Welcome to ME 170C

Welcome to our LEGO-based autonomous robotics course. This course is a blend of Mechanical, Electrical, and Computer engineering subjects. Together, this application of engineering is known as “Mechatronics”. To introduce a variety of concepts within each field, groups will create small robots, and program them to do complex tasks in the context of a fun project competition. The purpose of this letter is to provide an overview of the course and serve as a reference for many common questions. There are many important issues addressed in this document so read it carefully and thoroughly.

Course / Instructor Information

Course Location: Room 2163, Eng II
Instructor: Casey Hare
Lab Engineer: Verne Parmenter
Email: casey@engineering.ucsb.edu, verne@engineering.ucsb.edu
Office: 2163, Eng. II
Lecture: TR 5:30 – 6:00
TAs: Michael Nip, mdnip@engineering.ucsb.edu; Chris Hammetter, cih@engineering.ucsb.edu

Philosophy

The ME170C/ECE181C course has always been built around a core of mutual trust and respect. It is expected that you treat ALL the equipment in the lab with a GREAT degree of respect, and to ask questions when needed for proper use. ALL the LEGO parts and computer parts are to be returned in good working order. You will be charged for missing pieces. Over $15,000 has been spent to obtain these parts. In return, the TA’s and professor will make every effort to make this class educational, fun, convenient, and accessible. You will be allowed to take LEGO kits and Handy Board systems home to work on them, after properly signing out for them and promising to return them.

You will have to spend much more time on the project development than what is available during lab sessions. Good time management is CRUCIAL!

1.0 Course Overview

The purpose of this course is to gain hands-on experience in building a robot. Along the road you will gain valuable skills in debugging and trouble-shooting your design. You will also be introduced to a real-world design process for these types of devices. Within this context you will be expected to present and defend your design choices, and explain why the drawbacks are tolerable. With this being said, the main goal is to have an enjoyable experience while learning about and how to build robots.

Quizzes may also be given from time to time. Quizzes will be announced in class prior to the day they are given. The material tested on the quizzes will be based on lectures and/or material from the handouts, so read them (including this one) carefully!

1.1 Competition

The competition takes place on the robo rat table (the large white table in the rear corner of the lab). Groups will compete one-on-one, and the group with the most points at the end of the round wins. Each competition is two minutes long and the groups are not allowed to touch their rat during the competition. This means that the rat has to be pre-programmed on what to do for the entire two minutes, taking into account obstacles, collision and gaining points. The way that the rats earn points is by collecting “cheese.” The cheese is a two in. square foam blocks with a hole through each face. The rats can either collect the cheese and store it on the robot, or place their cheese on their home wall. Different cheese are worth different values, and where the rat stores the cheese makes a difference. If your opponent knocks off the
cheese you placed on your wall, too bad! Sabotage is part of the game. The point break down is as follows:

<table>
<thead>
<tr>
<th></th>
<th>Stored on Rat</th>
<th>Placed on Home wall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your cheese</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Center Cheese (Yellow)</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>Opponents Cheese</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

Your cheese is the cheese on your side of the table. Your opponent’s cheese is at the other end of the table. As you can see from the point breakdown, placing your opponent’s cheese on your wall is the most lucrative way to earn points. However, this task is rather challenging when you consider that for two minutes the cheese is being scattered everywhere by the robots. Strategy is one of the most important aspects to the competition. More details on the competition are described in section 3.0. In addition, the board will be augmented to add very high value hanging cheese.

1.2 Robo-Rat Progression

There are many tactics that the robo-rat has to accomplish in order to function on its own during the competition. It has to be able to move around the course (without being completely lost), it has to know when there is cheese in front of it so it can collect it, and it has to know when it has hit a wall so that it will turn around and continue collecting cheese. All of these tactics seem relatively easy, but you will be surprised how difficult it is to make sure everything performs the way it is meant to when the time comes. Therefore, during the first two weeks of class, as we are learning the basics, we will focus on a few key concepts and will ask you to build simple mechanisms to achieve the goal. Below are the capabilities we will be focusing on:

1.2.1 Modify Code

On the first day you will be asked to modify code that works properly. When downloaded to the Handy Board, the code will allow the Handy Board to perform certain functions. You will be asked to modify the code to change some functions that the Handy Board performs. This lesson will allow you to get familiar with the Handy Board and the IC computer language. If you are not familiar with C programming language you should get a C programming reference. The code used in the class usually involves just basic C functions, but finding ways to make your program modular and efficient can be extremely important when the robot needs to run all sub-systems at once. (Motors and various sensors.)

1.2.2 Motor/Gearing/Encoder Test

In this exercise you will learn about a motor – its no load speed, torque and how to manipulate the speed or torque for you application. You will be assessing the performance by learning how to use an encoder to estimate velocity.

1.2.3 Line Follower

The goal of this exercise is to get the robo-rat to start at one end of the table, follow the black line that runs down the center of the table over the ramp, reach the other side of the table and return in the shortest time possible. This requires that the robo-rat has a running drive train that allows the robo-rat to turn on relatively sharp corners, and that the optical sensors are mounted and working. Furthermore, a code must be written in IC and downloaded onto the Handy Board that will tell the robo-rat to perform certain tasks, such as turn, according to the information read in from the sensors. The handy Board must be mounted on the rat as well. This exercise introduces the basics of integrating the program to the robo-rat.

1.2.4 Wall Follower

This exercise is one that develops the most basic maneuvering system that most robo-rats will employ for their rat. In this exercise the robo-rat must start on one end of the table, follow the wall all the way around the table twice, turning when required, and return back to the beginning.
There are a few obstacles at the opposite end of the starting position that must be avoided. This challenge requires that the analog sensors are mounted and calibrated so that the robo-rat will know how far it is from the wall depending on the output of the sensors.

2.0 Laboratory Equipment and Software

2.1 Handy Board

Each station has the following Single Board Computer (SBC), and peripheral card:

1) Handy Board SBC using the Motorola 68HC11 microprocessor. There is an Expansion Board and LCD already installed on top of this board. They are made by Gleason Research (www.gleasonresearch.com) The power switch “off” position is toward the back of the SBC module. Do your wiring on the SBC’s “off,” and check for correct power connections before turning them “on.”

2) Serial Interface/Battery Charger Board plugged into a COM1 or COM2 port on the Pentium host. Note that a Power Supply module is plugged into this board. The battery charger switch on this board should be left in “Zap” mode, which fast-charges the battery pack beneath the Handy Board. In this way, your batteries will be refreshed while you are developing system solutions. Please see the Gleason manual for more details. Be sure to use the “normal” switch position for overnight charging.

3) Key Features of the Handy board and extender
   a. 32K static RAM of program space
   b. 16x2 character LCD useful for user input and debugging
   c. 9 digital inputs
   d. 9 digital outputs
   e. 4 DC motor outputs
   f. 6 servo motor inputs
   g. IR input and output
   h. 7 analog inputs (approximately 16 more on the expansion board)
   i. Start and Stop buttons
   j. Knob

2.2 Software

This class is using the Interactive C Development Environment known as “IC version 8.0.2” from Newton Research Labs (www.newtonlabs.com) The IC software resides in C:\IC on the host PC. In general, you will be writing code to control the LEGO-based robots. The code is downloaded to the memory on the Handy Board and it is run there autonomously. The 32K size RAM built into the 68HC11 processor will retain the program after the power is turned off.

One possible area of confusion is that various versions of IC are available, and tracing the evolution of the MIT 6.270 course, from the early days to the Handy Board days, can be quite challenging. Keeping track of your own code, and learning to use various libraries and functions available from the web, will be important toward creating a successful project. It is recommended that you always load the “libexpbd.icb” (binary) file as well, because this contains other useful functions like the one that allows DIGITAL OUT commands. The listing in the “.lis” file is what determines which libraries are loaded. You can create your own, trimmed down, efficient version of this file. For example, it would be wise to not load the “LEGO” libraries since we do not use any of those modules and drivers.

2.3 Host PC’s

Each host PC is connected to the campus network and Internet Explorer is installed. All the PC’s should have the power left on, but with screen savers running. All the PC’s have a “share directory”. You should be able to see this under the Network Neighborhood icon. It is encouraged to share data sheets and other helpful information with other lab groups, but don’t share your robot strategies and project code.
2.4 Security

The door to the lab MUST be locked after 5PM. The last person working in the lab has this responsibility. Someone will check with you from time to time to verify who will be taking on that responsibility. The windows to the lab must also be closed. Please be sure to turn your Fluke 37 Multimeters OFF before leaving, as they are run on batteries. Leave the computers on, but turn the monitors off. Keys to the lab will be available for a security deposit.

*Again, you will have to spend much more time on the project development than what is available during lab sessions. Good time management is CRUCIAL!* 

2.5 Hand Tools:

Each group is provided with the following hand tools:
1) Digital test probe. (The “red” wire connects to +5volts, “black” wire to ground.)
2) IC puller.
3) Small standard screwdriver.
4) Medium standard screwdriver.
5) Phillips blade screwdriver.
6) Wire cutter.
7) Wire stripper.
8) Tracing pencil.
9) Small pliers.
10) Small lock, and two keys.

Please be sure you have these items and then sign the confirmation sheet the TA presents to you which states you promise to return these items.

2.6 Wiring:

At some time during the course, you will most likely be required to build some circuitry on a proto-board. It is important that the class adheres to a few basic wiring standards to both increase the chances of your success, and help the TA’s help you! Furthermore, you MUST wire from a schematic diagram. Keep track of circuit revisions by writing notes along side the schematics in a notebook, and note the date and time. When you wire a circuit, use the colored tracing pencil to trace over each wire on the schematic as it is installed. Do this on a PHOTOCOPY of the schematic…NOT in a databook, etc.!! By wiring this way, you will be far less likely to make a wiring error. This is important. There are two wire colors that are crucial: “RED” for +5, and “BLACK” for GROUND (or “COMMON”). It is also a good idea to choose “yellow” for +V of some other voltage (ie., +12V, or +15V), while choosing “violet” for –V (ie., -12V, or –15V). The wire for proto-boards and general purpose short runs is AWG no. 22 solid. You will find it in yellow bins around the lab.

One thing which may seem tedious, but is actually quite important is to be sure and use the “header strips” to plug motors and sensors into the Handy Board. See page 73 of the “yellow” M.I.T. 6.270 Notes for a picture. The reason this is crucial is that the 22 gauge solid wire will break after time and leave a stub stuck in the Handy Board connector. It then makes this port unusable! Again, be sure to use the header strips!

The MIT 6.270 reference material has some pointers on assembly techniques, including soldering. Be sure to turn off the solder irons after use. Use safety googles, located at the front of the class, to protect your eyes from hot solder. Note that the reference material was written for a group expected to build an earlier version of this SBC. A lot of the material will not apply to this class. You will have to extract the useful sections by noting the context and consulting with TA’s, the professor, and fellow classmates.

There will be more reference material passed out as we proceed. The data sheets on various sensors and devices should be saved and studied before trying to use these parts. Connecting power incorrectly is especially damaging, and this may result in your robot not being able to use this part at all.
3.0 Robo-Rat Competition Rules

1) Scoring is as follows:

<table>
<thead>
<tr>
<th></th>
<th>On Your Robot</th>
<th>On Your Wall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your Cheese</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Center Cheese</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>Opponents Cheese</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

2) The robot must be no larger than one cubic foot.

3) Contestants must place their Robo-rats facing “left”, as referenced from the contestant’s end of the table. The referee will turn on the “start lamp” under each starting circle when both contestants are ready.

4) There will be no touching the robots, except as outlined in rule 5 below.

5) In the event that the referee determines that a collision is about to occur, each contestant must quickly activate a micro switch mounted on top of the robot. More details about the micro switch will be handed out in the lab. When the switch is activated, the robot must go into a “pause” state. In the event that neither robot had made it up the ramp (touching the ramp), both robots will be placed back on the starting circle with the lamp “on”. This should tell your software that a collision avoidance relocation was necessary, and the robot is at the start circle again. You can place the robot facing any direction at this time. In the event that your robot was touching the ramp (about to proceed up the ramp, or already on the ramp), and ahead of your competitor, you will not have to move your robot. In this case, your robot would of course not detect the start lamp. So, when the referee instructs the contestants to reactivate the micro switch, the robot on the ramp may simply continue on with its previous program.

6) The contest will go on for 2 minutes, and this time will not include any collision avoidance time out.

7) The contest is a double elimination, and this can take quite a long time. We need to have a battle every 3 minutes to be able to determine a winner in 60 minutes!