

Covid Assistant

(Μαργαρίτα)

Καλημέρα σας. Είμαστε τα παιδιά του Δημοτικού Σχολείου Νέας Φιλαδέλφειας και θα σας μιλήσουμε για την ρομποτική κατασκευή που φτιάξαμε στα πλαίσια του μαθήματος της Πληροφορικής. Ασχοληθήκαμε και δημιουργήσαμε ένα ρομποτικό κουτί, που στόχος του είναι η συμβολή στην διατήρηση της υγιεινής και του ασφαλούς ελέγχου θερμοκρασίας του εκάστοτε χειριστή.

Όπως βλέπεται και στην οθόνη, το Covid Assistant αποτελεί ένα πολύ-εργαλείο το οποίο το κατασκευάσαμε και το τοποθετήσαμε στην είσοδο του σχολείου. Από τον αισθητήρα υπέρυθρης μέτρησης θερμοκρασίας, ο κάθε μαθητής μπορεί να μετρήσει την θερμοκρασία του ανέπαφα, αν έχει ξεχάσει την μάσκα του ή έχει αχρηστευτεί αυτή που έχει μπορεί να προμηθευτεί μια καινούργια χωρίς να χρειαστεί να ακουμπήσει το κουτί, καθώς και το αντίστοιχο αν χρειαστεί γάντια. Πιστεύουμε ότι το Covid Assistant μπορεί να συμβάλει αρκετά και στην έγκαιρη ενημέρωση για θετικό κρούσμα (με την απομακρυσμένη και από απόσταση μέτρηση της θερμοκρασίας των μαθητών), αλλά και στη μείωση της μετάδοσης του ιού μια και η όλη χρήση του γίνεται ανέπαφα.

(Συμέλα)

Για την δημιουργία της κατασκευής μας, χρησιμοποιήσαμε τη δημοφιλή ρομποτική πλακέτα Arduino Uno, λόγω του ότι είναι μια φθηνή, αξιόπιστη, και εύκολη στη χρήση λύση για ρομποτικές κατασκευές. Επίσης αποτελεί και μάθημα στο μάθημα της πληροφορικής του σχολείου μας.

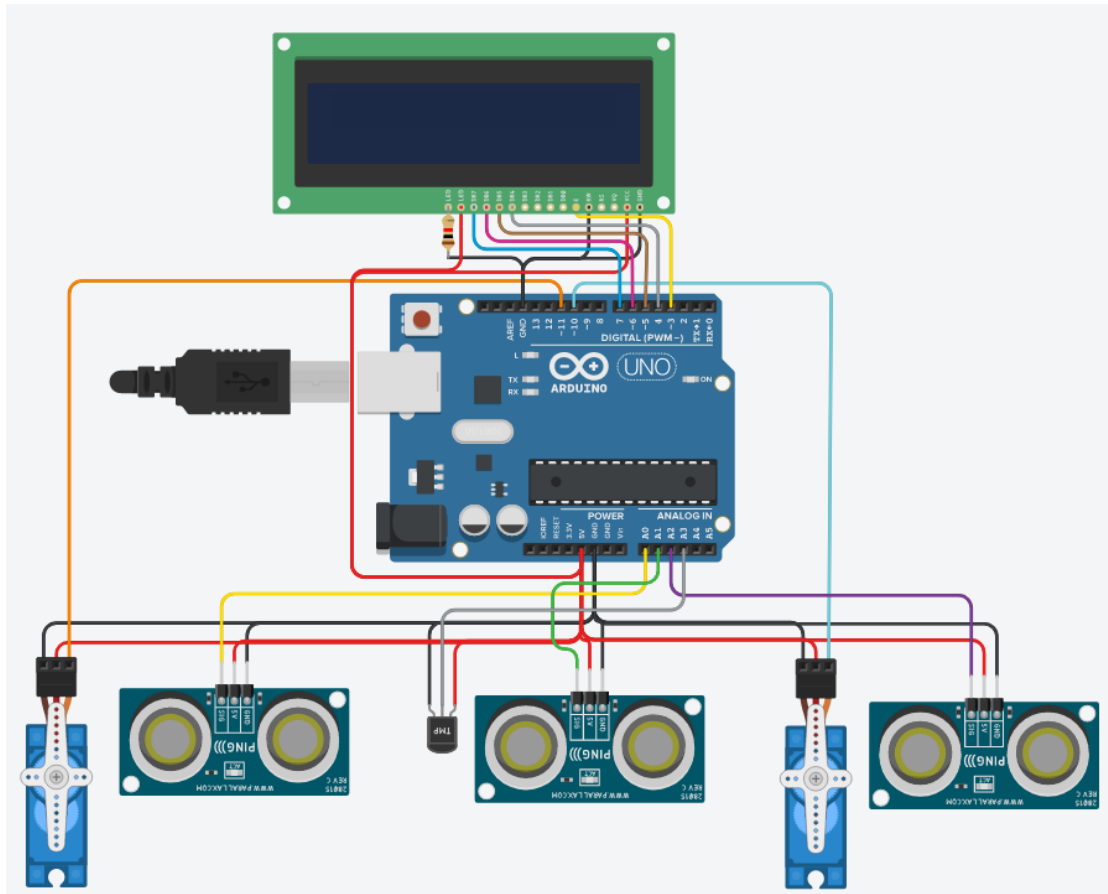
Για την μέτρηση της θερμοκρασίας, χρησιμοποιήσαμε τον αισθητήρα υπέρυθρων (MLX90614) ο οποίος δίνει τη δυνατότητα μέτρησης από απόσταση, και η προβολή της θερμοκρασίας γίνεται σε μία τυπική 16x2 LCD οθόνη της εταιρίας Adafruit. Η μέτρηση του αισθητήρα δεν γίνεται συνεχόμενα αλλά μόνο όταν ο ενδιαφερόμενος πλησιάσει στον αισθητήρα. Για να το επιτύχουμε αυτό, χρησιμοποιήσαμε έναν αισθητήρα απόστασης (Ultrasonic Distance Sensor) ο οποίος ελέγχει αν κάποιος πλησιάσει στην συσκευή και μόλις η απόσταση είναι στα 10 εκατοστά, δίνει εντολή στον αισθητήρα θερμότητας να κάνει μέτρηση και να την δείξει στην οθόνη. Οι αισθητήρες που χρησιμοποιήσαμε είναι σχετικά απλή και φθηνοί αλλά κατά την χρήση αυτών παρατηρήσαμε ότι ο αισθητήρας υπέρυθρης

θερμοκρασίας (ο MLX90614) δίνει σημαντικές διαφορές στη μέτρηση θερμοκρασίας ανάλογα με την απόσταση που βρίσκεται το αντικείμενο μέτρησης. Σε πολύ κοντινές αποστάσεις και κυρίως σε ίδια απόσταση, δίνει ικανοποιητικές τιμές.

(Μαρκέλλα)

Για την αυτόματη και ανέπαφη διάθεση масκών και γαντιών, χρησιμοποιήσαμε 2 σερβο-κινητήρες 1.8 κιλών οι οποίοι προωθούν το κάθε κουτί προς την αντίστοιχη σχισμή της συσκευής ανάλογα αν πρόκειται για μάσκα ή για γάντια. Εντολή στον σερβο-κινητήρα για να κινηθεί, δίνει και πάλι ένας αντίστοιχος αισθητήρας απόστασης (Ultra Sonic Distance Sensor) ο οποίος, όπως και στον αισθητήρα μέτρησης θερμοκρασίας που περιγράψαμε παραπάνω, ανιχνεύει αν υπάρχει εμπόδιο στα 5 εκατοστά. Με αυτόν τον τρόπο, ο ενδιαφερόμενος που θέλει να προμηθευτή μάσκα ή γάντια αντίστοιχα, αρκεί να βάλει το χέρι του μπροστά από τον αισθητήρα σε απόσταση λίγων εκατοστών αλλά χωρίς να τον αγγίζει, και η κατασκευή μας θα κάνει εξαγωγή της μάσκας ή των γαντιών που επιθυμεί.

Η μάσκες και τα γάντια είναι τοποθετημένα μέσα και χάρτινα κουτάκια, έτσι ώστε να διασφαλίζεται και η υγιεινή του υλικού, αλλά και για να μπορεί ο μηχανισμός να εξάγει την μάσκα από την αντίστοιχη σχισμή, χωρίς να τσαλακώνεται ή κολλάει μέσα στο μηχάνημα. Τα κουτάκια αυτά τοποθετούνται μέσα στο κουτί το ένα πάνω στο άλλο, και ο σερβο-κινητήρας σπρώχνει το κάτω κουτί κάθε φορά. Με την απομάκρυνση του κουτιού, τα υπόλοιπα κουτάκια μετακινούνται προς τα κάτω λόγω της βαρύτητας, και το σύστημα είναι ένα εξυπηρετήσει εκ νέου των επόμενο ενδιαφερόμενο.



(Τρύφωνας)

Η σχεδίαση του ηλεκτρονικού κυκλώματος έγινε καταρχάς την πλατφόρμα προσομοίωσης TinkerCad της AutoDesk. Στην πλατφόρμα αυτή, μπορέσαμε να κάνουμε την απαραίτητη συνδεσμολογία των εξαρτημάτων με την ρομποτική πλακέτα Arduino, να γράψουμε τον απαραίτητο κώδικα για την σωστή λειτουργία και, το κυριότερο, να ελέγξουμε σε περιβάλλον προσομοίωσης αν το έχουμε κάνει σωστά. Όπως βλέπεται και στο σχεδιάγραμμα, αφού δώσουμε ρεύμα (κόκκινα καλώδια) και γειώσουμε τα εξαρτήματα μας (μαύρα καλώδια) στα pin 5V και GND αντίστοιχα, συνδέουμε το κάθε εξάρτημα με τον εξής τρόπο:

Ο 1^{ος} σερβο-κινητήρας ο οποίος χρησιμοποιείται για να εξάγει τις μάσκες συνδέεται στην ψηφιακή θύρα 11 του Arduino (πορτοκαλί καλώδιο) η οποία είναι μια PWM αναλογική έξοδος, οπότε μπορεί να εξυπηρετήσει την κίνηση του κινητήρα μας. Στη θέση A0, συνδέουμε τον αισθητήρα απόστασης (κίτρινο καλώδιο) και έχουμε τελειώσει με τις μάσκες.

Αντίστοιχα κάνουμε και για την συνδεσμολογία του 2^ο σερβο-κινητήρα ο οποίος χρησιμοποιείται για να εξάγει τα γάντια, χρησιμοποιώντας τις θέσεις 10 (γαλάζιο καλώδιο) και A2 (μωβ καλώδιο) αντίστοιχα για τον αισθητήρα απόστασης που τον καθοδηγεί.

Ο αισθητήρας θερμοκρασίας, συνδέεται στην αναλογική θύρα A3 (γκρι καλώδιο) και ο αντίστοιχός του αισθητήρας απόστασης στην θύρα A1 (πράσινο καλώδιο).

Η συνδεσμολογία της οθόνης μας είναι σχετικά απλή. Εκτός από το ρεύμα (κόκκινο καλώδιο) που το συνδέουμε στις επαφές VCC και LED (το 1^ο που είναι η άνοδος) της οθόνης μας, γειώνουμε αντίστοιχα (με μαύρο καλώδιο) τις επαφές GND (προφανώς), RW (μια και μόνο γράφουμε στην οθόνη), και το 2^ο LED (που αποτελεί την κάθοδο). Από τα Data Pin της οθόνης, χρειαζόμαστε μόνο τα DB7, DB6, DB5, DB4 τα οποία τα συνδέουμε στα αντίστοιχα ψηφιακά pins του Arduino μας (7, 6, 5, 4). Την επαφή Enable (E) την συνδέουμε στο pin 3 και τέλος την επαφή Register Select (RS) της οθόνης μας την συνδέουμε στο ψηφιακό pin 2 της πλακέτας.

(Νίκος)

Ο κώδικας του κυκλώματος μας δεν είχε ιδιαίτερες δυσκολίες. Αφού ενσωματώσουμε τις απαραίτητες βιβλιοθήκες Servo και LiquidCrystal για την σωστή λειτουργία των σερβο-κινητήρων και της οθόνης, δημιουργούμε τις μεταβλητές μας:

dMask: για την απόσταση του αισθητήρα των масκών

dGloves: για την απόσταση του αισθητήρα των γαντιών

dTemp: για την απόσταση του αισθητήρα θερμοκρασίας

vTemp: για την τιμή που θα μας δώσει ο αισθητήρας θερμοκρασίας

lcd: για την οθόνη στην οποία δηλώνουμε τα pins στα οποία την συνδέσαμε και που είναι με την σειρά τα pins (RS, E, DB4, DB5, DB6, DB7).

Και τέλος της μεταβλητές **servo_10** και **servo_11:** για χρήση των σερβο-κινητήρων μας.

Δημιουργήσαμε μια function την readUltrasonicDistance για να μας επιστρέφει τα μικροδευτερόλεπτα που χρειάζεται ο παλμός για να επιστρέψει το σήμα και που πολλαπλασιασμένα με το 0.01723 (η ταχύτητα του ήχου διά 2) θα μας δώσει σε εκατοστά την απόσταση.

Αφού ορίσουμε τα αντίστοιχα pins και αρχικοποιήσουμε τα αντικείμενα μας, στο τμήμα του loop εκτελείται ο ουσιαστικός κώδικας του κυκλώματός μας.

Ατέρμονα, διαβάζουμε τις αποστάσεις των τριών αισθητήρων απόστασης, και όταν η απόσταση του αισθητήρα των масκών (dMask) είναι μικρότερη ή ίση των 10 (εκατοστών) τότε δίνουμε εντολή στον σερβο-κινητήρα να περιστραφεί για 90 μοίρες, αλλιώς (αν απομακρυνθεί το χέρι και η απόσταση είναι μεγαλύτερη των 10 εκατοστών, να γυρίσει στην

αρχική του θέση. Το ίδιο γίνεται και για τον έτερο σερβο-κινητήρα της μεταβλητής dGloves. Όταν ο αισθητήρας απόστασης dTemp (αυτός του θερμόμετρου) μας δώσει τιμή μικρότερη των 10 εκατοστών, τότε διαβάζουμε την θερμοκρασία από το Pin A3 και την μετατρέπουμε σε βαθμούς κελσίου. Αντίστοιχα, εκτυπώνουμε στην οθόνη μας στην πρώτη γραμμή το λεκτικό «Body Temp», και στην δεύτερη γραμμή την θερμοκρασία που πήραμε.

Σας ευχαριστούμε πολύ!