



Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

MILESTONE 1: AUTONOMOUS MAP EXPLORING CAR

Καλογεράκης Στέφανος | Ζερβάκης Αρης

Εισαγωγή

Στα πλαίσια του πρότζεκτ εξαμήνου του μαθήματος ενσωματωμένα συστήματα μικροεπεξεργαστών (ΗΡΥ411) επιλέξαμε, μετά από πολλή σκέψη και αλλαγές, την υλοποίηση ενός αυτόνομου αυτοκινήτου με την δυνατότητα εξερεύνησης τυχαίων λαβυρίνθων και τελικά την απεικόνιση τους. Η τεχνολογία στην οποία είναι βασισμένη το αυτόνομό μας αυτοκίνητο είναι Lego Mindstorms. Στα πλαίσια του πρώτου milestone, δώσαμε έμφαση στην εξοικείωση μας με την διαθέσιμη τεχνολογία και την τελική διαστασιολόγηση του προβλήματος, με σκοπό τον ακριβή καθορισμό των ζητούμενων και την μεταβολή των αρχικών προδιαγραφών σε μια ρεαλιστική υλοποίηση βάσει εξοπλισμού και πολυπλοκότητας

Περιγραφή

Παρακάτω περιγράφονται τα βήματα που ακολουθήσαμε κατά την υλοποίηση του πρώτου milestone

Κατασκευή αυτόνομου αυτοκινήτου

Σε ένα πρώτο στάδιο, οφείλαμε να κατασκευάσουμε το όχημα μας. Να επισημάνουμε αρχικά, ότι υπήρχε διαθέσιμο έτοιμο και κατασκευασμένο αυτοκίνητο το οποίο μπορούσε να μας διατεθεί για την διευκόλυνση μας. Θεωρήσαμε όμως ότι η συγκεκριμένη επιλογή δεν ήταν βέλτιστη και θα μας δημιουργούσε προβλήματα στην συνέχεια, λόγω του μεγάλου μεγέθους του αυτοκινήτου. Έτσι, επιλέξαμε να πάρουμε από το εργαστήριο μόνο το προγραμματιζόμενο NXT brick, και κάποια components που χρειαζόμασταν επιπλέον και τελικά κατασκευάσαμε δικό μας προκειμένου να είναι πιο λειτουργικό και να καλύπτει τις προδιαγραφές μας.

Το τελικό αυτοκίνητο που προέκυψε φαίνεται στην εικόνα του Παραρτήματος

Τα components που απαιτήθηκαν τελικά είναι τα εξής:

- 2 Motor/Rotation Sensor έναν για τον κάθε ένα από τους μπροστινούς τροχούς
- 2 Light Sensors που χρησιμοποιούν grayscale για την ανίχνευση χρωμάτων
- 1 RGB Light Sensor με δυνατότητα ανίχνευσης χρωμάτων
- 1 UltraSonic Sensor

Τεχνολογία που χρησιμοποιήθηκε

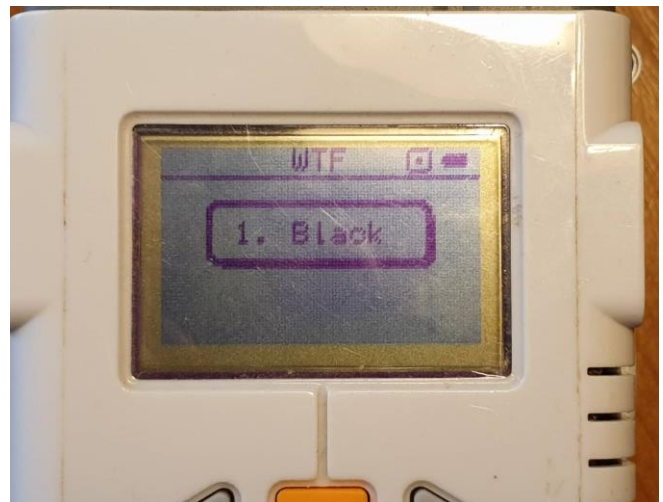
Η ανάπτυξη του κώδικα για το αυτοκίνητο θα γίνει με τη βοήθεια της πλατφόρμας ROBOTC. Η πλατφόρμα, βασισμένη στη γλώσσα C, διαθέτει όλα τα απαραίτητα εργαλεία για το χειρισμό και τον προγραμματισμό ρομποτικών συστημάτων και μας δίνει την δυνατότητα άμεσης επικοινωνίας με το ρομπότ. Στη ROBOTC θα πραγματοποιηθούν όλοι οι υπολογισμοί για τον κώδικα που θα αναπτυχθεί, λαμβάνοντας υπόψιν τις μετρήσεις που λαμβάνουμε από το NXT Brick. Ακολουθεί ένα απλό παράδειγμα κώδικα στο ROBOTC, που χρησιμοποιήσαμε στις πρώτες μας δοκιμές, όπου δίνουμε στα 2 μοτέρ του αυτοκινήτου σταθερή ταχύτητα όσο η μέτρηση που παίρνουμε από τον light sensor είναι μεγαλύτερος ή ίσος της τιμής 45.

```
#pragma config(Sensor, S1,      LIGHT,      sensorLightActive)
#pragma config(Motor,  motorA,      LMOTOR,      tmotorNormal, PIDControl, encoder)
#pragma config(Motor,  motorB,      RMOTOR,      tmotorNormal, PIDControl, encoder)
/*!!Code automatically generated by 'ROBOTC' configuration wizard    !!*/

task main()
{
    int velocity = 80;
    int brightness = SensorValue(LIGHT);
    //Motor's velocity range is between 0 and 100

    while(brightness >= 45){
        motor[LMOTOR] = 80;
        motor[RMOTOR] = 80;
    }
}
```

Έτσι η κίνηση του αυτοκινήτου πάνω στη γραμμή του λαβυρίνθου, καθορίζεται πλήρως από τις ανάλογες μετρήσεις των αισθητήρων φωτός όπως και του αισθητήρα RGB στο μπροστά μέρος του αυτοκινήτου, ο οποίος μας δίνει το ακριβές χρώμα του μπλοκ στο οποίο βρισκόμαστε. Μαύρο αν είμαστε πάνω στην γραμμή ή άλλο χρώμα σε αντίθετη περίπτωση.

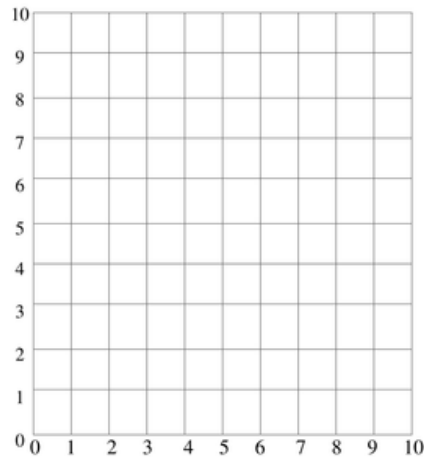


Τελικός καθορισμός προδιαγραφών

Όπως επισημάνσαμε κατά την παράδοση της πρότασης μας, η αρχική μας ιδέα απαιτούσε μια σειρά από αλλαγές και διευκρινήσεις αναφορικά με τις τελικές προδιαγραφές οι οποίες γίνονται εδώ μετά και από την πρώτη επαφή μας με τις διάφορες τεχνολογίες που θα έρθουμε αντιμέτωποι κατά την υλοποίηση του προτζεκτ.

Οι τελικές προδιαγραφές είναι οι ακόλουθες

- Όπως επισημάνσαμε και στην πρόταση μας, δεν θα χρησιμοποιεί τοιχώματα ως εμπόδια. Θα υπάρχουν τοιχώματα μόνο περιμετρικά του λαβυρίνθου τα οποία θα οριοθετούν τον λαβύρινθο μας.
- Ο αλγόριθμος που θα χρησιμοποιήσουμε θα λειτουργεί εξαντλητικά και να επισκεφτεί όλα τα σημεία με μαύρη γραμμή που βρίσκονται μέσα στο οριοθετημένο πλαίσιο του λαβυρίνθου.
- Προαιρετικά, θα υπάρχει επιλογή τερματικής συνθήκης πρόωρα η οποία θα είναι ένα κόκκινο σημείο πάνω στην μαύρη γραμμή και σε αυτό το σημείο θα σταματάει το όχημα μας. Προφανώς, ο χάρτης που θα δημιουργεί δεν θα είναι ολόκληρης της πίστας αλλά μόνο μέχρι το σημείο που βρήκε την τερματική συνθήκη. Σε περίπτωση που δεν βρεθεί τερματική συνθήκη θα σκαναριστεί όλο ο χάρτης και το όχημα μας θα σταματήσει αφότου έχουν σκαναριστεί όλα τα σημεία και θα μπορέσει να παρουσιάσει ολόκληρο τον χάρτη.
- Ο λαβύρινθος που θα μπορεί να δοκιμαστεί η λειτουργία του αυτοκινήτου μας θα είναι τυχαίος με την προϋπόθεση να φέρει τις μαύρες γραμμές που θα μπορέσει να κινείται το όχημα μας και το οριοθετημένο πλαίσιο.
- Το αυτοκίνητο θα μπορεί να εκκινήσει από τυχαία σημεία στον χάρτη και όχι μια συγκεκριμένη αφετηρία
- Η τελική πίστα μπορούμε να θεωρήσουμε ότι θα αποτελεί ένα νοητό grid διαστάσεων $10 * 10$ όπως φαίνεται στην εικόνα ενώ το κάθε «κελί» να έχει διαστάσεις $10\text{cm} * 10\text{cm}$. Άρα τελικά προκύπτει πίστα διαστάσεων $100\text{cm} * 100\text{cm}$ ικανή για την πραγματοποίηση πολλών και διαφορετικών τυχαίων συνδυασμών.



- Η μεταφορά των δεδομένων θα πραγματοποιηθεί με την χρήση USB από το nxt στον υπολογιστή. Να σημειώσουμε ότι αποπειραθήκαμε να μεταφέρουμε δεδομένα μέσω της λειτουργίας του Bluetooth από το nxt στον υπολογιστή χωρίς επιτυχία σε κάθε περίπτωση. Στην τελική παράδοση του πρότζεκτ, τα δεδομένα θα μεταφέρονται ΠΑΝΤΑ με την χρήση USB και σε περίπτωση που καταφέρουμε να λύσουμε τα προβλήματα που αντιμετωπίσαμε θα παρουσιάσουμε ως επιπλέον την λειτουργία την μεταφορά δεδομένων μέσω Bluetooth.
- Σε τελικό στάδιο αποφασίσαμε, μετά την μεταφορά των δεδομένων στον υπολογιστή την γραφική απεικόνιση τους μέσω δικής μας εφαρμογής. Στο πλαίσιο αυτό, θα δημιουργήσουμε μία απλή εφαρμογή σε Java ικανή να απεικονίσει αυτά τα δεδομένα.

Milestone 2

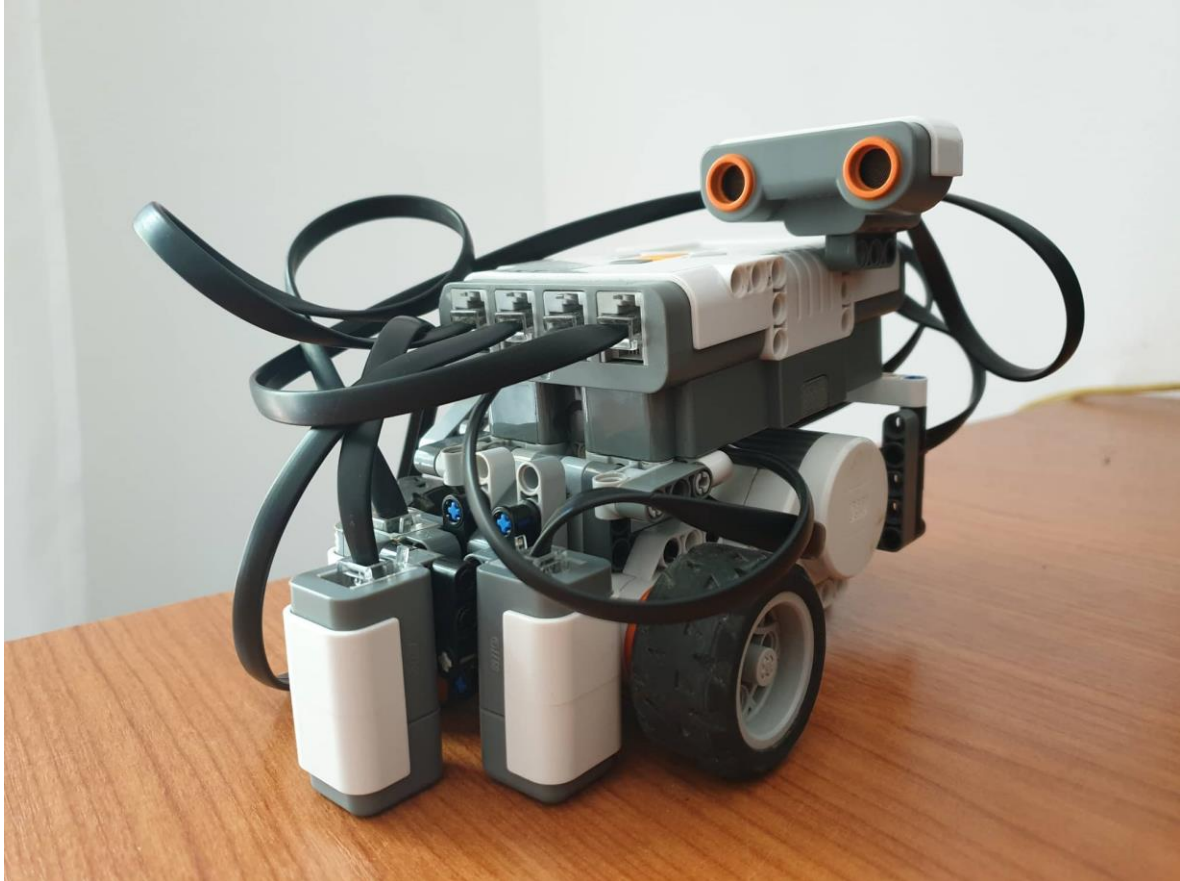
Μετά από υπόδειξη του διδάσκοντα, προκειμένου να γίνει πιο σωστός καταμερισμός εργασιών μεταξύ των milestones προβαίνουμε σε διαφοροποίηση της αρχικής μας πρότασης.

Συγκεκριμένα, τα παραδοτέα του milestone 2(**Ημ. Παράδοσης: 10 Μαΐου**) θα είναι :

Υλοποίηση εξαντλητικού αλγόριθμου για λαβύρινθους σε γλώσσα προγραμματισμού C, εφαρμογή σε Java για την απεικόνιση των τελικών δεδομένων, απλή κίνηση αυτοκινήτου σε μονοπάτι πάνω στον λαβύρινθο

Παράρτημα

Κατασκευή αυτόνομου αυτοκινήτου με χρήση τεχνολογίας Lego Mindstorms



Βιβλιογραφία

Βασικές γνώσεις σχετικά με ROBOTC: <http://help.robotc.net/WebHelpMindstorms/index.htm>

Forum με καθοδήγηση για προβλήματα που συναντήσαμε :

<http://www.robotc.net/forums/viewforum.php?f=1&sid=6decc57fb332bd1c7039d8a0840df4bb>

Line follower nxt: http://www.nxtprograms.com/line_follower/steps.html

Line follower vid: <https://www.youtube.com/watch?v=qLDKEjKZPDg>