California State University, Chico
College of Engineering, Computer Science, and Technology

Intelligent Systems Laboratory
O'Connell Technology Center, Room 431
400 West First Street
Chico, California 95929-0410
Tel. 530 898-4619 Fax. 530 898-5995

DATE: September 26, 2004

TO: Dr. Ken Derucher, Dean, College of ECT

cc: Dr. Orlando Madrigal, Chair, Computer Science
Dr. Greg Kallio, Chair, Mechanical and Mechatronic Engineering
Mr. Jesse Medrano, Administrative Analyst, CSUC Research Fdn.

isL
intelligent
systems
laboratory

FROM: Dr. Benjoe Juliano, Director, ISL
Associate Professor, Computer Science
Juliano@csuChico.edu

SUBJECT: NSF Progress Report, Spring – Summer 2004

I am pleased to share with you a copy of our NSF Progress Report for MRI/RUI Project #0321385, *Acquisition of robotics equipment for an Intelligent Systems Laboratory*, which encompasses Spring and Summer 2004.

On behalf of my co-PIs, Drs. Renee Renner and Ramesh Varahamurti, and the ISL Research Team, I would also like to take this opportunity to thank you all for your continued encouragement and support of our NSF Project. We all look forward to a fruitful and successful Fall 2004 semester as we wrap up the first half of our three-year project.

Benjoe Juliano

got bots?
http://isl.ecst.csuchico.edu/

NSF Progress Report
Spring & Summer 2004

Acquisition of robotics equipment for an Intelligent Systems Laboratory
(MRI/RUI 0321385)

August 15, 2004

NSF Program: EIA – MRI/RUI

NSF Award Number: 03-21385

Report period coverage: January 15, 2004 – August 15, 2004

PI Name: **B.A. Juliano**, Juliano@csuChico.edu
California State University, Chico

Co-PIs: **R.S. Renner**, Renner@csuChico.edu
California State University, Chico

R. Varahamurti, rVarahamurti@csuChico.edu
California State University, Chico

PI Address: Department of Computer Science
California State University, Chico
400 West First Street
Chico, CA 95929-0410

1 OVERVIEW

This progress report covers the second half of the first year of a project that aims to acquire robotics equipment for an *Intelligent Systems Laboratory* (ISL) that will facilitate the development of cross-disciplinary courses and provide exciting collaborative research opportunities for both students and faculty at California State University, Chico. This report covers various ISL activities utilizing basic and intermediate robotics equipment, including information regarding a course in *Robotics and Machine Intelligence* offered by the PIs last Spring 2004 semester and a *Summer Robotics Camp* for junior high school girls offered last June 2004.

2 ORGANIZATION

For the Spring 2004 semester, the following students were (re-)hired as ISL student research assistants (RAs):

- Matt Bauer, Mechanical and Mechatronic Engineering, undergraduate
- Felipe Jauregui, Computer Science, graduate
- Elena Kroumova, Computer Science, graduate
- Andy Whaples, Computer Science, undergraduate
- Alexis Winston, Computer Science, graduate

The three ISL PIs and all the ISL student RAs collectively make up the ISL research group. Andy had already been involved with numerous ISL activities as a volunteer. Matt, Felipe, Elena, and Alexis were re-hired from the Fall 2003 semester. Elena changed status from an undergraduate student to a graduate student. Although the NSF proposal only allotted for two (2) graduate RAs, we accounted for three (3) because both Alexis and Felipe have reduced work hours[1] for the Spring 2004 semester. Hence, Elena was a full-time graduate RA with the ISL, while Alexis and Felipe were half-time graduate RAs.

Most of the ISL activities for the Spring 2004 semester were dedicated to the following tasks:

- setup and organization of ISL equipment in
  - the *Intelligent Systems/Mechatronics Laboratory* in OCNL 431
  - the ISL "Closet Space" in OCNL 244A
- preparations, planning and course delivery of CSCI 244, *Robotics and Machine Intelligence*; and
- preparations and planning for the ISL Summer Robotics Camp for junior high school girls.

1 Felipe was hired by the Department of Computer Science to work part-time as an instructor and as system administration personnel.

For the Summer 2004 semester, the following students were (re-)hired as ISL student research assistants (RAs):

- Matt Bauer, Mechanical and Mechatronic Engineering, undergraduate
- Charissa Garcia, Mechanical and Mechatronic Engineering, undergraduate
- Dan Hirsmuller-Counts, Computer Science, undergraduate
- Felipe Jauregui, Computer Science, graduate
- Elena Kroumova, Computer Science, graduate
- Alexis Winston, Computer Science, graduate

Additionally, the ISL had two visiting international students from the Polytechnic University of Marseilles, France, who participated in ISL activities June-July 2004:

- Marin Laviolette
- Quentin Pelen

Andy Whaples graduated in the Spring 2004 semester, so his position was awarded to Dan who had already been involved with numerous ISL activities as a volunteer. Matt, Felipe, Elena, and Alexis were re-hired from the Spring 2004 semester. Although the NSF proposal only allotted for two (2) graduate RAs, we accounted for three (3) because both Alexis and Elena have reduced work hours[2] for the Summer 2004 semester. Hence, for the summer, Felipe was a full-time graduate RA with the ISL, while Alexis and Elena were half-time graduate RAs. We accounted for three (3) undergraduate RAs by keeping Matt as a full-time undergraduate RA with the ISL, while Charissa and Dan were half-time undergraduate RAs.

Additionally, since the PIs were not granted summer stipends by the College of ECST, Felipe Jauregui was appointed lead RA of the student RAs. Felipe did an outstanding job in spear-heading the curriculum development and organization of the Summer Robotics Camp. He also met with the other student RAs regularly to ensure that everyone was on track.

Most of the ISL activities for the Summer 2004 semester were dedicated to the following tasks:

- offering of the Summer Robotics Camp for junior high school girls;
- assessment of CSCI 244, *Robotics and Machine Intelligence* taught in the Spring, implementing any necessary and forseeable adjustments;
- development of research agenda for members of the ISL research group; and
- continue setup and organization of ISL equipment in
    - the *Intelligent Systems/Mechatronics Laboratory* in OCNL 431
    - the ISL "Closet Space" in OCNL 244A.

2 Elena was working part-time in another position on campus.

3 FINANCIAL REPORT

3.1 Overview

Table 1 below summarizes overall expenditures (NSF grant award) for the first year of this project. The information in Table 1 is limited to expenditures related to (a) equipment and operating expenses, and (b) salaries for student research assistants.

Table 1: Summary of Year #1 NSF Grant Expenditures.

	Fall 2003	Spring 2004	Summer 2004	Totals
A. Equipment and Operating Expenses				
Basic Robot Kits	$3,425.18	$0.00	$0.00	$3,425.18
Intrm Robot Kits	$44,091.43	$1,784.75	$0.00	$45,876.18
Advan Robot Kits	$0.00	$0.00	$0.00	$0.00
Subtotal	**$47,516.61**	**$1,784.75**	**$0.00**	**$49,301.36**
B. Salaries for Student Research Assistants				
Undergraduate	$1,848.00	$2,768.00	$5,716.00	$10,332.00
Graduate	$4,140.00	$3,180.00	$4,992.00	$12,312.00
Subtotal	**$5,988.00**	**$5,948.00**	**$10,708.00**	**$22,644.00**
Totals	**$53,504.61**	**$7,732.75**	**$10,708.00**	**$71,945.36**

3.2 Equipment and Operating Expenses

As mentioned in the prior *Progress Report* for this project [7], the focus of the first year of this project was the immediate acquisition of both Basic and Intermediate Robot Kits. The main reasons for these purchases were (a) to quickly supply the ISL with robotics equipment; (b) to expose ISL student research assistants to robotics kits that are relatively easy to get familiar with; and (c) to avail of robotics kits that can readily be used for workshops, seminars, and short courses. These kits included LEGO *Mindstorms Robotics Invention System 2.0* [22], TAB Electronics *Build Your Own Robot Kits* [29] powered by NetMedia *BasicX-24* microcontrollers [25], Parallax *BOE-Bot Full Kits* [27], Lynxmotion *Carpet Rover / Lynx 5 Combo Kits* [23], Lynxmotion *Hexapod 1 OOPic-R Combo Kits* [23], Sony *AIBO ERS 220A* quadrupeds [28], and *K-Team Khepera II* minirobots [21]. The $49,301.36 expenditure for robotics equipment for the first year represents approximately 28.26% of the $174,482 total awarded for robotics equipment.

As planned and proposed in the original NSF proposal, advanced robotics equipment are scheduled to be purchased during the Fall 2004 semester.

3.3 Salaries for Student Research Assistants

As seen in Table 1, expenditures for salaries for undergraduate research assistants exhibited an upward trend (49.78% increase from the Fall 2003 to Spring 2004, and 100.06% increase from Spring 2004 to Summer 2004). This upward trend slightly corresponds to the number of students hired at those times: two (2) for Fall 2003, two (2) for Spring 2004, and three (3) for Summer 2004. The discrepancy between expenditures for the Fall 2003 and Spring 2004 semesters, despite having the same number of undergraduate RAs, is explained by the fact that one of the students failed to submit timesheets for the Fall 2003 semester even though this student put in numerous hours working for this project. Three (3) undergraduates were hired for Summer 2004 to make up for the "slack" in hours of the prior two semesters.

From Table 1, the trend for expenditures for salaries for graduate RAs is different (23.19% decrease from Fall 2003 to Spring 2004, and 56.98% increase from Spring 2004 to Summer 2004). There were two (2) graduate RAs for Fall 2003, three (3) for Spring 2004, and the same three (3) for Summer 2004. The trend for expenditures for salaries for graduate RAs does not correspond with the number of graduate students hired because one of the three graduate RAs for the Spring 2004 and Summer 2004 semesters was incorrectly classified by the University Payroll system as an undergraduate. So, this student was getting paid the undergraduate RA rate (further explaining the trend for salaries for undergraduate RAs). Further, another of the three graduate RAs did not consistently turn in timesheets since this RA was not able to contribute much hours to this project.

Overall, only 63.18% of the monies allocated for salaries for student RAs for the first year of the project was actually spent. This does not reflect, in any way, the total number of student hours devoted to the first year of the project. CSU, Chico is a primarily teaching institution where students (and faculty) are not used to research assistantships. Students would sometimes either forget to submit a timesheet for a particular pay period, or feel embarrassed to submit actual hours spent working on the project. The PIs will keep closer tabs on these for the remainder of the project.

3.4 Other Expenditures

Through a $4,468 CSUC Office of Sponsored Programs research grant [6] awarded to Dr. Juliano last December 8, 2003 the PIs were able to purchase items they overlooked in their original proposed NSF budget. The items acquired through this internal grant were subdivided into four categories: (1) items for storing robotic equipment, accessories, and spare parts; (2) rechargeable power sources; (3) materials for robotics competitions; and (4) computing equipment. After allocating $364.88 for photocopying needs (primarily materials for the Summer Robotics Camp), the rest of the internal research award was allocated as presented in Table 2. Each category is elaborated on next.

Table 2: Allocations for the CSUC internal research award.

Qty	Description	Unit Price	Subtotal
Category 1: Items for storing robotic equipment, accessories, and spare parts			
12	16" toolbox, Keter	$4.99	$59.88
30	19" toolbox, Stanley	$7.97	$239.10
6	22" toolbox, Rubbermaid	$7.00	$42.00
1	6" tilt bin	$9.93	$9.93
1	Small parts box	$9.97	$9.97
6	12 pc tool kit	$16.95	$101.70
1	12 sq ft padding material	$2.97	$2.97
Category 2: Rechargeable power sources			
2	AA 36 pack	$13.49	$26.98
6	AA 4 pack rechargeable	$8.97	$53.82
3	Battery charger	$19.98	$59.94
6	AA 8 count rechargeable	$24.99	$149.94
2	NiMh RC charger	$21.99	$43.98
Category 3: Materials for robotics competitions			
1	6 pack 60 min mini digital tape	$36.99	$36.99
2	½ x 75 mounting tape	$3.97	$7.94
2	35 electric tape, red	$2.97	$5.94
2	35 electric tape, yellow	$3.27	$6.54
2	35 electric tape, green	$3.27	$6.54
2	Mailing tape	$2.97	$5.94
1	Tape	$5.97	$5.97
4	Tape	$5.98	$23.92
1	Tape 2"x45yd	$3.97	$3.97
Category 4: Computing equipment			
1	Toshiba A45S151 IP4 2.8GHz notebook	$1,349.99	$1,349.99
1	Toshiba A45S250 IP4 2.8GHz notebook	$1,499.00	$1,499.00
1	Samsonite L45 Plus notebook case	$49.99	$49.99
		Subtotal	$3,802.94
		Tax	$275.71
		TOTAL	$4,078.65

Category 1: Storage for equipment, accessories, and spare parts.

The materials in this category are being used for proper storage of ISL robotics kits in closets and for organized distribution in the classroom/lab. Ten of the (small) 16" toolboxes are being used for storing ten Parallax *BOE-Bot full kits* [27]. The remaining two are being used to store rechargeable batteries and battery chargers (see Category 2 below). The 30 (medium) 19" toolboxes are being used for storing 30 LEGO *Mindstorms* kits. Finally, the six (large) 22" toolboxes are being used for transporting various other robots, accessories, and spare parts when members of the ISL Research Group are invited to present their robots in a lecture/demo. These 22" toolboxes will be useful when students begin competing in robotics competitions. When travel is involved with these demos/lectures/competitions, padding material will be used in the toolboxes to protect the delicate circuitry of the robots. The bin and small parts box are being used for storing accessories and robot spare parts.

Category 2: Rechargeable power supplies.

Most robot kits purchased and used in this project drain their power sources (batteries) rather quickly. To reduce/eliminate student costs associated with the purchase of batteries, the ISL has a supply of rechargeable batteries and battery chargers. These were useful not just in the classroom, but also in the Summer Camp hosted by the ISL.

Category 3: Competition supplies.

Robotics competitions were the highlight of the robotics class co-taught by the PIs. The most fundamental competitions involved 2-dimensional mazes constructed by using (multi-)colored tape to represent walls on the floor. Sometimes, these 2-D mazes were made on multiple pieces of construction paper taped together to form a larger maze. Mounting tape was used to secure obstacles on the 3-D maze developed by a member of the ISL Research Team. Mini digital tapes were used to record/document various ISL-related teaching, research, and outreach activities.

Category 4: Computing equipment.

Two notebook computers make up this category. The main reason for purchasing these laptops is to facilitate downloading programs to ISL robots during robotics competitions, presentations, and/or demonstrations on-site or on the road. Two notebooks were purchased primarily in anticipation of having multiple teams entered in student robotics competitions. Additionally, ISL Research Team members are expected to participate in presentations, lectures, demonstrations, and research conferences which may coincide with robotics competitions.

4 EDUCATIONAL ACTIVITIES

4.1 CSCI 224: Robotics and Machine Intelligence

During the Spring 2004 semester, the PIs offered CSCI 224, *Robotics and Machine Intelligence* [12] as a co-taught course. CSCI 224 was scheduled for Tuesdays and Thursdays, 11:00 am to 12:15 pm in OCNL 431, the College of ECST's Intelligent Systems and Mechatronics Laboratory where the ISL is housed. The course was scheduled to be co-taught by the three PIs, but Dr. Varahamurti was not able to participate due to unavoidable health-related circumstances (he was not able to teach in the Spring 2004 semester). Hence, Dr. Juliano and Dr. Renner employed the help of the ISL Project's lead graduate student research assistant, Felipe Jauregui. The course was designed to provide students with a multidisciplinary experience working hands-on with more than one robotics platform in solving problems that relate to search and rescue.

The robotics platforms used in CSCI 224 during the Spring 2004 semester were:

- Parallax *Boe-Bot Full Kit* [27]; and
- LEGO *Mindstorms RIS 2.0 Kit* [22].

The Parallax *Boe-Bot Full Kit* includes one textbook published by Parallax [10] (replacing the two books [8,9] available with the earlier version) that was used to supplement the CSCI 224 curriculum. Negotiations with Parallax resulted in a bulk-rate discount rate of $165/kit. These were purchased through CSUC's A.S. Bookstore and were sold to students for $199/kit under the *Computerworks* section of the A.S. Bookstore. The LEGO *Mindstorms RIS 2.0 Kits* used were kits of the ISL.

Thirty-three (33) students were signed up for CSCI 224 last Spring 2004. The distribution of students' registered in the course is given in Table 3.

Table 3: Distribution of Students' Class Standing in CSCI 224, Spring 2004.

Class Standing	Count	Percentage
Freshman	0	0.00%
Sophomore	0	0.00%
Junior	4	12.12%
Senior	19	57.58%
Graduate	10	30.30%
Total	33	100.00%

Table 4: Distribution of Undergraduate Majors in CSCI 224, Spring 2004.

Undergraduate Major	Count	Percentage
Computer Information Systems (CINS)	2	8.69%
Computer Engineering (CMPE)	1	4.35%
Computer Science (CSCI)	5	21.74%
Computer Science Option in Computer Graphics Programming (CSCI/CSGA)	4	17.39%
Computer Science Option in Math/Science(CSCI/CSMS)	2	8.69%
Electrical Engineering (EE)	2	8.69%
Mathematics Option in Applied Mathematics (MATH/APLD)	1	4.35%
Mechanical Engineering (ME)	1	4.35%
Mechatronic Engineering (MECA)	5	21.74%
Total	23	100.00%

The distribution of majors for the 23 undergraduate students registered in the course last Spring 2004 is summarized in Table 4. According to Table 4, of these 23 undergraduate students, 14 (60.87%) were from the Department of Computer Science, six (26.07%) were from the Department of Mechanical Engineering, Mechatronic Engineering, and Manufacturing Technology, and one (4.35%) from a department outside the College of ECT.

Figure 1: Sample "Convoy Line Follower" setup.

Most of the earlier part of the semester was devoted to short lectures followed by the distribution of Activity Sheets. These Activity Sheets were designed to encourage hands-on activities where students had to investigate and experiment with their robotics kits to answer questions and solve problems. Some of these activities involved control and calibration of servo motors, distance control and ramping, using photoresistors and infrared detectors, frequency sweeps, and others.

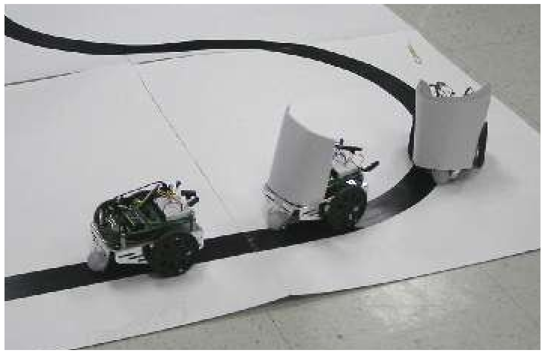


Figure 2: A “convoy” of *Boe-Bots* navigating a line.

Another major component of the course were the competitions. For each robot platform, the students had an opportunity to compete individually and as a team. Further, for each competition, students were required to write a report (see Appendix A for some samples) detailing their design, implementation, and suggestions for improving their robot based on its performance in the competition.

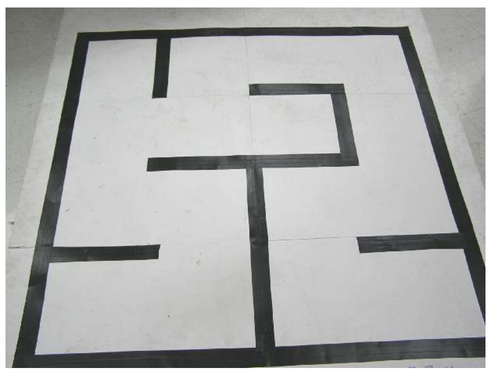


Figure 3: Sample “2D Maze Challenge” setup.

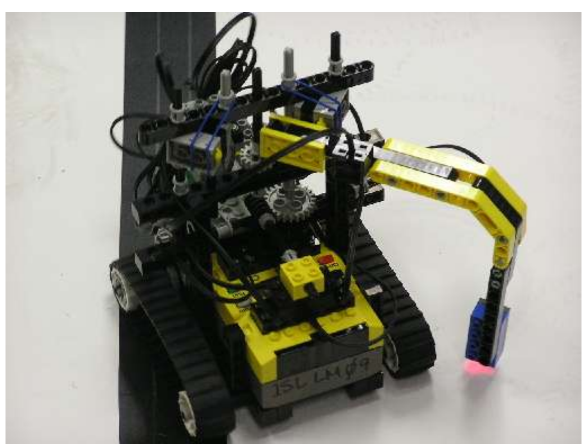


Figure 4: A *Mindstorms* robot scanning for “2D Maze” walls.

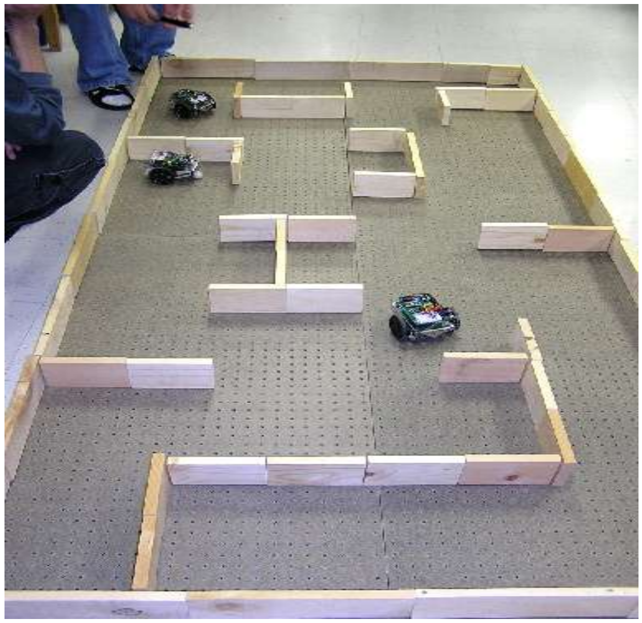


Figure 5: Sample “3D Maze Challenge” setup.

In the Student Evaluation of Teaching (SET), students commented that the class was fun, challenging, and interesting. Students also noted lots of opportunity to experiment and learn (by discovery) on their own. Some pointed out that the course provided great hands-on opportunities to program/work with something tangible.

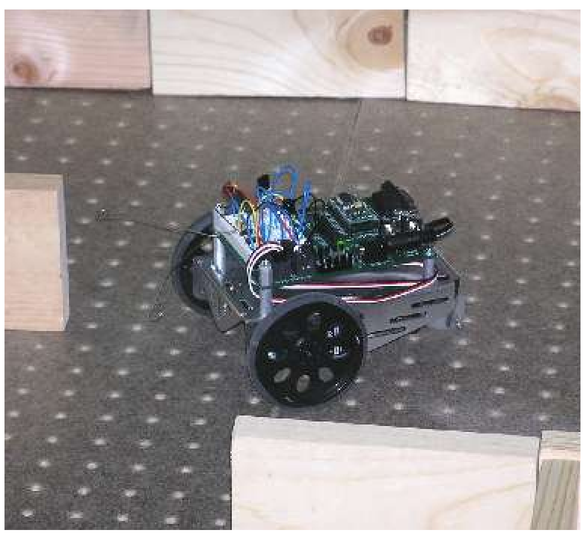


Figure 6: A *Boe-Bot* attempting the “3D Maze Challenge”.

The most common suggestions from the students were:

1. Spend additional time on machine intelligence algorithms.
2. Consider replacing the Parallax *Boe-Bot* platform with one that is more "reliable."
3. Consider requiring the LEGO *Mindstorms* platform as the "textbook" rather than the Parallax *Boe-Bot*.

The PIs also noted that students were not used to projects pertaining to the design and implementation of autonomous agents. In particular, some students were eager to get "precise" specifications for the competitions (which were called "Challenges"). The PIs had to use the Mars Exploration Rover Mission [20] to remind students that the most interesting missions for autonomous robots are those where the environment is unknown. In such cases, all that may available is some (possibly unreliable) information about the environment.

4.2 CSCI 233: Graphical User Interface Implementation

Dr. Challinger was awarded a CSUC internal grant [3,4] to explore potential research proposal development and classroom use of robotics equipment in the ISL. Last Spring 2004, Dr. Challinger used one of the ISL's K-Team *Khepera II* minirobots [21] in a programming project (see Appendix B for detailed specifications) for CSCI 233, *Graphical User Interface Implementation*.

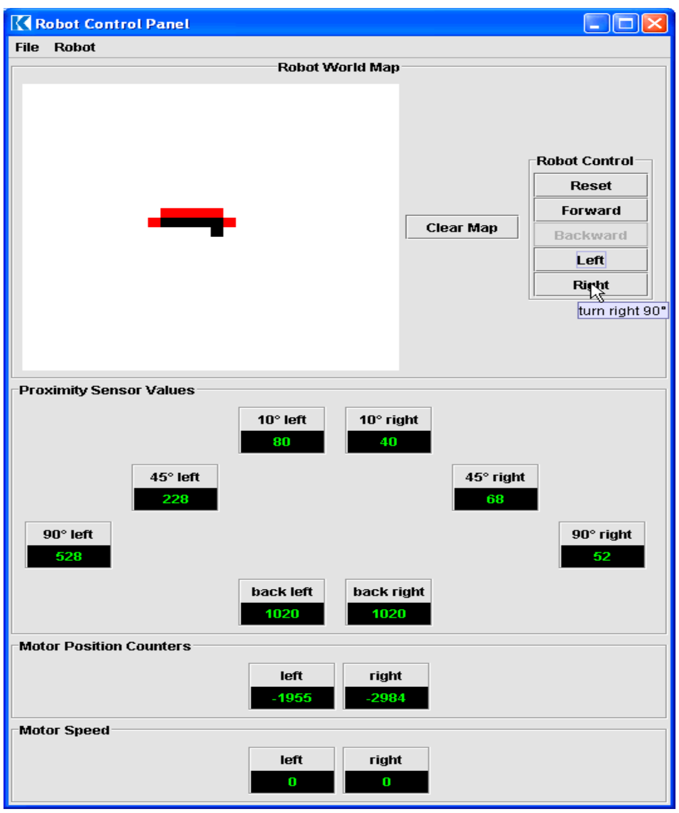


Figure 7: Prototype GUI for teleoperation of a *Khepera II*.

In this class project, students designed and implemented a GUI (see Figure 7) to control a *Khepera II* robot that was remotely located in Dr. Challinger's office, connected to a computer via a serial port. Dr. Challinger had a robot server running that accepted connections over the network and performed the necessary communications with the robot. A JAR file was provided with the robot client software.

An online survey (see Appendix B) was conducted to assess students' response to the GUI project. Overall, the CSCI 233 students found the project relevant, interesting, and they learned a lot regarding GUI design and implementation for teleoperation of robots.

5 RESEARCH ACTIVITIES

While most of Fall 2003 and portions of Spring 2004 were spent on purchasing robotics equipment, setup, assembling, and inventorying, the ISL Research Team managed to pursue various research activities most especially over the Summer 2004.

5.1 Precision Farming

Precision farming is the use of unmanned, autonomous vehicles in farming. The purpose of this project [1] is to focus on capacity building research that would facilitate the development of an intelligent ground vehicle capable of autonomously navigating in a field for a variety of tasks (e.g. pesticide application, plowing, irrigation, harvesting, etc.). This project has received CSU Agricultural Research Initiative (ARI) funding on the condition that the proponents acquire industrial partners.

5.2 Robotics and Visualization

Intelligent robotics systems need intelligent vision systems in order to be considered intelligent. In particular, autonomous mobile robots for search and rescue missions need highly coordinated and sophisticated vision systems. The focus of this project is to secure and incorporate visualization equipment in the ISL. Dr. Judy Challinger (Computer Science) is working with Dr. Juliano for both graphics-related uses/integration of ISL robots as well as potential funding to secure visualization equipment for research and instruction [3,4].

5.3 Lynxmotion Hexapod Development Platform

Matt Bauer, Mechatronics Engineering undergraduate RA, has devoted a significant amount of time developing and understanding the Lynxmotion *Hexapod 1 OOPic-R Combo Kit* [23]. One of his projects is to get two or more Hexapods to walk and navigate with Infrared sensors in the same environment. Each Hexapod employs a two emitter and single collector infrared proximity detector module. The problem with the IRPD modules is that when two or more Hexapods outfitted with the same sensors are in the same environment, signals emitted by each IRPD module affect the others around it. A solution to this issue could be to code each emitted signal with a unique sequence of bits. Hence, a robot would emit a unique sequence of bits and only register an object/obstacle if this robot collects the same sequence of bits.

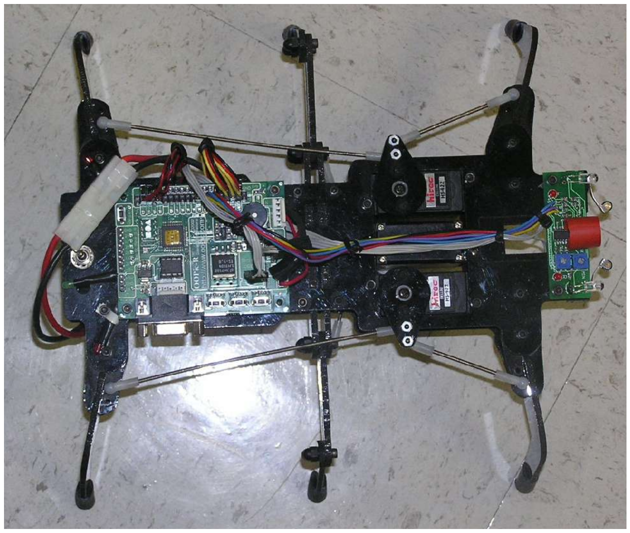


Figure 8: One of ISL's Lynxmotion *Hexapod 1*s [23].

5.4 Lynxmotion Carpet Rover Development Platform

Matt Bauer and Dan Hirsmuller-Counts, Computer Science undergraduate RA, have assembled and experimented with the Lynxmotion *Carpet Rover / Lynx 5 Combo Kits* [23]. The first minor problem stemmed from the servos in the kit not being continuous rotation servos. A plastic stop inside each servo had to be cut off and the polarity of the left servo had to be reversed by unsoldering the wires to the motor, flipping them, and then resoldering the wires back to the motor. Dan also had problems with the gripper being unbalanced and the kit not having enough standoffs to mount both the IRPD and the line following modules.

Dan also wrote code to test the robot's functionality. These tests included movement with the servos, obstacle avoidance with the IR sensor, line following with the line following module, and picking up an object with the robotic arm. The servos and the IRPD module were the easiest to test. The line following module seems to be pre-calibrated and relies strongly on having a white surface with a black line. Tests show that the line following module is calibrated to be much too sensitive to the white level that anything even slightly off white is registered as black. The spacing of the sensors on the line following module is very good for this particular robot, and the middle sensor allows the robot to detect a "T" in the line with great ease.

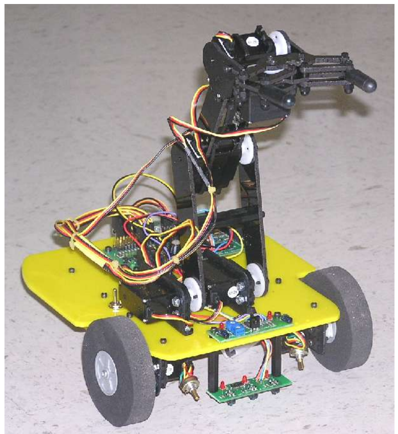


Figure 9: One of ISL's Lynxmotion *Carpet Rover*s [23].

The arm seems to be the most complicated *Carpet Rover* part to program. All the servos for the arm are controlled by a servo controller through a serial input and clock line. Although it is relatively easy to create and send the correct serial packets to the arm servos, care must still be taken so as not to move the arm in such a way as to damage either the arm or the robot.

5.5 Building Sensors for the Parallax Sumobot

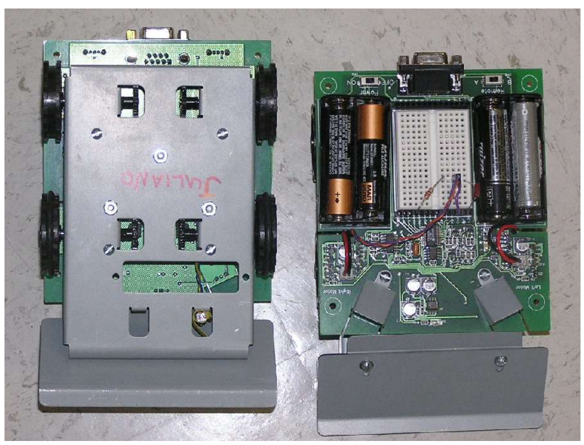


Figure 10: Two of ISL's TAB Electronics *Sumo Bot*s [30] each outfitted with an additional photo cell sensor.

Matt, Dan, and Felipe Jauregui, Computer Science graduate RA and lead RA, are experimenting with the TAB Electronics *Sumo Bot* [30] platform. The primary objective is to have a line of battle-ready Sumo Bots that abide by Tournament Rules and Regulations of the FSI (FujiSoft ABC Inc.) – All Japan Robot-Sumo Tournament [15]. The student RAs had to develop and implement the use of photo cell sensors to detect the border of the sumo ring (*Dohyo*). Using a photo cell allows the robot to receive a variable numerical value to distinguish between the black arena or white border of the *Dohyo*.

Figure 11: Matt, Dan, and Felipe discussing ways to install a photo cell sensor on a *Sumo Bot* [30].

Currently, the ISL Sumo Bots employ an automatic calibration behavior for their initial movements. Specifically, assuming that the sumo bot always starts on a white border region, the sumo bot approximates the border region by sampling the white region then traveling forward into the black region and sampling it.

5.6 NetMedia BasicX-24 Port for the Tab Electronics BYORK

Based upon suggestions from fellow roboticists, the PIs decided to purchase TAB Electronics *Build Your Own Robot Kits* [29] (a.k.a. BYORKs) powered by NetMedia *BasicX-24* microcontrollers (BX24) [25]. The BYORKs are typically controlled by a Parallax *Basic Stamp 2* module (BS2) [26], the microcontroller for which this robot platform was designed to operate with. Out of the box, the BYORK is designed to be used with the BS2. One can use the BS2 to control all the actions of the BYORK by using the standard communication protocol of the PBasic language [8].

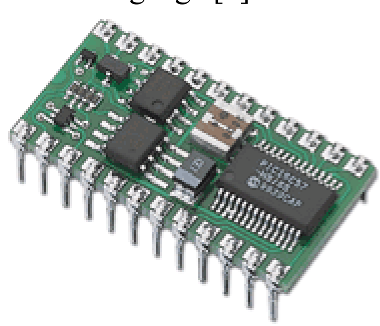


Figure 12: Parallax *Basic Stamp 2* module [26].

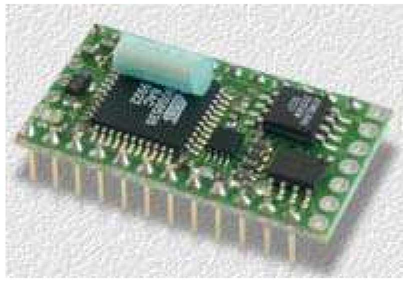


Figure 13: NetMedia *BasicX-24* microcontroller [25].

The BX24 is a more superior microcontroller than the BS2. For example, the BX24 has 32KB of EEPROM, 400 bytes of RAM, executes 65000 instructions per second, and has a maximum program length of 8000+ instructions; whereas, the BS2 has only 2KB of EEPROM, 32 bytes of RAM, executes 4000 instructions per second, and has a maximum program length of about 500 instructions. Both the BX24 and the BS2 are packaged as 24-pin DIP modules.

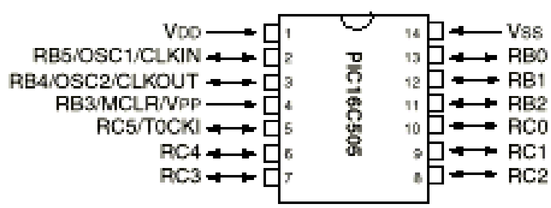


Figure 14: Microchip *PIC16C505* microcontroller [24].

Onboard the BYORK is a Microchip PIC 16C505 [24] microcontroller that has built in capabilities for controlling the BYORK's motors, infrared module and the CDS photocells. Also coded on the co-processor are four behaviors. These four behaviors, and the values associated with the infrared module and photocells, can be easily called from and controlled by a BS2; this was not the case for the BX24. The BX24 had a similar set of communication protocols, but they were too fast that the BX24 was communicating faster than the PIC co-processor could perform any tasks. Dr. Juliano and Matt Bauer ventured to create a new communication protocol for the BX24 that will allow it to send and receive data from the BYORK's co-processor. The resulting BX24 module (see Appendix D), based on the BS2 template written by Myke Predko (see Appendix C), has been fully tested to get all the data and send all the commands needed to fully control the BYORK from a BX24. This BX24 module will be released to the public this Fall.

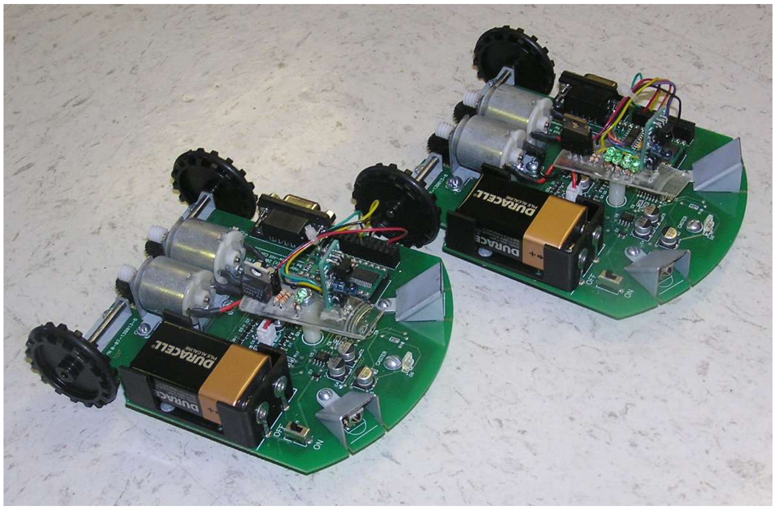


Figure 15: Two of ISL's TAB Electronics *BYORK*s [29] outfitted with additional "home brewed" sensors.

The development of a BX24 module for the BYORK facilitates the next project with the BYORK platform. The next project entails multi-robot, swarm-style communication with a master/slave system that would allow a master robot to control fifteen slave robots. Data sharing/remembering and teamwork search and rescue are concepts that will be explored.

5.7 Experiments with the SONY AIBO Development Platform

Figure 16: Felipe working on a Sony AIBO ERS-220 [28].

Dr. Renner is working with Felipe in an attempt to develop a mobile facial recognition system on the Sony AIBO platform. Felipe has been exploring the limitations of the platform and its programming capabilities, including wireless communication and interface, agility, movement, speed, and processing power. The overall intention is to have an AIBO roam an area and scan for people. Once a person has been detected, the AIBO would take a picture of the facial region. This image would then be scaled down using Fuzzy logic and identified against any known images by a Neural Network.

An additional research project is to get the AIBO robots programmed for a future RoboCup competition. This involves communication between AIBOs as well as tactics that can be modified on the fly given an opponents' formation.

5.8 Setting up a Linux Cluster for Parallel Soft Computing Applications

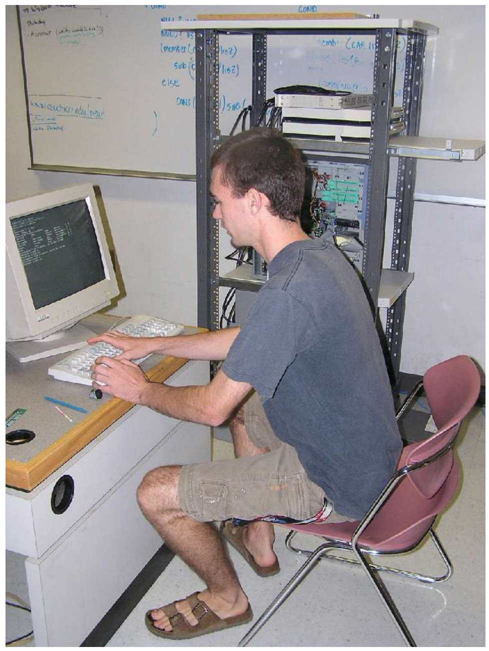


Figure 17: Dan configuring the ISL Linux Beowulf cluster.

Dan has been working on configuring the ISL Linux Beowulf Cluster with PVM software [11,16]. The cluster consists of four (4) dual-processor HP Kayak workstations. Three of the workstations are powered by dual Pentium II processors running at 200 Mhz each. The fourth machine runs two Pentium III processors both running at 600 Mhz.

The HP hardware was somewhat difficult to work with. The Gentoo Linux distribution [17] failed to install due to write access problems with the hard drives. An install of the Debian Linux distribution [13] worked, however the kernel that ships with Debian's "Woody" release does not have SMP support. Attempts to re-compile the Debian kernel with SMP support worked, but the network was lost! This issue was not resolved, and since the network is the other crucial part of a machine that is to be clustered, Dan decided to give up on Debian. However, the Slackware Linux distribution [28] is working great; the default kernel has been re-compiled to add SMP support. Additionally, Dan has successfully cloned the Pentium II machines (besides the original) over the cluster's network.

6 OUTREACH AND DISSEMINATION ACTIVITIES

6.1 Half-Day Workshop for Richvale Elementary School

Last May 28, the ISL Research Team sponsored an "Introduction to Robotics" seminar for thirty GATE (Gifted And Talented Education) kids in grades 6-8 from Richvale Elementary School (300 B St., Richvale, CA 95917). The ISL Research Team provided these kids with a two hour long mini-seminar that was hosted at OCNL 431 in the CSU, Chico campus. The main objective was to teach the kids (and due to interest, the teachers and parent chaperons as well) about robotics. Felipe Jauregui delivered the materials and the rest of the Team monitored kids as coaches.

The platform used in the seminar was the Lego Mindstorms Robotics Invention System 2.0 kits [22]. This platform was exciting for the kids, as well a great learning aid. The seminar began with an introductory lecture about existing robots and robots they may have seen on television and/or the movies. Robot navigation through sensors and actuators was also presented. This part of the presentation led to the focus of how a robot could "find" its way out of a maze. This was demonstrated with coding principles. Participants were grouped into teams of two and were given the challenge of programming their robot to solve two different walled mazes: a simpler maze and a more difficult maze. Due to time limitations, pre-built *Tank-Bots* (see Figure 18) were made available for the participants to use. We recognized the fact that building the robot is half the fun; therefore, we had the participants build the touch sensor whiskers that would allow them to program their robot to "bump" its way through a maze.

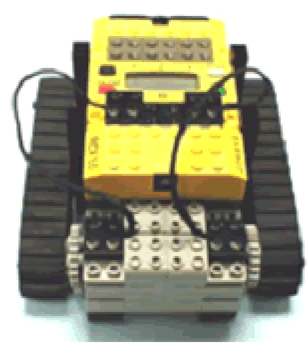


Figure 18: Pre-built *Tank-Bot*.

Although a tutorial on programming was given, the students were given a template RIS drag-and-drop GUI program that could be modified to solve the maze. This template had to be modified in terms of deciding the direction and speed of the two motors given a particular sensor input. This modification was made using the ideas presented in the lecture, as well as a trial and error process.

The kids were able to navigate the "easy" maze and two groups were able to solve the "difficult" maze. Overall, the students were very pleased with the seminar. The contact instructor immediately asked if it would be possible to return next semester with a different group of kids for the same presentation, as well as the same group for an advanced seminar.

6.2 Career Workshop for Hamilton Union High School

Last April 30, Felipe Jauregui gave a short forty-five minute presentation to high school students at Hamilton Union High School (Hwy. 32 And Canal St., Hamilton City, CA 95951). The robotic presentation was a part of a larger seminar geared towards promoting a college education. The students present were of Hispanic descent and would all be first generation college students. Thus, the main focus was to show them what they could learn in college and the benefits of a higher education.

The presentation focused on what they could do in college with robotics, how a robot works, real versus television robots, majors related to robots and careers in robotics. The learning experience was enhanced by demonstrating four robotic platforms: LEGO Mindstorms [22], Tab Electronics BYORKs [29], Parallax Boe-Bot [27] and the TAB Electronics Sumo Bot [30]. The students were very excited about seeing a real robotic interact with a new environment. They had several questions and asked about how to start learning about robotics in High School.

6.3 Research Internship with École Polytechnique Universitaire de Marseille

The PIs were able to secure an agreement for a research initiation internship with the École Polytechnique Universitaire de Marseille's Department de Genie Industriel et Informatique (Industrial and Computer Engineering). Part of this agreement is to have two of their students, Marin Laviolette and Quentin Pelen, come to Chico State and work with the ISL Research Team during the months of June and July 2004 (see Appendix E for copies of related documents). These two students experienced working with a research team in the US. They participated as coaches for our Summer Robotics Camp (see the next subsection) and conducted preliminary research regarding some of the ISL-related problems of interest. Their experience culminated with their submission of individual research papers on search and rescue experiments with ISL robots.

6.4 Week-Long Summer Robotics Camp for Junior High School Girls

The first ISL-sponsored *Summer Robotics Camp* for junior high school girls was held June 14-18, 2004 in the CSU, Chico campus rooms OCNL 431 and OCNL 124. The ISL Research Team developed a week-long, interactive *Summer Robotics Camp* to provide girls going into the 8th grade with the unique opportunity to learn more about science and math. The camp was designed to introduce these young women to the fields of Computer Science, Engineering, and Mechatronics through their learning, experimentation, building and use of robots via LEGO Mindstorms Robotics Invention System 2.0 kits [22]. Daily activities and lessons allowed participants to work as a member of a team in the lab where they learned more about the science of robotics.

In addition to the eight (8) members of the ISL Research Team for the summer, there were four (4) volunteers who helped in coordinating this effort. Two of the volunteers were research interns from France, and the other two were Chico State students (a Computer Science graduate student and a Mechatronics Engineering undergraduate student) who applied for volunteer positions for the Summer Camp (see the Mentor Application Form in Appendix F). Hence, there was a total of twelve staff members: two faculty (PIs Juliano and Renner), four graduate students, and six undergraduate students. The PIs designated Felipe Jauregui as Head Coach while the rest of the staff were Coaches. All twelve staff members were present for the week-long camp.

Figure 19: Staff for the ISL Summer Robotics Camp for Junior High School Girls.
Left to right: Sandeep Batra (CSCI grad), Marin Laviolette (visiting from France), Quentin Pelen (visiting from France), Matt Bauer (MECA undergrad), Joel Amato (seated; MECA undergrad), Felipe Jauregui (CSCI grad), Charissa Garcia (seated; MECA undergrad), Elena Kroumova (CSCI grad), Dr. Renner (PI), Dan Hirsmuller-Counts (CSCI undergrad), Dr. Juliano (PI), and Alexis Winston (CSCI grad).

Of the fifteen girls who applied and expressed interest in participating in the Summer Camp, twelve girls showed up. The ISL Research Team was very pleased with this group of girls not only because the number was good for the first offering of the Camp, but also because they were able to provide one-on-one interaction with the girls when needed. The distribution of Summer Camp participants is summarized in Table 5. The data in Table 5 points out that a majority (75%) of the Summer Camp participants were from cities other than Chico. Although all but one of the participants came from Butte County, the staff was very happy to have a participant come all the way from Siskiyou County!

Table 5: Distribution of Summer Camp Participants, Summer 2004.

Grade 7th	Grade 8th	School Name	School District	County
4		Biggs Middle School	Biggs Unified	Butte
1		Bidwell Junior High School	Chico Unified	Butte
1	1	Chico Junior High School	Chico Unified	Butte
3		Durham Intermediate School	Durham Unified	Butte
1		Paradise Intermediate School	Paradise Unified	Butte
1		Grenada Elementary	Grenada Unified	Siskiyou

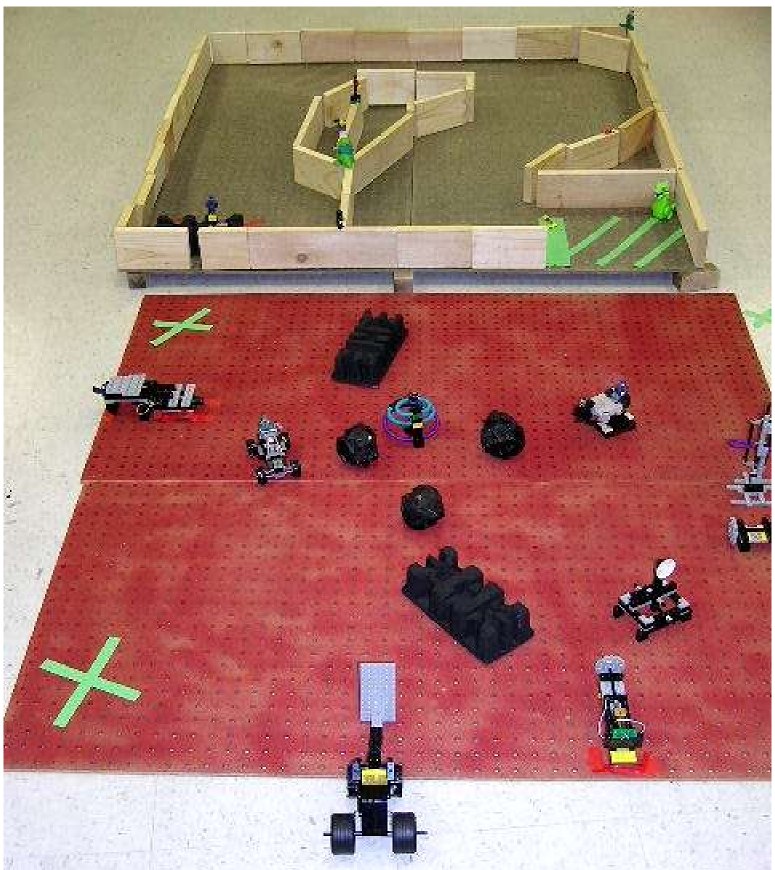


Figure 20: Simulated "Mission Mars" setup in OCNL 124.

To facilitate management of the camp, the girls were assigned into groups of two. This allowed the staff to form a "lunch crew" for each Camp day without heavily affecting the number of coaches available to assist each group. Additionally, coaches assigned to groups were rotated daily so that participants got to interact with every coach in the staff.

Figure 21: ISL Summer Camp participants navigating their Mars Tank-Bot on the non-autonomous portion of the "Mission Mars" challenge.

Since the NSF grant award only covers funding for robotic equipment for the ISL and student RA salaries, the ISL Research Team had to solicit additional funding to provide lunch to the Summer Camp participants. Felipe Jauregui spearheaded the effort to acquire funds through a Lunch Donations Program (see Appendix F). Drs. Gary and Judy Sitton, members of the College of ECST Advisory Board, provided enough funding for lunch for the duration of the camp. Dr. Kenneth Derucher, Dean of the College of ECST, provided gifts for the participants to take home and a pizza party for the last day of the camp. Dr. Orlando Madrigal, Chair of the Department of Computer Science, provided partial funds for T-shirts, which were purchased at discounted rates from a local company, Sundog Screenprints. The Associated Students of CSUC provided ice for drinks distributed during the Camp.

The Camp culminated in a simulated "Mission Mars" Challenge (see Figures 20 and 21) where the participants had to rescue fellow astronauts involved in a failed Mars mission. This Challenge consisted of a teleoperated segment and an autonomous segment. Prior to the competition, participants had to practice operating their robots via remote control to simulate teleoperation of their robots similar to the Mars rover teleoperation performed by NASA. Tasks requiring simulated teleoperation included clearing an area for landing, transferring the rescued astronauts to a safe location, and others. For the autonomous segment of the competition, participants had to write code that allowed their robots to autonomously navigate through a walled maze (simulating a cave in Mars) using only bumpers connected to touch sensors. Samples of the related website content and supplementary documents for the Summer Robotics Camp are provided in Appendix F.

The ISL Research Team considers the Summer Robotics Camp a huge success mostly due to immediate feedback from the participants. Additionally, the Camp was featured at both a write-up on a local paper [2] and coverage in a couple of local news channels. At least a couple of participants indicated they would someday like to be an engineer working on robots.

6.5 Other Ventures on Schedule

There are other ongoing projects the ISL Research Team are involved in. One of them is a presentation on “Careers in Robotics” for Live Oak High School (2351 Penington Road, Live Oak, CA 95953) scheduled for this coming Fall semester. Also for this Fall is a presentation for the CSUC Anthropology Forum entitled “Anthropology and Intelligent Systems.” Still another one is a project devoted to creating a simulator for the Parallax *Boe-Bot* [27] in a maze environment. The objective of this project is to develop a simulation program, similar to eyeWyre’s *Simulation Studio* [14], that students could use to test their code in a virtual maze that would be similar to a real competition maze. The simulator would facilitate faster results of performance, and thus allow the student to better understand maze solving algorithms.

7 REFERENCES

ISL-Related Research Proposals, Presentations, and Progress Reports

[1] J. Arthur, M. Spiess, and R. Varahamurti, "Autonomous farming using intelligent ground vehicles (precision farming)," CSU Agricultural Research Initiative (ARI) proposal, 2003.

[2] R.H. Aylworth, "Girl power: 12 teenagers save mock Mars mission," *Enterprise Record*, June 19, 2004.

[3] J. Challinger, "Exploring our new robotics lab for research proposal development and classroom use," CSUC Office of Sponsored Programs faculty development grant proposal, 2003.

[4] J. Challinger, "Establishment of a research program on cooperating autonomous mobile robots," CSUC Office of Sponsored Programs research proposal, 2003.

[5] M. Harmon, "Robots come to Chico," *Chico Statements*, Volume 10, Number 1, pp. 8-11, Spring 2004 http://www.csuchico.edu/pub/cs/

[6] B.A. Juliano, "Acquiring additional equipment for the Intelligent Systems Laboratory (ISL)," CSUC Office of Sponsored Programs faculty development grant proposal, 2003.

[7] B.A. Juliano, R.S. Renner, and R. Varahamurti, "Acquisition of robotics equipment for an Intelligent Systems Laboratory," NSF Progress Report, December 2003.

Other Referenced Documents

[8] Parallax, Inc., *BASIC Stamp Manual, version 2.0*, http://www.parallax.com/detail.asp?product_id=27218.

[9] Parallax, Inc., *Robotics with the Boe-Bot! Text, version 1.5*, http://www.parallax.com/detail.asp?product_id=28154.

[10] Parallax, Inc., *What's a Microcontroller? Text, version 2.1*, http://www.parallax.com/detail.asp?product_id=28152.

Miscellaneous URLs

[11] Computer Science and Mathematics (CSM) Division, Oak Ridge National Laboratory, Parallel Virtual Machine (PVM) http://www.csm.ornl.gov/pvm/

[12] CSCI 224: *Robotics and Machine Intelligence* website http://www.ecst.csuchico.edu/~juliano/csci224/

[13] Debian Linux distribution http://www.debian.org/

[14] eyeWyre Corporation, Simulation Studio http://eyewyre.com/studio/

[15] FujiSoft ABC Inc. (FSI) – All Japan Robot-Sumo Tournament, *Tournament Rules and Regulations* http://www.fsi.co.jp/sumo-e/out/outa0000.html

[16] A. Geist *et al.*, *PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing*, Oak Ridge National Laboratory http://www.netlib.org/pvm3/book/pvm-book.html

[17] Gentoo Linux distribution http://www.gentoo.org/

[18] The Institute for Research on Intelligent Systems (IRIS) website http://iris.ecst.csuchico.edu

[19] The Intelligent Systems Lab (ISL) website http://isl.ecst.csuchico.edu

[20] Jet Propulsion Laboratory, California Institute of Technology, "Mars Exploration Rover Mission," http://marsrovers.jpl.nasa.gov/

[21] K-Team website http://k-team.com

[22] The LEGO Mindstorms website http://mindstorms.lego.com/eng/

[23] Lynxmotion website http://lynxmotion.com

[24] Microchip PIC 16C505 website http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010119

[25] NetMedia BasicX-24 website http://www.basicx.com/

[26] Parallax BASIC Stamp 2 Module http://www.parallax.com/detail.asp?product_id=BS2-IC

[27] Parallax Boe-Bot Full Kit – EDU Discount http://www.parallax.com/detail.asp?product_id=EDU-28132

[28] The Slackware Linux Project http://www.slackware.com/

[29] Sony AIBO website http://www.us.aibo.com

[30] TAB Electronics Build Your Own Robot Kit http://www.tabrobotkit.com or http://books.mcgraw-hill.com/getbook.php?isbn=0071387870

[31] TAB Electronics SUMO BOT http://www.tabrobotkit.com or http://books.mcgraw-hill.com/getbook.php?isbn=0071411933

A P P E N D I C E S

Appendix A: Sample course materials from Computer Science 224, *Robotics and Machine Intelligence*, Spring 2004.

Appendix B: Programming Assignment #3, Specifications for developing a GUI for teleoperation of a Khepera II minirobot, Computer Science 233, *Graphical User Interface Implementation*, Spring 2004.

Appendix C: PBASIC code template for a TAB Electronics *Build Your Own Robot Kit* controlled by a Parallax *Basic Stamp 2*.

Appendix D: BasicXpress module for a TAB Electronics *Build Your Own Robot Kit* controlled by a NetMedia *BasicX-24*.

Appendix E: Documents on the research internship agreement between Chico State's ISL and École Polytechnique Universitaire de Marseille, June-July, 2004.

Appendix F: Sample materials from the *First ISL Summer Robotics Camp for Junior High School Girls*, June 14-18, 2004.

CALIFORNIA STATE UNIVERSITY, CHICO

CSCI 224: Robotics and Machine Intelligence

Abbreviated Syllabus for Spring Semester 2004

Visit http://www.ecst.csuchico.edu/~juliano/csci224 *for additional detail.*

Prerequisites

- **CSCI majors:**
  - CSCI 51A (Assembly Language Programming)
  - CSCI 151 (Algorithms and Data Structures)
- **ECE / ME / MECA majors:**
  - ECE 86 (Processor Architecture and Assembly Language Programming)
  - ECE 90 (Algorithms and Programs for Engineers)

Description

3 units. This course introduces students to the field of robotics by emphasizing the task of endowing machines with intelligence. Topics include various case studies of robot architectures and algorithms that facilitate embodying a robot with behaviors that are traditionally associated with human cognition (*e.g.*, perception, reasoning, intelligent navigation, vision, learning, etc.). Students will conduct robotics experiments and compete in robotics games.

TRACS Call #	Section	Days	Time	Room	Instructors
15273	CSCI 224-01	TR	11:00 am – 12:15 pm	OCNL 431	Dr. B.A. Juliano (Juliano@csuChico.edu) Dr. R.S. Renner (Renner@csuChico.edu) Dr. R. Varahamurti (rVarahamurti@csuChico.edu)

Instructor Information

Dr. B.A. Juliano (*a.k.a.* Dr. J)
OCNL 222, Tel 530 898-4619 / 6442 (dept office)
http://www.ecst.csuchico.edu/~juliano

Dr. R.S. Renner
OCNL 226, Tel 530 898-5419 / 6442 (dept office)
http://www.ecst.csuchico.edu/~renner

Dr. R. Varahamurti
OCNL 418, Tel 530 898-6353 / 5346 (dept office)
http://www.ecst.csuchico.edu/~rameshv

Office Hours: Please check with the instructors for details.

Required Textbook

Boe-Bot Full Kit – EDU Discount.
Stock Number EDU-28132.
Parallax Inc., Rocklin, CA.
http://www.parallax.com

Additional Requirements

1. Students are expected to open and maintain a Chico State Connection (CSC) Portal (see http://portal.csuchico.edu) account in order to access up-to-date on-line calendar of events, current scores, discussion board, etc.
2. Students are expected to have acquired their own *Boe-Bot Full Kit* by the second week of classes (week of February 2). Please note that the instructors negotiated with Parallax for a special volume discount for the *Boe-Bot Full Kit*. The kits will be available at this discounted price from the *ComputerWorks* section of the A.S. Bookstore.

Grade Evaluation

This is a project-centered course. A total of at least three (3) projects will be assigned during the semester. Some projects will be individual projects while others will be group/team projects. Additionally, some projects may involve competition with other individuals/teams in the class. Each project must be accompanied by a detailed written report and possibly a web-enabled version of the report. Students are expected to be ready to present their project(s) orally when asked to.

Additional Information

http://www.ecst.csuchico.edu/~juliano/csci224/
http://isl.ecst.csuchico.edu/
http://portal.csuchico.edu/

Appendix A

CALIFORNIA STATE UNIVERSITY, CHICO

CSCI 224: *Robotics and Machine Intelligence*

Spring 2004 Schedule Information

TRACS Call#	Section	Act	Days	Time	Room	Instructor(s)
15273	CSCI 224-01	DIS	TR	11:00-12:15	OCNL 431	Juliano Renner Varahamurti

Note: For instructor(s) office hours, room location(s), phone number(s) and e-mail(s), please click on the instructor(s) name(s) above to visit their respective web pages.

Prerequisites

CSCI majors	CSCI 51A (Assembly Language Programming) CSCI 151 (Algorithms and Data Structures)
ECE / ME / MECA majors	ECE 86 (Processor Architecture and Assembly Language Programming) ECE 90 (Algorithms and Programs for Engineers)

Description

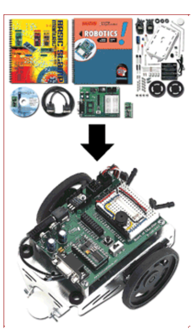
3 units. This course introduces students to the field of robotics by emphasizing the task of endowing machines with intelligence. Topics include various case studies of robot architectures and algorithms that facilitate embodying a robot with behaviors that are traditionally associated with human cognition (*e.g.*, perception, reasoning, intelligent navigation, vision, learning, etc.). Students will conduct robotics experiments and compete in robotics games.

Required Accounts

Students officially registered for the course will have their own *Chico State Connection (CSC Portal)* account.

Students are responsible for regularly checking their *WebCT* account (automatically generated through the *CSC Portal*) to access an up-to-date on-line calendar of events, current scores, on-line quizzes, etc.

Required Text(s)



Boe-Bot Full Kit - EDU Discount
Stock Number EDU-28132
Parallax, Inc., Rocklin, CA.

NOTE: The instructors negotiated a special volume discount for this item. The kits will be available at this discounted price from the *ComputerWorks* section of the A.S. Bookstore.

Course Objectives

The objectives of this course are to:

1. help students develop the necessary skills needed to apply learned fundamentals of robotics to the design, implementation, and analysis of simple robots;
2. introduce students to an algorithmic understanding of robot manipulation and control; and
3. provide students with the basic framework for further/advanced study and research in the field of robotics.

Course Outcomes

Upon successful completion of this course, the student shall be able to:

1. apply learned fundamentals of robotics to the design, implementation, and analysis of simple robots;
2. apply learned algorithmic understanding of robot manipulation and control; and
3. pursue further/advanced study and research in the field of robotics.

Grade Evaluation

This is a project-centered course. A total of at least three (3) projects will be assigned during the semester. Some projects will be individual projects while others will be group/team projects. Additionally, some projects may involve competition with other individuals/teams in the class. Each project must be accompanied by a detailed written report and possibly a web-enabled version of the report. Students are expected to be ready to present their project(s) orally when asked to.

Final Grades

Final grades shall be expressed as a percentage of the maximum possible score of all evaluated materials. Letter grades will be given according to the University definition of letter grading symbols (please refer to the *University Catalog* for detailed information).

Topical Coverage

Students will get hands-on experience working with the following programmable robot platforms:

- LEGO Mindstorms Robotics Invention System (RIS) Kit
- Parallax Boe-Bot Full Kit
- TAB Electronics Build Your Own Robot Kit (BYORK)

For each robot platform, students will learn:

1. robot platform design philosophy and robot anatomy
2. robot microcontroller features
3. microcontroller programming environment
4. I/O interfaces
5. control and robot programmable behaviors
6. programming intelligent solutions/algorithms

For additional information and resources for the above robotics kits, check out the *Intelligent Systems Lab* (ISL) website at `http://isl.ecst.csuchico.edu/`.

This page designed and maintained by Dr. J

Last revised: Tue, 27 Jan 2004 07:41:20 GMT

CALIFORNIA STATE UNIVERSITY, CHICO

CSCI 224 - Robotics and Machine Intelligence

Dr. Juliano Juliano@csuChico.edu
Dr. Renner Renner@csuChico.edu
Dr. Varahamurti rVarahamurti@csuChico.edu

(3.0 credit units) This course introduces students to the field of robotics by emphasizing the task of endowing machines with intelligence. Topics include various case studies of robot architectures and algorithms that facilitate embodying a robot with behaviors that are traditionally associated with human cognition (e.g., perception, reasoning, intelligent navigation, vision, learning, etc.). Students will conduct robotics experiments and compete in robotics games.

Syllabus
Notes
Calendar
Online Quizzes
Lab Activity Sheets and Projects
Discussion
Mail
Live Chat
Progress
Grades
Library Resources
Technical Support & Tutorials

Michael McMaster
CSCI 224
Dr. Renner, Dr. Juliano
March 11, 2004

Individual Boe-Bot Competition (Maze with Cheese) Report

Problem Description and Constraints

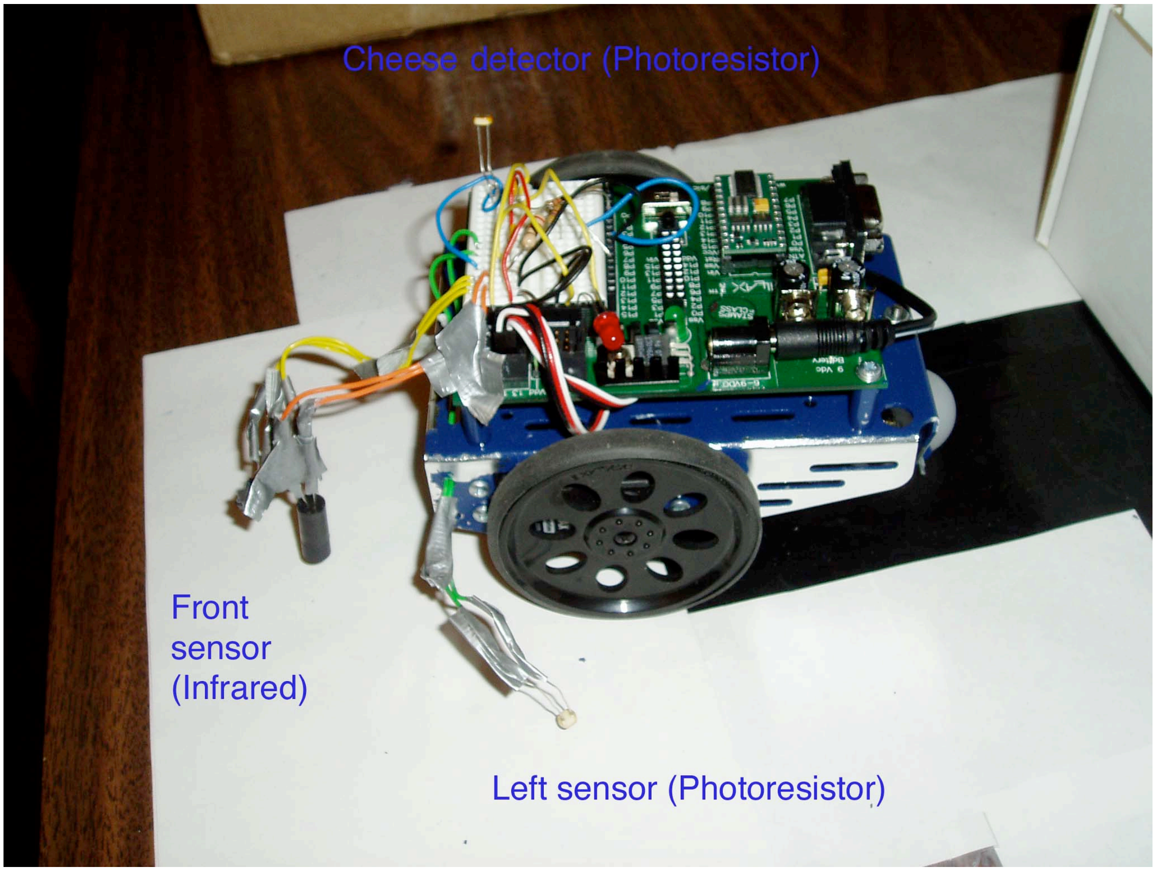
For this assignment, we were asked to modify the Boe-Bot so that it could successfully navigate a maze delimited by black stripes on white paper. The goal, called the "cheese," was a flashlight beam shining down somewhere in the maze. The robot should be able to stop moving upon detection of the cheese. The robot would receive a "fault" for going over a wall, and can only receive three faults before it receives a DNF (did not finish). We could use any sensors available to us in any arrangement we desired.

A score would be given based on the Boe-Bot's performance. Primarily, the robots were judged on the time it took to successfully find the cheese. Bonus points would be awarded for never crossing over a wall and successfully detecting and stopping near the cheese.

Proposed Solution and Methodology

For navigating the maze, I chose to utilize a wall-following algorithm. The robot used two sensors to sense the edges of the maze. The sensor mounted to the robot's left side would try to stay within the "wall" of black tape, while the other sensor, mounted in front, would try to stay outside of it. When one sensor would cross into territory that it shouldn't be in, the robot would turn accordingly. For example, if the left sensor finds white underneath, it means that the wall edge we're following is no longer parallel to our line of direction. We need to make a left turn until we find that wall again, and then we can continue to follow it normally. The same principle works for right turns when the front sensor finds itself "hitting" a wall.

I intended to keep the sensors and configuration of those sensors as simple as possible. I used a photoresistor for cheese detection, mounted from the right side of the breadboard and facing the ceiling. Mounting it on the right side put it as close to the center of the maze corridor as possible. The left sensor was also a photoresistor, aimed close to the floor about two inches off the left wheel axle. For the front sensor, I used an infrared LED and detector. I did this primarily because the Boe-Bot kit only included two photoresistors, and I was already using them.



Limitations of the proposed solution/algorithm

There are some drawbacks to a wall-following algorithm. The path to the goal can vary based on which wall you choose to follow. Also, wall-following can be time-consuming as the robot goes down every wrong path on the way to the goal. On the other hand, it ensures that the goal will eventually be found.

A drawback with my implementation was the decision to use the infrared sensor as the front sensor. A photoresistor is much better suited to the task of determining the color of the ground beneath it. While the photoresistor could be used to do this, it was occasionally incorrect in reading values, which could cause it to go through a wall or turn when it should not.

Analysis (and discussion) of the proposed solution

After much experimentation, I was finally able to align the sensors so that it would usually complete the test maze provided to us. The infrared sensor was the biggest challenge. The connections from the sensors to the breadboard sometimes came loose, causing a malfunction. With five such connections, I found myself repairing them more often than I would have liked. However, I was able to get it to perform correctly at a consistent enough level that I thought it should be ready for competition.

Unfortunately, the infrared sensor turned out to be too unreliable to be useful. It seemed that there was a noticeable delay between when the sensor should see the wall and when it would react. This high reaction time caused it to go over the wall several times on the day of the individual competition. The problem was not as severe before the competition, but it was not unexpected.

In the end, the robot faulted out of the competition.

Conclusions and Recommendations

I think the robot would have performed well if I had replaced the infrared sensor at the front with a photoresistor. This would have given it greater sensor reliability and a faster reaction time. I wanted to use the infrared sensor because I liked the challenge of using only the parts we were given in the kit. I was encouraged to continue using it because it gave accurate values in testing while at a standstill. Unfortunately, the physical stresses of moving and turning while running the maze caused the sensor to be too unreliable. Radio Shack had no photoresistors remaining by the time I decided I could use more. Even until the end, though, I wanted to see it work with the infrared sensor. It would have been very satisfying to overcome that challenge.

A compromise solution might have been to switch the positions of the sensors, so that the photoresistor was in front, and the infrared sensor on the left. This would have allowed the robot to react more quickly to oncoming walls. Left turns are not as critical to respond to as right turns with a left wall-following robot, so this may have helped to correct the situation somewhat. I did not think of this compromise until after the competition.

References

Parallax, Inc. Robotics! Student Workbook Ver. 1.5.

Appendix

Source code used for the maze follower:

```
'{$STAMP BS2}

' Program to navigate a maze delimited by black strips that acts
' as walls.  Uses two infrared sensors to hug the left wall, and
' one photoresistor to check for the presence of a bright light
' that indicates the goal.

' initialize
OUTPUT 10               ' LED
LOW 12                  ' servo
LOW 13                  ' servo
OUTPUT 1                ' right sensor
OUTPUT 7                ' left sensor
photo VAR Word
l_value VAR Bit
r_value VAR Word
counter VAR Nib
OUT10 = 1               ' LED off

main:

  ' do i see the cheese? if so, flash LED (P10) and stop

  HIGH 3
  PAUSE 3
  RCTIME 3,1,photo
  IF photo > 47 THEN dontstop

  OUT10 = 0
  PAUSE 200
  OUT10 = 1
  PAUSE 200
  GOTO main

  ' no cheese, continue
  dontstop:

  GOSUB check_sensors

' which way do i need to turn (if at all)?
  IF l_value = 0 AND r_value < 155 THEN turnLeft
  IF l_value = 0 AND r_value >= 155 THEN moveForward
  IF l_value = 1 AND r_value < 155 THEN turnRight
```

```
  IF l_value = 1 AND r_value >= 155 THEN turnRight

  moveForward:
    DEBUG "straight"
    PULSOUT 13,868
    PULSOUT 12,723
'     PAUSE 10
  GOTO main

  turnLeft:
    DEBUG "left    "
    FOR counter = 0 TO 8
      PULSOUT 13,750
      PULSOUT 12,750
'       PAUSE 10
    NEXT
  GOTO main

  turnRight:
    DEBUG "right    "
    FOR counter = 0 TO 11
      PULSOUT 13,791
      PULSOUT 12,781
      PAUSE 10
    NEXT
  GOTO main
' go forward a bit
'  PULSOUT 12, 768
'  PULSOUT 13, 768
'  PAUSE 20
GOTO main

check_sensors:

  l_value = 0
  r_value = 0

  FREQOUT 7, 1, 40990
  l_value = ~IN8

  HIGH 0
  PAUSE 3
  RCTIME 0,1,r_value


RETURN
```

Individual Challenge # 2 – Random Maze

Goal Locate "cheese" (a light source) within a random wooden maze using tactile and photo sensors.
Robot Name Pokey
Robot Trainer James Long
Hardware Used Standard BoeBot with some custom hardware. A special aluminum mount was created and additional hardware added for the front bumper and side wall sensors.
Algorithm IR range detection coupled with semi-random turns.

Problem Description & Constraints

The purpose of this challenge is to build a BoeBot that can successfully navigate a randomly-generated wooden maze. The maze is constructed of 6 inch-long wall segments attached to a pegboard floor. A "cheese" goal (a light source) is located at a random location within the maze. The robot must use tactile and/or photo-sensing techniques to navigate the maze, locate the light source, and stop.

Proposed Solutions & Methodologies

1. Initial Configuration - First and Second Trials

A simple design seemed to be the best for navigating the maze. The robot would have a difficult enough time traversing the physical environment, and a sophisticated algorithm would likely backfire (or worse, never have a chance of executing). Therefore the BoeBot was configured with a simple front bumper for detecting forward obstacles, and side-mounted whiskers for detecting and avoiding walls. The algorithm for navigating the maze makes random 90-degree turns once a forward obstacle is detected:

```
begin loop
   read front, side and photo sensors (front left, right, top photo)
   if top (photo) detected then stop
   elseif left (whisker) detected then turn pulse right
   elseif right (whisker) detected then pulse left
   elseif front (bumper) detected then
         back up
         choose random turn direction (left or right)
         turn 90 degrees
   else move forward
loop
```

listing 1.1 – Sensor detection loop – 1st and 2nd configurations

The BoeBot trainer attempted to implement a second BoeBot configuration using IR photo detectors, however after several hours of configuration and testing the trainer inadvertently shorted the IR sensors and was forced to use the same initial configuration and algorithm for the 2nd trail.

2. Final Configuration

With the generous loaning of a pair of working IR detectors, the trainer was able to successfully add IR range detection to the BoeBot for the 3rd and final trial. The BoeBot logic was modified to detect perform IR range detection at roughly 6-inch intervals, and when the forward bumper was activated. If an "opening" was detected to the left or right, the BoeBot turned 90 degrees in that direction. If a forward obstacle was detected, the BoeBot would look left and right to see which direction was clear, and then turn in that direction:

```
begin loop
   read front, side and photo sensors (front left, right, top photo)
   if top (photo) detected then stop
   elseif left (whisker) detected then turn pulse right
   elseif right (whisker) detected then pulse left
   elseif front (bumper) detected then
         back up
         detect left and right range via IR
         if left clear turn left 90 degrees
         elseif right clear turn right 90 degrees
         elseif neither clear
               choose random turn direction (left or right)
               turn random 90 degrees
   else
         if forward movement counter > 300 pulses then
               reset forward movement counter
               detect left and right range via IR
               if left clear turn left 90 degrees
               elseif right clear turn right 90 degrees
         increment forward counter
         move forward
loop
```

listing 1.2 – Sensor detection loop – Final configuration

Limitations to Proposed Solution/Algorithm

Considering the random element involved in choosing a turn direction, it is possible the BoeBot will never navigate close enough to the cheese to detect it. Also, the detection loop assumes that the BoeBot will not inadvertently get stuck due to some chance hangup in the maze (if the BoeBot gets stuck in a forward motion, it will attempt to move forward indefinitely). Lastly, the IR detection loop may cause the BoeBot to steer towards areas of the maze that are not on the solution path.

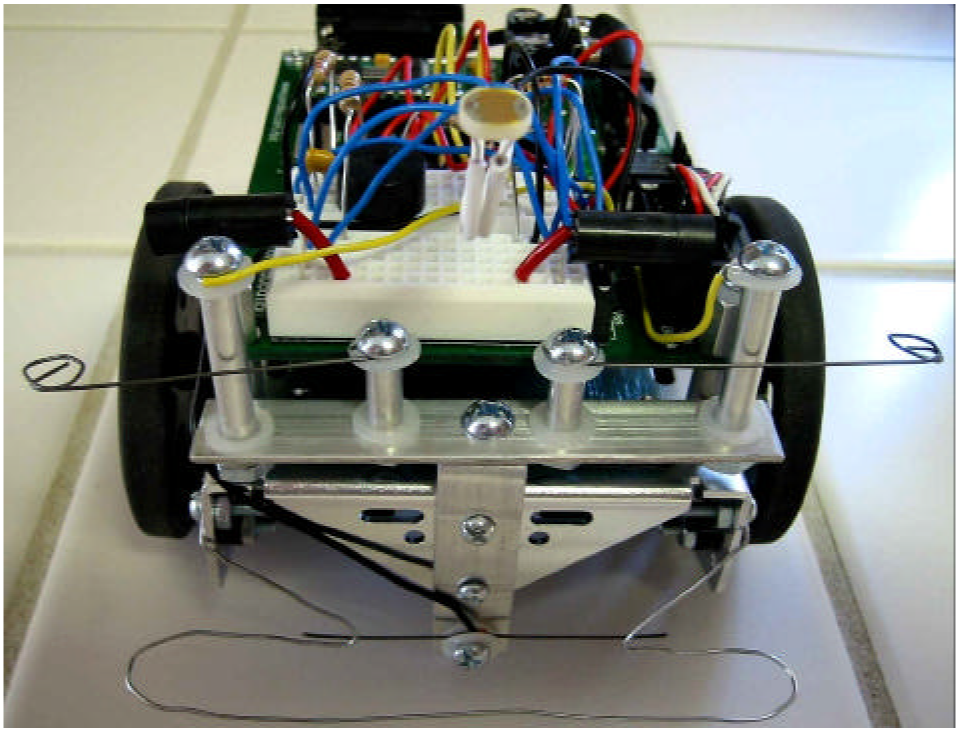


figure 1.1 – Final BoeBot configuration

Analysis of Proposed Solution/Algorithm

1. Initial Configuration - First and Second Trials

The BoeBot failed to successfully navigate the maze on the first and second trials. In both cases, the BoeBot became stuck on surface obstacles that were not accounted for in the algorithm logic. In the first trial, the BoeBot managed to wedge itself onto a corner where neither the front bumper nor side whisker could detect the obstacle (the whisker itself was hung up on the wall corner). The random navigation technique also proved to be a limiting factor, since the BoeBot never traveled near to the location of the goal. After this failure, the robot trainer adjusted the hardware to reduce any gaps between front bumper and side whisker detection, and the whiskers were shortened to prevent hanging up on side obstacles.

Due to the lack of any real change in hardware or algorithm (IR sensors were attempted but removed after they were shorted out), the second trial resulted in similar failure. The hardware adjustments did smooth out navigation, however the random motion of the BoeBot failed to move it close to the goal and it eventually snagged on an obstacle that the algorithm logic failed to account for.

2. Final Configuration

Unfortunately, the last trial was also unsuccessful. The IR sensors aided in navigating corner walls, which allowed the BoeBot turned in the correct direction to avoid a blocking wall to the left or right. The IR detection also helped in locating open portions of the maze, and the BoeBot was able to avoid some enclosed dead-end areas as a result. However, the random turn factor caused the BoeBot to turn away from paths leading to the cheese on several occasions, which indicates a weakness in the proposed algorithm. Eventually, even with IR range detection, the BoeBot hung up on a maze obstacle.

Conclusions and Recommendations

The random factor involved in the maze construction made this an extremely difficult maze to complete (as proven by the few successful BoeBots). Factors that helped some BoeBots navigate successfully included additional logic to “back up” the BoeBot if forward motion occurred for too long of a period. Such logic may have prevented the hang-ups that plagued this trainer’s BoeBot. Overall, the hardware configuration appeared to be adequate for navigating the physical environment of the maze (after modifications). The real limitation was in using an algorithm that did not account for obstacle hang-ups, and that did not use intelligence (mapping, breadth / depth searching) in determining and analyzing the layout of the maze.

If time had not been a factor, it would have been helpful to spend more time focusing on algorithm logic for solving the maze once successful physical navigation techniques had been worked out.

Appendix – Program Listing 1 – Lead BoeBot

```
'CSCI 224 - Robotics & Machine Intelligence
'James Long
'Maze #2 Competition
'
'{$STAMP BS2}
'{$PBASIC 2.5}

' ----------------------------------------------
' ----------------------------------------------
' -----           Declarations           -------
' ----------------------------------------------
' ----------------------------------------------

  ' ---------------------------------------------
  ' variables
  ' ---------------------------------------------
  IR_emmit_freq  VAR   Word  ' Stores frequency arg for freqout.
  left_IR        VAR   Byte  ' left IR sensor value.
  right_IR       VAR   Byte  ' right IR sensor value.
  left           VAR   Bit   ' sensor state for left  whisker.
  right          VAR   Bit   ' sensor state for right whisker.
  front          VAR   Bit   ' sensor state for front whisker.
  choice         VAR   Word  ' random turn direction.
  state          VAR   Nib   ' all sensor states.
  moves          VAR   Word  ' buffer for recent moves.
  photo          VAR   Byte  ' RC time for top photoresistor.
  pulse_count    VAR   Byte  ' used to set number of pulses delivered.
  loop_count     VAR   Byte  ' for...next loop counter.
  forward_count  VAR   Byte  ' forward pulse counter.

  ' ---------------------------------------------
  ' color constants
  ' ---------------------------------------------
  white          CON     22

  ' ---------------------------------------------
  ' move direction constants
  ' ---------------------------------------------
  backed_up      CON    %00
  turned_left    CON    %01
  turned_right   CON    %10
  moved_forward  CON    %11

  ' ---------------------------------------------
  ' movement pulse constants
  ' ---------------------------------------------
  stop_rw        CON    750  ' pulse for right wheel - stop.
  stop_lw        CON    750  ' pulse for left  wheel - stop.

  fwd_rw         CON    720  ' pulse for right wheel - half forward.
  fwd_lw         CON    780  ' pulse FOr left  wheel - half forward.
  bwd_rw         CON    775  ' pulse for right wheel - half backward.
  bwd_lw         CON    720  ' pulse for left  wheel - half backward.
```

```
  left_turn_rw   CON    500  ' pulse for right wheel - left  turn
  left_turn_lw   CON    750  ' pulse for left  wheel - left  turn
  right_turn_rw  CON    750  ' pulse for right wheel - right turn
  right_turn_lw  CON   1000  ' pulse for left  wheel - right turn


' -----------------------------------------------
' -----------------------------------------------
' -----          Initialization          -------
' -----------------------------------------------
' -----------------------------------------------
  OUTPUT 5                  ' Set P5 to output (left IR).
  OUTPUT 3                  ' Set P3 to output (right IR).
  OUTPUT 2                  ' Set P2 to output (speaker).

  FREQOUT 2, 500, 3000      ' Signal program is starting/restarting.
  LOW 12                    ' Set P12 and 13 to output-low.
  LOW 13

    ' initialize the forward pulse counter
    forward_count = 0

    ' initialize empty turn buffer
    moves = %0000000000000000

    ' seed the random variable
    choice = INH * 256 + INH


' -----------------------------------------------
' -----------------------------------------------
' -----            Main Routine          -------
' -----------------------------------------------
' -----------------------------------------------
main:
'    GOSUB check_IR
'    GOTO main

   ' initialize empty state
    state = %0000

    ' measure input from whiskers.
    left  = IN8
    right = IN6
    front = IN4

    ' measure RC time for top photoresistor.
    HIGH 1                  ' Set P9 to output-high.
    PAUSE 3                 ' Pause FOr 3 ms.
    RCTIME 1,1,photo        ' Measure RC time on P9.

    IF left  = 0      THEN state = state | %0001
    IF right = 0      THEN state = state | %0010
    IF front = 0      THEN state = state | %0100
    IF photo <= white THEN state = state | %1000
```

```
    ' cheese detected - STOP!
    IF state & %1000 = %1000 THEN
      FREQOUT 2, 500, 3000         ' Program start/restart signal.
      STOP
    ELSE

      SELECT state
       ' all sensors clear
        CASE %0000
          GOSUB forward_pulse

       ' left sensor on wall
       CASE %0001
          GOSUB right_turn

       ' right sensor on wall
        CASE %0010
          GOSUB left_turn

       ' front/both/all sensors on wall
        CASE %0100, %0110, %0111, %0011
          GOSUB pivot_90
     ENDSELECT
   ENDIF
GOTO main


' -----------------------------------------------
' -----------------------------------------------
' -----        Navigation Routines        -------
' -----------------------------------------------
' -----------------------------------------------

' -----------------------------------------------
' read IR sensors
' -----------------------------------------------
check_IR:
    ' reset left and right IR readings.
    left_IR  = 0
    right_IR = 0

    ' load sensor outputs into left and right IR readings using
    ' a for...next loop, a lookup table, and bit addressing.
    FOR loop_count = 0 TO 4

      LOOKUP loop_count,[37500,38250,39500,40500,41500], IR_emmit_freq

      FREQOUT 5,1, IR_emmit_freq
      left_IR.LOWBIT(loop_count)  = ~IN10

      FREQOUT 3,1, IR_emmit_freq
      right_IR.LOWBIT(loop_count) = ~IN0
    NEXT

    ' convert the readings
    left_IR = NCD(left_IR)
```

```
    right_IR = NCD(right_IR)
RETURN


' ----------------------------------------------
' default movement - forward
' ----------------------------------------------
forward_pulse:
  PULSOUT 12, fwd_rw
  PULSOUT 13, fwd_lw
  PAUSE 20

  forward_count = forward_count + 1

  IF forward_count > 72 THEN

    forward_count = 0
    moves = moves << 2
    moves = moves | moved_forward

    GOSUB check_IR

    IF left_IR > right_IR THEN
      GOSUB left_pivot
    ELSEIF left_IR < right_IR THEN
      GOSUB right_pivot
    ENDIF

  ENDIF
RETURN


' ----------------------------------------------
' turn direction picker
' ----------------------------------------------
pivot_90:

  forward_count = 0
  ' check IR readings before turn
  GOSUB check_IR
  GOSUB back_up

  IF left_IR > right_IR THEN
    GOSUB left_pivot
  ELSEIF left_IR < right_IR THEN
    GOSUB right_pivot
  ELSE
    RANDOM choice
    IF choice > 32368 THEN
      GOSUB right_pivot
    ELSE
      GOSUB left_pivot
    ENDIF
  ENDIF
RETURN
```

```
' ---------------------------------------------
' backup half a bot length
' ---------------------------------------------
back_up:
  PULSOUT 12, stop_rw
  PULSOUT 13, stop_lw
  PAUSE 20

  FOR pulse_count = 1 TO 34
    PULSOUT 12, bwd_rw
    PULSOUT 13, bwd_lw
    PAUSE 20
  NEXT

  PULSOUT 12, stop_rw
  PULSOUT 13, stop_lw
  PAUSE 20

  moves = moves << 2
  moves = moves | backed_up
RETURN


' ---------------------------------------------
' 90 degree turn left
' ---------------------------------------------
left_pivot:
  FOR pulse_count = 1 TO 30
    PULSOUT 12, bwd_rw
    PULSOUT 13, fwd_lw
    PAUSE 20
  NEXT
  moves = moves << 2
  moves = moves | turned_left
RETURN


' ---------------------------------------------
' 90 degree turn right
' ---------------------------------------------
right_pivot:
  FOR pulse_count = 1 TO 30
    PULSOUT 12, fwd_rw
    PULSOUT 13, bwd_lw
    PAUSE 20
  NEXT
  moves = moves << 2
  moves = moves | turned_right
RETURN


' ---------------------------------------------
' nudge turn left
' ---------------------------------------------
left_turn:
```

```
  FOR pulse_count = 1 TO 2
    PULSOUT 12, left_turn_rw
    PULSOUT 13, left_turn_lw
    PAUSE 20
  NEXT
RETURN


' ----------------------------------------------
' nudge turn right
' ----------------------------------------------
right_turn:
  FOR pulse_count = 1 TO 2
    PULSOUT 12, right_turn_rw
    PULSOUT 13, right_turn_lw
    PAUSE 20
  NEXT
RETURN

END
```

Robotic Competition #2

Authors: Jeremiah C. Anderson
Scott McMillan
Scott Rippee

Robot Team Name: Road Warriors
Robots: Bender
GRRR
Grintch

CSCI 224: Robotics and Machine intelligence
Project Evaluation/Report
Spring 2004

Department of Computer Science,
Department of Mechanical Engineering, Mechatronic Engineering, and Manufacturing Technology
California State University, Chico

Table of Contents

1 Problem description and constraints

1.1 Description

The goal of this competition is enable a group of robots to work as a team. Each human group must to devise and implement a software/hardware strategy which will accomplish this goal. The robot team forms a convoy; leader and two follower robots. Working as a team, the robots should solve a "maze"[1] and find a target. The maze is formed with black electrical tape on a white poster board floor. The target, or "cheese", is created by a small flashlight pointed in one cell of the maze[2].

1.2 Constraints

There are several constraints or "faults" in this project and each has a penalty. Three faults are allowed before disqualification. The group is allowed only 3 trials. If part of the team or entire convoy faults, then the entire team faults. This requires a restart from the start of the trial. Listed below are project requirements if these constraints are not met. The corresponding reductions for a fault can also be seen.

- Stops in area A, B, C, D before completing section E, Fig 1: 10%
- Starts on non-tactile command:10%
- All Boe-Bots in the convoy finish: 10%
- Failure to complete in 3 minutes.

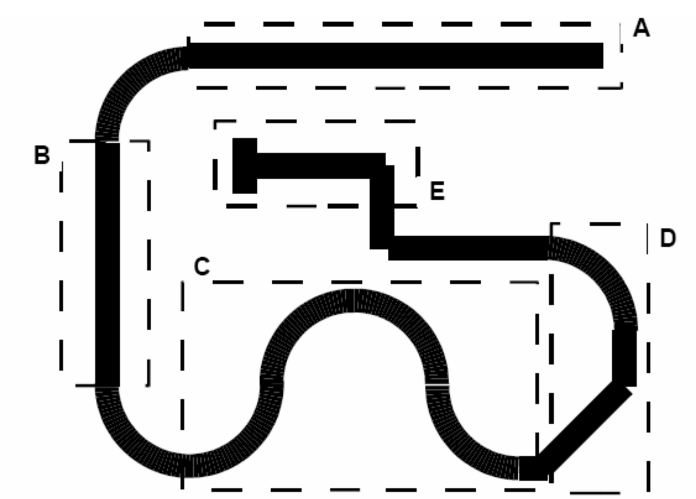


Figure 1 Competition Maze

2 Proposed solution and methodology

2.1 Concept Generation

The general concept to this project has been pre-selected by the competition rules. The general concept required "Shadow followers" from the ROBOTICS! [1]. Using this

[1] The "maze" is really a path or track with varying angles, curves, and straight lines.
[2] This section written by Scott McMillan

general model several conceptual features were generated. Non-tactile start command was implemented using a flashlight, sound, wireless radio signal, or mind control[3].

2.1.1 Leader concept: Line Follower

A line follower follows an existing line or wall. The concept is that the wall follower will travel along a wall until it reaches every cell or the "cheese." Using a simple wall following algorithm this design can be quite robust. Several concepts were generated to enable the leader to follow a line.

- One IR sensor on each side of the line with the Bot keeping the line in between the sensors.
- One photo sensor on each side of the line.
- One IR sensor on the line and one off.
- One photo sensor on the line and one off[4]

2.2 Concept Selection and Implementation

2.2.1 Concept Selection

Due to the maze architecture and contest rules, a line follower was selected for the leader. The line follower has many advantages such as its ability to only require a line to follow. Sensor feed back can be simple and robust.

2.2.2 Hardware Implementation

For Leader

The hardware was implemented so the sensors straddled the "wall[5]" of the maze. This method presented the problem that both of the sensor might end up being over a wall at one time. Usually this could be encountered when making turns (depending on wall width). In initial trial runs, it was determined that the sensors were detecting the lines frequently, reducing the Bot's forward progression velocity. In the next trial a different approach was implemented using one sensor on the wall and one off. The Bots forward velocity was increased. This method had the added benefit allowing both IR and the photo sensors to be implemented. More consistent data readings were obtained with the IR receivers when in close proximity to the ground. Therefore IR was used to follow the line. After these modifications to the design, the leader Bot moved quickly around the maze and had no problem following the lines. The last challenge was to detect the end of the maze which was identified by a line perpendicular to the one being followed by the Bot.

Photo-resistors were used detect this end line line. By extending them past the edges of the Bot and moving them an inch in front of the IR a wide enough spread so that the photo sensors would not both be over the line when making turns was obtained. They

---

[3] The group thought really hard, but seemed to have no luck in starting BOEBOTS. Later team members went home to find all spoons bent in their respective kitchens.??
[4] This section written by Scott Rippee
[5] "Walls" of black tape make up the maze and are the path which the leader follows

Figure 2 Leader Sensor Array

were also far enough in front of the IR to detect the perpendicular line before the IR detected it, which would have caused it to turn missing the end of the maze.

For Followers

Hardware requirements for the followers were minimal consisting of a left and right IR sender and receiver to detect the leader's position and a photo sensor to set the Bot's in motion with a flash light.

2.2.2.1 Leader Hardware[6]

- 3 x 220 ohm resistors
- 2 x 2k ohm resistors
- 2 IR senders
- 2 IR receivers
- 3 x .1uF capacitors
- 3 x Photo sensors (1 for flash light, 2 for finish line)
- extension rod for photo sensors
- speaker
- Wire various length

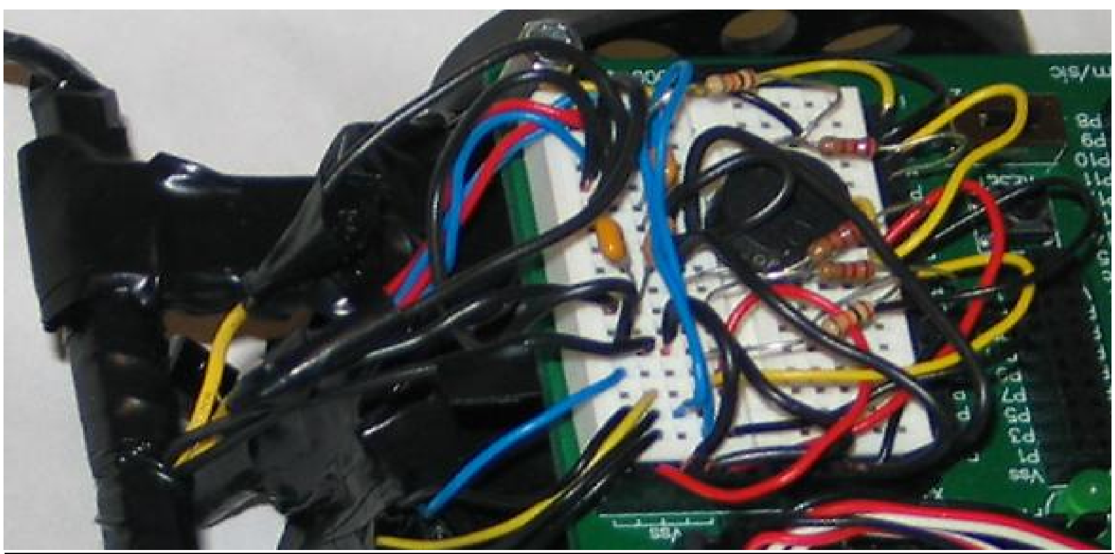


Figure 3 Leader Circuit

[6] Followers Hardware was pretty much the same as the leaders.

2.2.2.2 General Hardware/Equipment

Equipment used in this project include

- Weller Soldering Iron
- Solder
- Gas Lighter
- Wire cutters
- Leather man
- Digital Multi-Meter
- 3 BOE-BOT kit
- Small Maglite
- Heat shrink
- Several standoffs various sizes

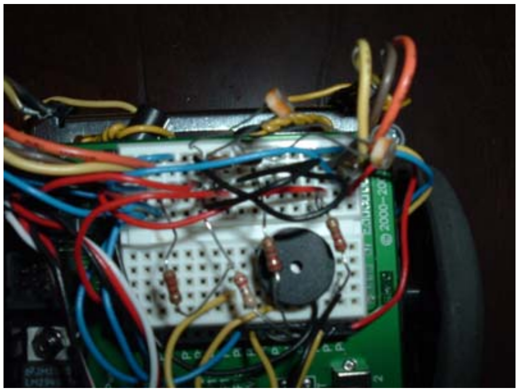


Figure 4 Follower Circuit

2.2.2.3 Non-tactile activation implementation

Photo-resistors were used as sensors to detect varying levels of light and darkness. The sensors use the characteristics of an RC network. The circuit will drain according to the resistance and capacitance of the circuit. The photo-resistor changes resistance depending upon light levels. The capacitor and resistors act as a constant variable allowing light to be the changing variable. The stamp attempts to determine when the circuit goes from a 1 to a 0. The circuit is checked every 2 u sec and the number of times the circuit is checked before it goes to 0, outputted as a value. Hence a value of 140 is actually 280 u sec. These values are used as sensor feedback.

2.2.3 Software Implementation

Leader Software Implementation

Following the line involved moving forward while keeping track of which IR sensor last detected the line. When the Bot begins moving and neither sensor have detected the line it moves forward until one of the IR sensor moves over the line. A flag is then set to save which IR sensor found it. The Bot then continues forward until both sensors do not detect the line or both sensors do detect the line. When either of these situations occurs it turns in a direction determined by the flag setting. The Bot detects the flashlight using a photo sensor that is pointed in the air. When the photo sensor returns a value under a threshold that was determined empirically the Bot starts in motion. Finally the detection of the finish line is accomplished by the two front photo sensors returning a value indicating that they are over the line.

Follower Software Implementation

The follower program works by using IR to determine where the Bot in front of it is. If one IR sensor detects the leading Bot then it will turn so that neither sensor detects the leader. If the leading Bot is not detected to the left or right then the follower Bot will continue forward. If the leader pulls ahead of the followers then the followers have a chance to catch up when the leader moves around a turn. If one of the followers gets to close to the leader it will stall until the leader moves further away and then continue moving forward.

2.2.3.1 Pseudo Code/Algorithm

Leader Pseudo Code

If top photo sensor detects light then start moving forward.
If photo sensors looking for end find perpendicular line then go to stop subroutine
Move forward until a sensor has found the line
If left IR on and right IR off then move forward and set flags
If right IR on and left IR off the move forward and set flags
If both IR are off then turn right if flags say that the right IR is suppose to be on the line
If both IR are off then turn left if flags say that the left IR is suppose to be on the line
If both IR are on then turn left if flags say that the right was suppose to be off
If both IR are on then turn right if flags say that the left was suppose to be off

Follower Pseudo Code

```
' {$Stamp bs2}
' Follow shadow bot

'---- Declarations
Declare all of the variables that are going to be used

'---- Initialization
Initialize the outputs
set the FREQUOUT values

'----- Main Routine
main:
call check sensors function
  IF photo < 20 THEN go to start label
 Else go to main
```

```
call check sensors function
  set the left servo value to 750 minus(plus for the other follower) ((3
minus the left IR value) times 125 )
  set the right servo value to 750 plus(minus for the other follower)
((3 minus the right IR value) times 125)

go to start

'---  Subroutines
check_sensors:
  set left and right values to 0
  set photo sensor value to a high value so bot does not turn on when
not supposed to
  set HIGH to 3
  set PAUSE to 3
  set RCTIME 3,1,photo          'run very slow
  ' Take 5 measurements for the distance at each IR pair.
  ' If you fine tuned your frequencies in Activity # 2 insert
  ' them in the look up tables.

  FOR counter = 0 TO 4
    check_left_sensor
    check_right_sensor
  NEXT

  set the ncd values of left and right sensor to a value between 0 and 5
   Return to calling function
```

Due to the different locations of the IR sensors on the two followers[7] we had to swap the values going to each of the servos to accommodate for difference. One of the followers IR sensors looks forward and the other one had the IR output crossed.[8]

3 Limitations

One limitation encountered was varying results we encountered with the IR accurately detecting the leading Bot. This aspect of the challenge required a lot of trial and error and never seemed to be perfect. The lack of ability for the follower Bot's to see the leader Bot can also be seen as a limitation. This was over come by adding a piece of white paper around the leading Bots.

On straight-aways the lead Bot could move faster than the followers. This provide to be a limitation because the follower Bots moved to the left and right to keep the leader in its center view, which slowed it down. Slowing the lead Bot down helped this problem but this could have still been a limitation if the straight-aways were longer and the leader

[7] Pseudo code and source code are exactly the same for both followers with the exception of a reversal on IR positioning on the trailing Bot. The only code difference is the switching of the left and right servo motor controls.
[8] Jeremiah C. Anderson contributed the follower pseudo code section.

managed to pull to far ahead. Having slowed down the leading Bot also had the effect of decreasing our overall time.

The unevenness of the maze floor also provided a limitation because the IR and photo sensor readings were not as accurate as they would have been on a completely flat surface. If both of the photo sensors encountered a raised area at the same time then they would see their shadow and determine that the end of the maze has been reached.

4 Analysis and Conclusion

We were successful in completing the objective for this team competition. As a result of this project we learned about the benefits and limitations of the Boe-Bot. In particular this project was beneficial in teaching us how to use the provided sensors accurately and for multiple purposes that we were able to creatively determine. The team aspect allowed us to gain insight into the potential for multiple Bots working together to complete a goal[9].

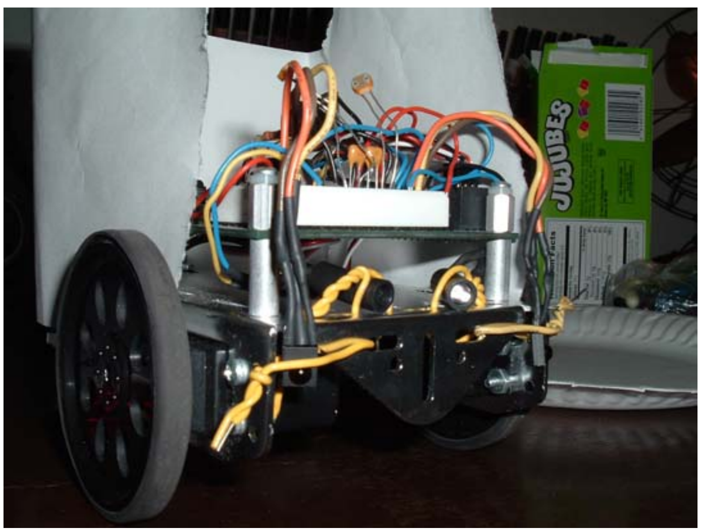


Figure 5 GRRR (Couldn't follow the broadside of a barn if it's life depended on it)

[9] Such as world domination, one line at a time.

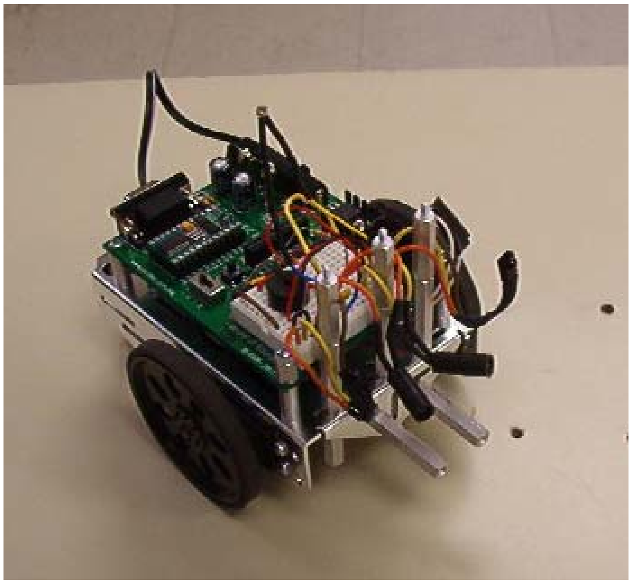


Figure 6 Bender (On a drunken binge again!)

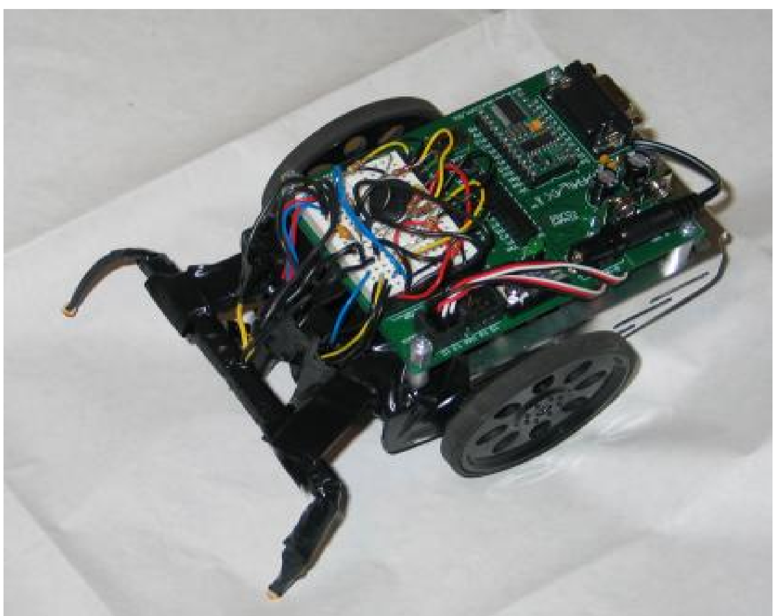


Figure 7 Gritnch (our fearless leader, that asshole! Follow me he says, lets go this way, never friggin asks for directions…)

Reference:

[1] Basic Stamp Manual Version 2.0b, Parrallax
[2] Robotics! with the Boe-Bot v1.5, Parrallax

Appendix A: Source Code for Leader

```
Appendix (Code For Leader)
' {$Stamp bs2}
' Scott Rippee
' Follow A black line with IR sensors
' Detect end of path with photo sensors
' Starts with a flashlight signal

'---- Declarations
counter   VAR Word
l_values  VAR Word
r_values  VAR Word
l_IR_freq VAR Word
r_IR_freq VAR Word
line_left   VAR  Bit
line_right  VAR  Bit
right_photo VAR  Word
left_photo  VAR  Word
start_photo VAR  Word
switch_up  VAR Bit
going      VAR Bit
right_on  VAR  Bit
left_on VAR    Bit

'---- Initialization
LOW 13
LOW 12
OUTPUT 2
OUTPUT 7
OUTPUT 1
FREQOUT 2, 500,3000

line_left = 0
line_right = 0
switch_up = 0
going = 0
right_on = 0
left_on = 0

'----- Main Routine
main:

  GOSUB check_sensors

  IF start_photo < 45 AND going = 0 THEN rob_start
```

```
  IF going = 0 THEN rob_wait

  'If it sees the END perp stripe
  IF right_photo > 550 AND left_photo > 150 THEN at_end

  'left on right off
  IF l_values <= 3 AND r_values >= 4 THEN forward_left_on
  'right on left off
  IF l_values >= 4 AND r_values <= 3 THEN forward_right_on

  'both off decide which way to turn
  IF l_values >= 4 AND r_values >= 4 AND right_on = 1 THEN right_turn
  IF l_values >= 4 AND r_values >= 4 AND left_on = 1 THEN left_turn
  IF l_values >= 4 AND r_values >= 4 AND right_on = 0 AND left_on = 0 THEN
forward
  'if both are on the line have to do something
  IF l_values <= 3 AND r_values <= 3 AND right_on = 1 THEN left_turn
  IF l_values <= 3 AND r_values <= 3 AND left_on = 1 THEN right_turn
  IF l_values <= 3 AND r_values <= 3 AND right_on = 0 AND left_on = 0 THEN
forward

GOTO main

'---  Nav Routines
forward:
  PULSOUT 13,800
  PULSOUT 12,700
  PAUSE 10
GOTO main

forward_right_on:
  right_on = 1
  left_on = 0
  PULSOUT 13,800
  PULSOUT 12,700
  PAUSE 10
GOTO main

forward_left_on:
  left_on = 1
  right_on = 0
  PULSOUT 13,800
  PULSOUT 12,700
  PAUSE 10
GOTO main
```

```
left_turn:
    PULSOUT 13,500
    PULSOUT 12,500
    PAUSE 10
GOTO main

right_turn:
    PULSOUT 13,1000
    PULSOUT 12,1000
    PAUSE 10
GOTO main

rob_start:
  going = 1
GOTO main

rob_wait:
  PAUSE 10
GOTO main

at_end:
  FOR counter = 0 TO 350
     PULSOUT 13, 770
     PULSOUT 12, 730
     PAUSE 10
  NEXT
GOTO rob_stop

rob_stop:
  PULSOUT 13, 750
  PULSOUT 12, 750
  PAUSE 1000
GOTO rob_stop


'===-----------------  Subs
check_sensors:
  l_values = 0
  r_values = 0

  FOR counter = 0 TO 4
    check_left_sensors:
      LOOKUP counter,[37500,38250,39500,40500,41500],l_IR_freq
      FREQOUT 7, 1, l_IR_freq
      l_values.LOWBIT(counter) = ~ IN8
```

```
  check_right_sensors:
    LOOKUP counter, [37500,38250,39500,40500,41500],r_IR_freq
    FREQOUT 1, 1, r_IR_freq
    r_values.LOWBIT(counter) = ~IN0

 NEXT

 HIGH 6
 PAUSE 3
 RCTIME 6,1,start_photo

 HIGH 3
 PAUSE 3
 RCTIME 3,1,right_photo

 HIGH 10
 PAUSE 3
 RCTIME 10,1,left_photo

 l_values = NCD l_values
 r_values = NCD r_values

RETURN
```

Appendix B: Source Code for Follower

```
Folllower CODE
' {$Stamp bs2}
' Follow shadow bot

'---- Declarations
Kp_r      CON 125
Kp_l      CON 125
set_point CON 3 'modify
x      VAR Word
photo VAR Word
counter   VAR Nib
l_values  VAR Word
r_values  VAR Word
l_IR_freq VAR Word
r_IR_freq VAR Word

'---- Initialization
OUTPUT  13
OUTPUT  12
OUTPUT  2
OUTPUT  7
OUTPUT  1

FREQOUT  2, 300, 3000

'----- Main Routine
main:
GOSUB check_sensors
  IF photo < 20 THEN start
 GOTO main
 start:
  GOSUB check_sensors

  l_values = kp_l * (set_point - l_values)  ' left proportional control
  r_values = kp_r * (set_point - r_values)  ' right ...
  'IF l_values AND r_values > 65411 THEN reversel
  PULSOUT 13,750 - r_values
  PULSOUT 12,750 + l_values

GOTO start

'---  Subroutines
check_sensors:
  l_values = 0
  r_values = 0
photo =50
  HIGH 3                ' makeing
  PAUSE 3
  RCTIME 3,1,photo            'run very slow
  ' Take 5 measurements for the distance at each IR pair.
  ' If you fine tuned your frequencies in Activity # 2 insert
  ' them in the look up tables.
```

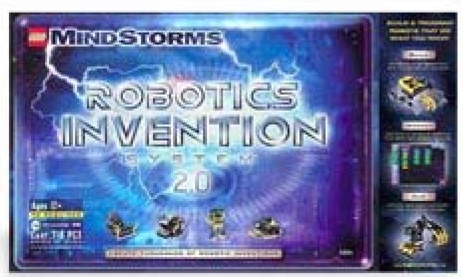
```
  FOR counter = 0 TO 4
    check_left_sensors:
      LOOKUP counter,[37500,38250,39500,40500,41500],l_IR_freq
      FREQOUT 7,1,l_IR_freq
      l_values.LOWBIT(counter) = ~IN8

    check_right_sensors:
      LOOKUP counter,[37500,38250,39500,40500,41500],r_IR_freq
      FREQOUT 1,1,r_IR_freq
      r_values.LOWBIT(counter) = ~IN0

  NEXT

  l_values = NCD l_values
  r_values = NCD r_values
RETURN
```

California State University, Chico
CSCI 224

Project

Lego-Mindstorm

Write-Up



Students:
Stephanie Liese
&
Matthew Myrick

Bot
El Roboto

May 20, 2004

Index

Problem Description

The Lego Mindstorm is required to navigate a maze delimited by black stripes formed with tape. The goal/cheese in the maze is a solid green circle within a green circle located somewhere within the maze. Scores are determined by the time needed to find the cheese in the maze. Each individual will be given one opportunity to "fault". A fault occurs when a Mindstorm wanders outside the maze. For each fault, the individual has one turn to do any hardware and/or software adjustments, after which the Mindstorm must restart the course. An individual is disqualified from competition on the second fault.

Proposed Solution

Hardware Modifications

The hardware modifications made for our robot include a swinging arm attached to a motor, a light sensor, and extensions to our axle to prevent the swinging arm from interfering with the tracks. The design was originally following the specifications for the tank bot. Treads were used, attached to small wheels, and allowed the robot to move with a good deal of agility and precision. Once the tank bot was built, we were able to add additional modifications in order to allow our bot to navigate a maze.

First, we needed to determine a method which would allow us to be able to discern the difference between a black wall and a white wall. In addition, we had to be able to detect the green (or gray, as the sensor would interpret) goal circle. To do this, the robot could not seek the gray at the same time as the black/white. This is because the sensor would see a slight gray in between the black and the white. Thus, it would register green and stop. To fix this, we used a swinging arm. When the arm was positioned on the right hand side, it sought a black line on a white background. Periodically, the arm would swing to the left hand side of the robot. When the arm hit the left hand side, it would check to see if the ground below it was green. If it saw green, it would register the goal, play a sound, and stop. If not, it would continue to try to follow the black line on the right hand side.

To keep the swinging arm stable, we used extensions to our axles to prevent the arm from swinging too far. Before the extensions were added, the arm would swing all the way to the treads and interfere with the robot's pace.


SYSTEM
2.0
Ages 12+
PC WITH USB REQUIRED
WINDOWS® 98/ME
Cont.718 PCS
BATTERIES NOT INCLUDED
3804

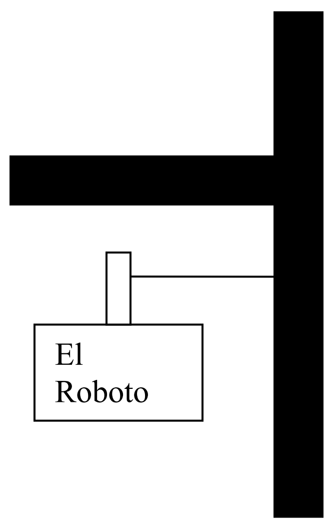
Limitations

The main limitations of this algorithm is that no matter how good at following a wall the Mindstorm was, it was still no more than a wall follower. The fact is that we were severely limited with only one light sensor. Even with the swinging arm algorithm it still left a great deal to chance. Other than this our Mindstrom was pretty much right on. We were able to keep up a good speed around the maze. In addition to this we had some very good code for actually detecting the cheese if we could have reached it.

The only other significant limitation would be the limited amount of hardware. If we would have had access to a few more light sensors we could have possibly implemented something a little more intelligent. In the future it would be best to let the students share or purchase additional Lego parts so that there was room for a little more creativity.

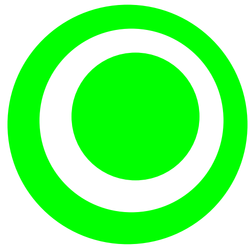
Analysis

Unfortunately, our robot was relatively unsuccessful on competition day. In trial, we had a few successful runs in which the robot was able to navigate a maze relatively successfully using line following. One problem we encountered in trial was that the robot could not successfully follow a line that was in the T shape. In the illustration below, the robot would be following on the vertical line, and come across a horizontal line in its tracks.



Now, the robot *should* turn left, with the new wall coming out horizontally from the one we are following. However, with one sensor, the robot will simply continue following the right hand wall and will roll right over the horizontal wall. This is a failing our robot had often in trial, and it was a problem as well on the final demonstration.

One aspect of our algorithm that was quite successful was the goal detection. Our first goal detection algorithm was extremely successful, though slightly more complex than the one we opted to use. Our first algorithm would wait until green was detected, once green was detected, it would roll forward for about 3 seconds. During the roll forward, it would check for white. Once it saw white, it would check for green again. This was optimal for the type of goal we were detecting, as illustrated below.



However, since our final algorithm only checked for the goal when it would be on a white background, we were able to use a simpler check to find the green background. We simply determined accurate thresholds for the color green, and checked to see if the light sensor.

Although our algorithm was relatively unsuccessful in practice, there were some parts of the project that made it fun and interesting. It was much more fun to code in a C like language than PBasic. Also, the speed of the lego robots was far superior to the Boe Bots, which made testing much more interesting and entertaining.

Conclusion

The implementation of our Mindstorm could be easily enhanced in a number of ways. First, we think that if we had more time we could have had a better opportunity to experiment with different configurations. It makes it very difficult to be limited to only class time for building and testing different configurations. At the same time we realize that college students can't be trusted with not losing all of the little pieces. Secondly, if we had another photo sensor to detect the other wall we think that we would have been able to develop a much better algorithm for wandering intelligently around the maze.

We believe that with a little more time and hardware, El roboto could have done better at traversing the maze. However, the fact that we successfully managed to come up with a unique approach for finding the cheese is good enough for us. This project has taught us a great deal about working with robotics. In the end, even though we didn't fair so well in the competition it was still a very educational experience.

References

[1] Baum, Dave. *Definitive Guide to LEGO Mindstorms*, pp 83-95.

[2] Lego Inc. "LEGO MINDSTORMS Robotics Invention System 2.0" Guide Book.

Appendix

Source code

Approach Standard Wall Following with moving sensor

```
// linebot2.nqc modification

// By Stephanie Liese & Matthew Myrick

// sensor and motors
#define EYE       SENSOR_2
#define LEFT      OUT_A
#define CENTER OUT_B
#define RIGHT     OUT_C

// thresholds
#define LEFT_THRESHOLD  42
#define RIGHT_THRESHOLD 53

#define STRAIGHT_TIME 10

task main()
{
      SetSensor(EYE, SENSOR_LIGHT);
      On(LEFT+RIGHT);
  int green = 0, count = 0, count1 = 0;
      while(true)
      {
    top:
    if (count++ > 25)
    {
        //we would move around the arm here
        PlaySound(SOUND_UP);
        Off(LEFT+RIGHT);

           //RIGHT
            On(OUT_B);
            SetPower(CENTER, 1);

            while(count1++ < 30) {
              Rev(OUT_B);
              Wait(1);
            }
            Off(OUT_B);
           Wait(25);


           count1 = 0;
   //LEFT

            if (EYE > 407 && EYE < 47)     //check for green
            {
              PlaySound (SOUND_UP);
```

```
          PlaySound (SOUND_DOWN);
          Off(LEFT+RIGHT);
          return;
         }
       On(OUT_B);
       SetPower(CENTER, 1);
       while(count1++ < 30) {
          Fwd(OUT_B);
          Wait(1);
       }
       Off(OUT_B);


 count1 = 0;
 count=0;
 On(LEFT+RIGHT);

 }


    /*
    if (EYE > 43 && EYE < 47)     //check for green
{
    green++;
    if (green > 2)
    {
            PlaySound (SOUND_UP);
            Off(LEFT+RIGHT);
            return;
    }
    Fwd(LEFT+RIGHT);
    until (EYE < 43 || EYE > 47);
}
*/




        if (EYE <= LEFT_THRESHOLD)
        {

  while(EYE < LEFT_THRESHOLD) {
   // if (count++ > 50)
   //   goto top;
    Rev(LEFT);
              }

              Fwd(LEFT);
              Wait(STRAIGHT_TIME);
        }
        else if (EYE >= RIGHT_THRESHOLD)
        {
  while (EYE > RIGHT_THRESHOLD) {
   // if (count++ > 50)
   //   goto top;
    Rev(RIGHT);
```

```
                        }

                        Fwd(RIGHT);
                        Wait(STRAIGHT_TIME);
                }
        }
}
```

Appendix B

CSCI 233 - Program #3
Program #3 is worth 200 points.

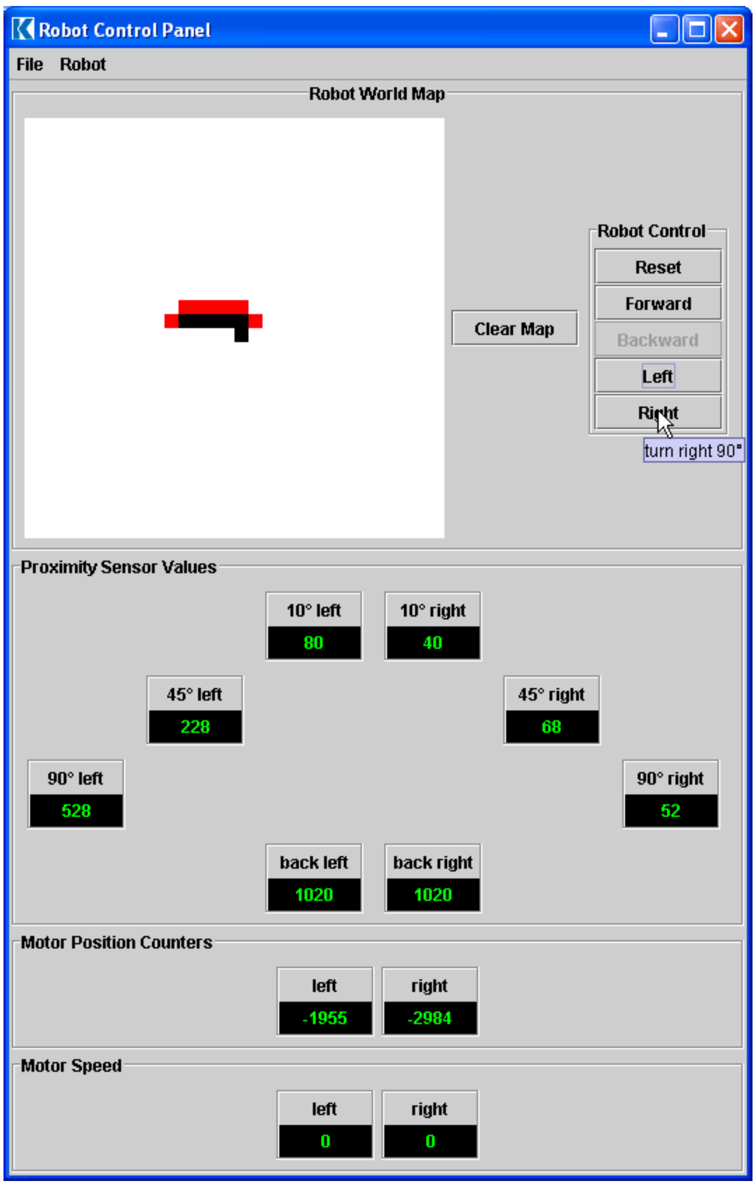
Please Note: **This assignment requires some specialized equipment. Let me know when you are ready to start working on this assignment and I will set up the robot table, robot, and web cam.**

Objectives:

- Implement a fairly complex GUI with several different panels and files.
- Customize a JPanel by extending and then overriding paintComponent( ).
- Use Action objects to create both buttons and menu items.
- Use TitledBorders to group and organize components.

Description:

You will be writing a GUI to control a Khepera II robot that is sitting in my office connected to my machine via the serial port. I have a robot server running that will accept connections over the network and perform the necessary communications with the robot. I will be providing a jar file with the robot client software. Your job will be to create the three source files that are described below, each implements a part of the GUI. Here is what your GUI will look like:



The Khepera II is small, only 70mm in diameter, and is round. The robot has 8 infrared proximity detectors - three to the front and left, three to the front and right, and two at the back.They can be seen around the outside circumference of the robot:

Appendix B

A large value from a sensor reading indicates that there is something (an obstacle) close by. In the GUI above you can see that the back left and back right proximity sensors are detecting something close.The robot has two wheels, each of which has a wheel motor position counter. A robot model class, **robot.shared.Robot**, will be provided in the jar file and will give you the following:

- a static constant - **Robot.N_SENSORS** - the number of sensors (8)
- a static constant - **Robot.OBSTACLE** - a value above which indicates an obstacle (800)
- a static constant - **Robot.LEFT_90** - index specifying a sensor - pass these to **getProximitySensor( )**
- a static constant - **Robot.LEFT_45**
- a static constant - **Robot.LEFT_10**
- a static constant - **Robot.RIGHT_10**
- a static constant - **Robot.RIGHT_45**
- a static constant - **Robot.RIGHT_90**
- a static constant - **Robot.RIGHT_BACK**
- a static constant - **Robot.LEFT_BACK**
- **String getSensorName(int s)** - returns the name of sensor s
- **int getProximitySensor(int s)** - returns the value of proximity sensor s
- **int getLeftMotorPosition( )** - returns the value of the counter for the left wheel motor
- **int getRightMotorPosition( )** - returns the value of the counter for the right wheel motor
- **int getLeftMotorSpeed( )** - returns the value of the left motor speed
- **int getRightMotorSpeed( )** - returns the value of the right motor speed
- **void moveForwardOne( )** - moves the robot forward one robot length
- **void moveBackwardOne( )** - moves the robot backward one robot length
- **void turnLeftNinety( )** - turns the robot left on its center by 90 degrees
- **void turnRightNinety( )** - turns the robot right on its center by 90 degrees
- **void sendMotorPosition(int left, int right)** - sets the wheel motor counters

You will use these constants and methods to implement the controls and displays in the robot control panel above.

Here is a zip file with the things you need to set up a development environment for this GUI assignment. Unzip it, maintaining the given directory structure. Because the server and client implementations share some classes, the files are organized into packages. After unzipping you should be able to compile, and even execute, the provided code. Use the batch file given, or at least look at it to see how to compile and execute with the jar file. There will be no GUI, however, when you execute. Please tell me about any problems you have. Included in RobotControl.zip are:

- the source files you will modify to implement the GUI, in the correct folders
  - robot\shared\RobotStatusPanel.java
  - robot\control\MapPanel.java
  - robot\control\RobotControl.java
- RobotControl.jar - contains several classes you will need for the Robot model and communication with the server
- robotcontrol.properties - the properties file with the IP address and port of the RobotServer
- khepera.gif - contains the graphic to use in the top-level window
- doIt.bat - a Windows batch file to compile and execute - type "doIt"

The parts of the GUI above should be implemented in these three files as follows:

- **RobotStatusPanel.java**
  - Contains a public class which extends JPanel and implements Observer.
  - It implements the three bottom portions of the GUI above: proximity sensor values, motion position counters, and motor speed.
  - This is just a status panel - there are no controls here to affect the robot.
  - The constructor for the RobotStatusPanel receives a reference to the Robot object.
  - The Robot extends Observable, so RobotStatusPanel can register to receive updates from the Robot.The robot server polls the robot twice per second and sends these updates out over the network. This will result in calls to your update method.
  - The reference to the robot can be used to retrieve the sensor values and such using the methods listed above.
- **MapPanel.java**
  - Implements a customized JPanel that provides the map of the robot's environment.
  - This map is being built up as you navigate the robot around the space.
  - The painting surface should be a 30 x 30 grid of cells where each cell is 10 x 10 pixels.
  - A white cell indicates the contents are unknown. A black cell indicates the cell is empty. A red cell indicates an obstacle occupies the cell.

Appendix B

  - A separate class for the map model will be implemented in this file. This class will maintain an array for the grid with values indicating the contents of each cell.
  - The view/controller for the MapPanel will also provide pass-through methods to set particular grid cells to empty/occupied, and to clear the grid to the unknown state.
- **RobotControlGUI.java**
  - Provides the JFrame, menu system, and control buttons, as well as providing a container for the MapPanel and RobotStatusPanel.
  - An Action object will be defined for each of the five robot actions: reset the robot, go forward one robot length, go backward one robot length, turn right 90 degrees, and turn left 90 degrees.
  - Each Action item will have a listener, tooltip text and accelerator key, and will be used to instantiate both a button and a menu item.
  - There are several instances (discussed below) when one or more buttons and menu items will be disabled. Action objects give us an easy way to do this and keep the menu and button states consistent.

The main method is implemented for you in RobotControl, one of the classes that provides communication with the robot server. This class will call the constructor for the RobotControlGUI after it establishes communication with the robot server, passing your constructor a reference to itself and to a Robot object. RobotControlGUI will create MapPanel and RobotStatusPanel.

Details, details, details...

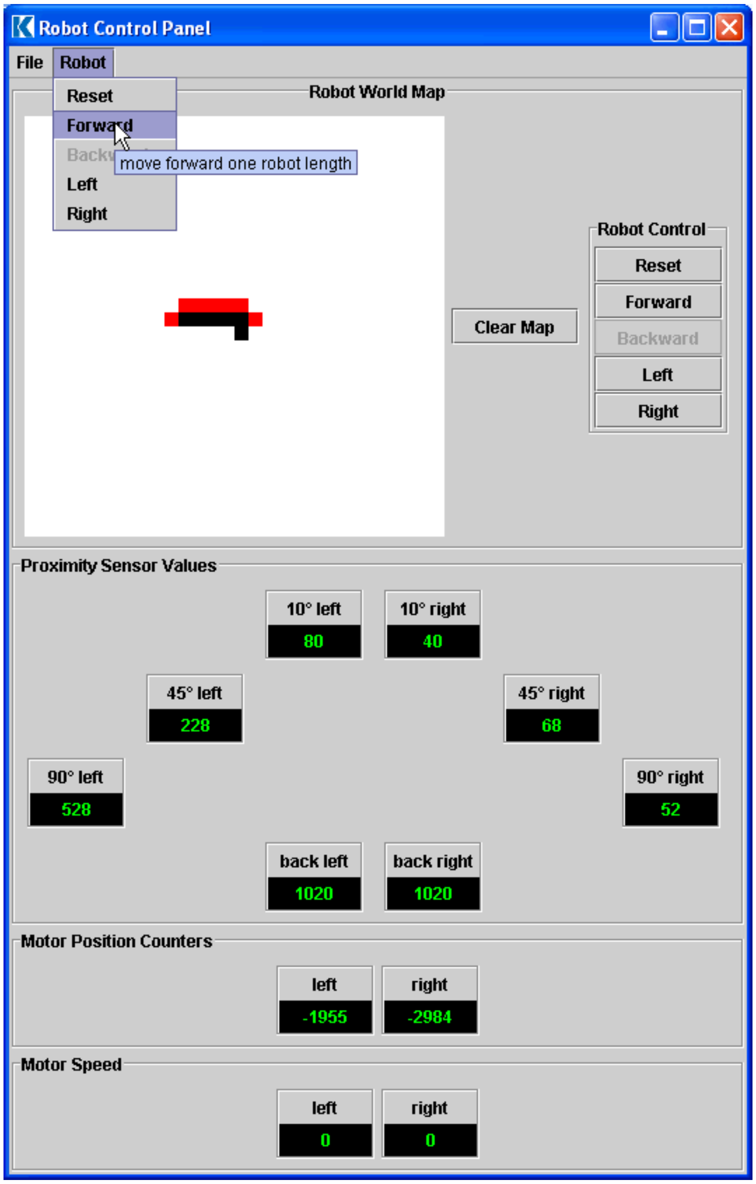
- There is only one menu item in the "File" menu, and that is "Exit". This is the only place you will need to use the reference to the RobotControl object that is passed in to the constructor for RobotControlGUI. Use it to terminate the program by invoking **robotControl.shutdown( )**.
- There are five menu items in the "Robot" menu, matching the five buttons in the "Robot Control" panel, as shown in the screen shot below. As discussed above, each button/menu item will be created with an Action object. The tooltips are as shown in the screen shots.
- The various panels and groups of controls are delineated using borders. All of the borders are of type TitledBorder and are created using an EtchedBorder.
- The panel titled "Robot World Map" containing the MapPanel, "Clear Map" button, and "Robot Control" panel is added to the "North" of the content pane.
- The RobotStatusPanel is added to the "South" of the content pane.
- You will need to keep track of "where" the robot is (it is relative to where it has been - no way to know absolutely). This is done in terms of a row and column number which corresponds to a cell in the map. You may just assume when the user clicks on "Forward" that the robot really did move forward the correct amount. (It may not have if it is stuck. You could check the motor position counters before and after, but it is not a requirement of this assignment.) After the robot moves, you will need to determine the new position (row and column). This is quite tricky. I used a boolean "swap" to keep track of whether I will be incrementing the row or the column when the robot moves forward. In addition, I used an int "forward" that is set to either 1 or -1 to keep track of whether I am adding or subtracting 1 to the row/column when the robot moves forward.I start out assuming the robot is facing the up direction on the screen (swap is false and direction is -1). Come see me if you have trouble with this part!
- The turn left and turn right methods cause the robot to turn on its center. It will not change map cells, just its orientation.
- Each time the robot finishes moving, it should check its proximity sensors for obstacles. Calls to methods in the MapPanel will set those neighboring cells that contain obstacles to indicate that they are occupied (red). A call should be made to indicate the the cell the robot is currently occupying is empty (black). All cells which have not yet been explored are unknown (white). The MapPanel object maintains this information in its model.
- Any time there is an obstacle directly behind the robot, the "Backward" button and menu item should be disabled. Any time there is an obstacle directly in front of the robot, the "Forward" button and menu item should be disabled.
- Whenever the robot is commanded to move (forward, back, turn left, turn right), all of the buttons and menu items that cause the robot to move should be disabled until the motor speed of both wheels is zero.
- The "Reset" button and menu item should set the wheel motor position counters to zero.
- The "Clear Map" button should set the robot position back to a default starting position in the map (15, 15) and clear all cells except the one the robot is currently occupying to the "unknown" state (white).
- If you want to you may color the cell the robot is currently on a different color, but it should turn to black once the robot moves off it.
- Layout of the "Proximity Sensor Values" panel in RobotStatusPanel was accomplished using a 6 x 4 GridLayout. An empty JPanel was added to each of the 24 grid locations, and then the status panels were added to the appropriate JPanels.
- **Very important!** Do not make any calls on the Swing components from the update( ) method. It will not be executing in the event dispatch thread. You must use SwingUtilities.invokeLater( ) from the update( ) method to schedule the updates to the GUI.

Other notes:

- It is common for the robot to get stuck. You can tell when this happens because when you click on "Forward" or "Backward" the motor position counters will not change, or will not change by much. If this happens try doing several turns and forward/backward motions. I'll be checking it frequently and will unstick it whenever I see that it needs it (or call or send me email).
- It is possible you will experience crashes and/or connection problems. For the most part, just retry. If you are stuck, alk to me and I will try to figure out what the problem is. This is the most complex assignment I have ever tried to give - so please be patient!
- Likewise, it is quite possible I have forgotten to tell you something important in this assignment. Writing the assignment was harder than writing the code! So please - don't hesitate to ask questions.

In this screen shot note that the "Backward" button and menu item are both disabled, and note the tooltip for the "Forward" button and menu item.

Appendix B



If you have any questions about how the Robot Control Panel should work, be sure to call or email!

Documentation

Note: Some code has been provided for you. You are responsible for all of the documentation.

See my Programming Assignment Expectations and Code Conventions for Java.

Turn-in Procedure

Send Professor Challinger an email at judyc@ecst.csuchico.edu. The subject line in your email should look like this:

CSCI 233 - Program 3 - Your Name

Your three source files (MapPanel.java, RobotStatusPanel.java, and RobotControlGUI.java) should be attached.

Robot Control Assignment Survey

Number of questions: 12

Finish Help

Question 1

This survey asks you to rate and give feedback on various aspects of the robot control assignment. Please answer all questions honestly. Surveys in WebCT are completely anonymous. The instructor can see that you completed a survey and can get the compiled responses of all completed surveys, but there is no information regarding which response came from which student. Thank you!

How clear were the learning goals for the robot control assignment?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 2

How clearly did the robot control assignment specify what you were to implement?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 3

How helpful was the robot control assignment in giving you direction as to how to get started?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 4

How appropriate was the level of difficulty of the assignment?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 5

How relevant was the robot control assignment to the course objectives and to your academic goals?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 6

How stimulating was the robot control assignment in terms of inspiring you to want to work on the project?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 7

How stimulating was the robot control assignment with respect to the level of interest it generated in you for further investigations into graphical user interfaces, robotics, or some other aspect of the assignment?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 8

How effective was the robot control assignment in helping you learn more about the implementation of graphical user interfaces?

- a. Excellent
- b. Good
- c. OK
- d. Not Good
- e. Terrible

Save answer

Question 9

What did you think of the robot control assignment overall?

a. Excellent

b. Good

c. OK

d. Not Good

e. Terrible

Save answer

Question 10

How effective was the robot control assignment at minimizing technical difficulties that might hamper your learning?

a. Excellent

b. Good

c. OK

d. Not Good

e. Terrible

Save answer

Question 11

Please share what you liked and/or didn't like about the robot control assignment. Be as specific as possible about why you liked or disliked any aspect of it.

Equation: Equation editor

Save answer

Question 12

Do you think this assignment, or one similar to it, should be used in this course in the future? Please share any suggestions you have for improving the robot control assignment, or make any other comments you would care to.

Equation: Equation editor

```
'   Template - Create your own TAB Electronics Build Your
'   Own Robot Kit BS2 application starting with this
'   file.
'
'  File Updated for SumoBot
'
'  Myke Predko
'
'  Copyright (C) 2001 & 2002 McGraw-Hill
'
' { $STAMP BS2 }


'  Variables
Temp var word                          '  Get a 16 Bit Variable


'  Mainline
      high SC                          '  Set the I/O Bits As O/P
      high SD                          '   and High
      high LED                         '  Turn off the LED

'  #### - Application Code Goes Here



'  Robot Interface Code Follows:
'
'  Myke Predko
'
'  Copyright (C) 2001 McGraw-Hill
'
'  Robot Commands
RobotStop     con  0                   '  Stop the Robot
Behavior1     con  1                   '  Random Movement
Behavior2     con  2                   '  Photovore
Behavior3     con  3                   '  Photophobe
Behavior4     con  4                   '  Wall Hugger/Maze Solver
RobotForward  con  5                   '  Move Forward for 200 msecs
RobotReverse  con  6                   '  Move Reverse for 200 msecs
RobotLeft     con  7                   '  Turn Left for 200 msecs
RobotRight    con  8                   '  Turn Right for 200 msecs
RobotLEDOn    con  9                   '  Turn on the Robot's LED
RobotLEDOff   con 10                   '  Turn off the Robot's LED
RobotPWM0     con 11                   '  PWM = 0% Duty Cycle
RobotPWM1     con 12                   '  PWM = 1st "Notch"
RobotPWM2     con 13                   '  PWM = 2nd "Notch"
RobotPWM3     con 14                   '  PWM = 3rd "Notch"
RobotPWM4     con 15                   '  PWM = 100% Duty Cycle
RobotPWM      con 16                   '  Return the Current PWM Value
RobotState    con 17                   '  Return the Executing State
RobotWhiskers con 18                   '  Return State of the "Whiskers"
                                       '   Bit 0 - Left "Whisker"
                                       '   Bit 1 - Right "Whisker"

RobotCDSL     con 19                   '  Return Value of Left CDS Cell
RobotCDSR     con 20                   '  Return Value of Right CDS Cell
RobotButton   con 21                   '  Return the Last Remote Button Press
                                       '   0 - No Buttons Pressed
                                       '   1 - Leftmost Button Pressed
                                       '   2 - Middle Button Pressed
                                       '   3 - Rightmost Button Pressed
                                       '  After "RobotButton" Operation,
                                       '   Button Save is Cleared


'  Robot Interface Pins
LED con 11                             '  LED, Negative Active On
RIR con 12                             '  Right Infra-Red Detector
LIR con 13                             '  Left Infra-Red Detector
SC con 14                              '  Define the I/O Pins
SD con 15
```

```
'  Robot Interface Variables
RobotData var byte                      '  Data Byte to Send to/Receive
                                        '   from Robot

'  Robot Operation Subroutines
RobotSend                               '  Send the Byte in "RobotData"
       low SC                           '  Hold Low for 1 msec before
       pause 1                          '   Shifting in Data
       shiftout SD, SC, LSBFIRST, [RobotData]
       high SC
       return

RobotSendReceive                        '  Send the Byte in "RobotData"
       low SC                           '  Hold Low for 1 msec before
       pause 1                          '   Shifting in Data
       shiftout SD, SC, LSBFIRST, [RobotData]
       pause 1                          '  Wait for Operation to Complete
       shiftin SD, SC, LSBPOST, [RobotData]
       high SC
       return
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'        TAB Electronics Build Your Own Robot Kit BX24 Interface Module
'                  by  Dr. J (Juliano@csuChico.edu)
'                  and Matt Bauer (MBauer1@csuChico.edu)
'                  California State University, Chico
'                  Intelligent Systems Laboratory
'                  Chico, CA 95929-0410
'                  http://isl.ecst.csuchico.edu
'
'  This module contains the RobotSend() and RobotSendReceive() subroutines for
'  communication with the PIC 16C505 which is the onboard co-processor for the
'  Build Your Own Robot Kit by TAB Robotics.
'
'  The code in this module is based on the BS2 programming template written by
'  Myke Predko to facilitate any BS2 to BX24 code migration.
'
'  Routines for serial pins:
'
'    Public Sub SetSerialPins   ( ByVal SC As Byte , ByVal SD As Byte )
'      - sets private interface variables SC as clock pin and SD as data pin
'    Public Sub GetSerialPins   ( ByRef SC As Byte , ByRef SD As Byte )
'      - gets the values of private interface variables into SC and SD
'    Public Sub SetSerialPinsHi ()
'      - sets private interface variables to pre-defined "high" value
'    Public Sub SetSerialPinsLo ()
'      - sets private interface variables to pre-defined "low" value
'    Public Sub InitSerialPins  ()
'      - sets private interface variables "high" for 1 msec
'    Public Sub ResetSerialPins ()
'      - sets private interface variables "low" then "high" for 1 msec
'
'  Routines for robot communication:
'
'    Public Sub ShiftOutSlowly  ( ByVal DataByte As Byte )
'      - slowed down ShiftOut() to match clock rate of PIC 16C505
'    Public Sub ShiftInSlowly   ( ByVal DataByte As Byte )
'      - slowed down ShiftIn() to match clock rate of PIC 16C505
'    Public Sub RobotSend       ( ByVal DataByte As Byte )
'      - send predefined command to the robot
'    Public Sub RobotSendReceive( ByRef DataByte As Byte )
'      - send predefined command to, and get result back from, the robot
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Attribute VB_Name = "BX24_byork"

Option Explicit

' Predefined Robot Commands

Public  Const RobotStop     As Byte =  0  '  Stop the Robot
Public  Const Behavior1     As Byte =  1  '  Random Movement
Public  Const Behavior2     As Byte =  2  '  Photovore
Public  Const Behavior3     As Byte =  3  '  Photophobe
Public  Const Behavior4     As Byte =  4  '  Wall Hugger/Maze Solver
Public  Const RobotForward  As Byte =  5  '  Move Forward for 200 msecs
Public  Const RobotReverse  As Byte =  6  '  Move Reverse for 200 msecs
Public  Const RobotLeft     As Byte =  7  '  Turn Left for 200 msecs
Public  Const RobotRight    As Byte =  8  '  Turn Right for 200 msecs
Public  Const RobotLEDOn    As Byte =  9  '  Turn on the Robot's LED
Public  Const RobotLEDOff   As Byte = 10  '  Turn off the Robot's LED
Public  Const RobotPWM0     As Byte = 11  '  PWM = 0% Duty Cycle
Public  Const RobotPWM1     As Byte = 12  '  PWM = 1st "Notch"
Public  Const RobotPWM2     As Byte = 13  '  PWM = 2nd "Notch"
Public  Const RobotPWM3     As Byte = 14  '  PWM = 3rd "Notch"
Public  Const RobotPWM4     As Byte = 15  '  PWM = 100% Duty Cycle
Public  Const RobotPWM      As Byte = 16  '  Return the Current PWM Value
Public  Const RobotState    As Byte = 17  '  Return the Executing State
Public  Const RobotWhiskers As Byte = 18  '  Return State of the "Whiskers"
                                    '   Bit 0 - Left "Whisker"
                                    '   Bit 1 - Right "Whisker"
```

```
Public  Const RobotCDSL     As Byte = 19  '  Return Value of Left CDS Cell
Public  Const RobotCDSR     As Byte = 20  '  Return Value of Right CDS Cell
Public  Const RobotButton   As Byte = 21  '  Return the Last Remote Button Press
                                   '   0 - No Buttons Pressed
                                   '   1 - Leftmost Button Pressed
                                   '   2 - Middle Button Pressed
                                   '   3 - Rightmost Button Pressed
                                   '  After "RobotButton" Operation,
                                   '   Button Save is Cleared
'  Robot Serial Interface Pins

Public  Const SC As Byte = 19            '  Default BX-24 clock pin
Public  Const SD As Byte = 20            '  Default BX-24  data pin

'  BX-24 private serial interface variables

Private ClockPin As Byte
Private Data_Pin As Byte


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set the robot serial interface pin numbers to SC for the clock pin
' and SD for the data pin.


Public Sub SetSerialPins( _
    ByVal SC As Byte, _
    ByVal SD As Byte    )

    ClockPin = SC
    Data_Pin = SD

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Get the robot serial interface pin numbers, storing clock pin to SC and
' the data pin to SD.

Public Sub GetSerialPins( _
    ByRef SC As Byte, _
    ByRef SD As Byte    )

    SC = ClockPin
    SD = Data_Pin

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set the robot interface pins as high.

Public Sub SetSerialPinsHi()

    Call PutPin( ClockPin , bxOutputHigh )
    Call PutPin( Data_Pin , bxOutputHigh )

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set the robot interface pins as low.

Public Sub SetSerialPinsLo()

    Call PutPin( ClockPin , bxOutputLow )
    Call PutPin( Data_Pin , bxOutputLow )

End Sub
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set the robot interface pins as output and high; this subroutine needs
' to be called before any application code.

Public Sub InitSerialPins()

    Call SetSerialPinsHi()
    Call Sleep ( 0.001 )

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Reset the robot interface pins as output and high.

Public Sub ResetSerialPins()

    Call SetSerialPinsLo()
    Call SetSerialPinsHi()
    Call Sleep ( 0.001 )

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Thanks to Don Kinzer of groups.yahoo.com/group/basicx for his suggestions on
' how to slow down the BX-24's ShiftOut() operation using PutPin(), PulseOut(),
' GetBit(), and PutBit() commands.  The slow-down is needed to match the clock
' rate of the BS2 for effective communication with the BYORK PIC 16C505
' microcontroller ...
' (Implemented as a fully unrolled loop; FOR loop implementation slower still.)

' Shift out data bits in DataByte to the robot in serial fashion, LSB-first.
' Data is 'rightshifted' into a register making the first bit sent the LSB.

Public Sub ShiftOutSlowly( ByVal DataByte As Byte )

'   Hold ClockPin and Data_Pin low for 1 msec before shifting in data
    Call SetSerialPinsLo()
    Call Sleep( 0.001 )

'   Send "DataByte" to the co-processor in serial fashion, LSB first;
'   sends an 8 bit high (10-12us) going clock pulse to co-processor
    Call PutPin( Data_Pin , GetBit( DataByte , 0 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 1 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 2 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 3 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 4 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 5 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 6 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutPin( Data_Pin , GetBit( DataByte , 7 ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )

'   Reset data and clock pins and wait for coprocessor
    Call ResetSerialPins()

End Sub
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Shift in data bits to DataByte from the robot in serial fashion, LSB-first.

Public Sub ShiftInSlowly( ByRef DataByte As Byte )

'   Reset data and clock pins and wait for coprocessor
    Call SetSerialPinsLo()
    Call Sleep ( 0.001 )

' Read an 8-bit value from co-processor in serial fashion, LSB first.
' Puts an 8 bit high(10-12)ms going pulse on the clock pin.
    Call PutBit( DataByte , 0 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 1 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 2 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 3 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 4 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 5 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 6 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )
    Call PutBit( DataByte , 7 , GetPin( Data_Pin ) )
    Call PulseOut( ClockPin , 10 , bxOutputHigh )

'   Reset data and clock pins and wait for coprocessor
    Call ResetSerialPins()

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Send the byte in "DataByte" to the robot

Public Sub RobotSend( ByVal DataByte As Byte )

'   Shift out data bits to the robot LSB-first ...
    Call ShiftOutSlowly( DataByte )

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Sends the byte in "DataByte" to the robot, using it for return value

Public Sub RobotSendReceive( ByRef DataByte As Byte )

'   Shift out data bits to the robot LSB-first and wait for completion ...
    Call ShiftOutSlowly( DataByte )

'   Read in result/status of requested operation
    Call ShiftInSlowly( DataByte )

End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

École polytechnique universitaire de Marseille

Département de Génie Industriel

INTERNATIONAL DEPARTMENT

Internship Office
CF/EC -SIR 2004
Marseilles, 3/02/2004

Director Benjoe A. Juliano
INTELLIGENT SYSTEMS
LABORATORY (ISL)
California State University, Chico
(CSUC), College of Engineering,
Computer Science, And Technology
(ECST),
400 West FIRST A venue CHICO CA
959290410
CA 95929-0410
U.S.A

Dear Professor,

Please find enclosed 3 copies of The Internship Agreement For Research Initiation for each Intern designated as follows :

LAVIOLETTE Marin PELEN Quentin that you welcome in your Laboratory

1er Juin à Août 2004

Please send back us 2 copies of each agreement signed by the Laboratory Supervisor.

Yours faithfully,

Claudia FRYDMAN
P/O. Internship Office

École **polytechnique** *universitaire de Marseille*

Département de Génie Industriel et informatique

AGREEMENT FOR RESEARCH INITIATION INTERNSHIP N° 2004- B 08

Article 1

The following agreement fixes the terms of the relationship

between the laboratory :

INTELLIGENT SYSTEMS LABORATORY (ISL)
California State University, Chico (CSUC), College of Engineering, Computer Science, And Technology (ECST),
400 West FIRST A venue CHICO CA 959290410
CA 95929-0410 U.S.A

represented by :Director Benjoe A. Juliano
hereafter referred to as LAB

and the **"Département de Génie Industriel et Informatique de l'Ecole Polytechnique Universitaire de Marseille (EPUM)"**

Domaine Universitaire de St Jérôme
Avenue Escadrille Normandie-Niémen
13397 MARSEILLE Cedex 20

represented by : Charles SANTONI, Director
hereafter referred to as THE DEPARTMENT

concerning the internship done
from 1er Juin à Août 2004
by : **LAVIOLETTE Marin**
hereafter referred to as INTERN

Article 2

The DEPARTMENT will make INTERN aware of this agreement and obtain, prior to initiating the internship, express consent to all of its articles.

Article 3

The minimum duration of the internship will be one month. The internship is part of the second year of studies at The DEPARTMENT and is to take place in June. If need be, it may extend into the next month or months, but under no account beyond August 31 st.

Article 4

The essential goal of this internship is to initiate INTERN into research activities by being in contact with a team of researchers.

Article 5

The particular program of this internship will be left to the initiative of the head of LAB, who is to inform The DEPARTMENT about it.

Article 6

Throughout the internship, the INTERN will still be considered a student. He will have to take out a specific insurance policy covering civil liability, health care and legal assistance, as well as

Article 7 The present agreement can only be signed by The DEPARTMENT after an insurance policy certificate has been produced as specified in article 6.

Article 8

Throughout the internship, INTERN will be subject to the regulations and discipline of LAB, particularly as concerns hours of work. He/She will make it a duty to be totally silent about all data and information that He/She might come across in the course of the internship.

Article 9

No remuneration is to be associated with the internship or it would nullify article 6. This also excludes any kind of allowance.

Article 10

The INTERN will deal directly with his insurance company if need be. Under no account will The DEPARTMENT intervene at this point.

Article 11

The internship work consists in analysing and synthesizing scientific documents dealing with the following topic :
Investigation of intelligent system approaches in the area of mobile robotics
These documents will be given to INTERN upon his arrival at LAB.

Article 12

This work will lead the INTERN in writing a summarised paper and supervised by the advisor according to the internship policy.

The internship advisor being : Dear Benjoe A. Juliano

Done in triplicate
in Marseilles on3/02/2004

LAB Representative	The DEPARTMENT Director
Benjoe Juliano Benjoe A. Juliano	POLYTECH Charles SANTONI
INTERN	In Charge of International Relation
LAVIOLETTE Marin	Claudia FRYDMAN

École **polytechnique** *universitaire de Marseille*

Département de Génie Industriel et informatique

AGREEMENT FOR RESEARCH INITIATION INTERNSHIP N° 2004- B 08

Article 1

The following agreement fixes the terms of the relationship

between the laboratory :
INTELLIGENT SYSTEMS LABORATORY (ISL)
California State University, Chico (CSUC), College of Engineering, Computer Science, And Technology (ECST), 400 West First Avenue, Chico,
CA 95929-0410 U.S.A
represented by :Director Benjoe A. Juliano
hereafter referred to as LAB

and the **"Département de Génie Industriel et Informatique de l'Ecole Polytechnique Universitaire de Marseille (EPUM)"**
Domaine Universitaire de St Jérôme
Avenue Escadrille Normandie-Niémen
13397 MARSEILLE Cedex 20

represented by : Charles SANTONI, Director
hereafter referred to as THE DEPARTMENT

concerning the internship done
from 1er Juin à Août 2004
by : **PELEN Quentin**
hereafter referred to as INTERN

Article 2

The DEPARTMENT will make INTERN aware of this agreement and obtain, prior to initiating the internship, express consent to all of its articles.

Article 3

The minimum duration of the internship will be one month. The internship is part of the second year of studies at The DEPARTMENT and is to take place in June. If need be, it may extend into the next month or months, but under no account beyond August 31 st.

Article 4

The essential goal of this internship is to initiate INTERN into research activities by being in contact with a team of researchers.

Article 5

The particular program of this internship will be left to the initiative of the head of LAB, who is to inform The DEPARTMENT about it.

Article 6

Throughout the internship, the INTERN will still be considered a student. He will have to take out a specific insurance policy covering civil liability, health care and legal assistance, as well as individual accident guarantee for the whole duration of his stay abroad in the Laboratory.

Article 7 The present agreement can only be signed by The DEPARTMENT after an insurance policy certificate has been produced as specified in article 6.

Article 8

Throughout the internship, INTERN will be subject to the regulations and discipline of LAB, particularly as concerns hours of work. He/She will make it a duty to be totally silent about all data and information that He/She might come across in the course of the internship.

Article 9

No remuneration is to be associated with the internship or it would nullify article 6. This also excludes any kind of allowance.

Article 10

The INTERN will deal directly with his insurance company if need be. Under no account will The DEPARTMENT intervene at this point.

Article 11

The internship work consists in analysing and synthesizing scientific documents dealing with the following topic :
Investigation of intelligent system approaches in the area of mobile robotics
These documents will be given to INTERN upon his arrival at LAB.

Article 12

This work will lead the INTERN in writing a summarised paper and supervised by the advisor according to the internship policy.

The internship advisor being : Dear Benjoe A. Juliano

Done in triplicate
in Marseilles on3/02/2004

LAB Representative	The DEPARTMENT Director
Benjoe A. Juliano	Charles SANTONI
INTERN	**In Charge of International Relation**
PELEN Quentin	Claudia FRYDMAN

polytech marseille

Visa de : E. CHOURAQUI. Tél.04 91 05 60 12
Ou **J.M. MERCANTINI.** Tél.04 91 05 60 15

Responsables des SIR

Le :
Signature :

PROPOSAL FOR AN INTRODUCTORY INTERNSHIP TO RESEARCH (S.I.R)

(Informations to fill convention of internship)

STUDENTS INDENTIFICATION

(to fill by the trainees)

Name : 1)Laviolette Marin…………………………… **Binomial Name** : 2) Pelen Quentin……………
Option : Génie Industriel et Informatique……………… **Option :** Génie Industriel et Informatique ………..

Personnal address and phone number (where we can contact you during the internship):

……………………………………………… ………………………………………………
……………………………………………… ………………………………………………
……………………………………………… ………………………………………………
……………………………………………… ………………………………………………

RESEARCH LABORATORY IDENTIFICATION

(to fill by the laboratory)

Name : Intelligent Systems Laboratory (ISL)
Address : California State University, Chico (CSUC), College of Engineering, Computer Science, And Technology (ECST), 400 West First Avenue, Chico, CA 95929-0410
Phone : 530 898-6442 .... **Fax :** 530 898-5995 ....

Subject of the study : Investigation of intelligent system approaches in the area of mobile robotics………..
………

Internship start : ……. June 2004.
Internship end : ………August 2004……………………………………………………

Contact :

Administrative person in charge of the internship :
Name : Dr. Benjoe A. Juliano
(Signatory of conventions) **Job or function :** ISL Director, Associate Professor
Phone : 530 898-4619

Tutor of the internship : **Name :** …Dr. Benjoe A. Juliano……………………………………………
(Encadrant les stagiaires) **Job or function** …ISL Director, Associate Professor………………………
Phone : …530 898-4619……………………………………………………
E-mail : …Juliano@csuChico.edu…………………………………………

With which to address conventions? (name, service...) …Dr. Benjoe A. Juliano………………………
………

Date and signature from the tutor of internship : Benjoe Juliano

Short description of the subject of the study:

The College of Engineering, Computer Science, and Technology (ECST) at California State University, Chico (CSUC) was recently awarded a National Science Foundation (NSF) grant to set up an Intelligent Systems Laboratory (ISL). Dr. Juliano is the Principal Investigator of the project and the Director of the ISL. The focus of ISL is to encourage research and studies in intelligent systems applied to search and rescue. This necessitates the investigation of the design, implementation, and testing of intelligent approaches (e.g. use of fuzzy logic, neural networks, genetic algorithms, etc.) in autonomous mobile robotics. The students in this internship to research will have the opportunity to work with other students and faculty on various robotics platforms, and the chance to attempt to solve a variety complex problems using mobile robots. The students will work under the direction/supervision of Dr. Juliano who will give the students specific tasks to complete during the internship period. Additionally, the students will also be individually required to prepare and submit a technical report/manuscript (in English) detailing their own work, experiences, and any findings.

Return the forms to:

Scolarité de POLYTECH - Service des Stages (SIR)
Domaine Universitaire de St Jérôme
13397 MARSEILLE Cedex 20

Fax 04 91 05 60 51

Appendix F

CALIFORNIA STATE UNIVERSITY, CHICO

College of Engineering, Computer Science, and Technology

INSTITUTE FOR RESEARCH ON INTELLIGENT SYSTEMS

Intelligent Systems Laboratory - Summer Robotics Camp

O'Connell Technology Center, OCNL 431

MISSION

The Chico State Intelligent Systems Lab (ISL) has developed a week long, interactive Summer Robotics Camp to provide girls going into the 8th grade with the unique opportunity to learn more about science and math. This camp is designed to introduce these young women to the fields of Computer Science, Engineering, and Mechatronics through their learning, experimentation, building and use of robots. Daily activities and lessons allow participants to work as a member of a team in the lab where they will learn more about the science of robotics and eventually design and build their own robot.

SPONSORS

National Science Foundation (NSF) MRI/RUI grant EIA-0321385
CSUC College of Engineering, Computer Science, and Technology
CSUC Department of Computer Science
Drs. Gary and Judy Sitton
Sundog Screenprints
Associated Students, CSUC

ISL | WELCOME | PEOPLE | FACILITIES | PROJECTS | SUMMER ROBOTICS CAMP | CONTACTS | LINKS | SITEMAP

Appendix F

CALIFORNIA STATE UNIVERSITY, CHICO

College of Engineering, Computer Science, and Technology

INSTITUTE FOR RESEARCH ON INTELLIGENT SYSTEMS

Intelligent Systems Laboratory - Summer Robotics Camp
O'Connell Technology Center, OCNL 431

GENERAL INFORMATION

QUALIFICATIONS:

- Applicants must be female students going into the 8th grade.
- Applicants must be able and willing to participate in camp from June 14th through June 18th, 9:00am to 3:00pm daily.
- Applicants must be willing to take on and apply themselves to intellectual challenges.

DATES:

Monday, June 14th, 2004 through Friday, June 18th, 2004 from 9:00am to 3:00pm daily.

LOCATION:

California State University, Chico campus. Participants will meet in front of the O’Connell Technology Center, located on the corner of Warner and West 1st Street.

COST:

This program is FREE to all participants and includes lunch daily.

TRANSPORTATION:

Participating students of the Summer Robotics Camp must be responsible for their own transportation to and from CSU, Chico. The Intelligent Systems Lab or any of its employees cannot provide transportation.

DEADLINE:

Completed applications and supplemental materials must be received by May 15th, 2004. Students should receive notification of acceptance by June 1st, 2004.

ISL | WELCOME | PEOPLE | FACILITIES | PROJECTS | SUMMER ROBOTICS CAMP | CONTACTS | LINKS | SITEMAP

Chico State Intelligent Systems Lab
Summer Robotics Camp 2004

General Information

The Chico State Intelligent Systems Lab (ISL) has developed a week long, interactive Summer Robotics Camp to provide girls going into the 8th grade with the unique opportunity to learn more about science and math. This camp is designed to introduce these young women to the fields of Computer Science, Engineering, and Mechatronics through their learning, experimentation, building and use of robots. Daily activities and lessons allow participants to work as a member of a team in the lab where they will learn more about the science of robotics and eventually design and build their own robot.

QUALIFICATIONS:

- Applicants must be female's going into the 8th grade.
- Applicants must be able and willing to participate in camp from June 14th through June 18th, 9:00am to 3:00pm daily.
- Applicants must be willing to take on and apply themselves to intellectual challenges.

DATES:

Monday, June 14th, 2004 through Friday, June 18th, 2004 from 9:00am to 3:00pm daily.

LOCATION:

California State University, Chico campus. Participants will meet in front of the O'Connell Technology Center, located on the corner of Warner and West 1st Street.

COST:

This program is FREE to all participants and includes lunch daily.

TRANSPORTATION:

Participating students of the Summer Robotics Camp must be responsible for their own transportation to and from CSU, Chico. The Intelligent Systems Lab or any of its employees cannot provide transportation.

DEADLINE:

Completed applications and supplemental materials must be received by May 15th, 2004. Students should receive notification of acceptance by June 1st, 2004.

CONTACT INFO:

The Chico State Intelligent Systems Lab welcomes any questions or concerns about our Summer Robotics Camp. Please contact us at csuc_isl@yahoo.com or visit the camp's website at www.ecst.csuchico.edu/isl/SummerCamp.html

The Chico State ISL in funded in part by a grant from the National Science Foundation
Grant # 0321385
Acquisition of robotics equipment for an Intelligent Systems Laboratory

Chico State Intelligent Systems Lab
Summer Robotics Camp 2004

Student Application Form

Please print legibly in ink or type.

PART 1: GENERAL INFORMATION	-To be completed by applicant

Name: Last First M.I.

Mailing Address: Number and Street Apartment #

City: Zip Code:

Phone: - Gender: F M Current Grade:

What school do you currently attend? ____________________

Father's Name: ____________________ Work Phone: ( ) ____________

Address: ____________________ Home Phone: ( ) ____________

Mother's Name: ____________________ Work Phone: ( ) ____________

Address: ____________________ Home Phone: ( ) ____________

Other Legal Guardian: ____________________ Work Phone: ( ) ____________

Address: ____________________ Home Phone: ( ) ____________

In case of emergency, contact: ____________________

Relationship: ____________________ Phone: ( ) ____________

Secondary emergency, contact: ____________________

Relationship: ____________________ Phone: ( ) ____________

PART 2: SHORT ANSWER RESPONSES	-To be completed by applicant

Describe three qualities or characteristics about yourself that make you an ideal candidate for participation in the Summer Robotics Camp.

What skills, abilities, or knowledge do you wish to gain from your participation in the Summer Robotics Camp?

What subjects and fields interest you? What do you think you might do after High School?

In three sentences or less, tell us why you want to learn about robots.

PART 3: STUDENT AGREEMENT -To be completed by applicant

I agree to participate in the ISL Summer Robotics Camp at CSU, Chico from Monday, June 14, 2004 through Friday, June 18, 2004 from 9:00am to 3:00pm daily. I will attend and actively participate in all scheduled activities such as classes, workshops, laboratories, and recreational activities. I will abide by the rules and regulations of the Summer Robotics Camp and California State University, Chico. I understand that my failure to abide by these rules or any behavior problems will result in my immediate dismissal from the program.

_______________________________________ ____________________

Applicant Signature Date

PART 4: PARENT CONSENT, AGREEMENT -To be completed by a parent/guardian

As the parent/guardian of ________________________________, I certify, with my signature below, that my daughter has my consent to participate in the ISL Summer Robotics Camp at CSU, Chico.

I understand and agree that:

1. The Summer Robotics Camp will be held at California State University, Chico, from Monday, June 14, 2004 through Friday, June 18, 2004 from 9:00am to 3:00pm.
2. I am responsible for dropping off and picking up my daughter from the Summer Robotics Camp on time daily. The ISL or its staff cannot provide transportation to participants of the program.
3. If I am unable to pick up my daughter, I will authorize a designee. I understand that my daughter will not be allowed to leave the Summer Robotics Camp with a designee unless I have completed an authorization form and submitted it to the camp coordinator beforehand.
4. The Summer Robotics Camp is a free program and participants will be provided lunch daily.
5. Participants must attend and are expected to actively participate in all scheduled activities.
6. Participants will be supervised in the lab by a trained adult staff and program officials.
7. Directors will dismiss a participant from the program for failing to abide by university or program rules and regulations or for any behavior problems. In the event of dismissal, the parent/guardian agrees to pick up the participant as soon as contacted.
8. Program staff will not be responsible for administering over-the-counter or doctor-prescribed medication to participants.
9. I will be responsible for any ISL items being lost, stolen or damaged due to my daughter's negligence.
10. The ISL Summer Robotics Camp and California State University, Chico are not responsible for any of my daughters items being lost, stolen, or damaged.
11. The ISL may use pictures taken of my daughter participating in Summer Robotics Camp activities for ISL website content and CSUC promotional purposes.
12. The ISL Summer Robotics Camp, California State University, Chico, staff and owners of properties used for the program activities shall not be held responsible for injuries to my daughter during the period of enrollment in the program.

_______________________________________ ____________________

Parent/Guardian Signature Date

PART 5: AUTHORIZATION TO TREAT A MINOR -To be completed by a parent/guardian

In the event that my daughter becomes ill or sustains an injury while in the care or under the supervision of the ISL Summer Robotics Camp, the adult supervisors of the activity are given my permission to administer first aid for her relief. If it is not practical to return her to me or receive my instructions for her care:

I, the undersigned parent or legal guardian of the aforementioned student, a minor, do herby authorize and consent to any X-ray examination, anesthetic, medical or surgical diagnosis or treatment, and emergency hospital care, which is deemed advisable by and if rendered under the general or special supervision of any member of the medical staff and emergency room staff licensed under the provisions of the Medicine Practice Act and on the staff of any acute general hospital holding a current license to operate a hospital from the State of California Department of Health. It is understood that effort shall be made to contact the undersigned prior to rendering treatment to the patient, but that any of the above treatment will not be withheld if the undersigned cannot be reached. This authorization is given pursuant to provisions of Section 25.8 of the Civil Code of California.

I further agree not to hold the ISL Summer Robotics Camp liable for the medical aid rendered and will reimburse the ISL Summer Robotics Camp for any medical or other expenses incurred in the care of my daughter.

Does your child have any physical or medical condition/needs of which the program staff should be aware or that can restrict her participation in certain type of activities? ☐ Yes ☐ No

If yes, please describe: ____

____

Are there any medications that your child takes regularly? ☐ Yes ☐ No

If yes, please describe: ____

____

Are there any medications that your child is allergic to? ☐ Yes ☐ No

If yes, please describe: ____

____

Do your child have any special food restrictions or preferences (i.e. allergies, vegetarianism, religious prohibitions, etc.)? ☐ Yes ☐ No

If yes, please describe: ____

____

Family Doctor: ______________________ Date of last Tetanus Booster: ____________

______________________________ ______________________

Parent/Guardian Signature Date

PART 6: STUDENT SURVEY	-To be completed by applicant

The following information will be important for program officials to know in the event that you are selected. The information that you provide will not affect the evaluation of your application. Please provide honest responses.

What is your preferred adult, unisex, T-shirt size? ☐ S ☐ M ☐ L ☐ XL

How would you rate your level of computer skills? ☐ None ☐ Novice ☐ Intermediate ☐ Advanced

How would you rate your level of robotic knowledge? ☐ None ☐ Novice ☐ Intermediate ☐ Advanced

What type of music do you enjoy listening to? ____________________ ____________________

☐ Check here if none ____________________ ____________________

PART 7: RECOMMENDATION	-To be completed by a teacher or counselor

Enclosed please find a recommendation form to be completed by a teacher or counselor. Applicants will need to fill out the top portion of the form. Include your completed recommendation form in its sealed envelope with this application.

COMPLETED APPLICATION	-Please read carefully

A completed application must include the following:

☐ A completed application form

☐ A completed recommendation form

All parts of this application must be completed and mailed together or the application will be considered incomplete. Incomplete or late applications will not be processed. All information provided will remain confidential.

Mail completed application to:

ISL Summer Robotics Camp
Department of Computer Science
California State University Chico
Chico, CA 95929-0410

Completed applications must be received by

Saturday, May 15th, 2004

Chico State Intelligent Systems Lab
Summer Robotics Camp 2004

Recommendation Form

PART A: APPLICATION IDENTIFICATION	-To be completed by applicant

Applicant: ____________________ Phone: ____________

PART B: RECOMMENDATION COMMENTS	-To be completed by teacher or counselor

The person whose name appears above has applied for admission to the Summer Robotics Camp at CSU, Chico. The selection committee would appreciate your completion of the questions below in a specific and candid manner. Please make no assumptions about selection criteria. If your relationship with the applicant does not allow you to make an evaluation of any particular item, please indicate "n/a."

Name: ____________________ Position: ____________________

School: ____________________ Phone: ____________________

How long have you known the applicant? ___________ Years __________Months

	Strongly Agree	Agree	Agree Somewhat	Disagree
Has positive self-image				
Demonstrates leadership capabilities				
Has intellectual curiosity				
Enjoys math and/or science				
Survives frustrating experiences; is tolerant of minor disappointments				
Is mature enough to participate in a Summer Camp				
Is well-behaved and respectful				

On the reverse, please provide any relevant comments or insight that would be useful to the selection committee.

____ ________________________

Signature Date

Upon completion, please place recommendation form in the enclosed envelope, seal, and place school stamp over seal and return to applicant. This form must be mailed with the application or the application will be considered incomplete and will not be processed. *THANK YOU FOR YOUR ASSISTANCE!*

Appendix F

CALIFORNIA STATE UNIVERSITY, CHICO

College of Engineering, Computer Science, and Technology

INSTITUTE FOR RESEARCH ON INTELLIGENT SYSTEMS

Intelligent Systems Laboratory - Summer Robotics Camp

O'Connell Technology Center, OCNL 431

DIRECTIONS

The Intelligent Systems Lab is located in the O'Connell Technology Center (OCNL). OCNL is located on Warner Street. Here are some brief directions of where OCNL is located.

- From I-5 going north or south
  Turn east on Hwy. 32 in Orland, stay on Hwy. 32 into Chico to West 2nd St. Turn left on West 2nd St. and continue to Chestnut St. and the university.
- From Hwy. 99 going north to Chico
  Stay on Hwy. 99 north to Chico. Exit Hwy. 99 at Hwy. 32 turning left onto East 8th St. to and through downtown Chico. Continue west to Chestnut St. Turn right on Chestnut St. and continue north for six blocks to West 2nd Street and the university.
- From Hwy. 99 going south to Chico
  Take the East 1st Ave. exit and turn right. Go west to the Esplanade and turn left onto the Esplanade into downtown Chico and onto Broadway. Turn right on 2nd St. and go three blocks to Chestnut St. and the university.

Visitors may purchase short term parking permits at the south-west corner of 2nd Street and Chestnut Avenue. For other parking information please contact University Police located at the corner of 2nd and Normal Streets, Chico, CA 95929-0133 (530)898-5372.

ISL | WELCOME | PEOPLE | FACILITIES | PROJECTS | SUMMER ROBOTICS CAMP | CONTACTS | LINKS | SITEMAP

Questions or problems regarding this web site should be directed to the Webmaster.

Chico State Intelligent Systems Lab Summer Robotics Camp 2004

Lunch Donations Program

The Chico State Intelligent Systems Lab (ISL) has developed a week long, interactive Summer Robotics Camp to provide girls going into the 8th grade with the unique opportunity to learn more about science and math. This camp is designed to introduce these young women to the fields of Computer Science, Engineering, and Mechatronics through their learning, experimentation, building and use of robots. In order to keep the Summer Robotics Camp free for participants, the ISL is looking for student organizations willing to provide and serve lunches during the week of the camp. Lunches will need to feed 30 participants, plus 10 staff members. Each participating organization will be assigned one day during the week of camp for which they will provide and serve lunch.

DATES:

Lunches need to be provided for Monday, June 14th, 2004 through Thursday, June 17th, 2004. Lunch will be served from 12:00 – 12:30 pm.

LOCATION:

California State University, Chico campus. Volunteer organizations will serve lunch in O'Connell 124.

REQUIREMENTS:

- Volunteers must be Chico State student organizations.
- Student organizations must provide four to six volunteers to set up and serve lunch.
- Volunteers are responsible for bringing all food and materials for lunches including drinks and plates, cups, napkins, silverware, etc.
- Student organizations are responsible for preparing the lunches themselves (the ISL does not have facilities available for the preparation of food and so encourages organizations to bring pre-prepared lunches or food requiring little to no preparation).
- Other responsibilities include bringing food to OCNL 124, setting up and serving lunch, and cleaning up after lunch is over.
- Menu selections must be approved by the ISL Summer Robotics Camp.

DEADLINE:

Completed registration forms must be received by April 1st, 2004.

CONTACT INFO:

The Chico State Intelligent Systems Lab welcomes any questions or concerns about our Summer Robotics Camp. Please contact us at csuc_isl@yahoo.com or visit the camp's website at www.ecst.csuchico.edu/isl/SummerCamp.html

The Chico State ISL in funded in part by a grant from the National Science Foundation.
Grant # 0321385
Acquisition of robotics equipment for an Intelligent Systems Laboratory

Chico State Intelligent Systems Lab
Summer Robotics Camp 2004

Student Organization Registration Form

Please print legibly in ink or type.

PART 1: GENERAL INFORMATION -To be completed by student organization

Organization:

Organization Contact:

Phone: - -

Email:

Volunteers Participating:

1. ______________________________ 4. ______________________________
2. ______________________________ 5. ______________________________
3. ______________________________ 6. ______________________________

Please rank, in order of preference, the day you would like to provide lunch for the Summer Robotics Camp:

______ Monday ______ Tuesday ______ Wednesday ______ Thursday

Description of the lunch the organization plans to provide: ______________________________

____

PART 2: STUDENT ORGANIZATION AGREEMENT -To be completed by student organization

By my signature below, I, as representative of ______________________, agree that my organization will provide, serve, and clean up lunch for the participants and staff of the ISL Summer Robotics Camp on ______________________. If, for any reason, my organization is unable to provide this service, we will give the ISL Camp Coordinator no less than 7 days notice.

______________________________ ______________________________

Student Organization Representative ISL Camp Coordinator

Chico State Intelligent Systems Lab
Summer Robotics Camp 2004

Mentor Information

The Chico State Intelligent Systems Lab (ISL) has developed a week long, interactive Summer Robotics Camp to provide girls going into the 8th grade with the unique opportunity to learn more about science and math. This camp is designed to introduce these young women to the fields of Computer Science, Engineering, and Mechatronics through their learning, experimentation, building and use of robots. The ISL is looking for volunteer mentors to help with the camp, assisting the girls in their activities, supervising daily events, and overall acting as role models for participants.

QUALIFICATIONS:

- Applicants must be Chico State Students.
- Applicants must be able and willing to participate in camp in some part from June 14th through June 18th, 9:00am to 3:00pm daily.
- Applicants must be responsible and dependable, able to take on challenges, and model good behavior and enthusiasm for Summer Robotics Camp participants.

DATES:

Monday, June 14th, 2004 through Friday, June 18th, 2004 from 9:00am to 3:00pm daily.

LOCATION:

California State University, Chico campus. Mentors and staff will meet in front of the O'Connell Technology Center, located on the corner of Warner and West 1st Street.

DEADLINE:

Completed applications must be received by April 1st, 2004. Mentors should receive notification of acceptance by April 15th, 2004.

CONTACT INFO:

The Chico State Intelligent Systems Lab welcomes any questions or concerns about our Summer Robotics Camp. Please contact us at csuc_isl@yahoo.com or visit the camp's website at www.ecst.csuchico.edu/isl/SummerCamp.html

The Chico State ISL in funded in part by a grant from the National Science Foundation.
Grant # 0321385
Acquisition of robotics equipment for an Intelligent Systems Laboratory

Appendix F

Chico State Intelligent Systems Lab
Summer Robotics Camp 2004

Mentor Application Form

Please print legibly in ink or type.

PART 1: GENERAL INFORMATION	-To be completed by applicant

Name: ______________________ ______________________ ____
Last First M.I.

Mailing Address: ____
Number and Street Apartment #

City: ______________________ Zip Code: __________

Phone: ___ ___ - ____ Gender: ☐ F ☐ M

Email: ____

What school do you currently attend? ____________________________Grade Level?__________________

What is your major/intended major?____

What is your current cumulative grade point average (GPA)? ☐ ☐☐

What are the highest levels of math and science that you have completed with a passing grade?__________

____

____

What days and times are you available to participate as a camp mentor?

________ Monday ________ Tuesday ________ Wednesday ________ Thursday ________Friday

Have you ever been convicted of a felony? ☐ Yes ☐ No

If yes, please explain:____

____

PART 2: MENTOR AGREEMENT -To be completed by applicant

I understand and agree that:

1. I agree to participate in the ISL Summer Robotics Camp at CSU, Chico from Monday, June 14th, 2004 through Friday, June 18th, 2004 from 9:00am to 3:00pm daily.
2. I will arrive on time and actively participate in all scheduled activities such as classes, workshops, laboratories, and recreational activities.
3. If I cannot attend the Summer Robotics Camp on a particular day, I will contact the ISL staff members to inform them ahead of time of my absence.
4. I will work with the staff of the Summer Robotic Camp in order to best meet the participants' needs.
5. I will abide by the rules and regulations of the Summer Robotics Camp and California State University, Chico. I understand that my failure to abide by these rules or any behavior problems will result in my immediate dismissal from the program.
6. I will be responsible for any ISL items being lost, stolen or damaged due to my negligence.
7. The Summer Robotics Camp and California State University, Chico are not responsible for any items of mine being lost, stolen or damaged.
8. The ISL may use pictures of me participating in Summer Robotics Camp activities for ISL website content and CSUC promotional purposes.
9. The ISL Summer Robotics Camp, California State University, Chico, staff and owners of properties used for the program activities shall not be held responsible for any injuries I endure during the period of participation as a mentor in the program.
10. All of the information I have provided in this application is honest and correct to the best of my knowledge.

______________________________________ ____________________

Applicant's Signature Date

PART 3: AUTHORIZATION TO TREAT -To be completed by applicant

I, the undersigned, do herby authorize and consent to any X-ray examination, anesthetic, medical or surgical diagnosis or treatment, and emergency hospital care, which is deemed advisable by and if rendered under the general or special supervision of any member of the medical staff or emergency room staff licensed under the provisions of the Medicine Practice Act and on the staff of any acute general hospital holding a current license to operate a hospital from the State of California Department of Health. It is understood that effort shall be made to contact the emergency contact prior to rendering treatment to the undersigned, but that any of the above treatment will not be withheld if the emergency contact cannot be reached. This authorization is given pursuant to provisions of Section 25.8 of the Civil Code of California.

I further agree not to hold the ISL Summer Robotics Camp liable for the medical aid rendered and will reimburse the ISL Summer Robotics Camp for any medical or other expenses incurred.

______________________________________ ____________________

Applicant's Signature Date

PART 4: MENTOR SURVEY	-To be completed by applicant

The following information will be important for program officials to know in the event that you are selected. The information that you provide will not affect the evaluation of your application. Please provide honest responses.

What is your preferred adult, unisex, T-shirt size? ☐ S ☐ M ☐ L ☐ XL

How would you rate your level of computer skills? ☐ None ☐ Novice ☐ Intermediate ☐ Advanced

How would you rate your level of robotic knowledge? ☐ None ☐ Novice ☐ Intermediate ☐ Advanced

Do you have any physical or medical condition/needs of which the program staff should be aware or that can restrict your participation in certain type of activities? ☐ Yes ☐ No

If yes, please describe: ____

____

Do you have any special food restrictions or preferences (i.e. allergies, vegetarianism, religious prohibitions, etc.)? ☐ Yes ☐ No

If yes, please describe: ____

____

In case of emergency, contact: ____

Relationship: ________________________________ Phone: ( ) __________________________

COMPLETED APPLICATION	-Please read carefully

All parts of this application must be completed and mailed together or the application will be considered incomplete. Incomplete or late applications will not be processed. All information provided will remain confidential.

Mail completed application to:

ISL Summer Robotics Camp
Department of Computer Science
California State University Chico
Chico, CA 95929-0410

Completed applications must be received by
Thursday, April 1, 2004