

Συσκευή Αναγνώριση Χρώματος

Σύνδεση LabView - Arduino

Κοτσίνης Δημήτριος
UP1059482-40 ΈΤΟΣ

Περιεχόμενα

Εισαγωγή	2
Υλικό	3
Arduino Elegoo Uno R3.....	3
MicroServo WG-SG90.....	3
ColourSensor TCS3200.....	4
Περιγραφή της συσκευής.....	6
Σχεδιασμός στο LabView και Arduino	8
LINX.....	8
Χρήσιμα VI.....	9
Angle2PMW.vi	9
Colour.vi.....	11
Distance.vi	12
Remap.vi	13
Pins	14
Καταστάσεις(Programm.vi)	14
Front Pannel	16
Αποτελέσματα	17
Προβλήματα κατά την κατασκευή	19

Εισαγωγή

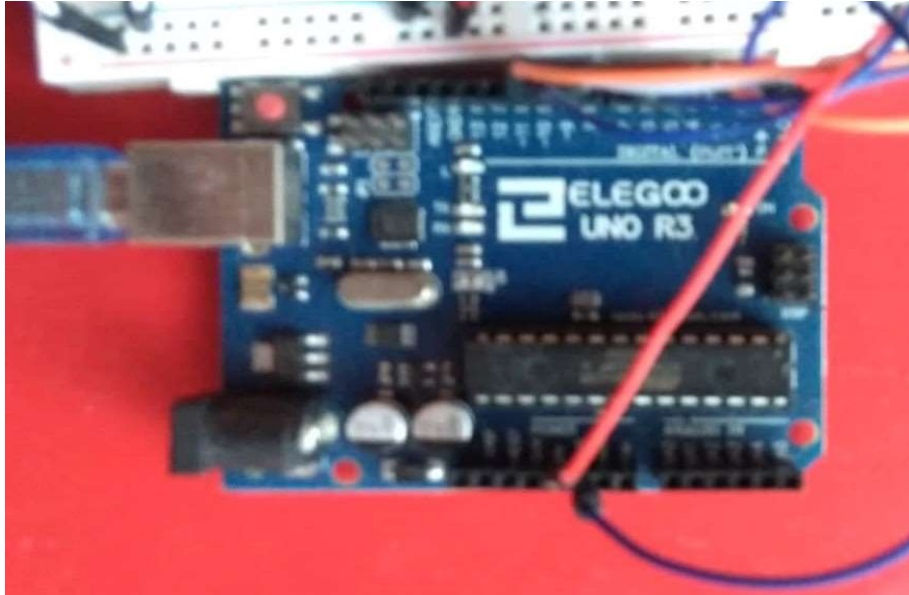
Στην παρούσα εργασία, περιγράφεται μία συσκευή η οποία αναγνωρίζει το χρώμα ορισμένων αντικειμένων, κόκκινο, πράσινο και μπλε. Η λειτουργία αυτή θα επιτευχθεί με την βοήθεια του προγράμματος LabView και του μικροεπεξεργαστή Arduino τα οποία θα συνδέονται με την βιβλιοθήκη LINX.

Το πρώτος μέρος της αναφοράς θα αφορά στα υλικά που χρησιμοποιήθηκαν για την υλοποίησή της. Το δεύτερο μέρος θα αναφέρεται στον σχεδιασμό της. Και τέλος στο τρίτο μέρος θα περιγράφει το πρόγραμμα στο LabView.

Υλικό

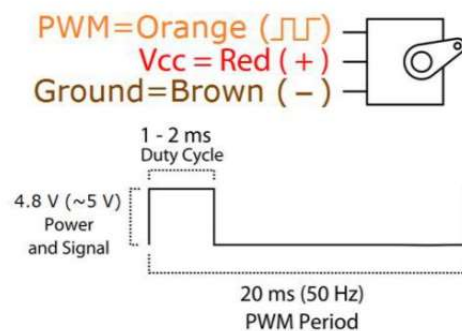
Για το μηχάνημα αυτό θα χρησιμοποιήσουμε έναν μικροεπεξεργαστή Arduino Elegoo Uno R3, έναν MicroServo WG-SG90 και ένας ColourSensor TCS3200.

Arduino Elegoo Uno R3



Το Arduino αυτό περιέχει 6 pins για αναλογικό σήμα και 13 για ψηφιακό σήμα και έναν μικροεπεξεργαστή ATMEGA328P. Εμείς θα χρησιμοποιήσουμε περισσότερο τα 6 pins που παίρνουν εισόδους και εξόδους σήματα PWM(Pulse Width Modulation).

MicroServo WG-SG90



Ο μικρό-σέρβο κινητήρας αυτός στρέφεται όταν έχει είσοδο σήμα PWM. Ανάλογα με την διάρκεια του παλμού στρέφεται στην κατάλληλη γωνία. Στον παρακάτω πίνακα φαίνεται η διάρκεια του παλμού με την αντίστοιχη γωνία.

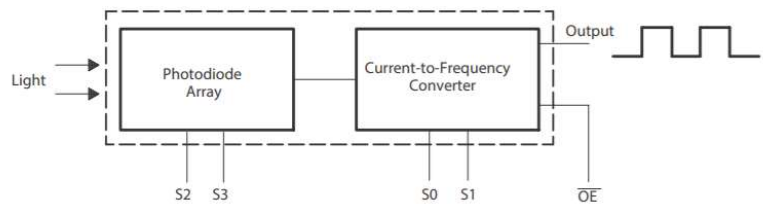
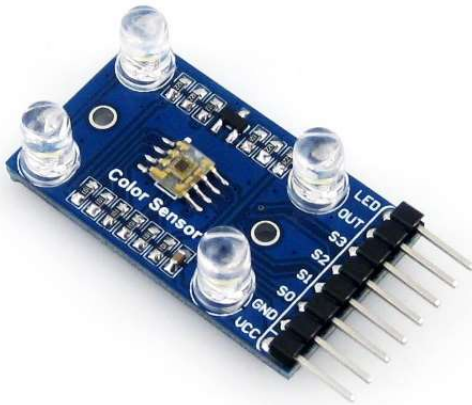
PMW(ms)	Γωνία
~1	-90°
~1.5	0°
~2	90°

Ο τύπος που συνδέει αυτές τις δύο μεταβλητές είναι:

$$PMW(Angle) = U16 \left[\frac{(Angle + 130.5)}{0.09} \right] (\mu s) \quad (1)$$

Τον τύπο (1) θα τον ορίσουμε αργότερα σε ένα VI.

ColourSensor TCS3200



Ο σένσορας παίρνει ως είσοδο το φως του περιβάλλοντος, μέσω ενός συνόλου φωτοδιόδων που έχει διάταξη πίνακα 8x8. Τα 16 από αυτά έχουν φίλτρα για το κόκκινο χρώμα, 16 επίσης για το πράσινο χρώμα, άλλα 16 για το μπλε και άλλα 16 που δεν έχουν φίλτρα. Το χρώμα που θα ανιχνευση καθορίζεται από τις εισόδους S2 και S3.

S2	S3	Χρώμα
L	L	Κόκκινο
L	H	Μπλε
H	L	Καθαρό
H	H	Πράσινο

Στην συνέχεια το σήμα από τις φωτοδιόδους μετατρέπεται σε σήμα PWM 50% duty cycle με μία συγκεκριμένη συχνότητα που σχετίζεται με το μήκος κύματος του αντίστοιχου χρώματος. Η κλίμακα της συχνότητας καθορίζεται από τις εισόδους S0 και S1.

S0	S1	Output Frequency Scaling (fo)
L	L	Κλειστό
L	H	2%
H	L	20%
H	H	100%

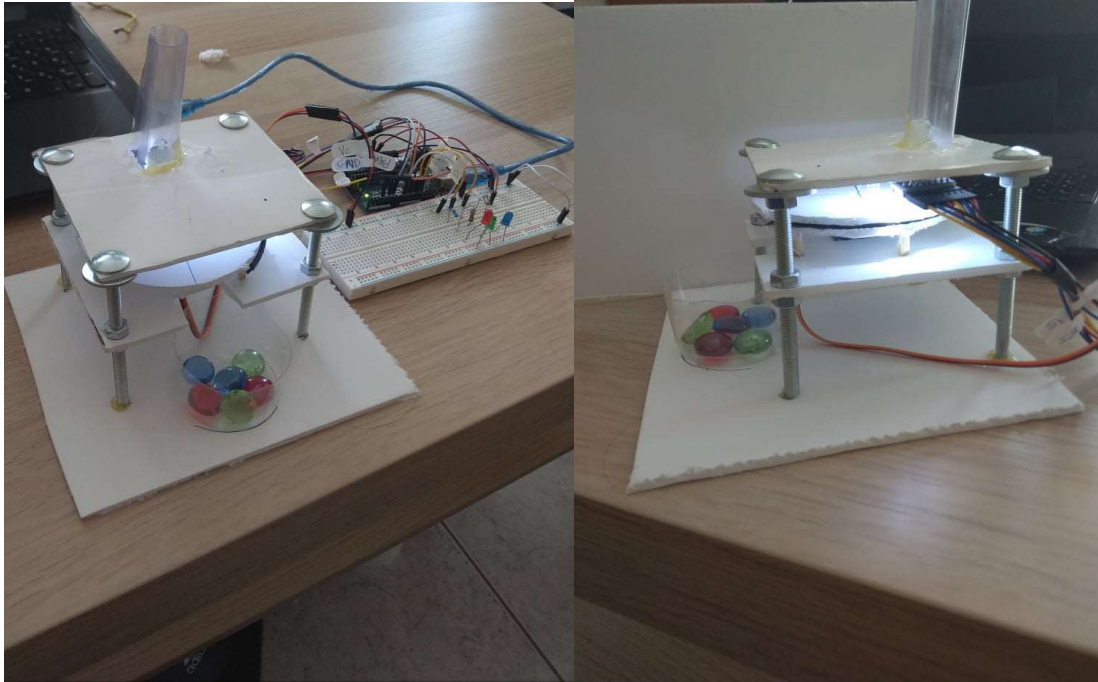
Θα χρησιμοποιήσουμε το σύστημα RGB για την αναγνώριση των χρωμάτων. Για να το κάνουμε αυτό θα πρέπει με τον παρακάτω τύπο να μετατρέψουμε την συχνότητα στην έξοδο σε τιμές του συστήματος αυτού. Αυτό το πραγματοποιούμε βάζοντας μπροστά στον αισθητήρα, στην κατάλληλη θέση και απόσταση από αυτόν, ένα λευκό αντικείμενο και ένα μαύρο για να βρούμε τι τιμές των ορίων των τριών χρωμάτων.

$$MAP(Colour, f_o) = \begin{cases} 255(1 - \frac{MinValueColour - f_o}{MinValueColour - MaxValueColour}), & MinValueColour \leq f_o \leq MaxValueColour \\ 0, & MinValueColour \geq f_o \\ 255, & MaxValueColour \geq f_o \end{cases} \quad (2)$$

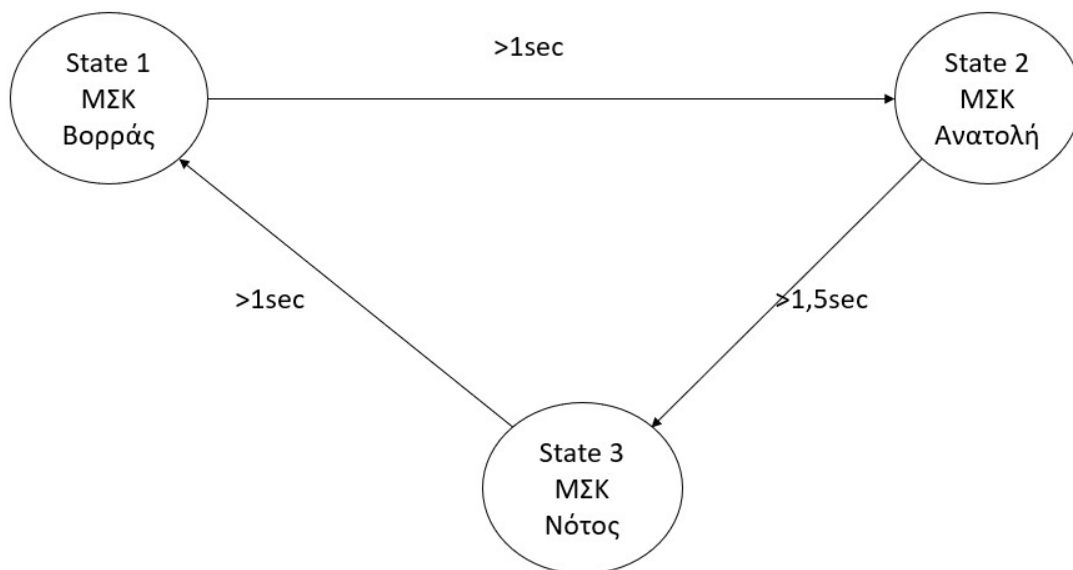
Όπου το f_o είναι συχνότητα εξόδου, $MinValueColour$ είναι το κάτω όριο και $MaxValueColour$ το πάνω όριο.

Περιγραφή της συσκευής

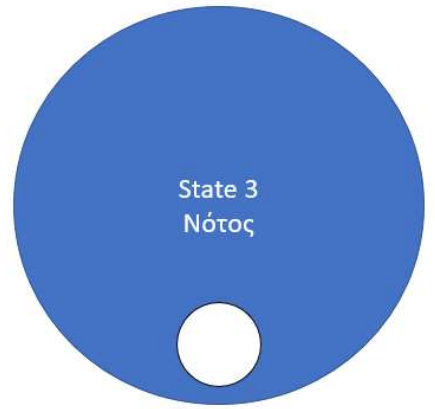
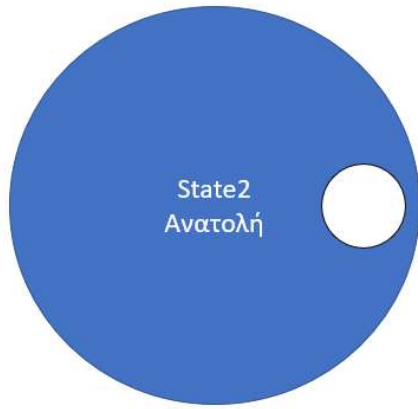
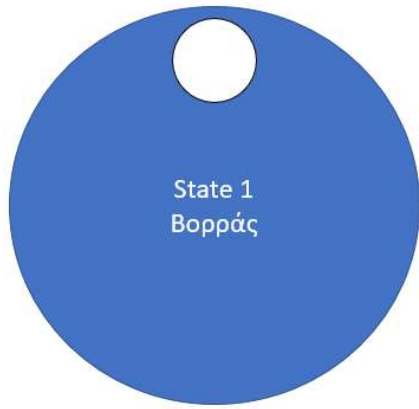
Η συσκευή που κατασκευάζουμε είναι:



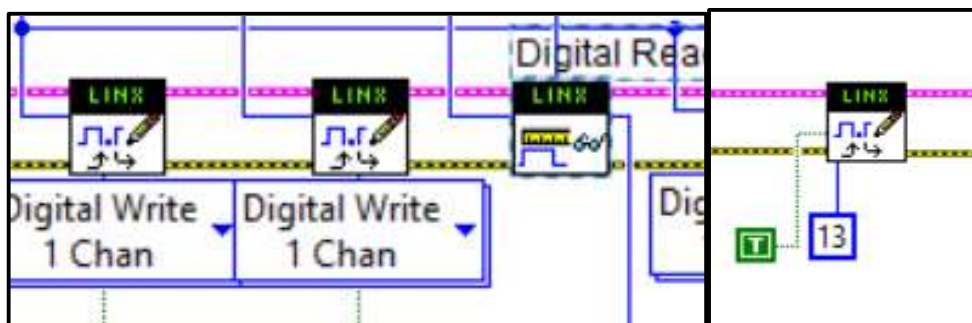
Οι καταστάσεις που περιέχει είναι τρεις:



Οι καταστάσεις αυτές σχετίζονται με την γωνία του μικρό-σέρβο κινητήρα. Στην κατάσταση 1 ο σέρβο κινητήρας, η τρύπα που εισχωρούν οι χάντρες με ένα συγκεκριμένο χρώμα (κόκκινο, πράσινο, μπλε) για αναγνώριση, βρίσκεται προ τα πάνω. Μετά από ένα δευτερόλεπτο, στην κατάσταση 2 είναι στις προς τα δεξιά (ανατολή) και γίνεται η ανίχνευση του χρώματος με τον αισθητήρα. Όταν εκτελέσει την μετατροπή του στο σύστημα RGB και το ανάμμα του LED που δείχνει το χρώμα του αντικειμένου (μετά από 1.5sec), ο σέρβο κινητήρας πηγαίνει στην τελευταία κατάσταση όπου η τρύπα είναι προς τα κάτω (νότος) και επιστρέφει στην κατάσταση 1 που συνεχίζει την ίδια διαδικασία μέχρι να τελειώσουν οι χάντρες.



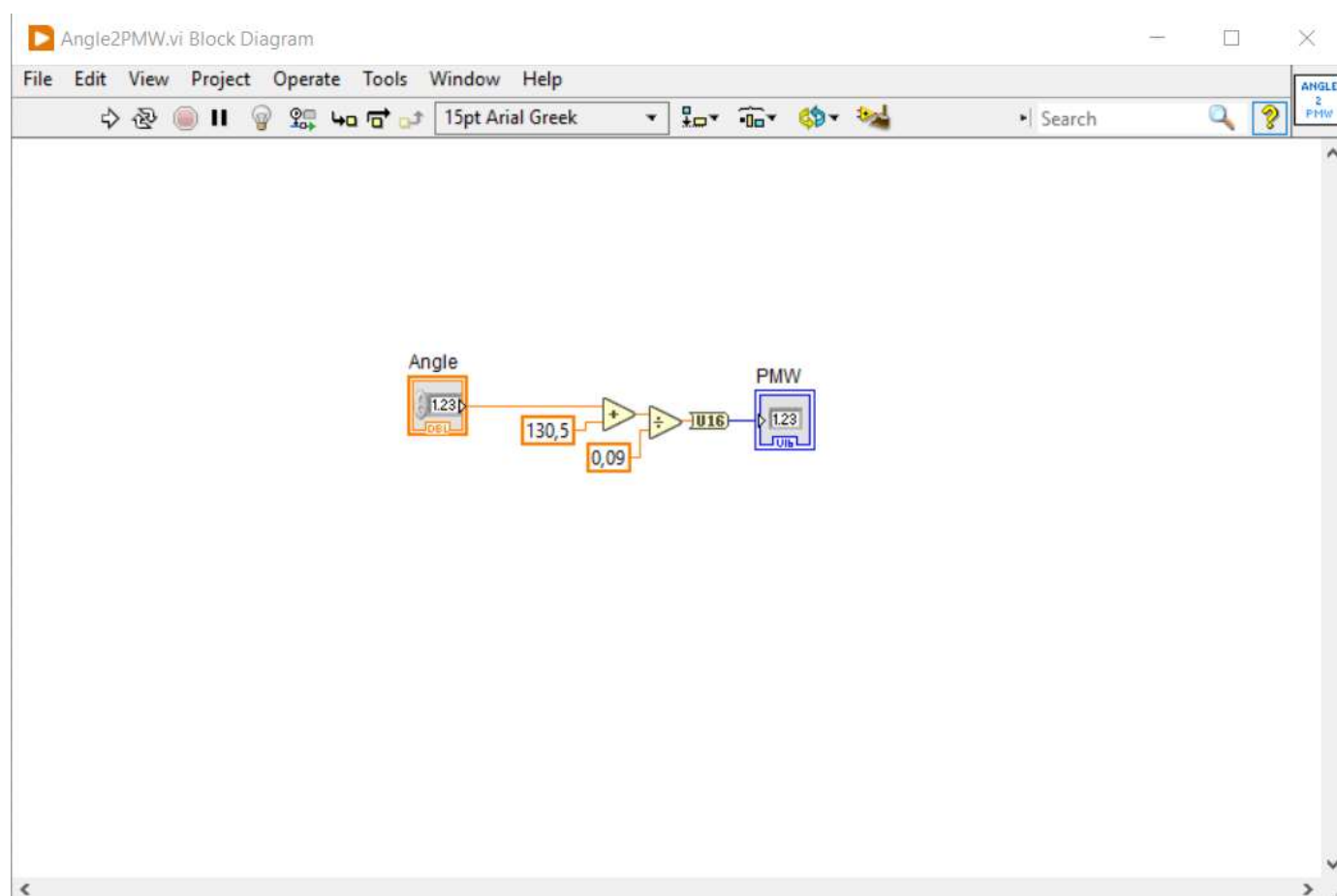
Για την ανίχνευση των χρωμάτων και για το άναμα του σωστού LED θα χρησιμοποιήσουμε τα Read και Write του LINX.

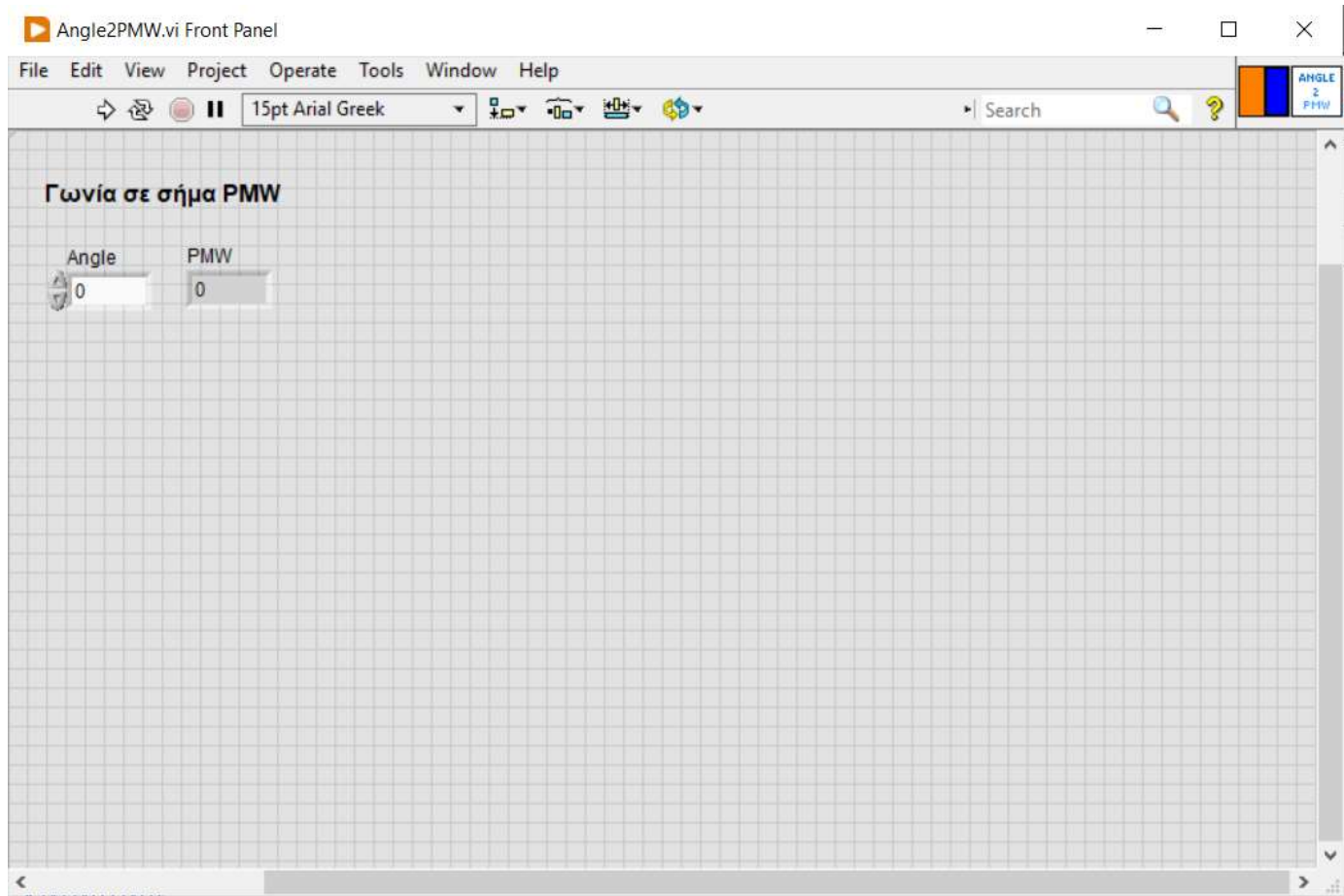


Χρήσιμα VI

Angle2PMW.vi

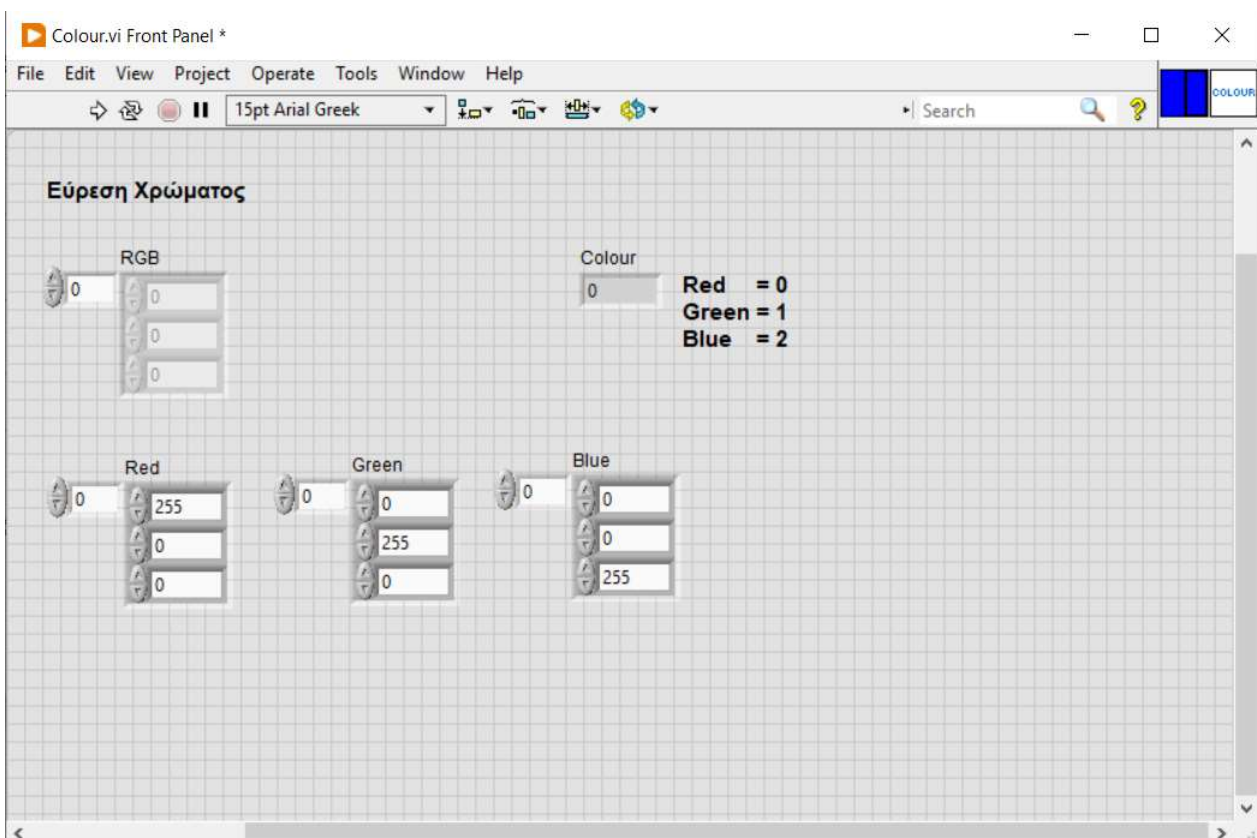
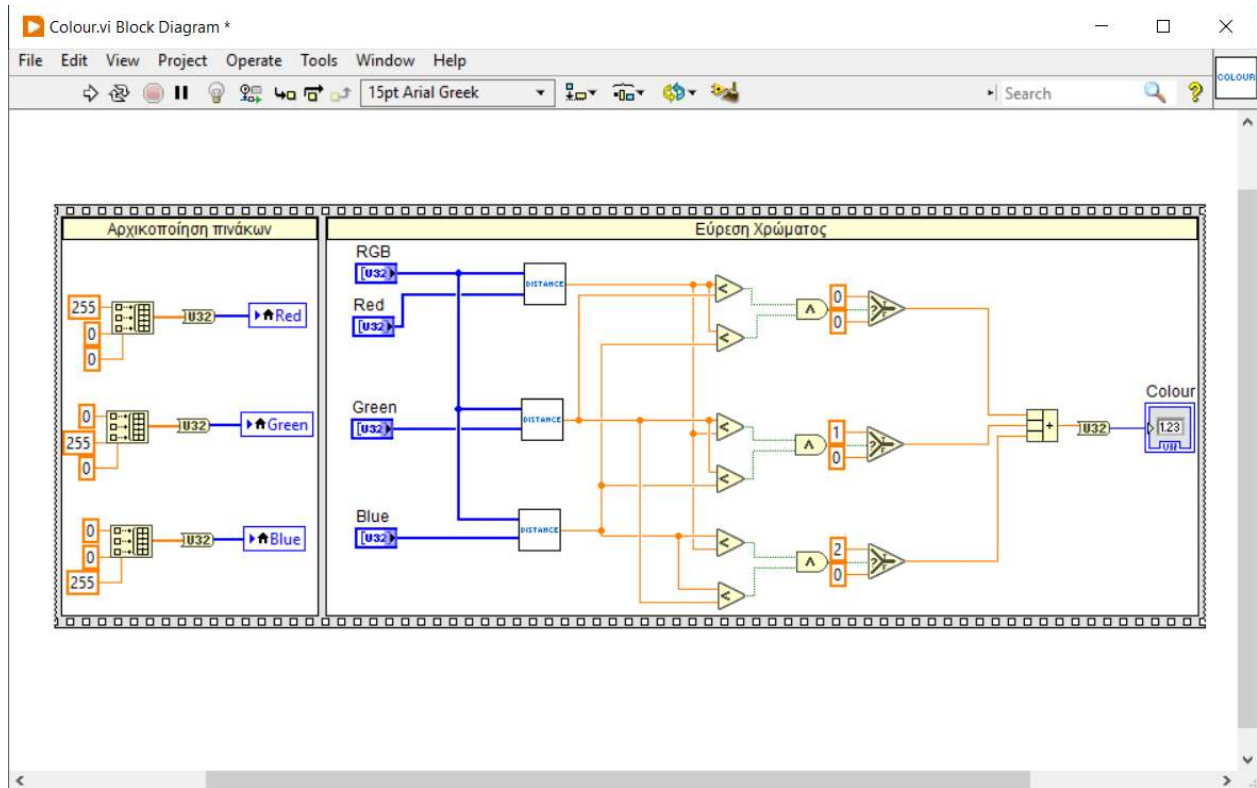
Το vi αυτό μετατρέπει την γωνία που θέλουμε να στρίψει ο σέρβο κινητήρας σε σήμα PWM, δηλαδή εφαρμόζει τον τύπο (1).





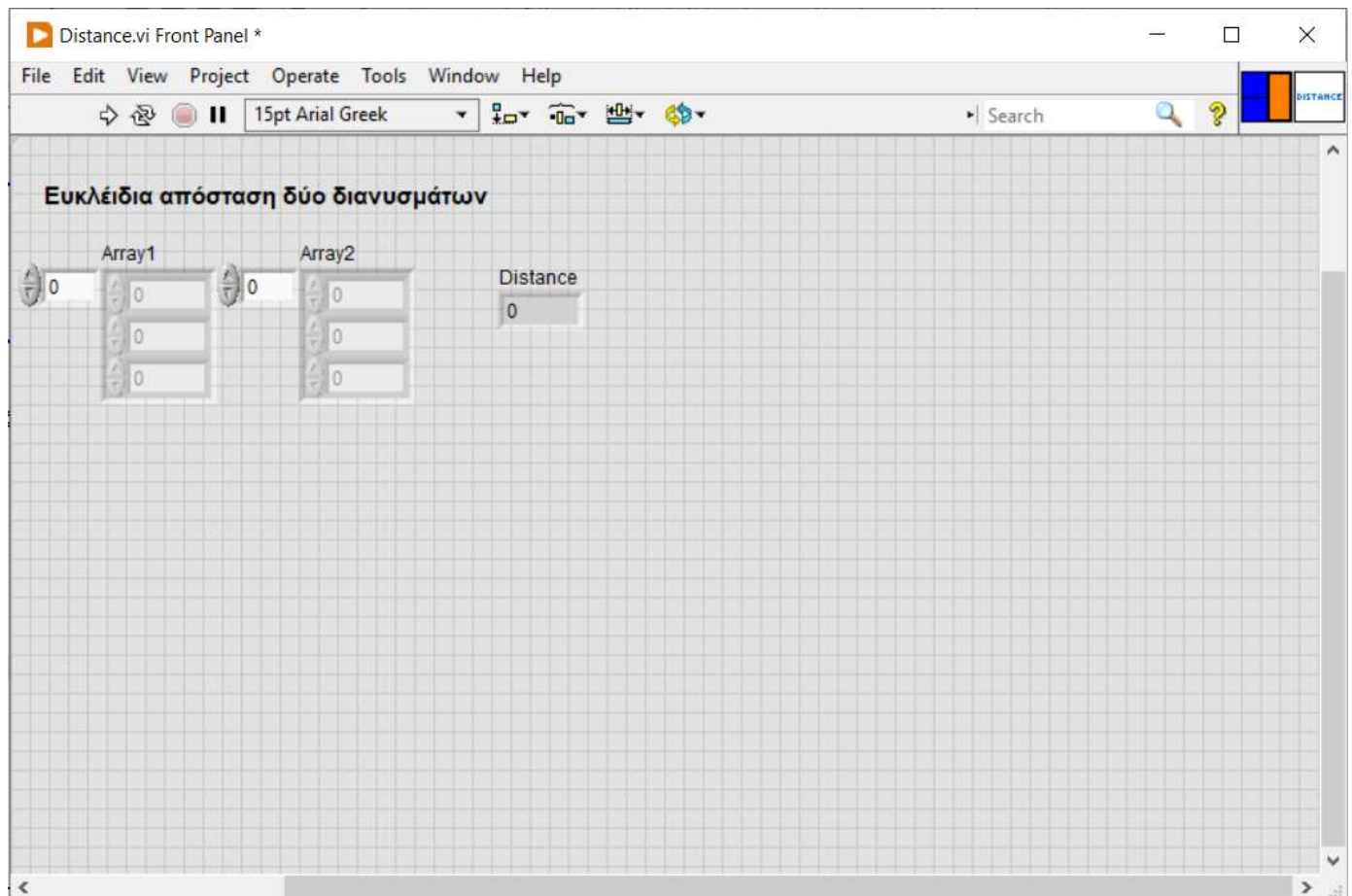
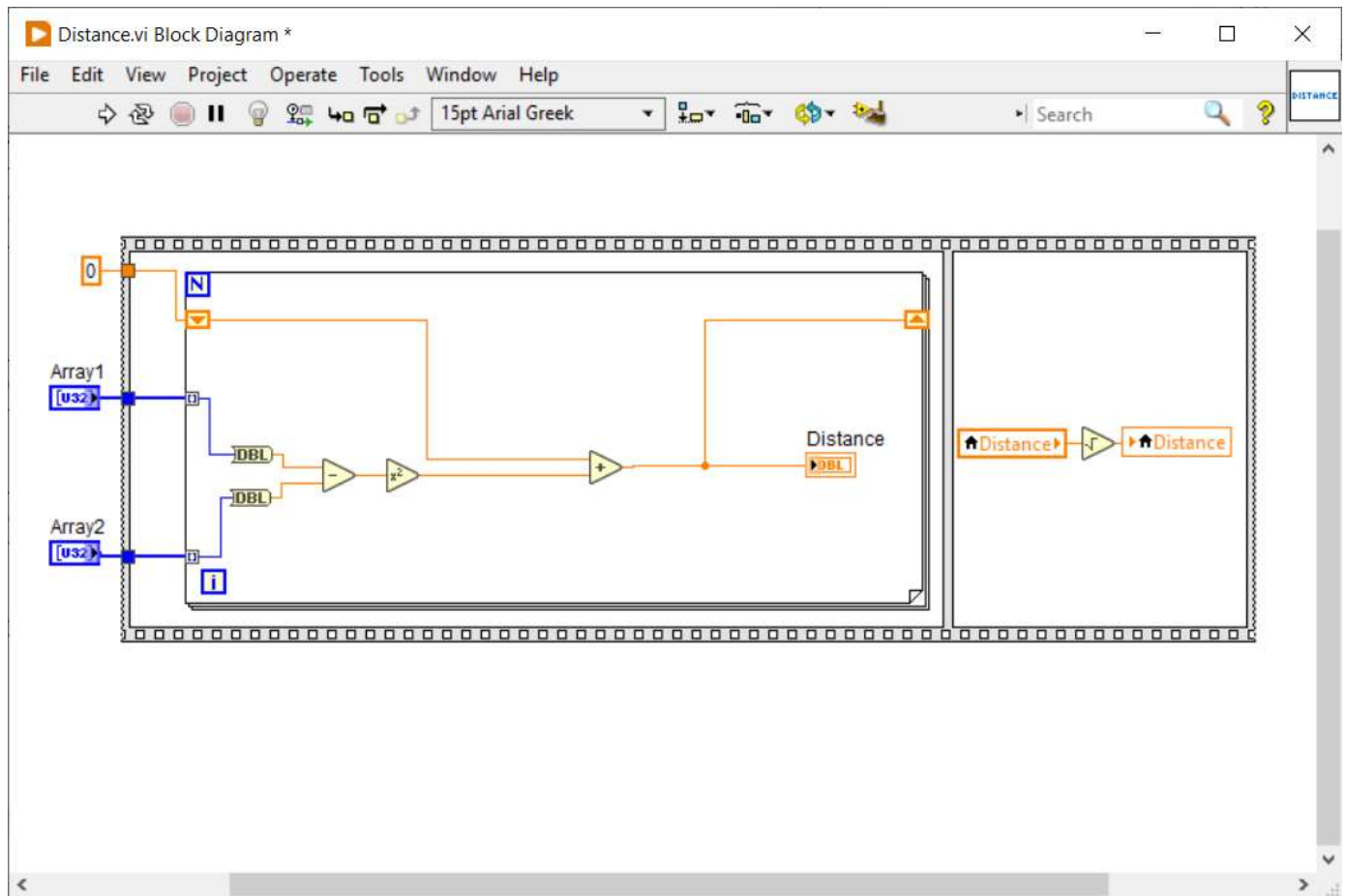
Colour.vi

Από τα δεδομένα που παίρνουμε από τον σένσορα και μετά την μετατροπή του στο σύστημα RGB, βρίσκουμε το χρώμα του αντικειμένου μέσω τις ευκλείδειας απόστασης. Δηλαδή ορίζουμε σαν πίνακες το κόκκινο, το πράσινο και το μπλε σύμφωνα με το RGB και υπολογίζουμε την απόσταση μεταξύ των δεδομένων και των χρωμάτων και όποιο έχει την μικρότερη απόσταση αυτό είναι το χρώμα του αντικειμένου.



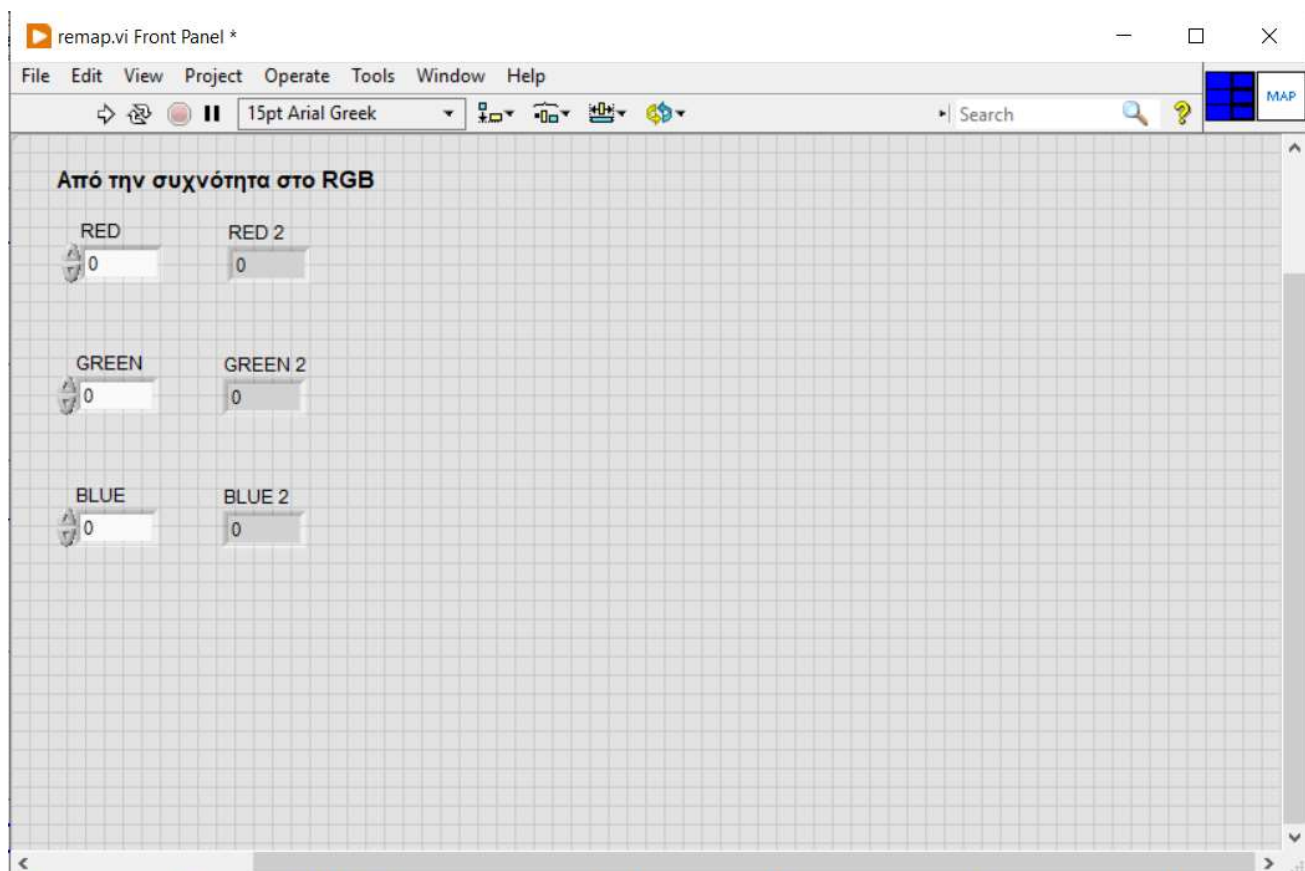
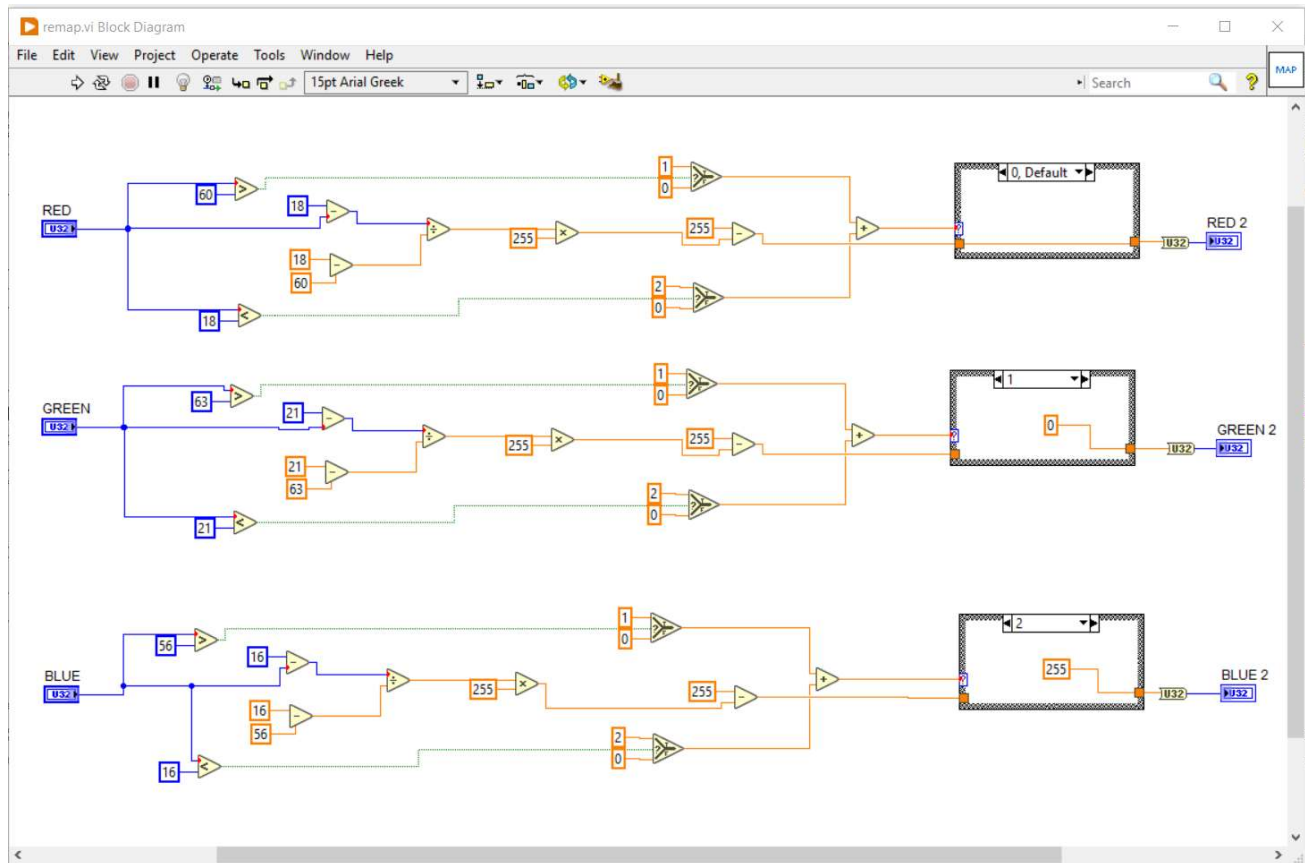
Distance.vi

Με αυτό το νι υπολογίζουμε την ευκλείδεια απόσταση δύο διανυσμάτων με διάσταση 3.



Remap.vi

Μετατρέπει την έξοδο του αισθητήρα χρώματος στο RGB και χρησιμοποιεί την εξίσωση (2).



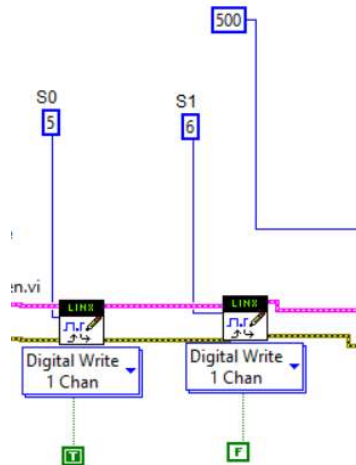
Οι συνθήκες στις δομές case είναι όπως και στην εξίσωση (2).

Pins

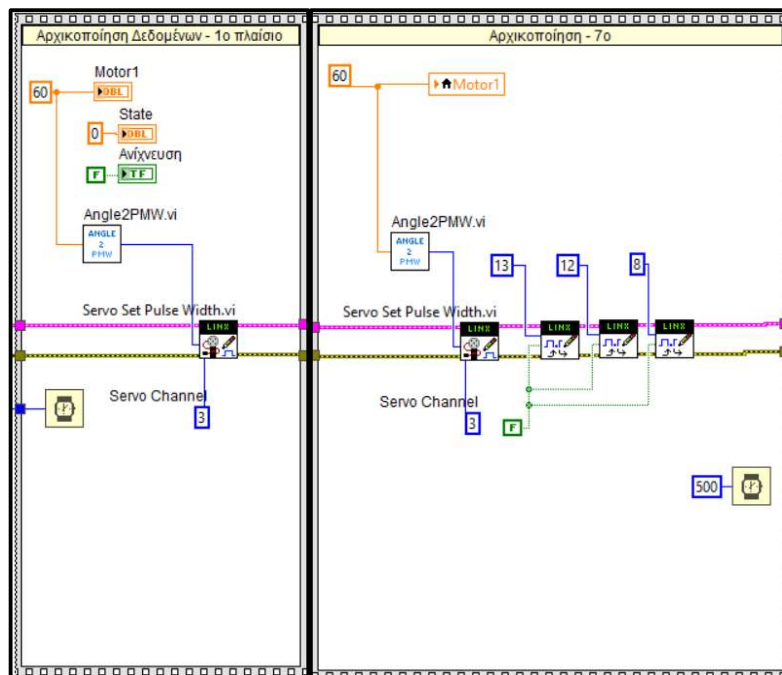
- 3 -> Μικρό-σέρβο κινητήρας
- 5 -> S_0
- 6 -> S_1
- 8 -> Red LED
- 9 -> S_2
- 10 -> S_3
- 11 -> Out
- 12 -> Green LED
- 13 -> Blue LED

Καταστάσεις(Programm.vi)

Αρχικά τοποθετούμε την κλίμακα 20% για την συχνότητα στην έξοδο του αισθητήρα χρώματος πριν την είσοδο στο while.

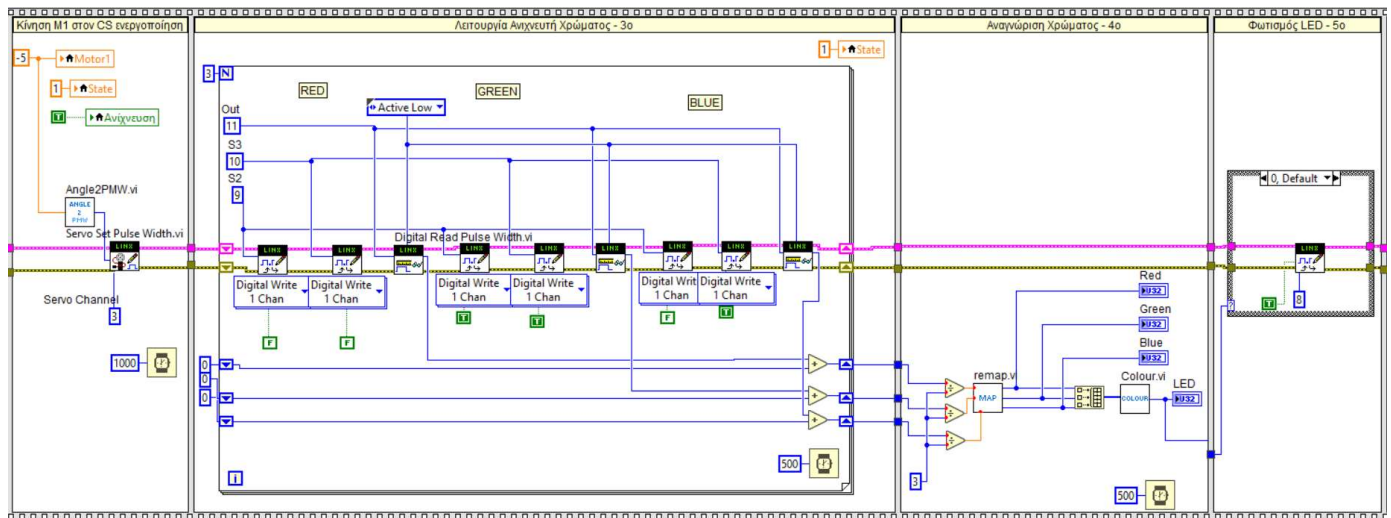


Με μία flat sequence ξεχωρίζουμε τις ενέργειες της κάθε κατάσταση. Η πρώτη κατάσταση είναι:



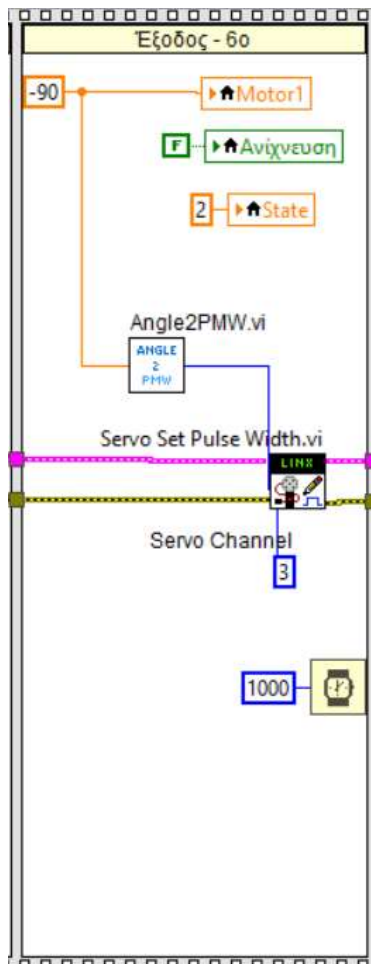
Αρχικοποιούμε την θέση του σέρβο κινητήρα στην πρώτη κατάσταση (Βορράς). Το ίδιο κάνουμε και στο τέλος του sequence ώστε στο τέλος της συσκευής να επιστρέφει πίσω στην ίδια κατάσταση. Στο τελευταίο επίσης θέτουμε False τα τρία led.

Στην δεύτερη κατάσταση:



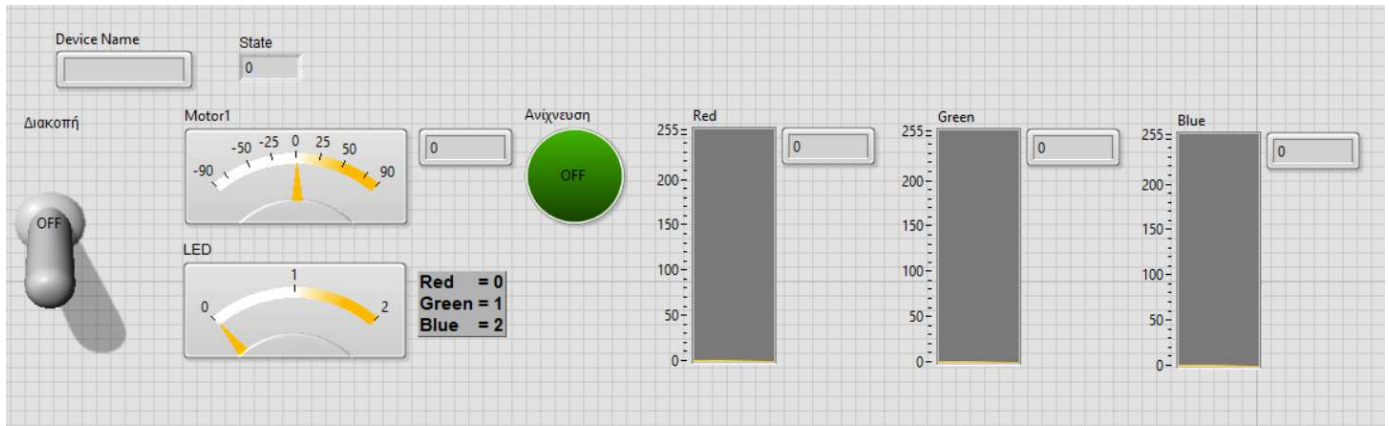
Στο 2^ο πλαίσιο στρέφουμε τον σέρβο κινητήρα στην 2^η κατάσταση(Ανατολή). Έτσι το αντικείμενο είναι κάτω ακριβώς από τον αισθητήρα και με την βοήθεια των Read και Write του LINX, στο 3^ο πλαίσιο παίρνουμε τρεις ή πέντε συχνότητες από κάθε χρώμα μέσω της δομής for και υπολογίζουμε τον μέσο όρο αυτών ώστε να το μετατρέψουμε μετά στο πλαίσιο 4 στο σύστημα RGB(remap.vi). Επίσης στο πλαίσιο 4, βρίσκουμε το χρώμα του αντικειμένου μέσω του Colour.vi. Όταν θα έχουμε το χρώμα από την έξοδο του Colour.vi, στο πλαίσιο 5 σε μία case δομή ανάβουμε το κατάλληλο LED.

Στην κατάσταση 3:



Φέρνουμε τον σέρβο κινητήρα στην κατάσταση τρία(Νότος) και το αντικείμενο πέφτει στο δοχείο.

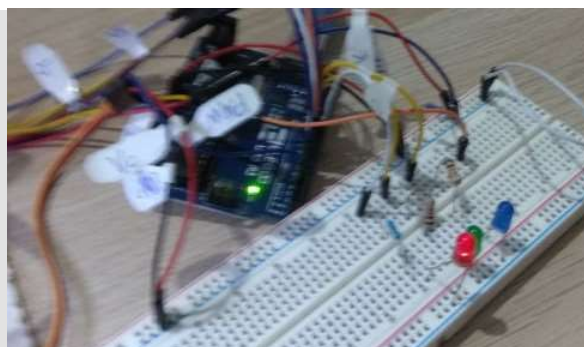
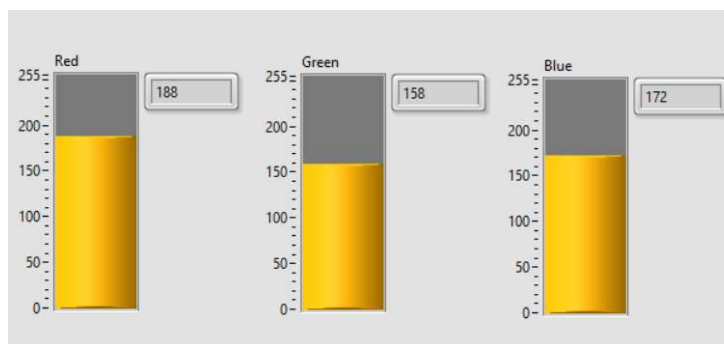
Front Pannel



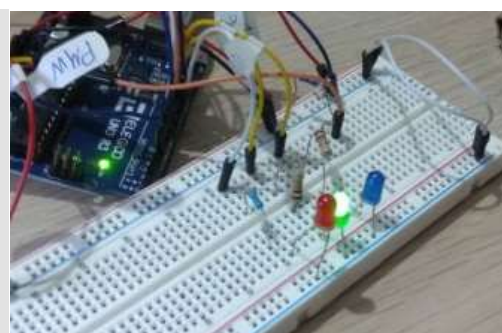
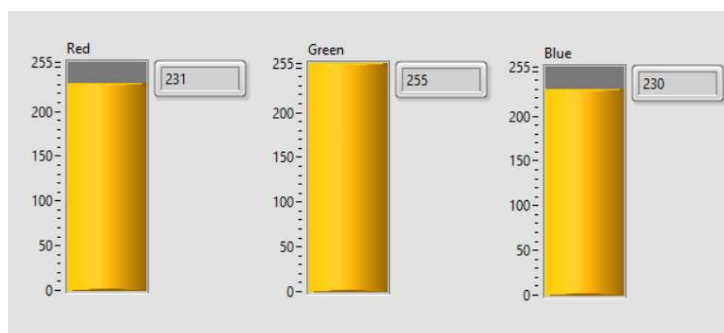
- *Device Name*: Το όνομα του Arduino
- *Διακοπή*: Όταν ενεργοποιείται, το πρόγραμμα τερματίζεται μόλις τελειώσει και η τελευταία κατάσταση.
- *State(Κατάσταση)*: Δείχνει σε ποια κατάσταση είμαστε(δείχνει τους αριθμούς με μείον ένα διαφορά)
- *Motor1*: Δείχνει την γωνία του μικρό-σέρβο κινητήρα.
- *LED*: Δείχνει ποιο από τα τρία λαμπάκια θα ανάψει.
- *Ανίχνευση*: ο σέρβο βρίσκεται στην κατάσταση 2(ανατολή).
- *Red, Green, Blue*: Οι μετρήσεις στο σύστημα RGB.

Αποτελέσματα

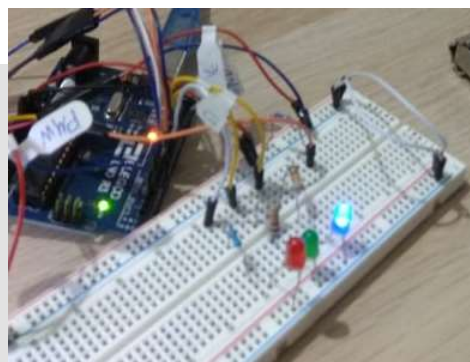
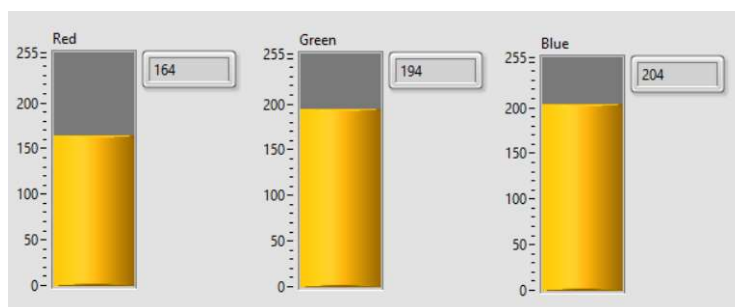
Για το κόκκινο χρώμα:



Για το πράσινο χρώμα:



Για το μπλε χρώμα:



Οι τιμές είναι μεγάλες και στα τρία χρώματα λόγω της ευαισθησίας που έχει ο αισθητήρας, καθώς και τα αντικείμενα αντανακλούν το λευκό φως.

Προβλήματα κατά την κατασκευή

- Η συσκευή κανονικά είχε και μία 'τσουλήθρα' η οποία λειτουργούσε με μικρό-σέρβο κινητήρα ώστε να ταξινομεί στο σωστό δοχείο το χρωματισμένο αντικείμενο. Όμως, λόγω προβλήματος στον κινητήρα, δεν προστέθηκε στην συσκευή. Για αυτό το λόγο έχουμε τα LED που δείχνουν πιο χρώμα πέρασε.
- Στο κατασκευαστικό κομμάτι που θα τοποθετηθεί ο κατάλληλος σένσορας.
- Και η επιλογή των αντικειμένων δυσκόλεψε την διαδικασία, διότι όπως αναφέραμε παραπάνω αντανακλά το φως και έχουμε τιμές κοντά στο άσπρο, με αποτέλεσμα να βρεθούν δύσκολα τα όρια για την μετατροπή στο RGB.