

Διαφάνεια 1<sup>η</sup> // ΚΩΣΤΗΣ

Καλησπέρα σας,

Είμαστε ο Κωστής Τζεβελεκάκης και ο Μιλτιάδης Στούρας και μαζί προγραμματίσαμε το Robot που διαδραματίζει τον ρολο του κυνηγού στην επίδειξη που σας ετοιμάσαμε, ενώ πρέπει να σημειωθεί ότι ο αλγόριθμος αυτός λειτουργεί για τον εντοπισμό ενός και μόνο στόχου.

Διαφάνεια 2<sup>η</sup> // Μιλτιάδης

Αρχικά θα σας μιλήσουμε για την διαταξη του ηλ.κυκλώματος. Το κύκλωμα αποτελείται από (1) τον μικροελεγκτή Arduino , το Arduino Shield , (2) δύο μοτέρ που κινούν το ρομπότ μας, (3) τον αισθητήρα απόστασης Ping Sensor , (4) το μοτέρ που κινεί τον αισθητήρα στον οριζόντιο άξονα , (5) την φωτοαντίσταση που είναι τοποθετημένη στο κάτω μέρος του ρομπότ όπως και το (6) LED.Αντίθετα με το προηγούμενο ρομπότ εμείς έχουμε τοποθετήσει και ένα (7) buzzer που παράγει έναν ήχο όταν εντοπίζουμε τον στόχο μας. Η διάταξη που πραγματοποιήσαμε τροφοδοτείται από μια μπαταρία Πολυμερούς Λυθίου ( Li-Po) 7.4Volt με 2 κελιά.

Διαφάνεια 3<sup>η</sup> // ΚΩΣΤΗΣ

Στην διαφάνεια μπορείτε να πάρετε μια εικόνα από τον αλγόριθμο που εκτελεί το ρομπότ μας , τον οποίο και θα αναλύσουμε στην συνέχεια.

Διαφάνεια 4<sup>η</sup> // ΚΩΣΤΗΣ

Όπως στην αρχή κάθε κώδικα , έτσι και στον δικό μας παρατηρούμε ότι με την εντολή #define ορίζουμε κάποιες σταθερές που θα χρησιμοποιήσουμε στο πρόγραμμά μας , ενώ με τις εντολές float και int ορίζουμε μεταβλητές που έχουν την μορφή κινούμενης υποδιαστολής και ακεραίου αντίστοιχα και δημιουργούμε δύο αντικείμενα Servo και New ping.

Διαφάνεια 5<sup>η</sup> // ΚΩΣΤΗΣ

Στην void setup() ορίζουμε κάποια από τα pin ως pin εξόδου και καλούμε την \* διαδικασία ThermalBalance(); Όστε το ρομπότ μας να αποκτήσει θερμική ισορροπία.

Διαφάνεια 6<sup>η</sup> // ΚΩΣΤΗΣ

Η void loop() είναι ο κύριος κορμός του προγράμματός μας και εκτελείται κυκλικά.Ξεκινάμε αρχικοποιώντας τις μεταβλητές μας , ενώ με την δομή επανάληψης for πέρνουμε τις μετρήσεις μας απο τον αισθητήρα. Ο ping sensor μας δίνει μια μέτρηση ανά 10 μοίρες για το εύρος 20 – 160 μοιρών , ενώ από τον μέσο όρο τους βρίσκουμε την θέση του στόχου.

Διαφάνεια 7<sup>η</sup> //ΚΩΣΤΗΣ

Η απόσταση που μας επιστρέφει κάθε μέτρηση αποθηκεύεται στην μεταβλητή distance. Ελέγχοντας εάν υπάρχουν τουλάχιστον 3 μετρήσεις που είναι κάτω από 900μs σε αυτό το εύρος καταλαβαίνουμε αν είμαστε κοντά στον στόχο , αν δηλαδή τον έχουμε εντοπίσει , ώστε το πρόγραμμα να τερματιστεί.

Διαφάνεια 8<sup>η</sup> //Μιλτιάδης

Ακόμα , ελέγχοντας εάν όλες οι μετρήσεις είναι μηδενικές , εάν δηλαδή ο στόχος μας βρίσκεται εκτός του εύρους των 20 με 160 μοιρών , δίνουμε την εντολή στο ρομπότ μας να εκτελέσει μια στροφή γύρω από τον άξονα περιστροφής του για να ψάξει τον στόχο στο υπόλοιπο εύρος μοιρών.

Διαφάνεια 9<sup>η</sup> //Μιλτιάδης

Στην συνέχεια καλούμε την (1) διαδικασία getServo\_Target() με παράμετρους Sw και S, η οποία μας επιστρέφει τις μοίρες της θέσης του στόχου. (2)Η διαδικασία getTurn(); με παράμετρο Servo\_Target , η οποία μέσω της εξίσωσης που φαίνεται στην εικόνα μας πληροφορεί εάν ο στόχος βρίσκεται αριστερά η δεξιά του ρομποτ μας.

Διαφάνεια 10<sup>η</sup> //Μιλτιάδης

Εάν η μεταβλητή turn είναι θετικός αριθμός , που σημαίνει ότι ο στόχος μας βρίσκεται στα δεξιά , το ρομπότ δίνει όλη την ισχύ που έχει το κύκλωμα στο αριστερό μοτέρ ενώ μηδενίζουμε την ταχύτητα του δεξιού μοτέρ για συγκεκριμένο χρόνο , ώστε το ρομπότ μας να στρίψει στην κατάλληλη θέση.

Διαφάνεια 11<sup>η</sup> //Μιλτιάδης

Αντίστοιχα εάν η μεταβλητή turn είναι αρνητικός αριθμός το ρομπότ μας εκτελεί τις αντίστοιχες κινήσεις με αποτέλεσμα να στραφεί στην επιθυμητή θέση.

Διαφάνεια 12<sup>η</sup> //Μιλτιάδης

Σας ευχαριστούμε για την προσοχή σας και θα συνεχίσουμε με την επίδειξη των ρομποτικών οχημάτων.