CS148
Building Intelligent Robots

Week 0 : LEGO Design and Intro to brickOS *Out: 8 Spt 2005*

Preliminary Tasks *before 15 Spt 2005, 9am*

You will be expected to complete this prior to the first lab. The purpose is for you to build a robot and install brickOS, as well as to get you acquainted with programming in brickOS. You will only be introduced to a limited set of commands, and will be expected to read through the documentation and understand the rest on your own.

Task:

- Install brickOS on your own computer (only one per group is necessary). You will need to follow the HOWTO / QuickStart guide, available on the web at:
  http://www.cs.brown.edu/courses/cs148/brickOS/quickstart.html
- You will need to download the demo program from your own computer
  (`C:\brickOS\demo\helloworld.lx`) to your RCX and run it. See page 2 for instruction on using a makefile.
- Build a robust tankbot with two treads, one motor controlling each tread, and a touch sensor. Look at the sample tankbot on the webpage for inspiration:
  http://www.cs.brown.edu/courses/cs148/tankbot.shtml

Some things to consider when designing robots for cs148:

- Black high-friction connector pegs vs. grey connector pegs
- Gear ratios
- Reinforcing structures and stabilizing the RCX brick

1 Setup

Introduction to brickOS

Pick a computer in the MS lab.

First you need to map the cs148 course directory to the 'L' drive. To do this, open up 'My Computer' and click on Tools -> Map Network Drive. Select 'L' fro m the Drive menu. In the folder type:

\\maytag\course0\cs148

and click finished.

We have configured brickOS in the MS Lab to be a little different from your dorm room installation. There should be an icon called "Cygwin" on the desktop after you log in. This is the bash shell that you want to use. Start it up.

You'll notice that when you open up Cygwin it looks and works like a shell on Linux. Right now you are in your home directory, the same home directory you would be on a Linux machine. Create a directory to work in. Then cd into that directory and copy the brickOS lab files from the course directory with

```
cp -r l:/asgn/2005/lab0/* .
```

Some general reminders about using the Makefile.

- To download the brickOS kernel to the RCX: `make brick`
- To compile a down a program: `make install`
- The Makefile works exactly the same if its on a Windows or Linux machine.

Lab 0: LEGO Design and Intro to brickOS *15 Spt 2005*

The rest of this assignment is not mandatory. It is intended to familiarize you with the brickOS commands. If you choose not to complete it, however, you will still be responsible for the material.

1.1

You will need to know how to control the motors so that the robot will go (approximately) where you want it to go.

Review the documentation on the course web site about using motors, timing, and playing sounds.

Task: Make the robot go forward for one second, then turn approximately 180 degrees and stop. After the robot has stopped moving, make it play a short tune of your own choice. If you want, you can use the skeleton file provided in *.../lab0/step1/step1.c*that you already copied from the course directory.

You may find the following commands helpful in this section:

- `motor_X_speed`
- `motor_X_dir`
- `msleep`, `sleep`
- `dsound_play`

1.2

Multitasking is a crucial ability that your robots will need to be able to do.

Review the web site documentation on task management.

The concept of threading is fairly simple. The idea is that you want to do multiple things at the same time, so to do this, you spawn a new thread. When a new thread is created, it begins execution at a specified point, and runs concurrently with other.

We have provided a skeleton file for you in *.../lab0/step2/step2.c*. Feel free to use this or write your own.

Task: Write a multi-threaded program. Your program should do the following:

- The main thread should create two child threads and then enter an endless loop.
- The first child thread should make your robot turn counterclockwise in place for one second, then turn clockwise in place for one second, and then repeat endlessly.

- The second child thread should play a tune of at least 4 notes in a loop... Keep a global variable that keeps track of how many times the tune has been played.

You will probably want to use the `execi` command.

1.3

Much of what your robot will do is to wait for things to happen. brickOS provides a `wait_event()` command that allows you to do just that - wait for something to happen. Review the web site documentation and make sure you understand the sample files. The sample files are also located in the course directory in a directory called *pub/samples*. This step will get you acquianted with events and handling them.

There are two steps involved in creating an event. First, you need the wakeup function. This is just a function that returns zero when your event hasn't happened yet, and a non-zero value when your event has happened. For example:

```
wakeup_t dkey_pressed(wakeup_t data);
```

This function will return non-zero only when a key has been pressed. To wait for this event, we would use the command `wait_event(dkey_pressed, 0);`

One thing to note about using `wait_event` is that it always takes a second parameter of type `wakeup_t`, but it almost never means anything. In almost all cases, you can safely use 0 as the value you pass to the `wait_event` function. The data value you pass it is passed to the wakeup function, but few of them actually use it.

Task: Modify your program from part two so that it does the following.

- The main thread should once again create two child threads. The two threads should do the same thing as before (turn back and forth endlessly, play a tune endlessly).
- Instead of having the main thread enter an endless loop, however, it should wait until the tune has been played 5 times. After the tune has been played five times, the main thread should kill the two child threads and then exit. You must use `wait_event` for this part. (hint: write a `wakeup_t` function that returns true when the tune has been played enough times)