

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ ΜΕ ΚΙΝΟΥΜΕΝΗ ΡΟΜΠΟΤΙΚΗ ΠΛΑΤΦΟΡΜΑ ΚΑΙ ΤΗ ΧΡΗΣΗ ΤΟΥ MSRS

Ευάγγελος Μπούκας, Αλέξανδρος Μπούκας, Αντώνιος Γαστεράτος,
Σπυρίδων Μουρούτσος

Τμήμα Μηχανικών Παραγωγής και Διοίκησης
Δημοκρίτειο Πανεπιστήμιο Θράκης
67100 Ξάνθη

ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή προτείνεται ένα προπτυχιακό μάθημα εργαστηριακού τύπου που ως σκοπό έχει την εισαγωγή και την εμπέδωση των βασικών αρχών λειτουργίας των ρομποτικών εξαρτημάτων και συστημάτων. Κατά τη διάρκεια του μαθήματος αυτού υλοποιείται μια εργασία με τη χρήση του περιβάλλοντος Microsoft Robotics Studio, που στόχο έχει την τηλε-πλοήγηση μιας ρομποτικής πλατφόρμας χαμηλής επεξεργαστικής ικανότητας, μέσω ενός εμπορικά διαθέσιμου επιταχυνσιόμετρου παιχνιδιομηχανής. Οι διδακτικοί στόχοι περιλαμβάνουν την κατανόηση του τρόπου λειτουργίας των βασικών μερών του MSRS δηλαδή του Concurrency and Coordination Runtime (CCR) και του Decentralized Software Services (DSS) καθώς και πιο εξεζητημένες μεθοδολογίες όπως η συνεργασία επεξεργαστών προγραμματισμένων σε διαφορετική γλώσσα, από το φοιτητή, όπως και την εξοικείωσή του με στοιχειώδες υλικό ρομποτικής.

Λέξεις κλειδιά: Εργαστηριακό μάθημα, Εκπαίδευση, Boe-Bot, Wiimote, Πλοήγηση, Επιταχυνσιόμετρο, MSRS, Microsoft Robotics Studio.

1 ΕΙΣΑΓΩΓΗ

Ο τρόπος λειτουργίας των ρομποτικών συστημάτων είναι αρκετά σύνθετος, αφού χρησιμοποιεί γνώση από πολλούς και διαφορετικούς τομείς της σύγχρονης τεχνολογίας. Ως εκ τούτου, η εκπαίδευση στις μοντέρνες ρομποτικές τεχνολογίες είναι μια επίπονη διαδικασία για το φοιτητή και, αν δεν εκτελεστεί με ολοκληρωμένες εκπαιδευτικές ασκήσεις, μπορεί να προκαλέσει σύγχυση ή ελλιπή γνώση σε κάποιον τομέα. Έτσι σε πολλά πανεπιστημιακά τμήματα οι εκπαιδευτικές ανάγκες στο μάθημα της ρομποτικής καλύπτονται με μαθήματα εργαστηριακού τύπου που διευκολύνουν τον φοιτητή στην κατανόηση και εμπέδωση της επιστημονικής γνώσης [1-5]. Στην παρούσα εργασία οργανώνεται μία τέτοια εργαστηριακή άσκηση, η οποία συνδυάζει την εξοικείωση του φοιτητή με μια απλή ρομποτική αυτοκινούμενη πλατφόρμα και με εξωτερικά αισθητήρια, που χρησιμοποιούνται στην πλοήγηση, καθώς επίσης και με ζητήματα προγραμματισμού που αφορούν στην κίνηση και τον έλεγχο των ρομπότ. Πιο συγκεκριμένα αρχικά παρουσιάζονται τα βασικά στοιχεία του μαθήματος, στη συνέχεια παρουσιάζεται το περιβάλλον εργασίας MSRS και τελικά περιγράφεται η διαδικασία υλοποίησης της εργαστηριακής άσκησης. Το πλεονέκτημα της προτεινόμενης άσκησης είναι ότι δεν απαιτεί ιδιαίτερα ακριβό εξοπλισμό για την υλοποίησή του, ενώ το λογισμικό που χρησιμοποιείται διανέμεται ελεύθερα από τον κατασκευαστή του.

2 ΒΑΣΙΚΟΙ ΑΞΟΝΕΣ ΤΟΥ ΜΑΘΗΜΑΤΟΣ

Τα ρομποτικά συστήματα αποτελούνται από διάφορα φυσικά εξαρτήματα και από το λογισμικό που καθορίζει τη λειτουργία τους. Η επιτυχημένη λειτουργία τους εξαρτάται, εκτός από την ποιότητα των φυσικών εξαρτημάτων, κυρίως από την ορθή παραλληλία και τον καλό συντονισμό των μερών του λογισμικού (Concurrency and Coordination) και από την ικανότητα διαχείρισης των σφαλμάτων (Error handling).

2.1 ΠΑΡΑΛΛΗΛΙΑ ΚΑΙ ΣΥΝΤΟΝΙΣΜΟΣ

Οι ανάγκες των ρομποτικών συστημάτων για παραλληλία και συντονισμό αποκλείουν τη χρήση ενός μονο-νηματικού (single-threaded) και σύγχρονου μοντέλου κώδικα καθώς συνήθως εκτελούνται διαδικασίες που είναι αλληλοεξαρτώμενες [7]. Αυτό σημαίνει ότι αν χρησιμοποιηθεί ένα τέτοιο μοντέλο θα πρέπει η κύρια διαδικασία να σταματά για να εκτελεσθεί μια επιμέρους εργασία που θα επιτρέπει την συνέχεια της κύριας. Για παράδειγμα στην πλοήγηση μιας ρομποτικής πλατφόρμας η απόφαση για την επόμενη κίνηση προϋποθέτει να έχει γίνει η επεξεργασία των δεδομένων που λαμβάνονται από τους αισθητήρες. Στο παράδειγμα αυτό αν χρησιμοποιήσουμε ένα τέτοιο μοντέλο θα πρέπει να σταματά η κίνηση της πλατφόρμας, να γίνεται η λήψη και η επεξεργασία των δεδομένων των αισθητήρων και στην συνέχεια να αποφασίζεται η επόμενη κίνηση. Είναι προφανές ότι μία τέτοια αντιμετώπιση προς το πρόβλημα είναι ασύμφορη και ως προς τον χρόνο εκτέλεσης της διαδικασίας αλλά και ως προς την αποτελεσματικότητα (π.χ. η ρομποτική πλατφόρμα δεν δέχεται καμία ανάδραση σε όλη τη διάρκεια της κίνησης της). Μία λύση αυτού του προβλήματος είναι η χρήση ενός πολύ-νηματικού ασύγχρονου (multi-threading asynchronous) μοντέλου, το οποίο όμως είναι πολύπλοκο, απαιτεί άριστη γνώση προγραμματισμού.

2.2 ΔΙΑΧΕΙΡΙΣΗ ΣΦΑΛΜΑΤΩΝ

Οι περισσότεροι προγραμματιστές ρομποτικών συστημάτων δοκιμάζουν τους αλγόριθμους τους σε προσομοιωτές κυρίως για λόγους ταχύτητας και ευκολίας, όμως συμφωνούν ότι τα αποτελέσματα της πραγματικής λειτουργίας διαφέρουν από τα αντίστοιχα μιας «αποστειρωμένης» προσομοίωσης. Αυτό συμβαίνει κυρίως λόγω των φυσικών συνθηκών που προκαλούν σφάλματα και λοιπές στοχαστικές διεργασίες. Στο παράδειγμα που προαναφέρθηκε ένας αντικατοπτρισμός ή κάποια άλλη συνθήκη μπορεί να αναγκάσει τους αισθητήρες να δώσουν δεδομένα μη ομαλά, εκτός ορίων ή ακόμα και κενά (null). Είναι προφανές ότι ένας συμβατικός κώδικας δεν θα μπορέσει να αντιμετωπίσει ικανοποιητικά τέτοιες συνθήκες. Στην επόμενη παράγραφο παρουσιάζεται η λειτουργία του MSRS και τονίζεται η ικανότητα του να ικανοποιεί τέτοιες ανάγκες.

3 MICROSOFT ROBOTICS STUDIO

Το MSRS είναι μια πλατφόρμα ανάπτυξης ρομποτικών εφαρμογών που βασίζεται σε 2 κύρια μέρη: το CCR και το DSS [6]. Το CCR παρέχει την υποδομή που επιτρέπει πολλαπλές εφαρμογές να εκτελούνται ταυτόχρονα σε έναν υπολογιστή, ενώ το DSS προσθέτει ένα επιπλέον επίπεδο για το συνδυασμό των εφαρμογών CCR σε δομές που ονομάζονται υπηρεσίες (services) και επιτρέπει την συνεργασία υπηρεσιών που εκτελούνται σε διαφορετικούς υπολογιστές που συνδέονται μέσω δικτύων.

3.1 CONCURRENCY AND COORDINATION RUNTIME

Το CCR είναι μία βιβλιοθήκη που παρέχει κλάσεις και μεθόδους που βοηθούν στο συντονισμό και τον ταυτοχρονισμό καθώς και στην διαχείριση των σφαλμάτων. Το CCR επιτρέπει να γράφονται τμήματα κώδικα που εκτελούνται ανεξάρτητα και στέλνουν μηνύματα (messages) όταν χρειάζεται να επικοινωνήσουν μεταξύ τους. Όταν λαμβάνεται ένα τέτοιο μήνυμα μπαίνει σε μια λίστα αναμονής (port) μέχρι να γίνει η επεξεργασία από το δέκτη (receiver). Κάθε ένα από τα τμήματα του κώδικα μπορεί να εκτελείται ταυτόχρονα και ασύγχρονα και δεν χρειάζεται να συγχρονιστούν λόγω της ύπαρξης των μηνυμάτων. Ο διαχειριστής (Dispatcher) είναι το μέρος του CCR που διαχειρίζεται τον υπολογιστικό φόρτο εργασίας. Ο διαχειριστής μπορεί να συγχρονίζει πολλαπλές διεργασίες με επιτυχία. Το CCR έχει ένα πολύ πρωτότυπο τρόπο να διαχειρίζεται τα σφάλματα, που ονομάζεται «αιτιότητα» (causality). Όποτε προκύψει κάποιο σφάλμα η αιτιότητα συνδέεται με το μήνυμα και το ακολουθεί σε όλη του την πορεία στα διάφορα τμήματα του κώδικα μέχρι το τέλος του προγράμματος. Μόλις εμφανιστεί το σφάλμα και δημιουργηθεί η αιτιότητα, ένα μήνυμα στέλνεται από τμήμα του κώδικα που την δημιούργησε και προς τα πίσω μέχρι το κώδικα που εκκίνησε τη διαδικασία.

3.2 DECENTRALIZED SOFTWARE SERVICES

Το DSS εισάγει ένα νέο όρο, τις υπηρεσίες οι οποίες είναι και το κύριο χαρακτηριστικό του MSRS. Οι υπηρεσίες είναι τμήματα κώδικα που συνεργάζονται μέσω του CCR για να υλοποιήσουν μία διαδικασία. Οι υπηρεσίες μπορούν να σχετίζονται με υλικό, να περιέχουν δηλαδή όλες τις διαδικασίες που επιτρέπουν στο MSRS να αναγνωρίσει μία συμβατή συσκευή ή να είναι τμήματα κώδικα που κάνουν υπολογισμούς, όπως η επεξεργασία των δεδομένων των αισθητήρων. Μια υπηρεσία απαρτίζεται από τα εξής δεδομένα: το μοναδικό αναγνωριστικό (Service Identifier), τον τύπο (Contract Identifier), τις διάφορες μεταβλητές (State), τα αναγνωριστικά των συνεργατών (Partners) και τον υποδοχέα αιτημάτων (Main Port). Ο υποδοχέας αιτημάτων δέχεται δεδομένα είτε από άλλες υπηρεσίες με τις οποίες συνεργάζεται για τον συντονισμό τους, είτε αυτόματα μηνύματα αλλαγής της κατάστασης μιας συνεργαζόμενης υπηρεσίας. Το DSS παρέχει το πλαίσιο συνεργασίας των υπηρεσιών το οποίο λειτουργεί μέσω του πρωτοκόλλου Decentralized Software Services Protocol (DSSP).

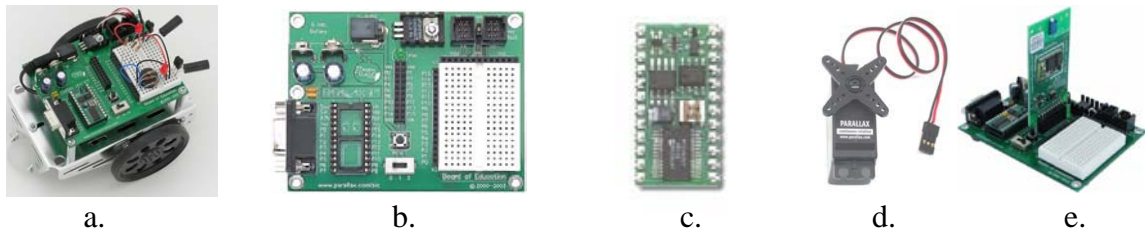
4 ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

Η εργαστηριακή άσκηση έχει ως σκοπό την πλοήγηση της ρομποτικής πλατφόρμας χαμηλής επεξεργαστικής ικανότητας στο περιβάλλον εργασίας του MSRS. Για τον τηλεχειρισμό της πλατφόρμας χρησιμοποιείται ένα επιταχυνσιόμετρο τριών αξόνων.

4.1 ΕΠΙΛΟΓΗ ΦΥΣΙΚΩΝ ΕΞΑΡΤΗΜΑΤΩΝ

Κατά την επιλογή των φυσικών εξαρτημάτων κύριος γνώμονας ήταν η απλότητα και η συμβατικότητα των υλικών έτσι ώστε οι φοιτητές να αισθανθούν οικεία ως προς την εργασία.

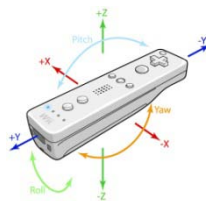
4.1.1 ΡΟΜΠΟΤΙΚΗ ΠΛΑΤΦΟΡΜΑ



Εικόνες 1: a. Boe-Bot, b. Boe-Bot's board, c. BasicStamp2, d. Servos, e. Bluetooth on board.

Η ρομποτική πλατφόρμα που επιλέχθηκε είναι το Boe-Bot της Parallax [8]. Ο επεξεργαστής του Boe-Bot είναι ο BasicStamp2 ο οποίος προγραμματίζεται σε μία ιδιότυπη γλώσσα τύπου Basic, την PBASIC. Δύο σέρβο-κινητήρες της parallax τοποθετημένοι back-to-back και συνδεδεμένοι σε δύο τροχούς προσφέρουν τη διαφορική κίνηση της πλατφόρμας, ενώ ένας μη μηχανικός τροχός στο πίσω μέρος παρέχει την απαιτούμενη στήριξη. Με ένα κατάλληλο κύκλωμα στο ενσωματωμένο breadboard του boe-bot η πλατφόρμα μπορεί να αποκτήσει υπέρυθρους αισθητήρες απόστασης για την αποφυγή σύγκρουσης. Για την σύνδεση με τον υπολογιστή εκτός από θύρα RS32 υπάρχει και το eb500 Bluetooth Module.

4.1.2 ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΟ



Εικόνα 2: Wiimote

Ως μονάδα τηλεχειρισμού επιλέχθηκε το τηλεχειριστήριο της Nintendo για την παιχνιδιομηχανή Wii, το Wiimote. Τα δεδομένα που λαμβάνονται από το Wiimote είναι ικανοποιητικά ακριβή, το κόστος του είναι μικρό και η αναγνωσιμότητα του από τους φοιτητές είναι μεγάλη.

4.2 ΠΛΑΝΟ ΕΡΓΑΣΙΑΣ

Αρχικά πρέπει να συνδέσουμε τις συσκευές μας. Συνδέοντας τις συσκευές το Boe-Bot θα αναγνωριστεί ως μια άγνωστη συσκευή Bluetooth και το Wiimote θα αναγνωριστεί κανονικά ως χειριστήριο παιχνιδιομηχανής. Για την υλοποίηση της εργασίας χρειάζονται η εξής διαδικασίες:

1. Αναγνώριση της άγνωστης συσκευής ως μικροεπεξεργαστή (Basic stamp 2),
2. Ορισμός του μικροεπεξεργαστή και των συνδεδεμένων κινητήρων και αισθητήρων ως ρομπότ διαφορετικής κίνησης και παροχή όλων εντολών για την κίνηση του.
3. Ανάγνωση των δεδομένων του επιταχυνσιόμετρου και μετάφραση σε εντολές κίνησης για το ρομπότ.

4.3 ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΥΠΟΛΟΓΙΣΤΗ, ΔΗΜΙΟΥΡΓΙΑ ΥΠΗΡΕΣΙΩΝ ΚΑΙ ΕΚΤΕΛΕΣΗ.

Όλες οι διαδικασίες αποτελούνται από τμήματα κώδικα που συγκροτούν υπηρεσίες. Συνεπώς για κάθε μία από τις διαδικασίες θα δημιουργηθεί μία υπηρεσία, μια για την πρώτη με το όνομα «Boe-Bot BASIC Stamp 2», μια για την δεύτερη με το όνομα «Boe-Bot Generic Drive» και μία για την τρίτη με το όνομα «Wiimote».

Ο κώδικας για τις δύο πρώτες υπηρεσίες προέρχεται από την προηγούμενη έκδοση του MSRS την 1.5 [6] και ανανεώθηκε από τους συγγραφείς για να είναι συμβατός με το MSRS 2008 R2 ενώ ο κώδικας για το Wiimote προέρχεται από μία αντίστοιχη εργασία του Alberto Bietti [9].

4.3.1 ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΥΠΟΛΟΓΙΣΤΗ

Για την υλοποίηση της εργασίας είναι απαραίτητη η χρήση ενός υπολογιστή με λειτουργικό Windows και με δυνατότητα σύνδεσης Bluetooth. Επίσης είναι απαραίτητη η εγκατάσταση του Microsoft Visual Studio 2008 [10] και του Microsoft Robotics Studio 2008 R2 [11].

Το επόμενο βήμα είναι η ενσωμάτωση (migration) των εργασιών (projects) που θα δημιουργήσουν τις υπηρεσίες. Αρχικά θα πρέπει να αντιγραφούν οι φάκελοι Wiimote και Parallax στη θέση "C:\Users\qubi\Microsoft Robotics Dev Studio 2008 R2\Projects". Έπειτα θα πρέπει να εκτελεστεί μέσω του DSS COMMAND PROMPT η εντολή: "dssprojectmigration <project_location>", με την οποία θα ανανεωθούν όλες οι ρυθμίσεις που είναι μεταβλητές από υπολογιστή σε υπολογιστή (όπως το όνομα χρήστη και η τοποθεσία εγκατάστασης του MSRS). Στη συνέχεια πρέπει να ανοιχθούν με το Microsoft Visual Studio τα αρχεία Basicstamp.sln και Wiimote.sln.

4.3.2 ΔΗΜΙΟΥΡΓΙΑ ΥΠΗΡΕΣΙΩΝ

Οι υπηρεσίες δημιουργούνται μέσω του Visual studio οικοδομώντας την εργασία που περιέχει ένα αρχείο κώδικα με συγκεκριμένη δομή με όλα τα χαρακτηριστικά τους. Ένα τέτοιο αρχείο είναι το BasicStamp2.cs. Τα αρχεία αυτά χρησιμοποιούν συγκεκριμένες βιβλιοθήκες που περιέχουν συνήθειες εντολές και πρέπει να δηλώνονται στην αρχή του κώδικα.

4.3.3 ΣΥΝΤΟΝΙΣΜΕΝΗ ΕΡΓΑΣΙΑ ΜΕΣΩ ΤΟΥ DSS HOST

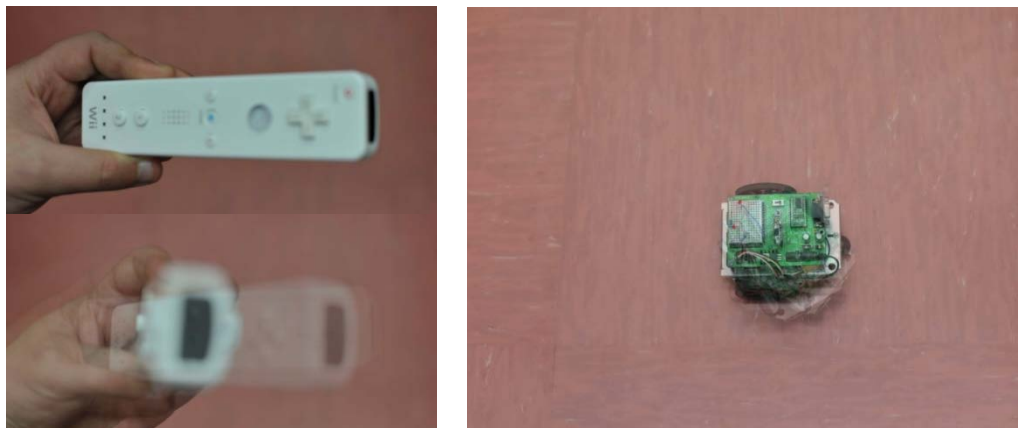
Σε αυτό το σημείο της εργασίας, αφού έχουν οικοδομηθεί και οι δύο εργασίες, έχουν δημιουργηθεί όλες οι υπηρεσίες του ρομποτικού συστήματος. Το τελευταίο βήμα είναι να φορτώσουμε το περιβάλλον εργασίας του DSS το DSS host και από αυτό να καλέσουμε όλες τις υπηρεσίες που είναι απαραίτητες για τη λειτουργία του συστήματος. Αν όλα τα στάδια έχουν εκτελεσθεί ορθά το Boe-Bot πρέπει να ανταποκρίνεται στην κίνηση του Wiimote.

4.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΓΑΣΙΑΣ.

Στις εικόνες που ακολουθούν παρουσιάζονται τα αποτελέσματα της εργασίας. Στο αριστερό τμήμα των εικόνων 3 και 4 απεικονίζεται η κίνηση του τηλεχειριστηρίου, ενώ στο δεξιό τμήμα απεικονίζεται το αποτέλεσμα αυτής στη ρομποτική πλατφόρμα.



Εικόνα 3: Κίνηση εμπρός



Εικόνα 3: Περιστροφή αριστερά

5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρουσιάστηκε μια εργαστηριακή άσκηση που υλοποιεί τον τηλεχειρισμό μιας πλατφόρμας Boe-bot με χρήση του χειριστηρίου Wiimote. Για την άσκηση επίσης απαιτείται ένας υπολογιστής με δυνατότητα σύνδεσης Bluetooth και εκτέλεσης του λογισμικού MSRS. Σκοπός της εργασίας ήταν να δημιουργηθεί ένα μάθημα το οποίο θα βοηθήσει τον καθηγητή στην παράδοση αλλά πολύ περισσότερο θα κεντρίσει το ενδιαφέρον των φοιτητών και θα τους βοηθήσει να κατανοήσουν με απλά βήματα τον σχεδιασμό, τον προγραμματισμό, και την λειτουργία ενός ρομποτικού συστήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Istvain Matijevics, Education of Mobile Robot Architecture, INES 2006- 10th International Conference on Intelligent Engineering Systems, London 2006, pp 180 – 183.
2. Bingchen Qi , Yanqiu Dong , Lina Chen , Wei Qi , O. Yoshikuni., The Impact of Robot Instruction to Education Informatization, 1st International Conference on Information Science and Engineering (ICISE), Nanjing 2009, pp 3497 – 3500.

3. D. Lukac, L. Banjanovic-Mehmedovic, Realization of the action-oriented education approach by programming a microcontroller steered mobile robot, XXII International Symposium on Information, Communication and Automation Technologies, ICAT2009, Bosnia 2009, pp 1 – 7.
4. C. Cardeira, J.S. da Costa, A low cost mobile robot for engineering education, 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005, p 6.
5. M. Sajkowski, T. Stenzel, B. Grzesik, Walking robot HEXOR® II - a versatile platform for engineering education, Power Electronics and Motion Control Conference, EPE-PEMC-2008, Poznan 2008, pp 956 – 960. Xuemei Li, Gang Xu, Interdisciplinary Innovative Education Based on Modular Robot Platform, ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM '08, Guangzhou 2008, Vol 2, pp 66 – 69.
6. Kyle Johns, Trevor Taylor (2008), “Professional Microsoft Robotics Developer Studio (Wrox Programmer to Programmer)”, Wiley Publishing, Inc., Indianapolis, Indiana.
7. Rahbaran, B., Steininger, A. (2009), Is Asynchronous Logic More Robust Than Synchronous Logic?, Dependable and Secure Computing, IEEE Transactions on, vol. 6, issue. 4, pp.282-294.
8. The Boe-Bot® robot from Parallax, official web page: <http://www.parallax.com/tabid/411/Default.aspx>
9. Alberto Bietti (2007), Lego NXT + wiimote with MSRS tutorial, retrieved from Alberto Bietti's blog: <http://alandtech.blogspot.com/2007/11/lego-nxt-wiimote-with-msrs-tutorial.html>
10. Microsoft Visual Studio official web page: <http://msdn.microsoft.com/en-us/vstudio/default.aspx>
11. Microsoft Robotics Studio official web page: <http://www.microsoft.com/robotics/#Robots>