CPE 470-670 – Autonomous Mobile Robots

Instructor: Monica Nicolescu

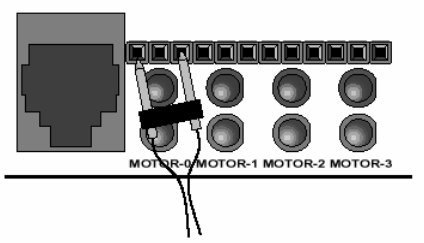
Lab 2 – Handout

1. Start using the HandyBoard

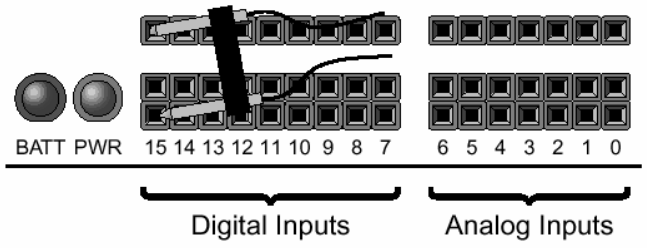
a) Make sure that the HandyBoard is **OFF** and connected to the serial port of your computer, using the Serial Interface board. The Serial Interface board connects to the computer using a standard modem cable; the HandyBoard connects to the Serial Interface using a standard 4-wire telephone cable. Plug in the charger for the Serial Interface board.

ATTENTION!!!
Do not leave the HandyBoard charging in ZAP mode for more than 3 hours. Use NORMAL mode for charging the HandyBoard.

b) **Connect the motors** to the HandyBoard, using the modified LEGO adapter cables. Connect one motor to port 1 and the other to port 3.
c) **Connect the sensors** to the HandyBoard, using the modified LEGO adapter cables. Handy Board has inputs for 9 digital (switch-type) sensors and 7 analog (continuously varying) sensors. Connect your touch sensors to digital inputs 10 and 11 (make sure you know which one is which).



HandyBoard motor board



HandyBoard sensor board

d) Start Interactive C on your computer.
- In the first window select Handyboard (with expansion).
- In the Port Selection Window select Connect Later
- From the Tools Menu select "Download Firmware" and follow the instructions
- Your HandyBoard should be ready to use

2. Test your HandyBoard

In the command line of your IC application, test the following instructions:

a) **2 + 2:** the HandyBoard will compute the answer and will print the answer in the IC application.
{beep(); sleep(2.0); beep();}
b) Load the first program: in your IC window, select "Open" and load the file called hello.ic. Download it to the Handyboard using the IC icons and then select Run Main. Your Handyboard will display your message to the world!!

3. Overview of relevant functions

a) Beeper (Appendix E, pp. 418 & 419)
- **beep();** - Causes the HandyBoard to beep
- **tone(frequency, length)** – produces a tone at pitch "frequency" Hertz for length seconds.

b) DC Motors (Appendix E, pp. 415)
- **fd(0);** - Motor 0 output port turns on, green LED on, motor spins
- **bk(0);** - motor spins in opposite direction
- **off(0);** - motor turns off
- **motor(0,50);** - turns motor port 0 on in the "fd" direction with a power level of 50% (port can be 0, 1, 2, 3) (power level ranges from -100 to +100; 0 is off, +100 is full on in the "fd" direction)

c) Digital Input (Appendix E, pp. 417)
- **digital(15);** Handy Board returns True/1/switch closed or False/0/switch open.
- **if (digital(15)) {beep();}** Tests the state of the sensor.

4. Write your first programs

- **Beeper program**:

Write a program to sound the beeper for 3 seconds with 500 Hz frequency, then be silent for 1 second, and then sound the beeper for 3 seconds with 100 HZ frequency, be silent for 1 second and finally sound the beeper for 2 seconds with 1000 Hz frequency.

- **Motor program**

First check that your fd() command actually make the robot drive forward. In the IC command line type {fd(1); fd(3);}. If the robot is not driving straight ahead, unplug the motor that is running backwards, turn it 180° and plug it back in.

Write a program to turn on the motor connected to port No. 1 with 50% power, work for 3 seconds, sound the beeper, reverse the motor with 50% power for 2 seconds, sound the beeper, turn off the motor for 2 seconds, then turn it on with full power for working 3 seconds, sound the beeper and turn it off.

- **Sensor program**

Write a program for your robot so that when one of its touch sensors is triggered, it will start rotating around itself until the same touch sensor is triggered for the second time. Your program should print a message on the handyboard LCD each time the touch sensor is triggered.

- **Obstacle avoidance**

Implement the "obstacle avoidance" program described in Section 2.3.2, and extend the program so that both touch sensors are active. For this program, the robot should start running at the press of the "Start" button and should stop at the press of the "Stop" button. (For information on how to use these buttons see page 417). **Maintain this capability for all your future programs!**

- **Multi-tasking program**

Rewrite your obstacle avoidance program as a multi-threaded process. You should implement the following processes: *check_sensors* to detect if a touch sensor has been pressed, *move_forward* to keep the robot moving forward if no sensors have been pressed and avoid to have the robot move backward and turn when a sensor has been pressed. For information about how to use Multitasking capabilities of IC and the Processes, refer to your book (pages 209, 419-421).

Submit all your programs through e-mail before midnight on Tuesday Sep 14 (only one copy per team). These will constitute a part of your evaluation for this lab. Use the following as the subject line: "Lab 2: team #".