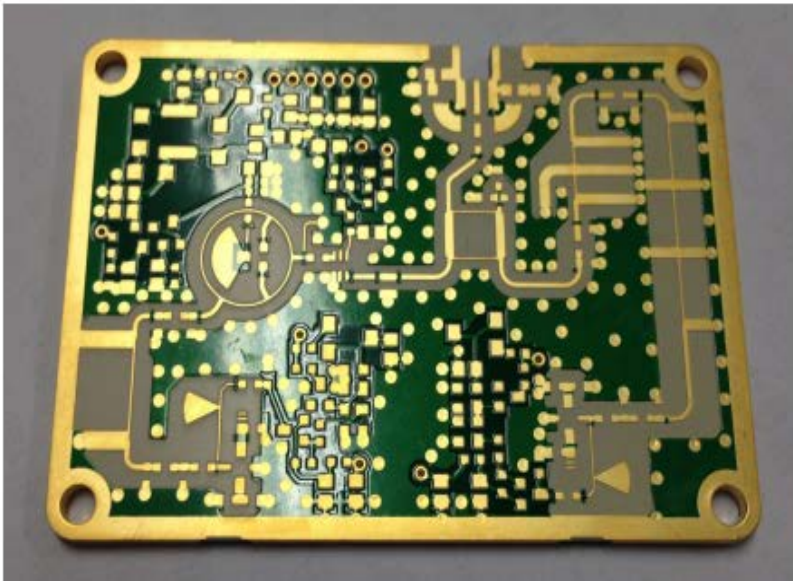
1) **Κυκλώματα με οπές μεταξύ των επιπέδων τους (Through-hole circuits).** Στα κυκλώματα αυτά, τα εξαρτήματα έχουν ακροδέκτες οι οποίοι τοποθετούνται σε οπές στις πλακέτες τυπωμένου κυκλώματος. Η κόλληση των ακροδεκτών στις δυο επιφάνειες της πλακέτας εξασφαλίζει την ηλεκτρική αγωγιμότητα. Αυτή η τεχνική όμως παρουσιάζει μερικά προβλήματα καθώς τα απλά και μικρά συστήματα κατασκευάζονται με αρκετή μεγάλη ευκολία, αλλά οι πλακέτες που κατασκευάζονται στα πολύπλοκα και μεγάλα συστήματα παρουσιάζουν προβλήματα ηλεκτρομαγνητικών παραβολών στις υψηλές συχνότητες.

2) **Υβριδικά κυκλώματα (Hybrid circuits).** Τα παθητικά εξαρτήματα που υλοποιούνται με τεχνολογία φιλμ (film technology)(*) και τοποθετούνται σε μια βάση σε συνδυασμό με τα ενεργητικά και τα παθητικά κυκλώματα που προστίθενται στο φιλμ για την ολοκλήρωση του κυκλώματος, συνθέτουν τα υβριδικά κυκλώματα. Η δομή της πλακέτας αυτού του τύπου δεν είναι ιδιαίτερα περίπλοκη, παρόλα αυτά οι πλακέτες αυτές χρησιμοποιούνται κυρίως σε διαδικασίες υψηλού κόστους.

3) **Κυκλώματα με εξαρτήματα επιφανειακής στήριξης(Surface mounted circuits).** Ο συνδυασμός των κυκλωμάτων με οπές μεταξύ των επιπέδων και των πλακετών τυπωμένου κυκλώματος δημιουργούν τα κυκλώματα με εξαρτήματα επιφανειακής στήριξης. Στα κυκλώματα αυτά, οι ακροδέκτες δεν διαπερνούν τη πλακέτα, καθώς στα εξαρτήματα τους δεν υπάρχουν ακροδέκτες με ακίδες, παρόλα αυτά τα εξαρτήματα τοποθετούνται έτσι ώστε να γίνεται η κόλληση των ακροδεκτών με τις αγώγιμες πίστες(pads). Θεωρείται η καλύτερη που κυκλοφορεί στην αγορά, καθώς μπορεί να προσφέρει πολύ υψηλές επιδόσεις, αλλά παρουσιάζουν μεγάλη δυσκολία στην παραγωγή και την επισκευή τους, καθώς οι διαδικασίες που απαιτούνται είναι πολύ υψηλού κόστους.



Εικόνα 4:Κύκλωμα με οπές μεταξύ των επιπέδων τους.



Εικόνα 5: Υβριδικό κύκλωμα



Εικόνα 6: Κύκλωμα με εξαρτήματα επιφανειακής στήριξης

1.6 Γενικά στοιχεία Μικροελεγκτών

Ο μικροελεγκτής είναι ένας τύπος επεξεργαστή, συγκεκριμένα ένας μικροεπεξεργαστής. Οι μικροελεγκτές μπορούν να λειτουργήσουν με ελάχιστα εξωτερικά εξαρτήματα, καθώς διαθέτουν πολλά ενσωματωμένα υποσυστήματα. Χρησιμοποιούνται σε όλα τα ενσωματωμένα συστήματα ελέγχου (embedded systems), κυρίως στους αυτοματισμούς, στα ηλεκτρονικά προϊόντα του εμπορίου (ψηφιακές φωτογραφικές κάμερες, παιχνίδια), σε ηλεκτρικές συσκευές αλλά και σε κάθε είδους αυτοκινούμενων τροχοφόρων οχημάτων. Οι μικροελεγκτές διακρίνονται από ορισμένα

πλεονεκτήματα, τα οποία τους καθιστούν τα πλέον κατάλληλα εξαρτήματα για αυτόματες εγκαταστάσεις. Τα πλεονεκτήματα αυτά είναι τα εξής :

- Αυτονομία. Πολλοί μικροελεγκτές δεν χρειάζονται κανένα ολοκληρωμένο κύκλωμα για να λειτουργήσουν, καθώς έχουν ενσωματωμένα περιφερειακά υποσυστήματα στους μικροελεγκτές.
- Μεγάλη αξιοπιστία, καθώς υπάρχουν ελάχιστες διασυνδέσεις.
- Οι μικροελεγκτές έχουν πολύ χαμηλό κόστος.
- Οι ελάχιστες διασυνδέσεις οδηγούν στην ευκολότερη υλοποίηση εφαρμογών. Το γεγονός αυτό, οδηγεί σε χαμηλή κατανάλωση ισχύος, μεγιστοποιείται η φορητότητα και μειώνεται το κόστος των μικροελεγκτών.
- Το συνολικό υπολογιστικό σύστημα μπορεί να περιλαμβάνεται σε πολύ μικρό μέγεθος.
- Η σύνδεση εξωτερικών περιφερειακών συστημάτων που πραγματοποιούνται χωρίς δεσμεύσεις, απελευθερώνουν περισσότερους ακροδέκτες για ψηφιακές εισόδους & εξόδους.
- Ο μικρότερος αριθμός, το μικρό μήκος των εξωτερικών διασυνδέσεων και η χαμηλή ταχύτητα λειτουργίας οδηγούν σε μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μικρότερη ευαισθησία στις παρεμβολές αυτές.

Ένας μικροελεγκτής θα μπορούσε να θεωρηθεί σαν ένας μικροϋπολογιστής. Οι μικροϋπολογιστές διαθέτουν επεξεργαστή, μνήμη, σύνδεση με περιφερειακές συσκευές και εκτέλεση προγραμμάτων. Με τον ίδιο ακριβώς τρόπο, λειτουργούν οι μικροελεγκτές καθώς διαθέτουν αυτά τα χαρακτηριστικά ολοκληρωμένα σε ένα chip. Οι μικροελεγκτές βρίσκουν εφαρμογή σε πολλά πεδία και είναι τα εξής :

- Σε συστήματα αυτοματισμού
- Σε τηλεπικοινωνιακά κυκλώματα
- Σε ηλεκτρικές & ηλεκτρονικές συσκευές
- Σε εφαρμογές ηλεκτρονικών ισχύος & δικτύων
- Σε συστήματα τηλεματικής
- Σε συστήματα συλλογής δεδομένων
- Σε συστήματα διασυνδέσεων Γενικότερα, η κύρια χρήση των μικροελεγκτών εντοπίζεται οπουδήποτε απαιτείται έλεγχος συστημάτων.



Εικόνα 7 : chip μικροελεγκτή

1.7 Γενικές πληροφορίες για το Arduino

Το Arduino αποτελεί μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα, η οποία βασίζεται σε εύκολο στη χρήση υλικό και λογισμικό. Οι πλακέτες Arduino έχουν την δυνατότητα να διαβάζουν εισόδους και να το μετατρέπουν σε εξόδους, μερικά παραδείγματα είναι η ανίχνευση φωτός μέσω ενός αισθητήρα και η ενεργοποίηση ενός LED, ή ακόμα και ένα δάκτυλο σε ένα κουμπί που μπορεί να ενεργοποιήσει τον κινητήρα. Η πλακέτα αντιλαμβάνεται τις ενέργειες που πρέπει να πραγματοποιήσει όταν ο χειριστής αποστέλλει ένα σύνολο οδηγιών του μικροελεγκτή της πλακέτας, την επίτευξη του σωστού συνόλου οδηγιών πρέπει να γίνει χρήση της γλώσσας προγραμματισμού του Arduino και του λογισμικού του Arduino (IDE), καθώς πρέπει να είναι ορθή και η επεξεργασία των εντολών της γλώσσας προγραμματισμού για να εκτελεστεί το σύνολο οδηγιών του μικροελεγκτή από την πλακέτα. Από το 2005 που δημιουργήθηκε το Arduino, και με το πέρασμα των χρόνων το Arduino αποτέλεσε βασικό εξάρτημα πολλών έργων, από τα καθημερινά αντικείμενα μέχρι και πολλά πολυσύνθετα επιστημονικά όργανα.

Ο Arduino γεννήθηκε το 2005 στο Ivrea Interaction Design Institute ως ένα εύκολο εργαλείο για τη δημιουργία πρωτότυπων σχεδίων. Η αρχική ιδέα απευθυνόταν σε φοιτητές που δεν έχουν εμπειρία στα ηλεκτρονικά και στο προγραμματισμό. Μόλις ξεκίνησε η πλακέτα να χρησιμοποιείται από μια μεγάλη κοινότητα ανθρώπων, ο Arduino ξεκίνησε να προσαρμόζεται στις νέες ανάγκες και προκλήσεις της εκάστοτε εποχής. Οι αλλαγές αυτές μετέτρεψαν τις απλές 8-bit κάρτες σε προϊόντα για εφαρμογές IoT, φορητές συσκευές, τρισδιάστατους (3D) εκτυπωτές και ενσωματωμένα περιβάλλοντα. Οι δυο ιδρυτές Massimo Banzi και David Cueartielles, ονόμασαν το αρχικό σχέδιο αυτό Arduin de Ivrea (Ο Αρντουίν της Ιβρέας). Το αρχικό σχέδιο του Arduino είναι ένα παρακλάδι της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα.

Προγραμματίζεται, χρησιμοποιώντας μια γλώσσα Wiring, παρόμοια με την γλώσσα C++ σε απλούστερη μορφή και με ορισμένες παραλλαγές, επίσης διαθέτει και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Όλοι οι πίνακες Arduino είναι ανοικτού κώδικα, με τον τρόπο αυτό επιτρέπουν στους χρήστες να κατασκευάζουν και να προσαρμόζουν τις πλακέτες στις ανάγκες τους. Μια από τις πιο γνωστές πλακέτες της εταιρίας είναι το Arduino Uno, βρίσκει χρήσεις σε πολλούς τομείς και είναι μια από τις οικονομικότερες εκδόσεις της εταιρίας.

1.8 Οι μικροελεγκτές Arduino: Uno & Motor Shield

Οι πλακέτες Arduino για την διευκόλυνση των χρηστών χωρίζονται σε δυο κατηγορίες που είναι ορατές και είναι εφικτό να χωριστούν για την κατανόηση της πλακέτας. Οι δύο κατηγορίες παρουσιάζονται παρακάτω αναλυτικά. Σε επίπεδο Υλικού(Hardware), η πλακέτα Arduino αποτελείται από έναν μικροελεγκτή Atmel AVR, καθώς και μερικά ακόμη εξαρτήματα που αφορούν τον προγραμματισμό και την σύνδεση του με άλλα κυκλώματα. Όλες οι πλακέτες εμπεριέχουν έναν ρυθμιστή τάσης στα 5 V και ένα κρυσταλλικό ταλαντωτή των 16 MHZ. Προκειμένου να μην υπάρχει εξάρτηση από εξωτερικό προγραμματιστή, ο μικροελεγκτής είναι προγραμματισμένος με ένα bootloader από την κατασκευή του. Η πλακέτα Arduino Uno διαθέτει δεκατέσσερις (14) ψηφιακές εισόδους/εξόδους, έξι εκ των οποίων μπορούν να

διαβάζουν αναλογικά δεδομένα και έξι που μπορούν να παράγουν διαμορφωμένα σήματα. Η συγκεκριμένοι πλακέτα, επίσης, διαθέτει ενσωματωμένο τροφοδοτικό και μπορεί να τροφοδοτηθεί από USB ή εξωτερική παροχή. Σε επίπεδο Λογισμικού(Software), το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μια εφαρμογή γραμμένη σε Java, λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το ολοκληρωμένο περιβάλλον ανάπτυξης της γλώσσας προγραμματισμού Processing σε συνδυασμό με το σχέδιο Wiring. Η εφαρμογή περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα και είναι σε θέση να μεταγλωττίζει και να φορτώνει τα προγράμματα στη πλακέτα με το πάτημα ενός κουμπιού.

Τα προγράμματα του Arduino είναι γραμμένα σε γλώσσα C ή C++. Το Arduino Motor Shield είναι βασισμένο στο chip L298, το οποίο είναι ένας διπλός οδηγός πλήρους γέφυρας, σχεδιασμένο για την οδήγηση επαγωγικών φορτίων όπως για παράδειγμα ηλεκτρονόμοι, ηλεκτρονικές βαλβίδες, DC και βηματικούς κινητήρες. Δίνει την δυνατότητα στον χειριστή να οδηγεί δυο ηλεκτροκινητήρες συνεχούς ρεύματος με την βοήθεια της πλακέτας Arduino, ελέγχοντας την ταχύτητα και κατεύθυνση καθενός ξεχωριστά. Μπορεί, επίσης, να μετρήσει την απορρόφηση ρεύματος κινητήρα από κάθε κινητήρα ξεχωριστά.

I) Αρχή λειτουργίας Arduino Uno

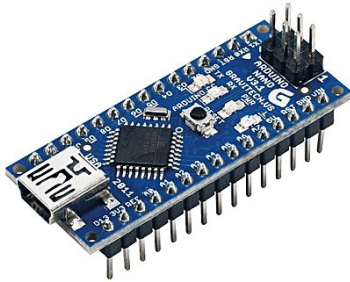
Ο Arduino Uno διαθέτει, από την κατασκευή του, αρκετούς τρόπους επικοινωνίας με έναν υπολογιστή, ή μια άλλη πλακέτα Arduino, ή ακόμα και από έναν άλλον μικροελεγκτή. Ο μικροελεγκτής ATmega328*, που είναι ενσωματωμένος στη πλακέτα του Arduino Uno, του παρέχει μια σειριακή επικοινωνία, η οποία διατίθεται σε ακίδες μηδέν (0) και ένα (1). Ένα microchip, το ATmega16U2, που υπάρχει πάνω στην πλακέτα μετατρέπει την σειριακή θύρα σε USB και εμφανίζεται ως εικονική θύρα, ονόματος COM, για το λογισμικό του υπολογιστή. Το υλικολογισμικό του 16U2 χρησιμοποιεί πρότυπα προγράμματα οδήγησης USB COM και δεν απαιτείται εξωτερικό πρόγραμμα οδήγησης. Μια σειριακή οθόνη επιτρέπει την αποστολή απλών δεδομένων κειμένου από και προς την πλακέτα, η σειριακή οθόνη αυτή περιλαμβάνεται από στο λογισμικό του Arduino (IDE). Όταν τα δεδομένα μεταδίδονται μέσω του chip USB to Serial (από USB σε σειριακή) και της σύνδεσης του υπολογιστή μέσω του USB, τότε οι ενδεικτικές λυχνίες των ακίδων (0-1) στη πλακέτα αναβοσβήνουν. Για να επιτευχθεί αυτό, η βιβλιοθήκη Software Serial επιτρέπει την σειριακή επικοινωνία με οποιαδήποτε από τις ψηφιακές ακίδες του Arduino Uno. Συνήθως, στην αυτόματη επαναφορά λογισμικού οι πλακέτες χρειάζονται φυσική πίεση του κουμπιού επαναφοράς πριν από τη μεταμόρφωση, εν αντίθεση η πλακέτα Arduino Uno κατασκευάστηκε με τέτοιο τρόπο, ώστε η επαναφορά του να πραγματοποιείται από το λογισμικό που εκτελείται από τον συνδεδεμένο υπολογιστή.

II) Αρχή λειτουργίας Arduino Motor Shield

Η πλακέτα Arduino Motor Shield αυτούσια δεν μπορεί να πραγματοποιήσει καμία λειτουργία, για τον λόγο αυτό χρησιμοποιείται κατά κόρον σε συνδυασμό με τον Arduino Uno. Η κύρια λειτουργία του είναι η οδήγηση, DC και βηματικών κινητήρων. Ο Arduino Motor Shield τοποθετείται στο πάνω μέρος ενός Arduino Uno μέσα στις ακίδες του (pins). Η σύνδεση των καλωδίων πλέον πραγματοποιείται από τους κινητήρες στην πλακέτα του Motor Shield καθώς και η τροφοδοσία γίνεται στην ίδια

πλακέτα. Με τον τρόπο αυτό, ο Arduino Uno αναλαμβάνει το κομμάτι του προγράμματος και αποδίδει τις συνδέσεις στο Motor Shield.

1.9 Arduino nano



Βασικά στοιχεία του Arduino Nano

Το Arduino Nano είναι ένας πίνακας μικροελεγκτών σχεδιασμένος από τον Arduino.cc. Ο μικροελεγκτής που χρησιμοποιείται στο Arduino Nano είναι ο Atmega328, ο ίδιος με τον Arduino UNO. Έχει ένα ευρύ φάσμα εφαρμογών και είναι ένας σημαντικός πίνακας μικροελεγκτών λόγω του μικρού μεγέθους και της ευελιξίας του. Τώρα, λοιπόν, ας δούμε τα βασικά χαρακτηριστικά του:

Βασικά χαρακτηριστικά του Arduino Nano

Εδώ είναι μερικά από τα βασικά χαρακτηριστικά του:

- Έχει 22 ακίδες εισόδου / εξόδου συνολικά.
- 14 από αυτές τις καρφίτσες είναι ψηφιακές καρφίτσες.
- Το Arduino Nano έχει 8 αναλογικές ακίδες.
- Έχει 6 ακροδέκτες PWM μεταξύ των ψηφιακών καρφίτσες.
- Διαθέτει κρυστάλλινο ταλαντωτή 16MHz.
- Η τάση λειτουργίας κυμαίνεται από 5V έως 12V.
- Υποστηρίζει επίσης διάφορους τρόπους επικοινωνίας, οι οποίοι είναι:
 - Σειριακό πρωτόκολλο.
 - Πρωτόκολλο I2C.
 - Πρωτόκολλο SPI.
- Έχει επίσης ένα μίνι USB Pin που χρησιμοποιείται για την αποστολή κώδικα.
- Έχει επίσης ένα κουμπί Reset σε αυτό.

Μνήμη στο Arduino Nano

Έχει κάτω από τις μνήμες που είναι ενσωματωμένες σε αυτό που χρησιμοποιούνται για διαφορετικούς σκοπούς και έχουν ως εξής:

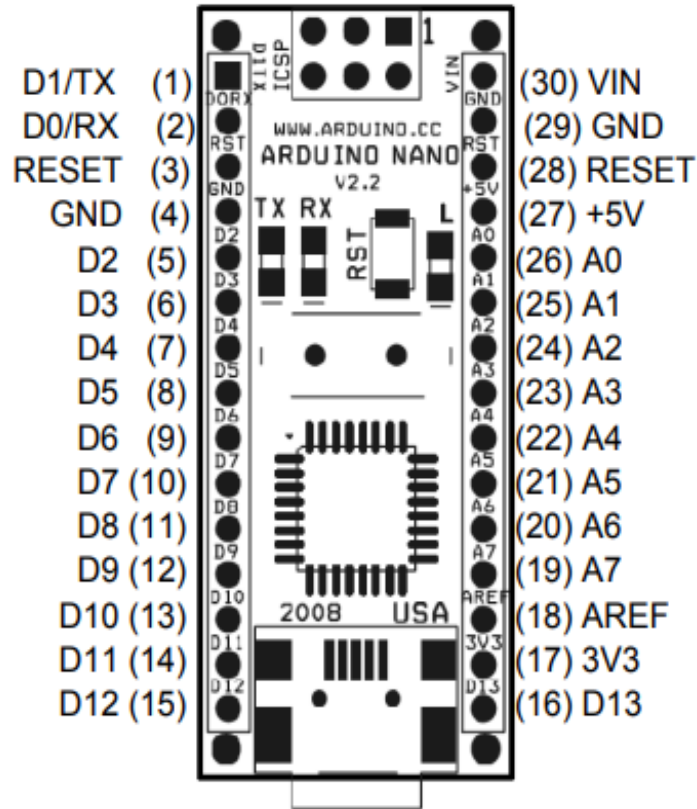
- Η μνήμη Flash του Arduino Nano είναι 32Kb.
- Έχει προεγκατεστημένο bootloader σε αυτό, το οποίο παίρνει μια μνήμη flash 2kb.
- Η μνήμη SRAM αυτής της κάρτας Microcontroller είναι 8kb.
- Έχει μνήμη EEPROM 1kb.

Εφαρμογές του Arduino Nano

Εδώ είναι λίγες από της εφαρμογές του:

- Ενσωματωμένα συστήματα.
- Αυτοματοποίηση.
- Ρομποτική.
- Συστήματα Ελέγχου.
- Ενοργάνιση.

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

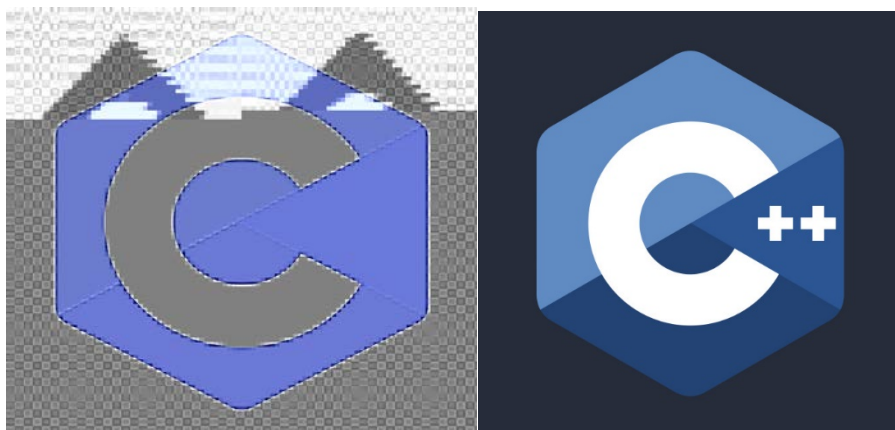
Κεφάλαιο 2: Η Γλώσσα C++

2.1: Η Ιστορία της γλώσσας C++

Η γλώσσα προγραμματισμού C αποτελεί μια γλώσσα προγραμματισμού γενικής χρήσης, η οποία δημιουργήθηκε μεταξύ του 1969 και του 1973 από τον Αμερικάνο επιστήμονα της πληροφορικής Dennis Ritchie (Ντένις Ρίτσι) στα εργαστήρια AT&T Bell Labs, όμοια με εκείνα που αναπτύχθηκε η C++. Ο αρχικός σκοπός της δημιουργίας αυτής της γλώσσας ήταν για να χρησιμοποιηθεί στην ανάπτυξη του λειτουργικού συστήματος Unix. Ο Ντένις Ρίτσι ήθελε να δημιουργήσει μια γλώσσα, στην οποία θα απαιτούνταν ένας μικρός αριθμός εντολών σε γλώσσα μηχανής για κάθε βασικό στοιχείο της, η διαδικασία αυτή ονομάζεται μεταγλώττιση. Ο Ρίτσι κατάφερε να επιτύχει τον στόχο του και έτσι πλέον η γλώσσα C μπορεί να αναπτύξει κώδικα σε χαμηλό επίπεδο προγραμματισμού με ακρίβεια ανάλογη της συμβολικής γλώσσας.

Το 1979 στα εργαστήρια Bell της εταιρίας AT&T,- αμερικάνικη εταιρία παροχής κινητής τηλεφωνίας-, ο Δανός επιστήμονας της πληροφορικής Bjarne Stroustrup (Μπιάρνε Στρούπστρουπ) ανέπτυξε την γλώσσα προγραμματισμού “C with Classes”, δηλαδή η γλώσσα προγραμματισμού C με κλάσεις, βασισμένος όμως στην αρχική ιδέα του Ρίτσι. Η γλώσσα αυτή επί της ουσίας προϋπήρχε, καθώς αποτελεί την βελτιωμένη έκδοση της γλώσσας προγραμματισμού C. Αργότερα, το 1983 ο Στρούπστρουπ μετονόμασε την γλώσσα αυτή σε γλώσσα προγραμματισμού «C++», καθώς μετά τις κλάσεις προέκυψαν και νέες βελτιώσεις όπως για παράδειγμα τα πρότυπα, οι εικονικές συναρτήσεις, η πολλαπλή κληρονομικότητα και η υπερφόρτωση των τελεστών. Το 1994, ο Στρούπστρουπ δημοσίευσε το βιβλίο «The Design and Evolution of C++» περιγράφει τα κριτήρια που χρησιμοποίησε για τον σχεδιασμό της C++, παρακάτω παρατίθενται αναλυτικά τα κριτήρια αυτά:

- Είναι μια γλώσσα γενικής χρήσης με στατικούς τύπους, τόσο αποτελεσματική όσο και φορητή.
- Μπορεί να υποστηρίξει άμεσα και γενικά πολλά είδη προγραμματισμού.
- Επιτρέπει στον προγραμματιστή πληθώρα επιλογών, ακόμη και αν είναι λανθασμένες.
- Η μετάβαση από τη γλώσσα C στη γλώσσα C++ είναι απλή, καθώς και οι γλώσσες είναι αρκετά συμβατές και όμοιες μεταξύ τους.
- Σχεδιάστηκε με τέτοιο τρόπο, ώστε να αποφεύγονται χαρακτηριστικά που δεν είναι γενικής χρήσεως ή που αναφέρονται σε συγκεκριμένες πλατφόρμες.
- Δεν δημιουργείται κανένα επεξεργαστικό κόστος.
- Η λειτουργία της γίνεται χωρίς κάποιο εξελιγμένο προγραμματιστικό περιβάλλον.



2.2 Γενικά στοιχεία της γλώσσας C++

Ο ηλεκτρονικός υπολογιστής είναι μια συσκευή, η οποία μπορεί να λαμβάνει αποφάσεις και να εκτελεί λογικές πράξεις μέσα σε ελάχιστο χρόνο. Οι υπολογιστές επεξεργάζονται δεδομένα, που αποτελούν ένα σύνολο οδηγιών. Τα δεδομένα αυτά ονομάζονται προγράμματα υπολογιστών. Οι προγραμματιστές, καθορίζουν το σύνολο των οδηγιών, δηλαδή το πρόγραμμα, έτσι ώστε ο υπολογιστής να εκτελέσει καθορισμένες ενέργειες. Στη γλώσσα των προγραμματιστών, τα προγράμματα που προαναφέρθηκαν χαρακτηρίζονται ως λογισμικό. Τα δεδομένα αποτελούνται από πολλαπλά στοιχεία. Τα στοιχεία δεδομένων μπορεί να είναι οτιδήποτε και για τον λόγο αυτό έχει δημιουργηθεί μια ιεραρχία δεδομένων. Η ιεραρχία αυτή αποτελείται από 6 διαφορετικά στοιχεία, τα οποία αναλύονται παρακάτω:

- 1) **Bit (Binary Digit)** : Αποτελεί το μικρότερο στοιχείο δεδομένων σε έναν υπολογιστή, έχει τη δυνατότητα να λαμβάνει τιμές 0 ή 1. Οι ηλεκτρονικοί υπολογιστές αντιλαμβάνονται αποκλειστικά τη δυαδική μορφή, όπως ονομάζεται το σύστημα αρίθμησης 0 ή 1. Με τη δυαδική μορφή οι υπολογιστές έχουν τη δυνατότητα να εκτελούν πράξεις, να εξετάζουν τη τιμή των Bits και γενικότερα να απλοποιούν το σύνολο των πράξεων στη μορφή αυτή.
- 2) **Χαρακτήρες** : Καθώς για τους ανθρώπους η επεξεργασία δεδομένων μέσω της χρήσης της δυαδικής μορφής είναι λίγο περίπλοκη, χρησιμοποιούν τα γνωστά για αυτούς σύμβολα, δεκαδικά ψηφία 0-9, γράμματα της αλφαβήτου Α-Ω και ορισμένα ειδικά σύμβολα (!,@,#,\$,%,&,*), το σύνολο των συμβόλων αυτών που χρησιμοποιούν οι άνθρωποι ονομάζεται Χαρακτήρες. Οι χαρακτήρες χρησιμοποιούνται τόσο για τη σύνταξη προγραμμάτων, όσο και για την αναπαράσταση στοιχείων δεδομένων. Γνωρίζοντας ότι οι υπολογιστές μπορούν να διαβάσουν αποκλειστικά τα δεδομένα σε δυαδική μορφή, δημιουργήθηκε ένα ειδικό σύνολο χαρακτήρων, που ονομάζεται Unicode, για τη μετατροπή των χαρακτήρων σε χαρακτήρες με τη μορφή Bit. Το γνωστότερο υποσύνολο χαρακτήρων της Unicode αποτελεί το σύνολο χαρακτήρων ASCII. Συγκεκριμένα, η C μπορεί να υποστηρίξει ένα μεγάλο αριθμό συνόλων χαρακτήρων.
- 3) **Πεδία** : Τα πεδία συνήθως αποτελούνται από χαρακτήρες και Byte. Μια ομάδα χαρακτήρων ή Bytes που μεταφέρουν μαζί τους κάποιο νόημα, ονομάζονται Πεδία. Η διαδικασία μετατροπής των πεδίων σε γλώσσα μηχανής

πραγματοποιείται με τον ίδιο τρόπο που αναφέρθηκε παραπάνω στους χαρακτήρες.

- 4) **Εγγραφές** : Μια εγγραφή δημιουργείται όταν συνδέονται πολλαπλά πεδία μεταξύ τους. Παράδειγμα μιας εγγραφής είναι ένα σύστημα μισθοδοσίας, στο οποίο αναγράφονται το όνομα, η διεύθυνση κατοικίας, ο αριθμός ταυτότητας του υπαλλήλου, το ωρομίσθιο, τα κέρδη του έτους ως σήμερα και το ποσό του φόρου που παρακρατήθηκε. Στο παραπάνω παράδειγμα τα πεδία αρχικά θα ανήκουν σε έναν υπάλληλο αλλά ύστερα θα πρέπει να πραγματοποιηθεί η ίδια διαδικασία για όλους τους υπαλλήλους της εταιρίας. Επομένως, πράγματι μια εγγραφή αποτελεί ένα σύνολο πολλαπλών πεδίων.
- 5) **Αρχεία** : Μια ομάδα αποτελούμενοι από εγγραφές ονομάζεται Αρχείο. Παράδειγμα για τα αρχεία είναι το συχνό φαινόμενο που συναντάται μια εταιρία ή ένας οργανισμός να έχει πολλά αρχεία που περιέχουν εκατομμύρια ή δισεκατομμύρια χαρακτήρες πληροφοριών.
- 6) **Βάση δεδομένων** : Μια ηλεκτρονική συλλογή δεδομένων που οργανώνεται για εύκολο χειρισμό και πρόσβαση, ονομάζεται Βάση Δεδομένων. Το πιο συνηθισμένο μοντέλο βάσης δεδομένων είναι η σχεσιακή βάση δεδομένων, όταν δηλαδή τα δεδομένα αποθηκεύονται σε πίνακες. Οι πίνακες αυτοί αποτελούνται από εγγραφές και πεδία. Η βάση δεδομένων συναντάται κυρίως σε εκπαιδευτικά ιδρύματα, μεγάλους οργανισμούς και εταιρίες.

Η C++ αποτελεί μια ισχυρή γλώσσα προγραμματισμού υπολογιστών. Αποτελεί μια από τις δημοφιλέστερες γλώσσες ανάπτυξης λογισμικού, καθώς έχει το πλεονέκτημα ότι είναι κατάλληλη τόσο για έμπειρους προγραμματιστές, που την χρησιμοποιούν για τη δημιουργία πληροφοριακών συστημάτων, όσο και για άτομα με ελάχιστη ή και καθόλου εμπειρία στον προγραμματισμό. Η εποχή που διανύουμε έχει ως κύριο χαρακτηριστικό την ανάπτυξη της τεχνολογίας σε όλους τους τομείς, το γεγονός αυτό οδηγεί τους ανθρώπους στην ενσωμάτωση της τεχνολογίας στην ζωή τους. Σύμφωνα με μια μελέτη eMarketer, που διεξήχθη το έτος 2012 και διήρκεσε ένα χρόνο, ο αριθμός χρηστών του Διαδικτύου αποκλειστικά σε κινητές συσκευές ξεπέρασε τα 134 εκατομμύρια, ενώ εκτιμάται πως χρησιμοποιούνται παγκοσμίως περισσότεροι από ένα δισεκατομμύριο υπολογιστές γενικών χρήσεων και 3 δισεκατομμύρια κινητά τηλέφωνα, smartphones και φορητές συσκευές. Τα δεδομένα και οι αριθμοί αυτοί καθιστούν κατανοητό το γεγονός ότι η ανάπτυξη της τεχνολογίας βαδίζει με ταχύτετους ρυθμούς, για το λόγο αυτό δεν θα μπορούσαν να μην εξελίσσονται τα πληροφοριακά συστήματα και οι γλώσσες προγραμματισμού. Από το 2011, δημοσιεύτηκε η τελευταία έκδοση της γλώσσας προγραμματισμού C++, ενώ μέσα στα επόμενα χρόνια αναμένεται μια νέα ακόμα πιο εξελιγμένη έκδοση. Η τωρινή έκδοση, γνωστή και ως C++, είναι αρκετά εξελιγμένη σε ότι έχει σχέση με την ανάπτυξη προγραμμάτων, καθώς έχει ορισμένες εντολές που δίνουν την δυνατότητα στους χρήστες να εξελίξουν τα προγράμματα τους, παραδείγματα είναι η αρχικοποίηση λίστας, η ανάθεση μεθόδων κατασκευής και διόρθωση του μεταγλωττιστή για ορισμένες εντολές σύμφωνα με τους πρότυπους τύπους.

Οι γλώσσες προγραμματισμού που γράφουν τις οδηγίες οι προγραμματιστές, είτε είναι άμεσα κατανοητές από τον υπολογιστή είτε απαιτούν ενδιάμεσα βήματα μετάφρασης. Τις γλώσσες, τις οποίες κατανοεί άμεσα ο υπολογιστής και συσχετίζονται με την αρχιτεκτονική του υλικού τους, ονομάζονται γλώσσες μηχανής ή κώδικας μηχανής. Οι γλώσσες μηχανής προκύπτουν από ακολουθίες ψηφίων 0 και 1, που στο

σύνολο τους δημιουργούν ακολουθίες αριθμών. Οι κώδικες μηχανής, όπως αλλιώς ονομάζονται είναι εξαιρετικά δύσχρηστες και ακατανόητες για τους ανθρώπους. Όπως προαναφέρθηκε η δημιουργία μια γλώσσας μηχανής ήταν εξαιρετικά δύσκολη και αργή διαδικασία για τους προγραμματιστές. Για τον λόγο αυτό, οι προγραμματιστές ξεκίνησαν να χρησιμοποιούν συντομογραφίες και συμβολισμούς που έμοιαζαν με λέξεις και γράμματα της αγγλικής γλώσσας για να αναπαριστάνουν ορισμένες από τις λειτουργίες των υπολογιστών. Οι συντομογραφίες αυτές αποτέλεσαν τη βάση για τη δημιουργία των γλωσσών συμβολομετάφρασης.

Για την μετατροπή των προγραμμάτων από γλώσσα συμβολομετάφρασης σε γλώσσα μηχανής, δημιουργήθηκαν και αναπτύχθηκαν ειδικά προγράμματα, γνωστά και ως συμβολομεταφραστές. Οι γλώσσες αυτές, μπορεί να είναι απόλυτα κατανοητές από τους ανθρώπους αλλά είναι τελείως ακατανόητες από τους υπολογιστές μέχρι να φθάσουν στο σημείο της μετάφρασης. Όσο ανέβαινε το επίπεδο του προγραμματισμού τόσο αυξανόταν και η ανάγκη για την επιτάχυνση της διαδικασίας του προγραμματισμού. Έτσι, αναπτύχθηκαν μερικές γλώσσες που επέτρεπαν στους προγραμματιστές να πραγματοποιούν σημαντικές και ουσιώδεις εργασίες με μια μόνο εντολή, οι γλώσσες αυτές ονομάζονται γλώσσες υψηλού επιπέδου. Οι γλώσσες αυτές, δίνουν στον προγραμματιστή τη δυνατότητα να γράφει εντολές στην αγγλική γλώσσα και περιέχουν ορισμένες μαθηματικές εκφράσεις. Μερικά παραδείγματα γλωσσών υψηλού επιπέδου είναι η Java, C++, Visual Basic και η C. Για την μετατροπή από γλώσσα υψηλού επιπέδου σε γλώσσα μηχανής χρησιμοποιούνται μερικά προγράμματα μετάφρασης, τα οποία ονομάζονται μεταγλωττιστές και εκτελούν αυτή τη λειτουργία. Σε αρκετές όμως περιπτώσεις, η διαδικασία της μεταγλώττισης ενός προγράμματος μπορεί να διαρκέσει αρκετό χρόνο. Οι διερμηνευτές είναι τα προγράμματα τα οποία εκτελούν απευθείας τις εφαρμογές που είναι γραμμένες σε γλώσσες υψηλού επιπέδου, συνήθως όμως είναι πολύ πιο αργά προγράμματα ως προς την εκτέλεση τους από τις μεταγλωττισμένες εκδοχές τους.

Τα τελευταία χρόνια έχουν αναπτυχθεί μερικές γλώσσες που υποβάλλονται σε επεξεργασία από διερμηνευτές, καθώς η λειτουργία τους είναι εξαιρετική έχουν γίνει δημοφιλείς και οι γλώσσες αυτές. Μερικά παραδείγματα αυτών των γλωσσών είναι οι γλώσσες του web JavaScript και PHP. Για την κατανόηση της λειτουργίας της γλώσσας C++ παρουσιάζονται παρακάτω οι φάσεις τις οποίες περνά ένα πρόγραμμα για να γίνει αντιληπτό από τον υπολογιστή και να εκτελεστεί. Αρχικά, τα συστήματα C++ αποτελούνται από τρία μέρη: 1) **περιβάλλον ανάπτυξης προγραμμάτων**, 2) **τη γλώσσα**, και 3) **τη Πρότυπη Βιβλιοθήκη C++**. Επιπλέον ένα πρόγραμμα διέρχεται από έξι φάσεις: 1) **σύνταξη**, 2) **προεπεξεργασία**, 3) **μεταγλώττιση**, 4) **διασύνδεση**, 5) **φόρτωση**, και 6) **εκτέλεση**. Παρακάτω ακολουθεί αναλυτική περιγραφή ενός τυπικού περιβάλλοντος ανάπτυξης εφαρμογών της γλώσσας C++ :

- 1) **ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ**: Κατά την διάρκεια της πρώτης φάσης με τη χρήση ενός προγράμματος δημιουργίας κώδικα, γνωστού και ως editor, συνιστάται η σύνταξη ενός αρχείου. Έκτος του κοινού προγραμματισμού σε γλώσσα C++, υπάρχουν ορισμένα ολοκληρωμένα περιβάλλοντα ανάπτυξης, γνωστά ως IDE. Τα IDE περιέχουν διάφορα εργαλεία, τα οποία υποστηρίζουν τη διαδικασία ανάπτυξης λογισμικού, τα οποία είναι προγράμματα για τη σύνταξη και επεξεργασία προγραμμάτων και προγράμματα εκσφαλμάτωσης. Ορισμένα από τα δημοφιλέστερα αυτά

προγράμματα είναι το Microsoft Visual Studio 2012, Dev C++, Eclipse, CodeLite, NetBeans και το Apple's Xcode.

- 2) **ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ:** Σε αυτή την φάση δίνεται η εντολή για τη μεταγλώττιση (compile) του προγράμματος. Στο σύστημα της γλώσσας C++, υπάρχει ένα πρόγραμμα που εκτελείται αυτόματα πριν ξεκινήσει η φάση της πραγματικής μετάφρασης από τον μεταγλωττιστή, το πρόγραμμα αυτό ονομάζεται προεπεξεργαστής. Ο προεπεξεργαστής της C++ εκτελεί ειδικές εντολές, οι οποίες υποδεικνύουν στο χρήστη ότι πρέπει να εκτελεστούν συγκεκριμένες λειτουργίες στο πρόγραμμα πριν την μεταγλώττιση του, οι ειδικές αυτές εντολές ονομάζονται οδηγίες προεπεξεργαστή.
- 3) **ΜΕΤΑΓΛΩΤΤΙΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ C++:** Ο μεταγλωττιστής, σε αυτή τη φάση, μεταφράζει τον πηγαίο κώδικα του προγράμματος C++ σε γλώσσα μηχανής, ή αλλιώς σε κώδικα αντικειμένου.
- 4) **ΔΙΑΣΥΝΔΕΣΗ:** Η φάση αυτή αποκαλείται διασύνδεση. Ο κώδικας αντικειμένου που παράγει ο μεταγλωττιστής της C++ εμπεριέχει ορισμένα κενά, καθώς τα προγράμματα που είναι γραμμένα στη γλώσσα C++ περιέχουν αναφορές σε συναρτήσεις και δεδομένα που είναι ορισμένα σε κάποια πρότυπη βιβλιοθήκη της γλώσσας ή σε κάποια ιδιωτική βιβλιοθήκη προγραμματιστών που συνεργάζονται για την υλοποίηση ενός έργου. Για την διασύνδεση ενός κώδικα αντικειμένου με τον κώδικα των συναρτήσεων που λείπουν χρησιμοποιείται ένας σύνδεσμος, ο οποίος παράγει ένα εκτελέσιμο πρόγραμμα που δεν έχει κανένα πλέον κενό. Στη περίπτωση που το πρόγραμμα διασυνδεθεί και μεταγλωττιστεί σωστά θα παραχθεί η εκτελέσιμη εικόνα του.
- 5) **ΦΟΡΤΩΣΗ:** Για να εκτελεστεί ένα πρόγραμμα πρέπει αρχικά να αποθηκευτεί στη μνήμη. Τη μεταφορά από την εκτελέσιμη εικόνα στο δίσκο στη μνήμη, την εκτελεί το πρόγραμμα φόρτωσης. Στη φάση αυτή, φορτώνεται στο πρόγραμμα οποιοδήποτε στοιχείο το υποστηρίζει.
- 6) **ΕΚΤΕΛΕΣΗ:** Λαμβάνοντας μια προς μια τις εντολές ο υπολογιστής, υπο τον έλεγχο της CPU του εκτελεί το πρόγραμμα. Μερικοί από τους σύγχρονους υπολογιστές μπορούν να εκτελέσουν περισσότερες από μια οδηγίες παράλληλα.

Κεφάλαιο 3^ο : Project Διαλογής Χρώματος

3.1 Ηλεκτρονικοί διαλογείς χρώματος

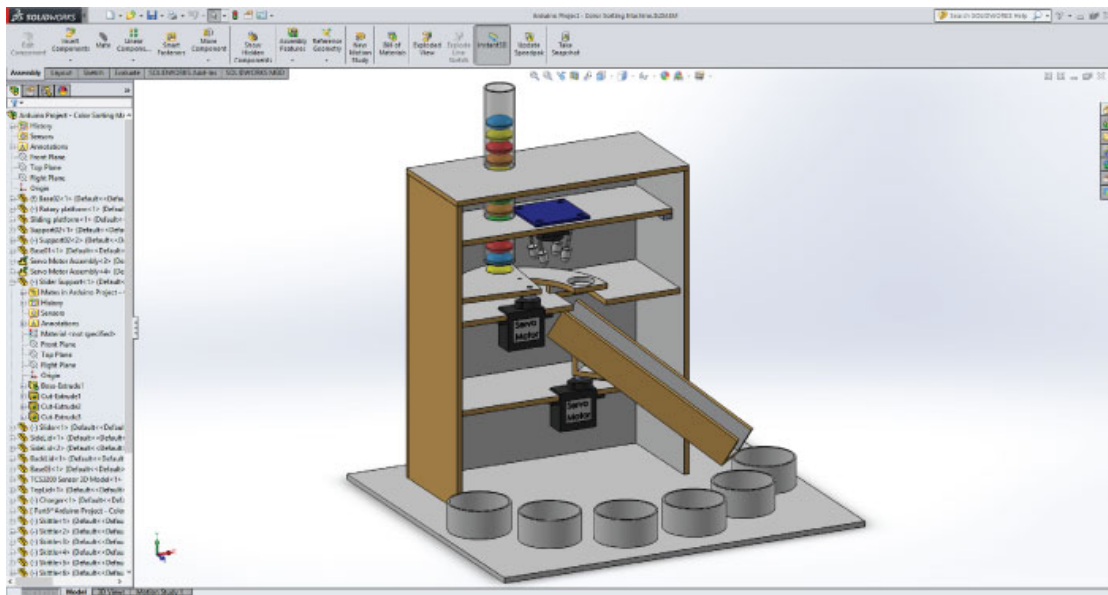
Οι ταξινομητές χρώματος ή οι διαχωριστές χρωμάτων (μερικές φορές αποκαλούνται οπτικοί ταξινομητές , ψηφιακοί διαλογείς ή ηλεκτρονικοί διαλογείς χρώματος) είναι μηχανές που χρησιμοποιούνται στις γραμμές παραγωγής για την επεξεργασία χύδην τροφίμων και σε άλλες βιομηχανίες. Διαχωρίζουν τα αντικείμενα ανάλογα με τα χρώματα τους, ανιχνεύοντας τα χρώματα των πραγμάτων που περνούν μπροστά τους και χρησιμοποιώντας μηχανικές ή πνευματικές συσκευές εκτόξευσης για την εκτροπή αντικειμένων των οποίων τα χρώματα δεν εμπίπτουν στην αποδεκτή περιοχή ή τα οποία επιθυμούν να σχηματίσουν ξεχωριστή ομάδα από τα υπόλοιπα.

Οι διαλογείς χρώματος χρησιμοποιούνται κυρίως για τη διαλογή σιτηρών (γεωργικά προϊόντα). Η βιομηχανία διαλογής ρυζιού είναι η πρώτη μεγάλη αγορά. Η τεχνολογία ταξινόμησης του ρυζιού είναι σύμφωνη με τις διαφορές χρώματος των υλικών ρυζιού (αποφλοιωμένου ορυζώνα), χρησιμοποιώντας έναν οπτικό αισθητήρα CCD υψηλής ανάλυσης για να διαχωρίσει τις πέτρες, το μαύρο ρύζι κλπ. Είναι το τελικό βήμα μετά το γυάλισμα του ρυζιού με έναν ρυμουλκούμενο ρύζι . Η δεύτερη αγορά διαλογής χρησιμοποιείται σε χοντρά σιτηρά, όπως σιτάρι, καλαμπόκι, φιστίκια, διάφορα είδη φασολιών, σουσάμι κλπ. Ένας τυπικός ταξινομητής χρωμάτων χρησιμεύει ως γεωργικός μηχανισμός για τη διαλογή σιτηρών ή άλλων στερεών σπόρων. Είναι μια μηχανή διαλογής υψηλής ακρίβειας της οποίας ο βιομηχανικός φακός μπορεί να ανιχνεύσει ακαθαρσίες 0,01 τετραγωνικών χιλιοστών σε γεωργική εφαρμογή. Μπορεί να χρησιμοποιηθεί για να αφαιρέσει ακαθαρσίες όπως άμμο, στέλεχος, γυαλί και σπασμένους σπόρους σιτηρών.

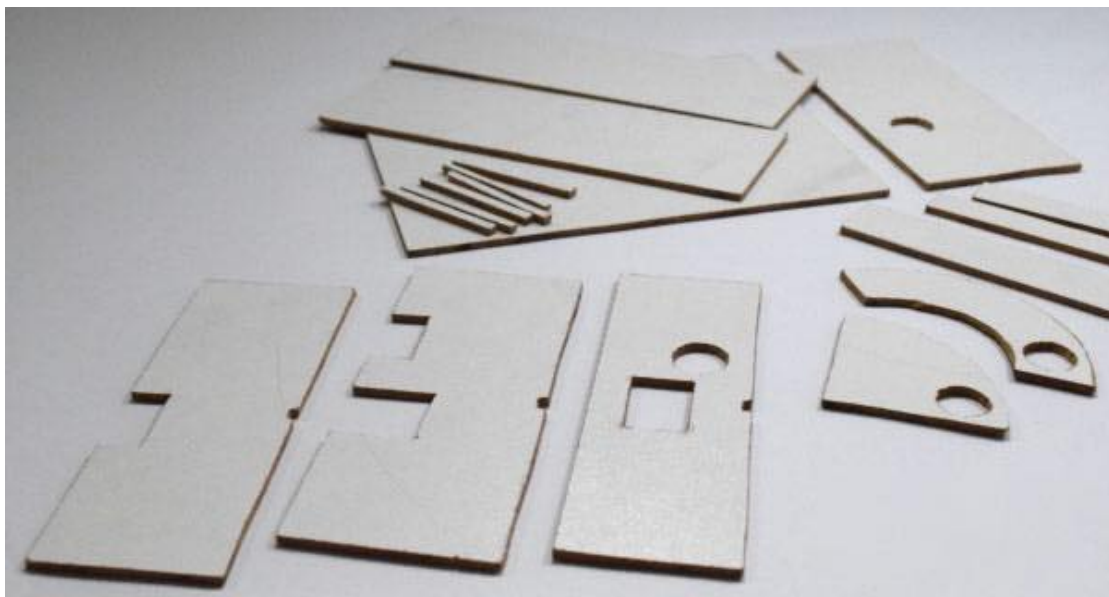
Οι διαλογείς χρώματος χρησιμοποιούνται για τη βιομηχανία επεξεργασίας τροφίμων, όπως ο καφές, τα καρύδια και οι ελαιούχες καλλιέργειες. Ο στόχος είναι ο διαχωρισμός των αντικειμένων που έχουν αποχρωματιστεί, τοξικά (όπως η ερυσιγγέρα), όχι τόσο ώριμα όσο απαιτείται, ή ακόμα με το κύτος μετά την αποσύνθεση όπως οι ηλιόσποροι. Σε σύγκριση με τη χειρωνακτική διαλογή, τα μηχανήματα εξοικονομούν εργασία και χρόνο, έχουν μεγαλύτερη απόδοση και έχουν χαμηλότερο κόστος επεξεργασίας. Οι διεργασίες αυξήθηκαν με τη χρήση νέων τεχνολογιών CCD και τώρα φτάνουν τα 100 t / h.

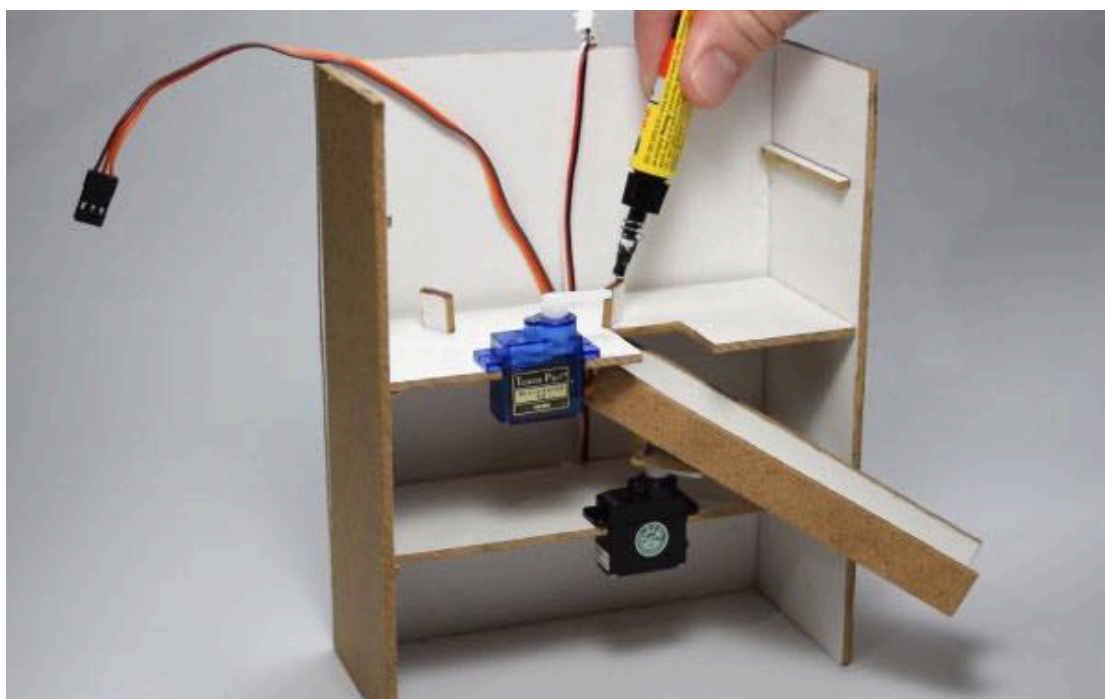
3.2 Λειτουργία

Το πρόγραμμα διαλογής χρωμάτων καταδεικνύει τις λειτουργίες των μηχανών διαλογής που χρησιμοποιούνται συνήθως από τις βιομηχανίες. Το έργο αυτό διαθέτει πίνακα Arduino και αισθητήρα χρώματος που μπορεί να οργανώσει αντικείμενα με βάση το χρώμα. Η φυσική κατασκευή είναι σχεδιασμένη σε 3D εφαρμογή που ταιριάζει ακριβώς στη λειτουργία.

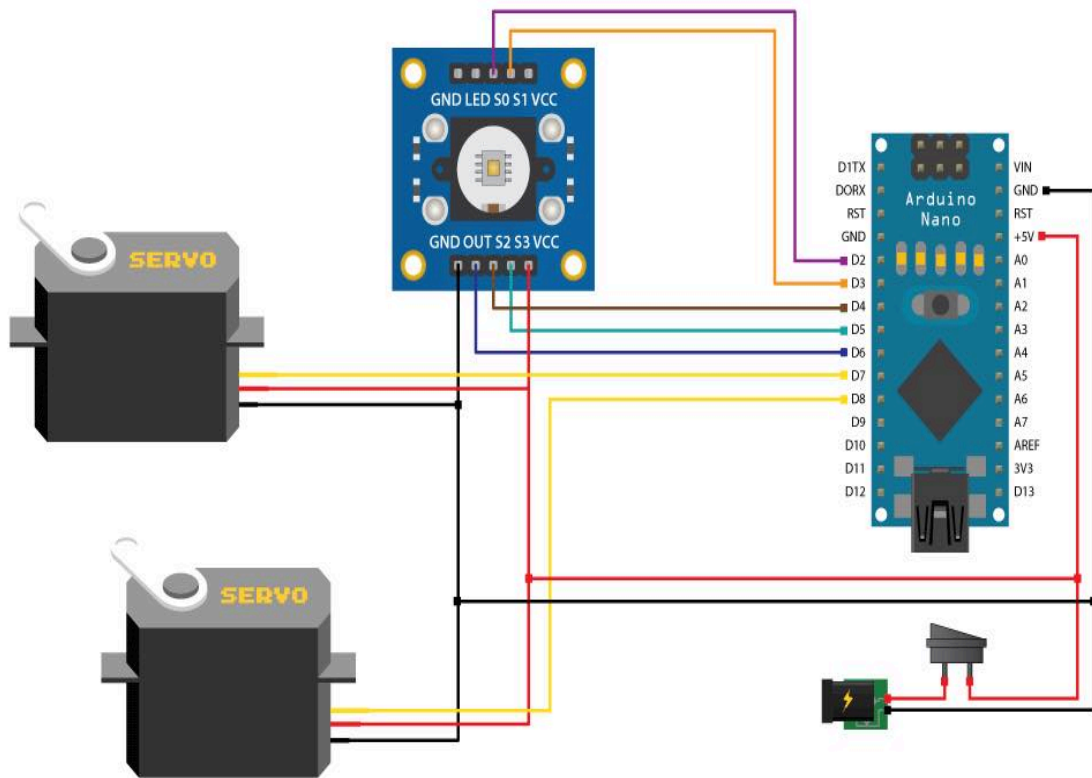


Η φυσική κατασκευή είναι κατασκευασμένη από ίνες προιονισμένη και κολλημένη σε σχέση με το 3D σχέδιο.



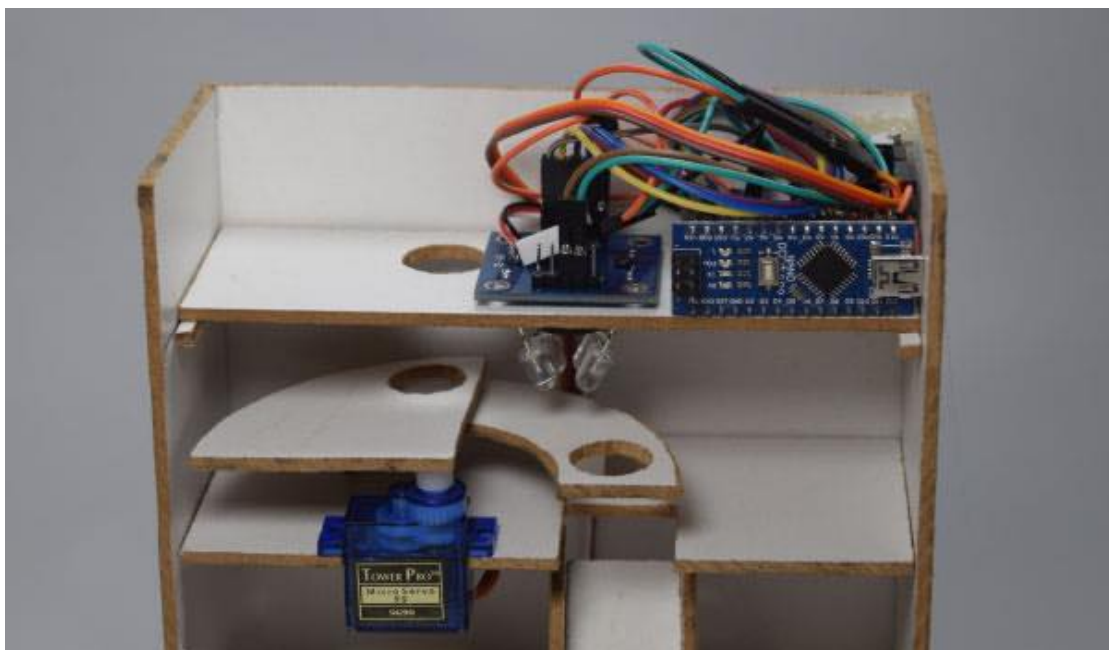


Το Arduino nano λειτουργεί ως ο κύριος ελεγκτής στον οποίο συνδέονται τα εξαρτήματα όπως ο αισθητήρας χρώματος και οι σερβοκινητήρες.



Σχηματικό συνδέσεων

Ανάλυση των δεδομένων που λαμβάνονται από τον αισθητήρα χρώματος και στη συνέχεια γίνεται μετάδοση εντολής στους σερβοκινητήρες. Ο αισθητήρας χρώματος είναι τοποθετημένος στραμμένος προς τα κάτω, καθώς έλεγξε το χρώμα του αντικειμένου και αυτός ο έλεγχος αντιστοιχεί σε μια χρωματική συχνότητα με μια προσαρμοσμένη συνάρτηση που συμβάλλει στην αναγνώριση του χρώματος του αντικειμένου.



Ο πρώτος σερβοκινητήρας εξασφαλίζει ότι ένα κομμάτι αντικειμένου παραδίδεται μετά την αναγνώριση του χρώματος του αντικειμένου, το οποίο μετακινεί το αντικείμενο από την πηγή στον αισθητήρα χρώματος και έπειτα στην οπή πτώσης με μια λαβή πλατφόρμας ολισθητήρα από έναν σέρβο πυθμένα. Αυτή η πλατφόρμα ολίσθησης ρυθμίζεται έτσι ώστε να παραδοθεί το αντικείμενο στο δοχείο ομάδας(κουτί) όπου αποθηκεύεται το έγχρωμο αντικείμενο.



Αυτό το πρόγραμμα διαλογής χρωμάτων Arduino εφαρμόζεται στην οργάνωση μικρών αντικειμένων λαμβάνοντας υπόψη τις δυνατότητες του υλικού που οδηγεί αυτό το έργο. Εκτός από την οργάνωση των μικρών αντικειμένων που βασίζονται στο χρώμα, είναι επίσης μια καλή αναφορά για την ανάπτυξη μεγαλύτερου συστήματος που είναι σε θέση να ταξινομήσει μεγαλύτερα αντικείμενα ειδικά στις βιομηχανίες.

Σερβοκινητήρες



Οι σερβοκινητήρες που χρησιμοποιήθηκαν είναι οι **SG90 Mini Gear Micro Servo 9g**. Έχει ταχύτητα λειτουργίας (4,8V χωρίς φορτίο): 0,12 δευτερόλεπτα / 60 μοίρες και η τάση λειτουργίας του είναι (3.0-7.2V).

Επιπλέον χαρακτηριστικά:

- Υψηλή ποιότητα και απόδοση υψηλού κόστους.
- Καλώδιο σύνδεσης 245 mm.
- Μέγιστη ροπή 1,6 Kg.
- Μικρό μέγεθος και ελαφρύ.

3.3 Κώδικας προγράμματος

Περιγραφή του κώδικα:

Όσον αφορά τον κώδικα του Arduino, πρέπει λοιπόν να συμπεριλάβουμε τη βιβλιοθήκη "Servo.h", να καθορίσουμε τις ακίδες στις οποίες θα συνδέσουμε τον αισθητήρα χρώματος, να δημιουργήσουμε τα σερβοαντικείμενα και να δηλώσουμε μερικές μεταβλητές που απαιτούνται για το πρόγραμμα. Στο τμήμα ρύθμισης πρέπει να ορίσουμε τις ακίδες ως Έξοδοι και Είσοδοι, να ορίσουμε την κλίμακα συχνότητας για τον αισθητήρα χρώματος, να καθορίσουμε τα σέρβο και να ξεκινήσουμε τη σειριακή επικοινωνία για την εκτύπωση των αποτελεσμάτων του χρώματος που διαβάζεται στη σειριακή οθόνη.

Στο τμήμα βρόχου, το πρόγραμμά μας ξεκινάει με τη μετακίνηση του πάνω σερβοκινητήρα στη θέση του φορτιστή κρουνούς. Σημειώστε ότι αυτή η τιμή των 115

ταιριάζει στην κατασκευή μου και στον σερβοκινητήρα μου, οπότε θα πρέπει να ρυθμίσουμε αυτήν την τιμή καθώς και τις ακόλουθες τιμές για τους σερβοκινητήρες ανάλογα με την κατασκευή.

Στη συνέχεια, χρησιμοποιώντας το βρόχο "για", θα περιστρέψουμε και θα φέρουμε τη βάση μετακίνησης προϊόντος στη θέση του αισθητήρα χρώματος. Χρησιμοποιούμε ένα βρόχο "για" έτσι ώστε να μπορούμε να ελέγξουμε την ταχύτητα της περιστροφής μεταβάλλοντας την ώρα καθυστέρησης στον βρόχο.

Στη συνέχεια, μετά από μια καθυστέρηση μισού δευτερόλεπτου, χρησιμοποιώντας τη λειτουργία custom-made, readColor (), θα διαβάσουμε το χρώμα της βάσης μετακίνησης χρωματιστού αντικειμένου. Εδώ είναι ο κώδικας της προσαρμοσμένης λειτουργίας. Χρησιμοποιώντας τους τέσσερις ακροδέκτες ελέγχου και τον ακροδέκτη εξόδου συχνότητας του αισθητήρα χρώματος διαβάζουμε το χρώμα της βάσης. Ο αισθητήρας διαβάζει 3 διαφορετικές τιμές για κάθε σκουλαρίκι, κόκκινο, πράσινο και μπλε και σύμφωνα με αυτές τις τιμές λέμε τι είναι το πραγματικό χρώμα

TCS230 Αισθητήρας χρώματος



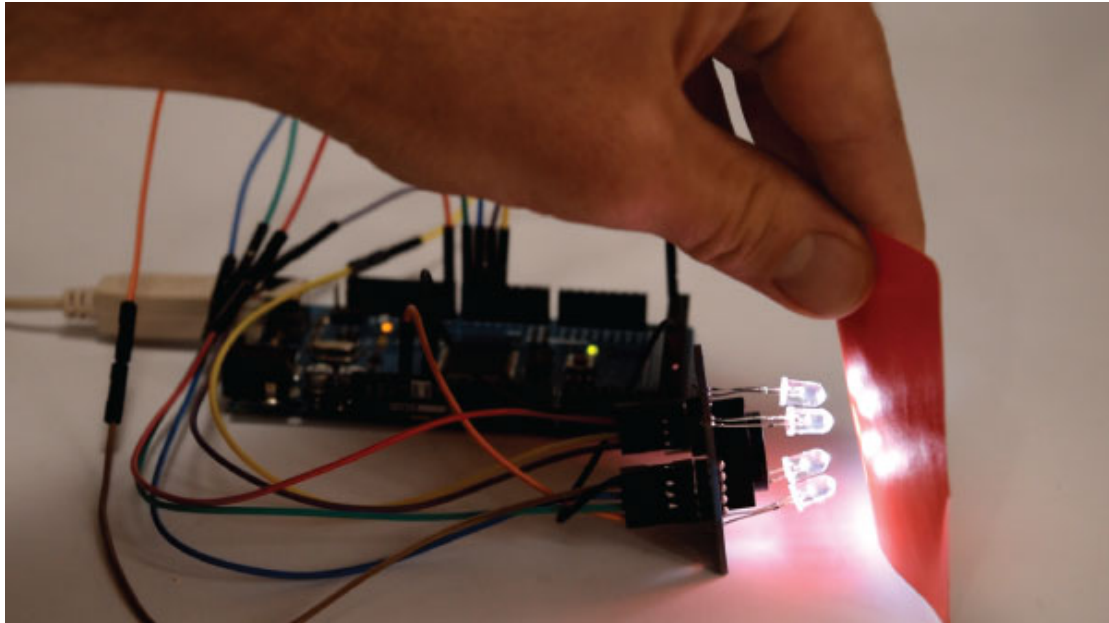
Κώδικας πηγής αισθητήρα χρωμάτων TCS230

Περιγραφή: Πρώτα πρέπει να ορίσουμε τις ακίδες στις οποίες συνδέεται ο αισθητήρας και να ορίσουμε μια μεταβλητή για την ανάγνωση της συχνότητας. Στο τμήμα εγκατάστασης πρέπει να ορίσουμε τις τέσσερις ακίδες ελέγχου ως έξοδοι και την έξοδο του αισθητήρα ως είσοδο Arduino. Εδώ πρέπει επίσης να ρυθμίσουμε την κλιμάκωση συχνότητας, για αυτό το παράδειγμα θα το ρυθμίσω στο 20% και θα ξεκινήσω τη σειριακή επικοινωνία για την εμφάνιση των αποτελεσμάτων στη σειριακή οθόνη.

Στο τμήμα βρόχου, θα ξεκινήσουμε με την ανάγνωση των κόκκινων διηθημένων φωτοδίοδων. Για το σκοπό αυτό, θα θέσουμε τους δύο ακροδέκτες ελέγχου S2 και S3 σε χαμηλό επίπεδο λογικής. Στη συνέχεια, χρησιμοποιώντας τη λειτουργία "pulseIn ()" θα διαβάσουμε τη συχνότητα εξόδου και θα την βάλουμε στη μεταβλητή "συχνότητα". Χρησιμοποιώντας τη λειτουργία Serial.print () θα εκτυπώσουμε το αποτέλεσμα στη

σειριακή οθόνη. Η ίδια διαδικασία ισχύει για τα άλλα δύο χρώματα, απλά πρέπει να ρυθμίσουμε τους ακροδέκτες ελέγχου για το κατάλληλο χρώμα.

```
1. #define S0 4
2. #define S1 5
3. #define S2 6
4. #define S3 7
5. #define sensorOut 8
6.
7. int frequency = 0;
8.
9. void setup() {
10. pinMode(S0, OUTPUT);
11. pinMode(S1, OUTPUT);
12. pinMode(S2, OUTPUT);
13. pinMode(S3, OUTPUT);
14. pinMode(sensorOut, INPUT);
15.
16. // Setting frequency-scaling to 20%
17. digitalWrite(S0,HIGH);
18. digitalWrite(S1,LOW);
19.
20. Serial.begin(9600);
21. }
22.
23. void loop() {
24. // Setting red filtered photodiodes to be read
25. digitalWrite(S2,LOW);
26. digitalWrite(S3,LOW);
27. // Reading the output frequency
28. frequency = pulseIn(sensorOut, LOW);
29. // Printing the value on the serial monitor
30. Serial.print("R= ");//printing name
31. Serial.print(frequency);//printing RED color frequency
32. Serial.print(" ");
33. delay(100);
34.
35. // Setting Green filtered photodiodes to be read
36. digitalWrite(S2,HIGH);
37. digitalWrite(S3,HIGH);
38. // Reading the output frequency
39. frequency = pulseIn(sensorOut, LOW);
40. // Printing the value on the serial monitor
41. Serial.print("G= ");//printing name
42. Serial.print(frequency);//printing RED color frequency
43. Serial.print(" ");
44. delay(100);
45.
46. // Setting Blue filtered photodiodes to be read
47. digitalWrite(S2,LOW);
48. digitalWrite(S3,HIGH);
49. // Reading the output frequency
50. frequency = pulseIn(sensorOut, LOW);
51. // Printing the value on the serial monitor
52. Serial.print("B= ");//printing name
```

Έτσι τώρα αν θέλουμε να αντιπροσωπεύσουμε τα ανιχνευόμενα χρώματα με το Μοντέλο RGB που έχει τιμές από 0 έως 255, θα χρησιμοποιήσουμε τη λειτουργία `map()` για να χαρτογραφήσουμε ή να μετατρέψουμε τις μετρήσεις στις τιμές από 0 έως 255.

1. `//Remaping the value of the frequency to the RGB Model of 0 to 255`
2. `frequency = map(frequency, 25,70,255,0);`

Η τιμή 70 θα αντιστοιχιστεί στο 0 και η τιμή θα είναι 25 έως 255. Η ίδια διαδικασία ισχύει και για τα δύο άλλα χρώματα.

Ακολουθεί ο τελικός κώδικας πηγής για αυτό το παράδειγμα:

1. `#define S0 4`
2. `#define S1 5`
3. `#define S2 6`
4. `#define S3 7`
5. `#define sensorOut 8`
- 6.
7. `int frequency = 0;`
- 8.
9. `void setup() {`
10. `pinMode(S0, OUTPUT);`
11. `pinMode(S1, OUTPUT);`
12. `pinMode(S2, OUTPUT);`
13. `pinMode(S3, OUTPUT);`
14. `pinMode(sensorOut, INPUT);`
- 15.
16. `// Setting frequency-scaling to 20%`
17. `digitalWrite(S0,HIGH);`
18. `digitalWrite(S1,LOW);`
- 19.
20. `Serial.begin(9600);`
21. `}`

```

22.
23. void loop() {
24. // Setting red filtered photodiodes to be read
25. digitalWrite(S2,LOW);
26. digitalWrite(S3,LOW);
27. // Reading the output frequency
28. frequency = pulseIn(sensorOut, LOW);
29. //Remaping the value of the frequency to the RGB Model of 0 to 255
30. frequency = map(frequency, 25,72,255,0);
31. // Printing the value on the serial monitor
32. Serial.print("R= ");//printing name
33. Serial.print(frequency);//printing RED color frequency
34. Serial.print(" ");
35. delay(100);
36.
37. // Setting Green filtered photodiodes to be read
38. digitalWrite(S2,HIGH);
39. digitalWrite(S3,HIGH);
40. // Reading the output frequency
41. frequency = pulseIn(sensorOut, LOW);
42. //Remaping the value of the frequency to the RGB Model of 0 to 255
43. frequency = map(frequency, 30,90,255,0);
44. // Printing the value on the serial monitor
45. Serial.print("G= ");//printing name
46. Serial.print(frequency);//printing RED color frequency
47. Serial.print(" ");
48. delay(100);
49.
50. // Setting Blue filtered photodiodes to be read
51. digitalWrite(S2,LOW);
52. digitalWrite(S3,HIGH);
53. // Reading the output frequency
54. frequency = pulseIn(sensorOut, LOW);
55. //Remaping the value of the frequency to the RGB Model of 0 to 255
56. frequency = map(frequency, 25,70,255,0);
57. // Printing the value on the serial monitor
58. Serial.print("B= ");//printing name
59. Serial.print(frequency);//printing RED color frequency
60. Serial.println(" ");
61. delay(100);
62. }

```

Παρακάτω αναγράφεται ο κώδικας του Arduino nano ολοκληρωμένος

```

1. #include <Servo.h>
2.
3. #define S0 2
4. #define S1 3
5. #define S2 4
6. #define S3 5
7. #define sensorOut 6
8.

```

```
9. Servo topServo;
10. Servo bottomServo;
11.
12. int frequency = 0;
13. int color=0;
14.
15. void setup() {
16. pinMode(S0, OUTPUT);
17. pinMode(S1, OUTPUT);
18. pinMode(S2, OUTPUT);
19. pinMode(S3, OUTPUT);
20. pinMode(sensorOut, INPUT);
21.
22. // Setting frequency-scaling to 20%
23. digitalWrite(S0, HIGH);
24. digitalWrite(S1, LOW);
25.
26. topServo.attach(7);
27. bottomServo.attach(8);
28.
29. Serial.begin(9600);
30. }
31.
32. void loop() {
33.
34. topServo.write(115);
35. delay(500);
36.
37. for(int i = 115; i > 65; i--) {
38. topServo.write(i);
39. delay(2);
40. }
41. delay(500);
42.
43. color = readColor();
44. delay(10);
45.
46. switch (color) {
47. case 1:
48. bottomServo.write(50);
49. break;
50.
51. case 2:
52. bottomServo.write(75);
53. break;
54.
55. case 3:
56. bottomServo.write(100);
57. break;
58.
59. case 4:
60. bottomServo.write(125);
61. break;
62.
63. case 5:
64. bottomServo.write(150);
65. break;
66.
```

```

67. case 6:
68. bottomServo.write(175);
69. break;
70.
71. case 0:
72. break;
73. }
74. delay(300);
75.
76. for(int i = 65; i > 29; i--) {
77. topServo.write(i);
78. delay(2);
79. }
80. delay(200);
81.
82. for(int i = 29; i < 115; i++) {
83. topServo.write(i);
84. delay(2);
85. }
86. color=0;
87. }
88.
89. // Custom Function - readColor()
90. int readColor() {
91. // Setting red filtered photodiodes to be read
92. digitalWrite(S2, LOW);
93. digitalWrite(S3, LOW);
94. // Reading the output frequency
95. frequency = pulseIn(sensorOut, LOW);
96. int R = frequency;
97. // Printing the value on the serial monitor
98. Serial.print("R= "); //printing name
99. Serial.print(frequency); //printing RED color frequency
100. Serial.print(" ");
101. delay(50);
102.
103. // Setting Green filtered photodiodes to be read
104. digitalWrite(S2, HIGH);
105. digitalWrite(S3, HIGH);
106. // Reading the output frequency
107. frequency = pulseIn(sensorOut, LOW);
108. int G = frequency;
109. // Printing the value on the serial monitor
110. Serial.print("G= "); //printing name
111. Serial.print(frequency); //printing RED color frequency
112. Serial.print(" ");
113. delay(50);
114.
115. // Setting Blue filtered photodiodes to be read
116. digitalWrite(S2, LOW);
117. digitalWrite(S3, HIGH);
118. // Reading the output frequency
119. frequency = pulseIn(sensorOut, LOW);
120. int B = frequency;
121. // Printing the value on the serial monitor
122. Serial.print("B= "); //printing name
123. Serial.print(frequency); //printing RED color frequency
124. Serial.println(" ");

```

```

125. delay(50);
126.
127. if(R<45 & R>32 & G<65 & G>55){
128. color = 1; // Red
129. }
130. if(G<55 & G>43 & B<47 & B>35){
131. color = 2; // Orange
132. }
133. if(R<53 & R>40 & G<53 & G>40){
134. color = 3; // Green
135. }
136. if(R<38 & R>24 & G<44 & G>30){
137. color = 4; // Yellow
138. }
139. if(R<56 & R>46 & G<65 & G>55){
140. color = 5; // Brown
141. }
142. if (G<58 & G>45 & B<40 & B>26){
143. color = 6; // Blue
144. }
145. return color;
}

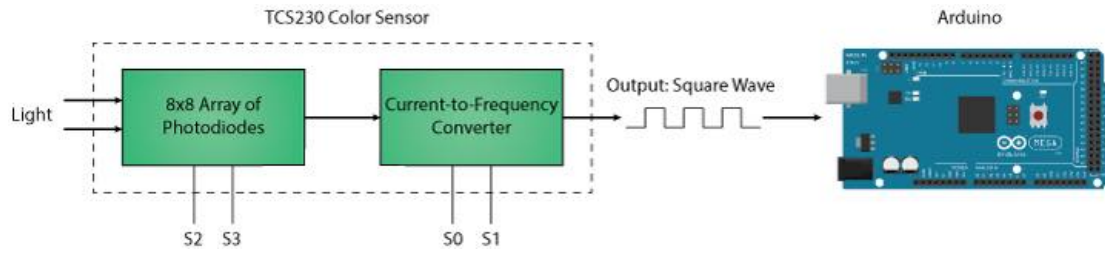
```

Εδώ είναι οι τιμές RGB που πήρα από τον αισθητήρα για κάθε σκούος. Σημειώστε ότι αυτές οι τιμές μπορεί να διαφέρουν επειδή οι αισθητήρες δεν είναι πάντα ακριβείς. Επομένως, χρησιμοποιώντας αυτές τις δηλώσεις "if" επιτρέπουμε στον αισθητήρα σφάλμα περίπου + -5 της δοκιμασμένης τιμής για το συγκεκριμένο χρώμα. Έτσι, για παράδειγμα, αν έχουμε μια κόκκινη σφαίρα, η πρώτη εντολή "if" θα είναι αληθής και η μεταβλητή "χρώμα" θα πάρει την τιμή 1. Έτσι κάνει αυτό η συνήθης λειτουργία readColor () και στη συνέχεια χρησιμοποιώντας μια " Δήλωση περιστρέφουμε το κάτω σέρβο στη συγκεκριμένη θέση. Στο τέλος περιστρέφουμε περαιτέρω τον ανώτερο σερβοκινητήρα μέχρι να πέσει η κασέτα στην σιδηροτροχιά οδηγού και να την στείλει και πάλι πίσω στην αρχική θέση, ώστε να επαναληφθεί η διαδικασία.

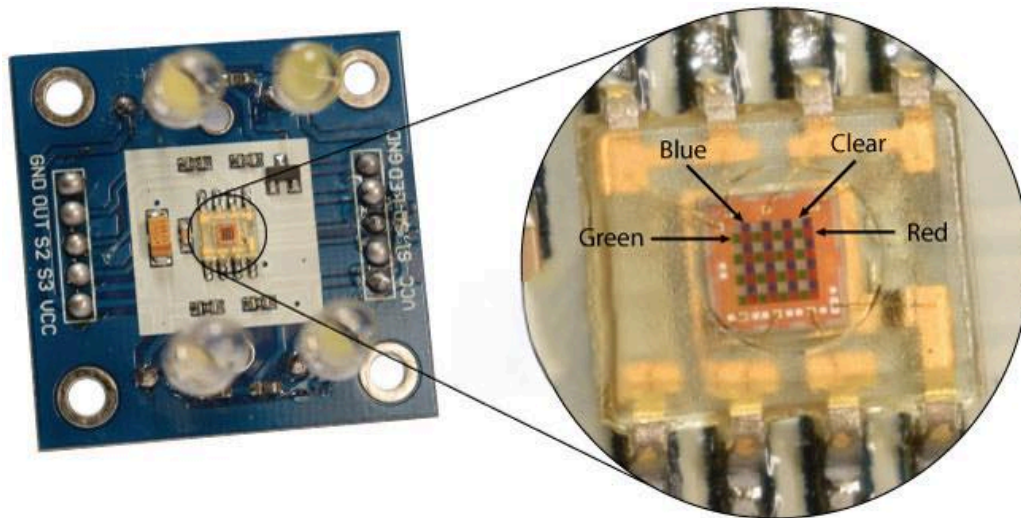
3.4 Πώς λειτουργεί ο αισθητήρας χρώματος

Πώς λειτουργεί ο αισθητήρας χρώματος TCS230

Το TCS230 αισθάνεται το έγχρωμο φως με τη βοήθεια μιας σειράς 8 x 8 φωτοδίοδων. Στη συνέχεια, χρησιμοποιώντας μετατροπέα ρεύματος προς συχνότητα, οι μετρήσεις από τις φωτοδίοδοι μετατρέπονται σε τετραγωνικό κύμα με συχνότητα ευθέως ανάλογη προς την ένταση του φωτός. Τέλος, με τη χρήση του πίνακα Arduino μπορούμε να διαβάσουμε την παραγωγή τετραγωνικών κυμάτων και να πάρουμε τα αποτελέσματα για το χρώμα.



Αν κοιτάξουμε προσεχώς τον αισθητήρα μπορούμε να δούμε πώς ανιχνεύει διάφορα χρώματα. Οι φωτοδιόδοι έχουν τρία διαφορετικά έγχρωμα φίλτρα. Δεκαέξι από αυτά έχουν κόκκινα φίλτρα, άλλα 16 έχουν πράσινα φίλτρα, άλλα 16 έχουν μπλε φίλτρα και οι άλλες 16 φωτοδιόδους είναι καθαρές χωρίς φίλτρα.



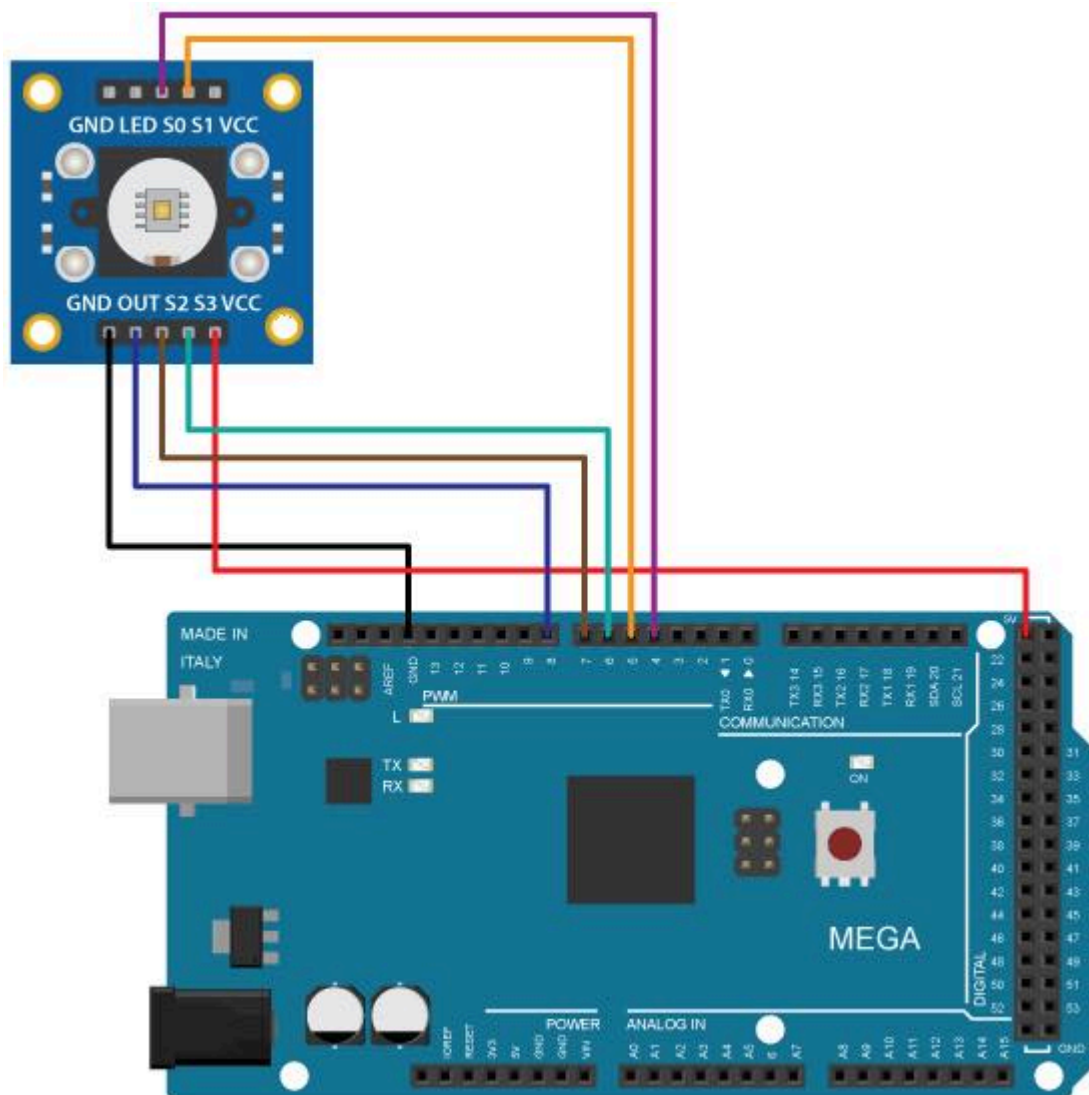
Κάθε 16 φωτοδιόδοι συνδέονται παράλληλα, έτσι χρησιμοποιώντας τους δύο ακροδέκτες ελέγχου S2 και S3 μπορούμε να επιλέξουμε ποιες από αυτές θα διαβαστούν. Έτσι, για παράδειγμα, εάν θέλουμε να ανιχνεύσουμε το κόκκινο χρώμα, μπορούμε απλώς να χρησιμοποιήσουμε τις 16 κόκκινες διηθημένες φωτοδιόδους θέτοντας τους δύο ακροδέκτες σε χαμηλό λογικό επίπεδο σύμφωνα με τον πίνακα.

S0	S1	Output Frequency Scaling
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

S2	S3	Photodiode Type
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Ο αισθητήρας διαθέτει δύο ακόμη ακροδέκτες ελέγχου, S0 και S1, οι οποίοι χρησιμοποιούνται για την κλιμάκωση της συχνότητας εξόδου. Η συχνότητα μπορεί να κλιμακωθεί σε τρεις διαφορετικές προκαθορισμένες τιμές 100%, 20% ή 2%. Αυτή η λειτουργία κλιμάκωσης συχνότητας επιτρέπει την βελτιστοποίηση της εξόδου του αισθητήρα για διάφορους μετρητές συχνότητας ή μικροελεγκτές.

Τώρα είμαστε έτοιμοι να προχωρήσουμε και να συνδέσουμε τον αισθητήρα TCS230 με τον πίνακα Arduino.



Τα σχήματα κυκλώματος και οι συνδεσμολογίες

Βιβλιογραφία

1. <http://es.uop.gr/esmet/images/domi-ptixiakis.pdf>
2. https://www.researchgate.net/publication/270878911_DEVELOPMENT_OF_AN_AUTO_MATIC_COLOR_SORTING_MACHINE_ON_BELT_CONVEYER
3. <https://www.eeweb.com/featured-projects/arduino-color-sorter-project>
4. https://www.researchgate.net/publication/294878316_Automatic_Color_Sorting_Machine_Using_TCS230_Color_Sensor_And_PIC_Microcontroller
5. <https://howtomechatronics.com/projects/arduino-color-sorter-project/>
6. <http://www.athinodromio.gr/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%BF-%CE%B1%CF%85%CF%84%CE%BF%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CF%8C%CF%82/#.Xjw3LTlzbIU>
7. http://imm.demokritos.gr/platon/AEOAAUAC_OOIOO_AOOIIAEOEIIIOO/aeoaaauac_ooio_o_aooiaoeoiiioo.html
8. <http://ikaros.teipir.gr/phyche/Subjects/Routoulas/AutomatismoiVIKTE/ViomixanikosAutomatismos.pdf>
9. <http://okeanis.lib.puas.gr/xmlui/bitstream/handle/123456789/4101/%CE%A8%CE%A9%CE%9C%CE%99%CE%91%CE%94%CE%97%CE%A3%20%CE%A1%CE%91%CE%A6%CE%91%CE%97%CE%9B-%CE%9D%CE%99%CE%9A%CE%9F%CE%9B%CE%91%CE%9F%CE%A3%20%CE%91%CE%9C.44544.pdf?sequence=1&isAllowed=y>
10. https://en.wikipedia.org/wiki/Colour_sorter
11. http://okeanis.lib2.uniwa.gr/xmlui/bitstream/handle/123456789/2066/aut_201400651.pdf?sequence=1&isAllowed=y
12. <http://okeanis.lib2.uniwa.gr/xmlui/bitstream/handle/123456789/4895/%CF%80%CE%B%CE%B1%CE%BA%CE%B1%CF%83.pdf?sequence=3&isAllowed=y>
13. <http://ir.lib.uth.gr/bitstream/handle/11615/47130/16955.pdf?sequence=1>
14. <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>
15. <https://www.rs-online.com/designspark/basics-of-arduino-nano>
16. <http://www.teilar.gr/dbData/ProfAnn/profann-968a4b3c.pdf>
17. https://en.wikipedia.org/wiki/Arduino_Uno
18. <https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE>
19. <http://www.electronics.teipir.gr/personalpages/papageorgas/download/2/shmeiwiseis/p>
20. https://en.wikipedia.org/wiki/Arduino_Uno
21. <https://store.arduino.cc/arduino-motor-shield-rev3>
22. https://en.wikipedia.org/wiki/Ultrasonic_transducer
23. <https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CF%84%CF%>
24. <http://users.sch.gr/manpoul/docs/arduino/ProgrammingArduino.pdf>