Chris Werner Final Project:

Controlling a Robot with Linux via Bluetooth...

Meet the ROBOT!

The robot that is going to be used for this project is a Lego MINDSTORMS NXT Robot. LEGO MINDSTORMS NXT is a robotics toolset that provides endless opportunities for armchair inventors, robotics fanatics and LEGO builders to build and program robots that do what they want. The heart of the system is the NXT brick, an autonomous 32-bit LEGO microprocessor that can be programmed using a PC, or a Mac. But this Guide will show how Linux can be used to control it remotely via Bluetooth.

Read more > Lego MINDSTORMS

Getting Started

Items used in this tutorial (what you will need to follow along)

- Lego MINDSTORMS Robot
- USB Bluetooth Adaptor
- A Linux Box
- An Internet connection (for initial setup)
- and this all mighty tutorial

Linux Box Setup

The First thing that I did was YUM bluetooth to see what was on my system...

```
[user@host ~]$ besu
[user@host ~]# yum install bluetooth
```

The Next thing that you need to install is BlueZ. BlueZ is the official Linux Bluetooth Protocol stack. We are going to install two things:

- BlueZ Library
- BlueZ Utilities

Click on each item in the above list to download the package.

Now navigate to where you downloaded these packages and install the 'bluez-lib' one.
Once that has completed, go ahead and install the 'bluez-utils'.

Bluetooth Test

Now we can test to see if we can communicate to our Robot using Bluetooth
First we need to turn on the NXT Robot and turn on bluetooth.
Then open a Terminal and run these commands...

```
[user@host ~]$ hcitool scan
```

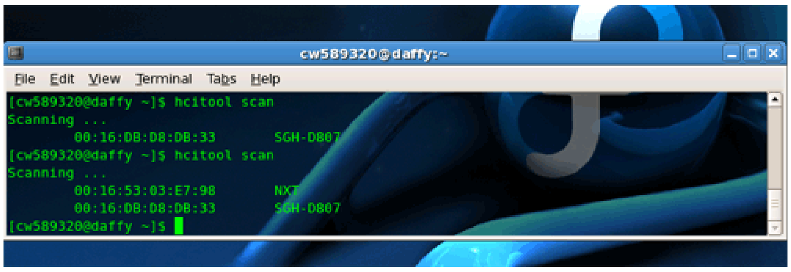
If everything worked right you should see this...

```
[user@host ~]$ hcitool scan
Scanning...
```

```
                xx:xx:xx:xx:xx:xx       NXT
```

You should write these numbers down for future reference.

Here is an screenshot before and after the NXT is turned on:



```
[cw589320@daffy ~]$ hcitool scan
Scanning ...
        00:16:DB:D8:DB:33       SGH-D807
[cw589320@daffy ~]$ hcitool scan
Scanning ...
        00:16:53:03:E7:98       NXT
        00:16:DB:D8:DB:33       SGH-D807
[cw589320@daffy ~]$
```

Perl Bluetooth Interface

Now we want to install a Perl - Bluetooth Interface. The reason for this is because
we are going to be using Perl scripts to communicate with the NXT Robot.
To start this step, download the Net::Bluetooth Package...

- NET :: Bluetooth

Now you need to unzip and untar this file. You can do it in your GUI, or you can use the
terminal. I will show below how to do it in the terminal. First navigate to the directory where
it downloaded to. Then run these commands.

```
[user@host ~]$ gzip -dc Net-Bluetooth-0.39.tar.gz | tar -xof -
```

After trying to install this about 383 times, I finally figured out why it wasn't working.
As a wild guess I changed the permissions of the whole directory, and it worked!

Now to install Net-Bluetooth run these commands:

```
[user@host ~]# chmod 777 -R Net-Bluetooth-0.39
[user@host ~]# cd Net-Bluetooth-0.39
[user@host ...]# perl Makefile.PL
[user@host ...]# make
[user@host ...]# make test
[user@host ...]# make install
```

LEGO :: NXT API

The final thing that needs to be installed is the NXT API. This module provides low-level
control of a LEGO NXT brick over bluetooth using the Direct Commands API. This API will not
only enable you to run programs on the NXT, but it will connect to the NXT and issue real-time
commands that turn on/off motors, retrieve sensor values, play sound, and more.
download the LEGO :: NXT API...

- LEGO :: NXT API

Again, you need to unzip and untar this file. You can do it in your GUI, or you can use the
terminal. First navigate to the directory where it downloaded to. Then run these commands...

```
[user@host ~]$ gzip -dc LEGO-NXT-2.00-1.tar.gz | tar -xof -
```

Now to install the NXT API run these commands:

```
[user@host ~]# chmod 777 -R LEGO-NXT-2.00-1
[user@host ~]# cd LEGO-NXT-2.00-1
```

```
[user@host ...]# perl Makefile.PL
[user@host ...]# make
[user@host ...]# make test
[user@host ...]# make install
```

Test Run

Its time to test the connection between the Linux box and the NXT through the Perl API. To do this we can run a test script, which should play a sound that is stored on the robot. You will need the numbers that hcitool scan returned, as well as this test script:

- test.pl

I suggest making a new folder to store all the scripts that we will make.

once you have this downloaded, turn on the robot, go to that directory and run this command:

```
[user@host NXT]$ perl test.pl <xx:xx:xx:xx:xx:xx>
```

You should have heard the robot make a sound if you followed these steps correctly.

Custom Commands

Now that all of the setup work is done, we want to set up a nice little environment to control the robot using simple command line commands.
To do this we will write our own Perl scripts that will control the robot. First create a file called say (this is a Perl file, but I am choosing to leave off the .pl ending. This will make things convenient later). In this file, copy or type this code into it:

```
use LEGO::NXT;
use LEGO::NXT::BlueComm;
use LEGO::NXT::Constants qw(:DEFAULT);
use Data::Dumper;
use Net::Bluetooth;
use strict;

my $filename = $ARGV[0].".rso";
my $addr = "xx:xx:xx:xx:xx:xx";
my $port = 1;

die "No Bluetooth Address Specified!\n" if !$addr;

$| = 1;

my $nxt = LEGO::NXT->new( new LEGO::NXT::BlueComm($addr,$port) );

my $returnVal;


print "NXT at Bluetooth Address: $addr\n";
print "Filename: $filename\n";

print "searching for: $filename...\n";
$returnVal = $nxt->sys_find_first($filename);
print Dumper($returnVal);


$nxt->play_sound_file($NXT_RET, 0, $filename);


exit;
```

You need to substitute the address of your NXT robot in for the xx:xx:xx:xx:xx:xx.

He Talks!!

Lets make him talk to us. Go to a terminal and go to the directory that you made to store the NXT scripts. enter this command:

```
[user@host NXT]$ perl say hello
```

You should have heard the robot say 'hello' to you! You can also give him this command:

```
[user@host NXT]$ perl say goodbye
```

What this script does is take the argument after 'say', and plays the audio file on the robot with that filename. There are a few audio files pre-installed on the robot. You could also upload more to him to give him a bigger vocabulary.
Out of the box, the possible arguments are: 'hello', 'goodbye', 'woops', 'play', and 'music'. There are a few other audio files on the robot, but they are not words.

perl => nxt

I dont like having to type in 'perl say hello'. 'perl' isn't very robot-like. So what we can do is maka an alias for perl. You will need to edit the file which has your aliases. For me this happens to be /etc/profile.d/99local.sh . Open this file and make an alias... you can make it whatever you wish... I'm going with 'nxt'.

```
alias nxt='perl'
```

You should now hear the robot say 'hello' when you type this command:

```
[user@host NXT]$ nxt say hello
```

He Walks Too!

From this point forward, these scripts which I have written will assume that you have built the humanoid robot named Alpha Rex. Instructions on constructing him should have come with your NXT kit. Using the Lego provided software create a program which simply makes him walk. (this is also in the humanoid directions).
Once you have him built, and the walk program on the robot, we can execute it remotely.
You can download my walk program here...

- walk.rbt

We could write the walk program in the perl method, but for simplicity, and to save time, we'll go with what we have for now.
Now we need to write a perl script to run this program on the nxt. Here is what I've come up with, its named 'walk' (again, no .pl):

```
use LEGO::NXT;
use LEGO::NXT::BlueComm;
use LEGO::NXT::Constants qw(:DEFAULT);
use Data::Dumper;
use Net::Bluetooth;
use strict;

my $filename = "walk.rxe";
my $addr = "xx:xx:xx:xx:xx:xx;
my $port = 1;

die "No Bluetooth Address Specified!\n" if !$addr;

$| = 1;

my $nxt = LEGO::NXT->new( new LEGO::NXT::BlueComm($addr,$port) );

my $returnVal;

print "NXT at Bluetooth Address: $addr\n";
print "Filename: $filename\n";
print "searching for: $filename...\n";
$returnVal = $nxt->sys_find_first($filename);
print Dumper($returnVal);

$nxt->start_program($NXT_NORET,$filename);

exit;
```

Now we can make him walk by typing in this command:

```
[user@host NXT]$ nxt walk
```

STOP!

Now that we can execute programs that reside on the robot, the next thing that would be needed is the ability to stop him. For example, when we give him the walk command, we don't want him to walk forever. But we also need this script to be universal, I don't want this to only stop him if he is walking. It should stop him no matter what program he is running. First, using the Lego software, make a blank program called stop and load it onto the robot. Then copy this script and call it 'stop'

```
use LEGO::NXT;
use LEGO::NXT::BlueComm;
use LEGO::NXT::Constants qw(:DEFAULT);
use Data::Dumper;
use Net::Bluetooth;
use strict;

my $filename = "stop.rxe";
my $addr = "00:16:53:03:E7:98";
my $port = 1;

die "No Bluetooth Address Specified!\n" if !$addr;

$| = 1;

my $nxt = LEGO::NXT->new( new LEGO::NXT::BlueComm($addr,$port) );

my $returnVal;


print "NXT at Bluetooth Address: $addr\n";
print "Filename: $filename\n";

print "searching for: $filename...\n";
$returnVal = $nxt->sys_find_first($filename);
print Dumper($returnVal);

$nxt->stop_program($NXT_NORET);
$nxt->start_program($NXT_NORET,$filename);


exit;
```

First make him walk...

```
[user@host NXT]$ nxt walk
```

Now, make him stop by typing in this command:

```
[user@host NXT]$ nxt stop
```

The End

That's about it. We have successfully talked to the robot (and had him talk back!) via Bluetooth from our Linux

Box.
You can download my nxt directory which contains the scripts used for the robot... as well as extras here

Videos

Here I have some videos of the robot in action... You will need flash installed to view them

NXT says Hello

NXT says Goodbye

NXT Walks, then Stops on command

(He doesn't move forward in the video because the table's surface is too slick...
I didn't want him walking away in the video)

NXT Dances!! ...the robot.

(He asks you to 'Play music' first)

NXT says hello, Does a little dance, then says goodbye