# LEGO Prototype of the LineScout Robot


**Team Members:**

**Daniel Been**

**Sartsawat Dendandome**

**Sean Ding**

**Project Sponsor:**

**Dr. Janos Toth**

**Applied Science 479**

**Engineering Physics**

**The University of British Columbia**


**Jan 10, 2011**

**Project Number: 1073**

# Preface

An assumption that the project group made was that the LEGO bricks are easy to obtain given a design in LEGO Digital Designer file. As it turned out, LEGO.com only supplies less than 20% of the bricks we needed for our design. We spent roughly another two weeks looking for vendors who sell those other parts.

After placing the orders to LEGO, they made some mistakes and held our shipments in the factory for three weeks. The shipments did not arrive for additional two weeks. Our project could not officially start until early-mid November 2010. This left us with about one month to work on the project before the final exams. One of the main parts, the driving wheels, did not arrive until late November. The status of the project at the completion of this report is not as completed as we would have imagined.

Thanks to Shawn Crockett, previous BC Hydro co-op, for his preliminary design on LEGO Digital Designer. His work had saved us a lot of time trying to design a prototype LEGO that has the same functionality of a LineScout.

Thanks for Bernhard for building the test environment for the LineScout to hang on and as well, recommendations for alternations of our design. Thanks to Jon for providing various feedback and comments whenever possible.

## Executive Summary

This report outlines the construction of a replica of the LineScout robot. The LineScout is a remotely operated live-line inspection and maintenance robot for high-power electrical transmission lines developed by the Institut de Recherche d'Hydro-Québec in conjunction with BC-Hydro. The project sponsor, Dr. Janos Toth of BC-Hydro, desires a scale model that emulates the basic functions of the LineScout Robot for demonstration purposes.

We construct such a replica, primarily using commercially available LEGO construction parts, based on the design done in LEGO Digital Designer by Shawn Crockett over the past summer as part of his co-op work at BC-Hydro. The prototype is remotely controlled by an operator, be able to move along a rope or cable, and cross obstacles in its path. The prototype consists of two frames that can translate laterally relative to each other. One frame includes the drive wheels that move the robot along the line, while the other has two grippers that can hold the robot in place when the wheels are not in contact with the line. During an obstacle crossing, the grippers will grip the line on either side of an obstacle, and the wheels will be rotated away from the line, and moved to the other side of the obstacle. The grippers then release to allow the robot to move on.

The result performance of the LEGO LineScout is not robust due to the fact that the LEGO pieces deform when under load. An improvement can be seen the most when the wheel frame can be tilted forward more.

# Table of Contents

## List of Figures

# 1. Introduction

## *1.1 Background and Motivation*

Industrialized nations rely heavily on universal access to reliable electrical power. This puts a huge onus on power utilities to operate continuously, while serving large, and growing, power demands. This means that electrical distribution networks must be monitored and maintained very carefully. Maintaining an electrical grid is an expensive and dangerous endeavour. It requires highly trained technicians to properly inspect and repair power lines. Working on the power lines can be very dangerous for personnel due to the very high voltages needed for efficient long-distance power transmission. Often, the transmission line must be shut down for the safety of maintenance crews, which results in interrupted service for customers. The problem is further compounded for Canadian power companies because of the vast distances that transmission lines must cover to provide service to all those who need it. The result of this low population density is that more miles of transmission lines are required relative to the number of customers served. British Columbia is particularly daunting, even compared to other provinces, because of the rugged and inaccessible terrain that some power lines must traverse. These problems have resulted in a growing interest in robotic tools for inspection and maintenance of transmission systems.

One of the most advanced and sophisticated live-line inspection robots is the LineScout developed at the research institute at Hydro-Québec. Recently, Hydro-Québec has been collaborating with BC-Hydro on the LineScout project, and much of the testing has taken place in British Columbia. The LineScout, seen in Figure 1 below, weighs 100kg, is 1.37m long, travels along a single conductor at up to 1 meter per second (3.6 km/h), can cross obstacles on the line up to 84cm in length, and can operate for up to 8 hours a day.

**Figure 1. LineScout on an inclined transmission line (left). LineScout crossing an obstacle (right).**

The LineScout's primary function is conductor inspection, and so is equipped with four visible-light cameras and an infra-red camera. It is also equipped with a highly maneuverable manipulator arm that can be fitted with a variety of sensors and tools. The cameras transmit in real-time to the operator, who can be up to 5km away from the robot.
Because of the LineScout's size and weight, and the demand for it to be in use, it is not practical to use it for live demonstrations. BC-Hydro wishes to have a portable model of the LineScout

which is capable of demonstrating the basic capabilities of this remarkable new technology. To this end we propose to build a model of the LineScout which can fulfill this role.

A preliminary design for a model has already been completed by Shawn Crockett, an Integrated Engineering student, during his co-op work term at BC-Hydro. The design was done in the LEGO Digital Designer software. This is a 3-D CAD software that facilitates the design of structures to be made with commercially available LEGO pieces, and generates step-by-step assembly instructions for custom designs. The decision to use LEGO was made for a number of reasons:

- Using LEGO pieces minimizes the need for machining custom parts.
- Because the model is primarily made of standard LEGO pieces, it can be easily replicated.
- LEGO produces a number of high-quality components for robotics applications; most importantly microcomputers and servomotors.
- If the model is featured at BC-Hydro's display at Science World (as hoped), the use of LEGO would increase its appeal to children.
- If the model will be displayed in public, it may be possible to persuade LEGO to sponsor the project by providing discounted components, thus decreasing the cost to BC-Hydro

## *1.2 Objective*

- Deliver to the project sponsor a working scaled-model of the LineScout Robot that:
    - Can move along a single suspended rope or cable
    - Can cross obstacles on the wire that block its path
    - Can be remotely operated by a single person
    - Automated obstacle crossing sequence

- Provide the project sponsor with full documentation regarding the construction, programming, capabilities, and operation of the completed model.

At the completion of this project, it will provide BC-Hydro with a valuable tool to help show the applicability of robotics to modern power transmission networks. It will aid immensely in explaining the capabilities of the LineScout robot. Instead of just using words and pictures, a demonstrator will be able to show a manager, a client, or the public, something that they can touch and see in real life. Even though the model will be quite different from the LineScout in many respects, one can use the model as a starting point to properly describe the LineScout, in conjunction with pictures and videos. If the user interface is robust enough, it will allow people to get a taste of what is required to operate the LineScout when inspecting lines or crossing obstacles.

### *1.3 Organization*

This report is organized into the following sections:

- Discussion – In this section, technical background, final design, obstacle crossing sequence, and results are presented.
- Conclusion – In this section, the important accomplishments are presented at the completion of the project.
- Project Deliverables – In this section, the materials to be handed over to project sponsor and the summary of budget are outlined.
- Recommendations – In this section, some features are outlined for improvements if future projects are to carried out.

# 2. Discussion

## *2.1 Technical Background*

The LEGO Minstorm NXT Brick is the LEGO only micro-controller based on 32-bit ARM7 processor. It each provides 4 input and 3 output ports. The communication between each brick is by wire via RS485 or wireless via Bluetooth. The computer controlling the robot communicates to the master brick on the wireless Bluetooth protocol.

There are several options in programming the NXT brick. The LEGO provides the NXT-G Programming software when the LEGO Minstorm NXT kit is purchased. The software is based on graphical flow control and suitable for basic programming and easy to use. However, to bring NXT to the next levels, LabVIEW Toolkit, alternative programming software, could be used to provide advanced featured textual/graphical-based programming environment. For the purpose of this project, we require the robot to be controlled wirelessly from a computer. Many LEGO hobbyists have come up with different programming language to control the NXT through Bluetooth. Because of the background of the project members, we chose to use the Anders' C++ Communication library to implement a program to remote control the NXTs[1]. In addition, it offers some example programs that work well and complete library for almost everything we can think of.

There are five major moving parts that we refer to often in this report and they are: the base frame, the horizontal slider, the gripplers, the wheel frame and wheel frame support. The base frame holds the two NXTs at its bottom, three servomotors that control the tilting of wheel frame, and wheel frame support, and extending/retracting of the horizontal slider. The horizontal slider slides on top of the base frame during obstacle crossing. It contains one servomotor that raises/lowers two gripplers that are attached to it. Each grippler has a servomotor to open or close the hook to the cable. The wheel frame has a Power Function Motor that drives the robot along the cable. It can be tilted back to cross obstacles when the gripplers are engaged.

## *2.2 Final Design*

In this section, we discuss about the major modifications to the original design due to technical reasons explained in each part.

### 2.2.1) Base Extension

The base that guides the horizontal slider is extended so that the gripplers never slide out of the base. This modification was implemented to prevent the gripplers from catching the edge of the base and hence stopping the horizontal sliders from coming back to the compact position.

### 2.2.2) Gripper Arm Extension and Line Reinforcement

LEGO beams are not rigid enough for a project of this scale. The evidence is when horizontal slider slides out and the weight sits on one side, the base bends and twists towards outside. The far grippler cannot reach the cable with original height. In addition, a line is used to pull the base back up when the horizontal slider is extended to its fullest.

### 2.2.3) Height Extension

The height of wheel frame is increased so the robot is able to cross larger obstacles. With original design, the grippler arms are about the same height as the wheel.

### 2.2.4) Driving Mechanism Modification

The original design for the driving wheels was using O-rings. During our tests we discovered that one of the O-rings cannot provide enough friction to turn the driving wheels. Therefore gear sets are used to replace that O-ring.

### 2.2.5) Tilting Mechanism Modification

The original design that connects the base frame to driving frame cannot provide rigid force to keep the wheel frame and base frame in the desire angle. Although the modification has been made to reduce the gear ratio dramatically, the mechanism still does not provide enough force to prevent the base from tilting so the locking/support mechanism is introduced into the system.

### 2.2.6) Base Gear Rail Modification

The original design of the gear rail has all the gears on the rail to be able to drive the horizontal slider out. Due to backlash of multiple gears, some torques is lost and may or may not be able to extend/retract the slider frame. Because of the fact that the horizontal slider has continuous racks and its one end never slides across the middle of the base, we made the end gears free rotating.

### 2.2.7) Gripplers Raising Mechanism Gear Ratio Change

The gripplers need to be able to lift the entire robot during the obstacle crossing sequence. The original gear design was one to one ratio. A gear ratio reduction was made on the slider unit.

### 2.2.8) Grippler Gear Modification and Hook Addition

The original worm gear design has slippage problem when the gripplers try to close on the cable as the axle that holds the worm gear tilts. It is replaced by a row of gears and the gripplers were added a little hook so that the motors do not need to exert any force once the gripplers are hooked onto the cable.

### 2.2.9) NXT mounts Modification

The original design did not take into account the rechargeable battery, which has an extra thickness than when using AA batteries. Locations of the mounts are still at the bottom of the base.

## 2.3 Experimental Equipment / Obstacle Crossing Sequence/ Algorithms

Bernhard has created a testing platform for us that consist of a frame, that is easy to uninstall and become portable, and a tightened rope that stays reasonably straight when the LEGO LineScout is on.


**Figure 2. The testing platform.**

**Obstacle Crossing Sequence**

To cross the obstacle, the robot would slide out the horizontal slider by turning the servo on the base frame which drives the gear rail which push out the rack on the horizontal slider until half of the horizontal slider passes through the obstacle. Both grippers will be raised by the servo on the horizontal slider until the gripplers can close on the cable. The servo on each grippler would drive the gears closing the gripper to clamp the gripper on the wire. The same motor on the horizontal slider lifts up the robot; then, another motor on the base frame swings the Rolling Unit off to the side preventing it from hitting the obstacle after the frame support unit is disengaged. The motor on the base frame would drive the rack moving the base over the obstacle. The rolling servo would roll the wheel frame back on the cable and then the frame support back in place. The gripplers are released and lowered; finally, the horizontal slider slides back to original position then the robot continues down the line.

The series of diagrams below demonstrates how the LEGO LineScout across an obstacle.

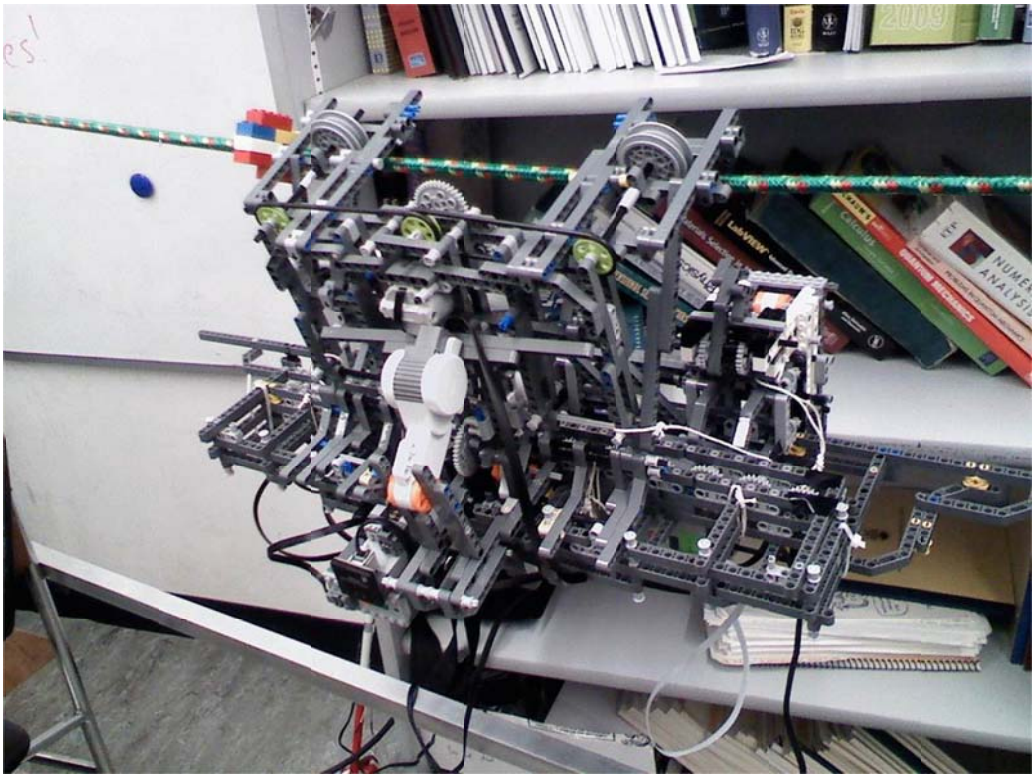

**Figure 3. The robot approaches the object from the right and stops just before the left wheel touch it.**

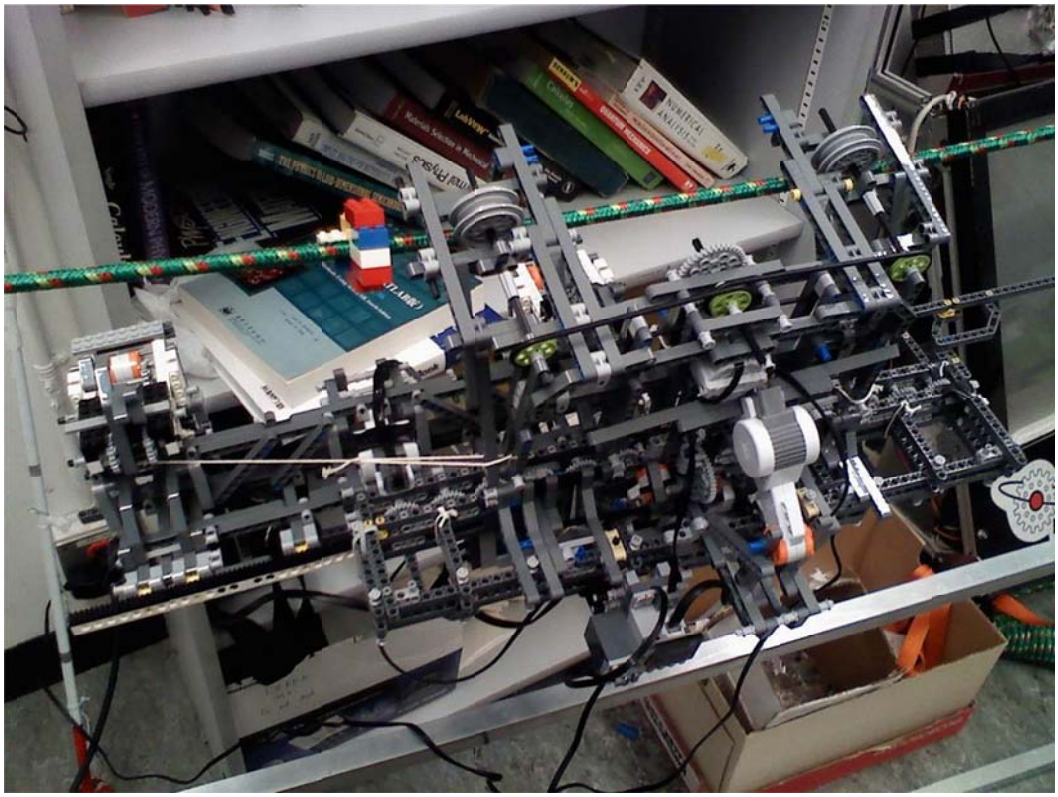

**Figure 4. The horizontal slider slides all the way across the obstacle.**

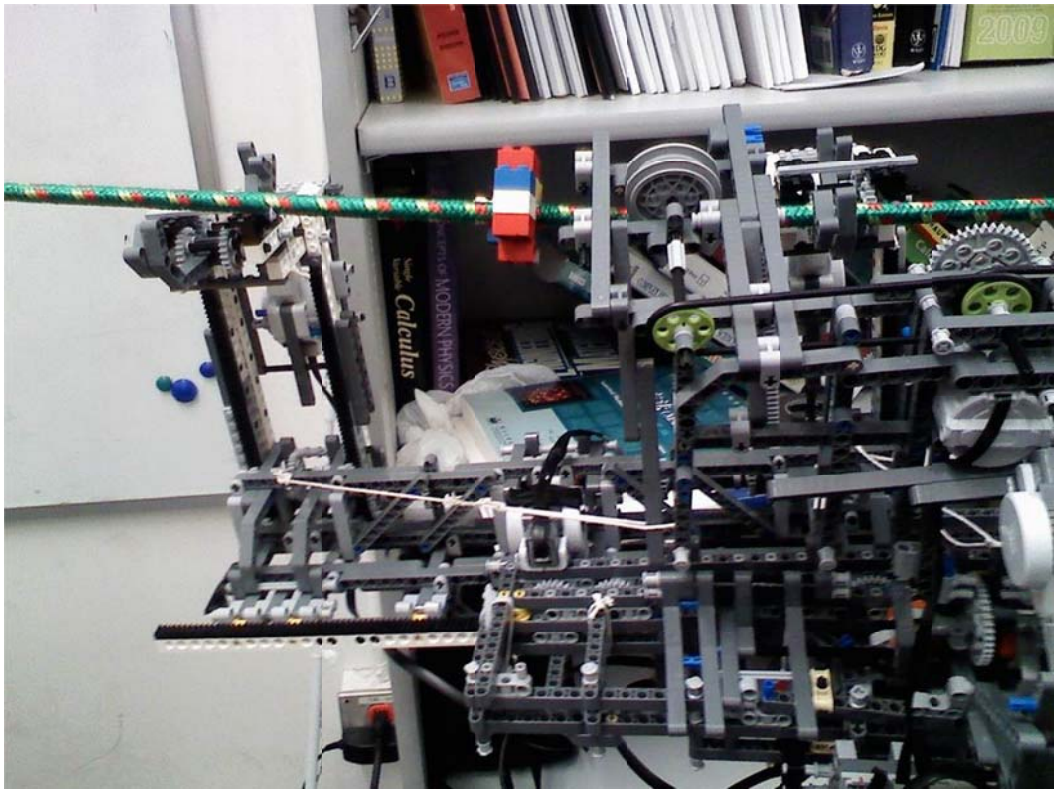


**Figure 5. Both gripplers raise up to grab the cable.**

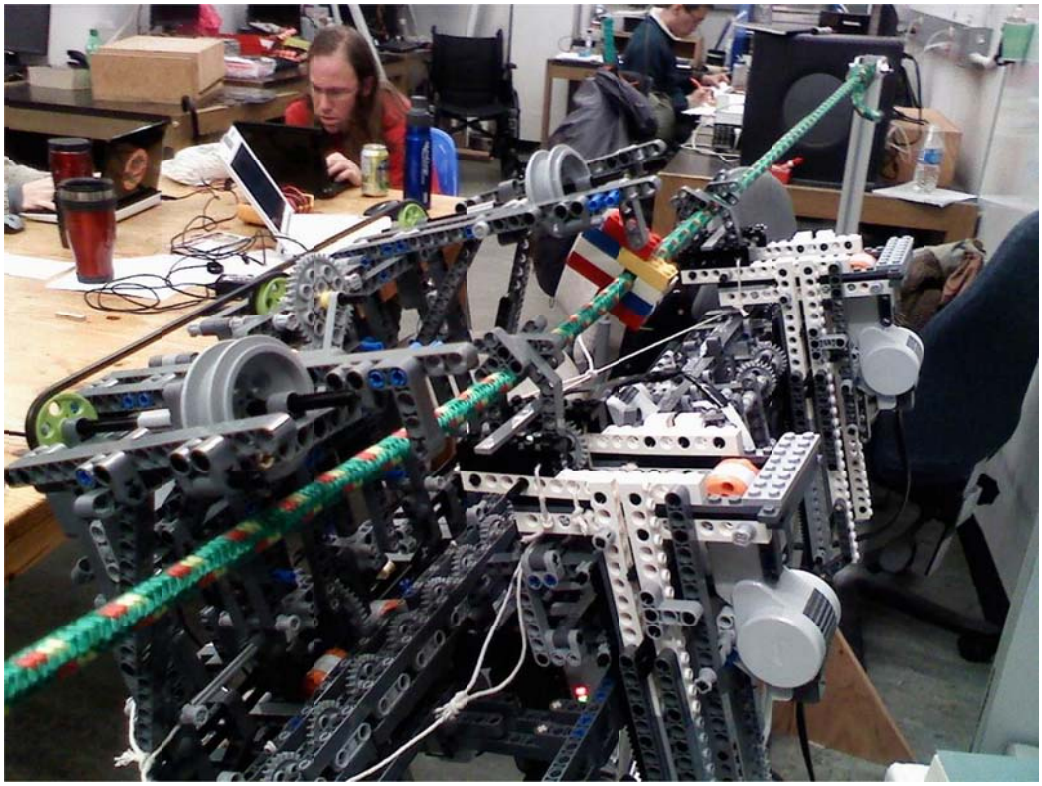

**Figure 6. The gripplers close and then lift the wheels off the cable.**

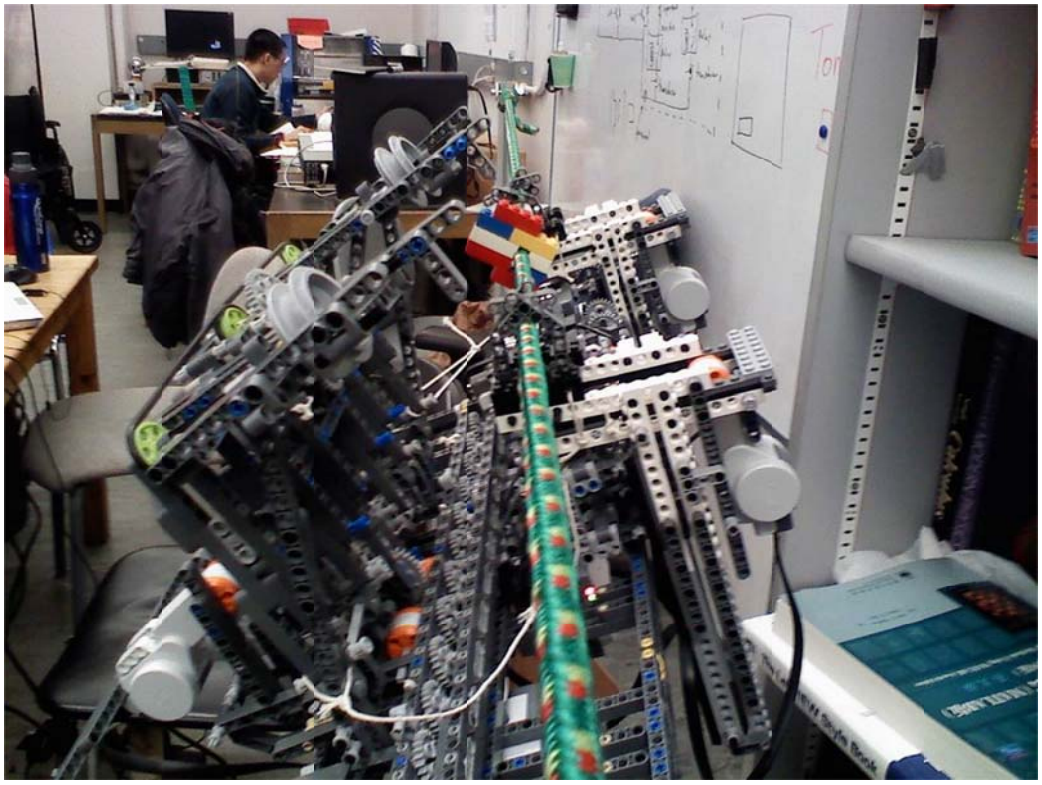

**Figure 7. The wheel frame is further tilted backward and the frame support disengaged.**

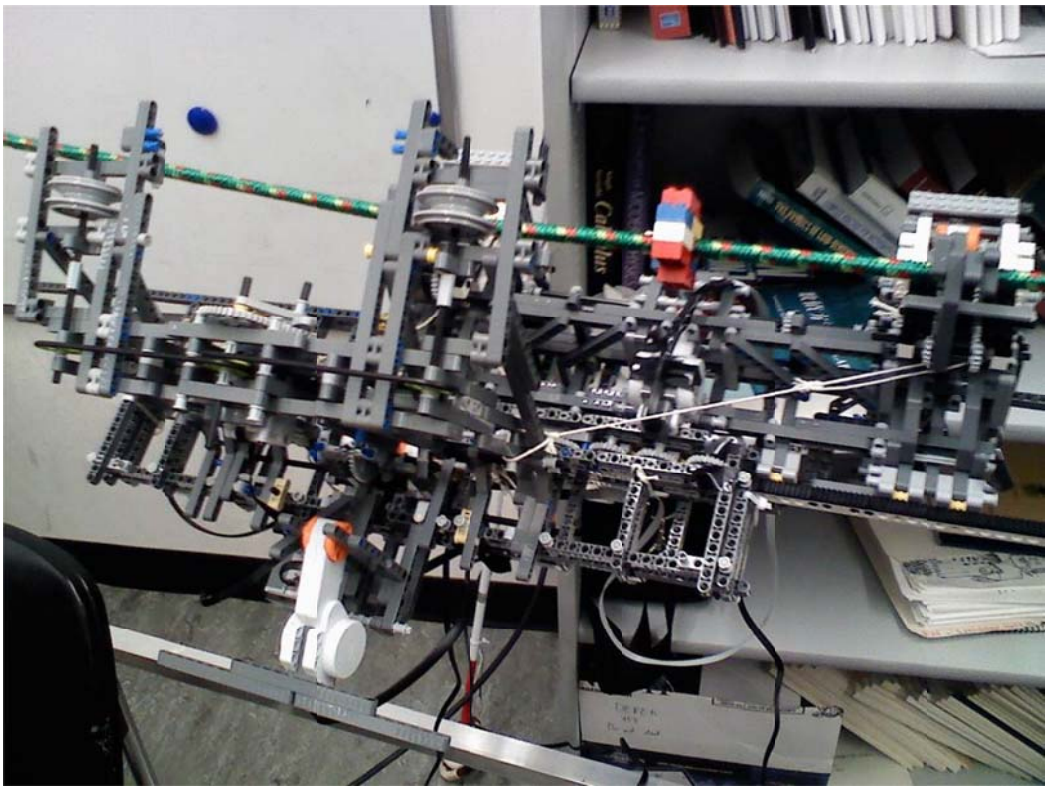

**Figure 8: The wheel frame slides to the other side of the obstacle.**

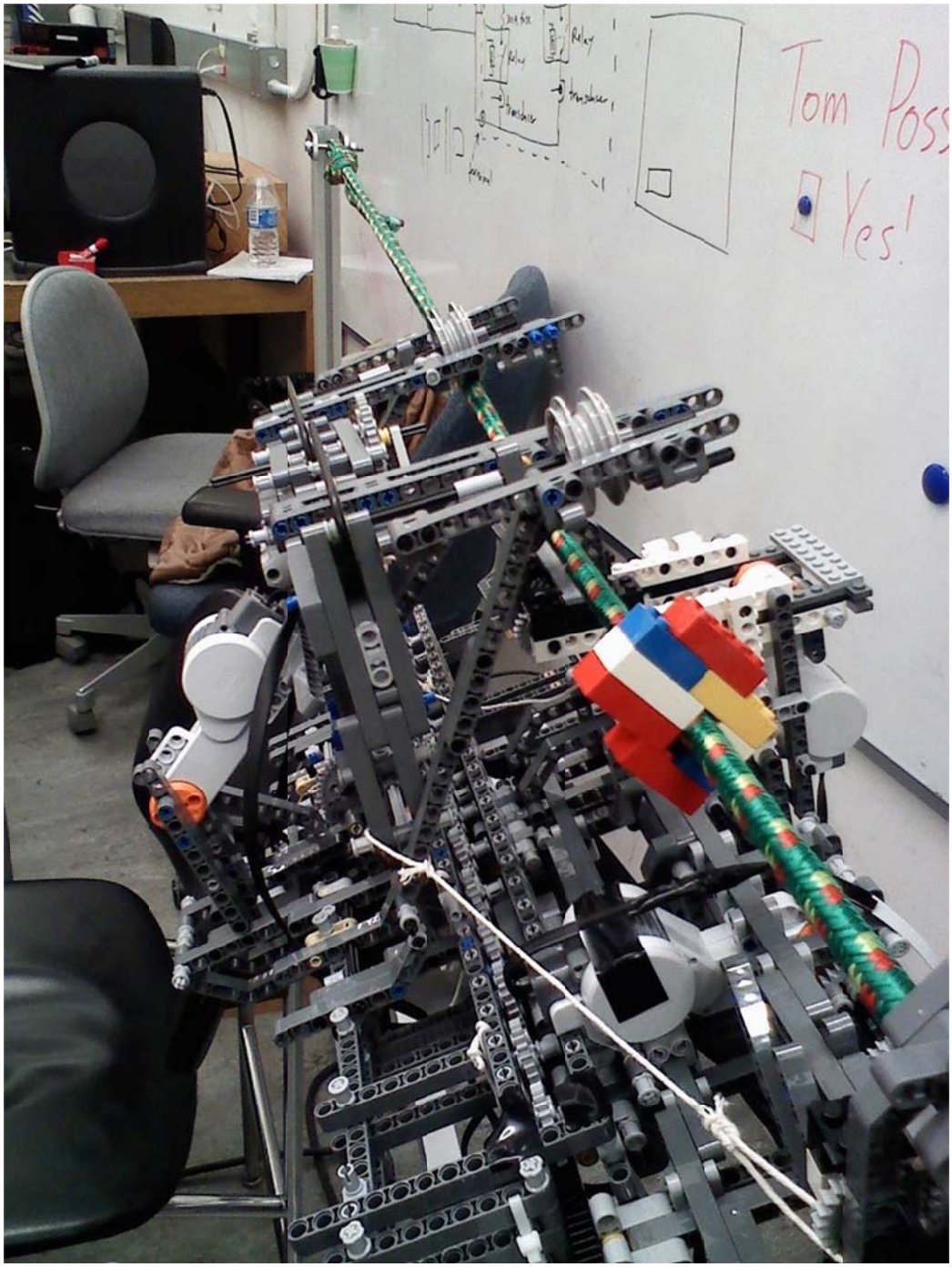


**Figure 9: The wheel frame and frame support are tilted back onto the cable.**

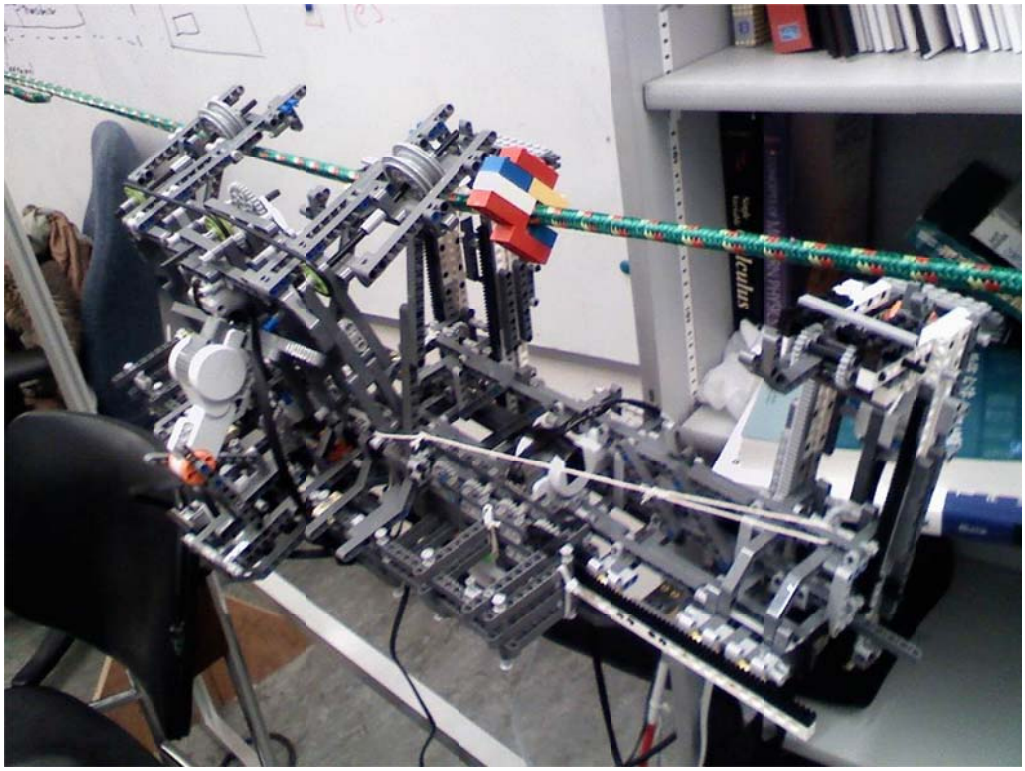

**Figure 10: The wheel frame is lowered back onto the cable and then gripplers open.**

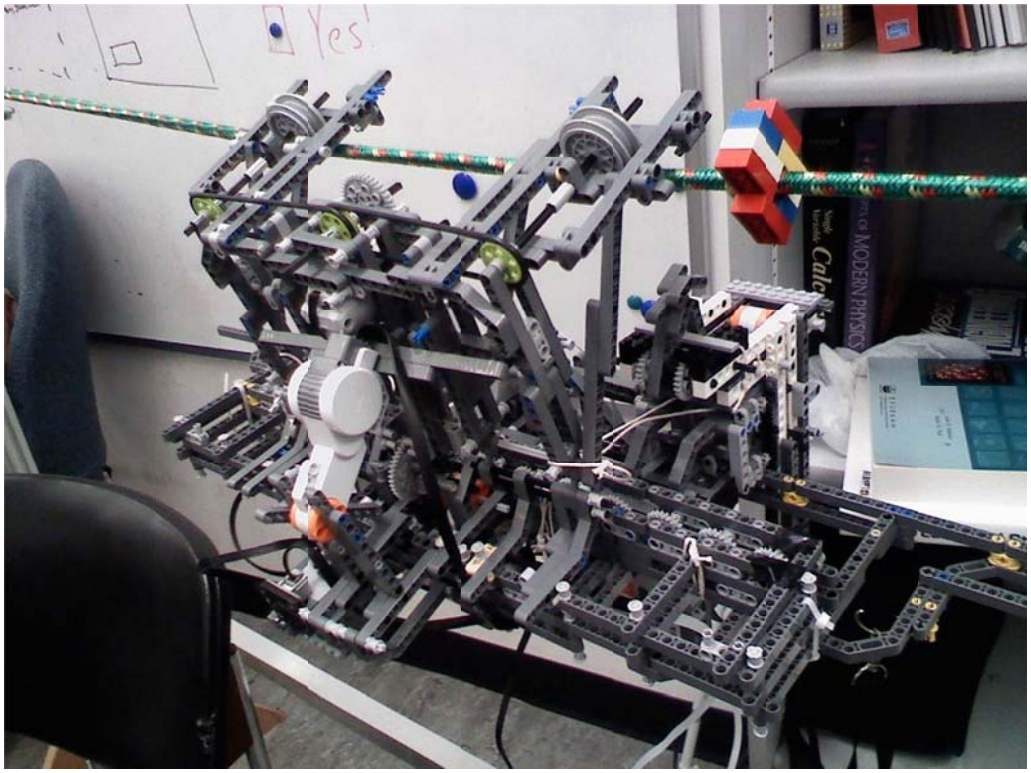


**Figure 11: The robot continues to travel down the other side of the cable after returning to original position.**

**Algorithm**

The control of the LEGO LineScout is done manually over Bluetooth wirelessly on a computer. Direct commands are sent to the two NXTs to control all the motors. This enabled us to program in our familiar and also algorithm-efficient C/C++. The library we are using was developed by Anders' Mindstrom[1]. An exception to the direct motor command is the Power Function Motor. In order to use the Power Function Motor, which was designed for previous LEGO product, we used a RF remote control sensor called PFmate connecting to one of the NXTs' sensor port. Due to the fact there is no controlling library for PFmate from Anders, an alternative implementation was developed. A NXT-G program is downloaded to the NXT connecting to the PFmate. This program reads the Bluetooth inbox message and if the received messages is forward or reverse, the NXT sends the required signal through PFmate over RF to control the Power Function motor accordingly. If there is no message in the inbox, by default, the motor coasts to a stop. The code for the Bluetooth control interface and the NXT-G program are both archived and attached with this report.

## 2.4 Result and Discussion

Upon testing, there are some uncertainties in disengaging the wheel frame lock. This mainly was caused by the heavily loading of the entire robot. Due to the fact that the weight of the robot is mostly in the base frame, the lock and joints in wheel frames undergoes bending. Nonetheless, this issue may be resolved by tiling the wheel frame and the locking mechanism insides a few times in order to get the lock to release. Due to this reason, automation has not been implemented since the procedure to unlock the locking mechanism is uncertain. There are no quantitative results for the required time to cross an obstacle for this reason. If the wheel frame lock can be successfully disengaged, the replica can cross obstacles of size roughly the size of a tennis ball.

During the testing stage, the LEGO LineScout often showed more inconsistent performance among the days. Lots of efforts were made to discover that the structure strength of the LEGO Technics is just not that strong for our design. Snaps that lock two beams together cannot hold the two beams tightly in place when they are under load. This causes the horizontal slider failed to slide smoothly. It also bends and twists when it is extended. The above two were some examples of the problems we encountered. Strings were used to temporary fix the issues. Before each run, it is a good practice to make sure all the snaps tightly connect the beams.

## 3. Conclusion

The replica of the LineScout has been built at the completion of this report. LEGO parts are used for the construction of this replica and the preliminary design was completed in the LEGO Digital Designer. Much effort had been put in to locate the necessary parts for the design as no one vendor sells them all, including LEGO. This put the project on hold for roughly two months.

The result of the final LEGO LineScout can cross obstacles of a size of a tennis ball. However, its robustness is still not very good. The major reason is because LEGO parts suffer from deformation under load. However, further tests should be conducted to find out if there are more reasons other than the structure strength of the LEGO pieces if time allows.

There are a few features that the real LineScout has that the Replica does not. The features that are lacking are the variable cameras, the three-frame design and a robotic arm. Every motion performed on the LEGO LineScout is inspected by the eyes of the operator directly. This can produce an issue when the cable is far away from the operator. The three-frame design that the real LineScout has allows it to cross larger obstacles and to be more balanced. However implementing this on the replica will definitely signify the increasing in weight as more motors are introduced. With current two-frame design the LEGO parts are almost at their limit. It may not be a good idea to implement the three-frame design unless efforts are made to lighten the structures. Finally, the robotic arm is used to repair or to inspect the obstacles crossed. Since this is a rather familiar technology to the public, it may not be worthwhile implementing such feature on the LEGO LineScout.

# 4. Project Deliverables

## *4.1 List of Deliverables*

- LEGO LineScout prototype robot
- Complied controlling software
- C/C++ Source Code and NXT-G program
- Excess LEGO parts
- LEGO Digital Designer Design File

## *4.2 Financial Summary*

The table below summarizes the major costs for the project, where, and who is it from.

| Description | Quantity | Vendor | Cost | Purchased By: | Funded By: |
|---|---|---|---|---|---|
| LEGO Parts | | LEGO | Approx $1500 | BC Hydro | BC Hydro |
| Testing Platform | 1 | | | Project Lab | Project Lab |
| Bluetooth Dongle | 1 | ASUS | | Project Lab | Project Lab |

## *4.3 Ongoing Commitments by Team Members*

Currently, the LEGO design files need to be updated to match the actual design. Sartsawat is taking responsibility for this task and will give it to the Sponsor by the January 31, 2011. If there are chances of the project going into the Science World, both Sartsawat and Sean are willing to participate in the demo runs provided time allows.

# 5. Recommendations

There are several features that are not implemented in the current project design. In this section, we have listed some goals that can be carried on for future projects based on the results of this project. In addition, some issues are pointed out that affect the robustness of the obstacle crossing sequence.

## *5.1 Implement GUI Control*

The current project uses a text-based program to reduce the development time. By implementing the graphical user interface (GUI), it will provide ease of use for the operator to control the LEGO LineScout. This will allow younger operators to use the replica.

## *5.2 Install Cameras to Aid Obstacle Crossing*

The actual LineScout robot has several wireless cameras mounted on several place to assist the operation of the robot since it can go out-of-sight easily; also they can take images (still or motion) of the inspection process for records.

## *5.3 Use Alternative Material or Design for Bending Issue*

The LEGO prototype twists and bends when the horizontal slider is extended. One may consider alternative materials that are more rigid because LEGO parts suffer bending greatly when the load is applied. Alternatively, the robot may also be lightened to reduce the bending issue.

## *5.4 Redesign Wheel Frame for Rotation Issue*

The current wheel frame cannot rotate forward past 90 degrees with the body frame. This is due to several reasons such as the axles bending. The major cause is due to a supporting rod physically stops the frame from rotating forward. By allowing the wheel frame to rotate in more, the frame support can disengage easily during the obstacle crossing sequence.

## *5.5 Individual Motors to Raise/Lower the Gripplers*

When the horizontal sliders are extended, the centre of gravity of the robot is shifted and the gripplers do not travel the same height to reach the cable. By allowing individual control of the vertical positions of the gripplers will enhance the flexibility. However by introducing an extra motor, the overall weight of the robot must be increased and needs to be considered.

# Appendices

## *A1. Procedures to setup Bluetooth on the PC*

In order for Bluetooth communication with NXTs possible, the PC must have Bluetooth connectivity. With Bluetooth enabled on the PC, pair the first NXT with default passcode "1234". Then you need to enable the service "Device B" for the NXT you just connected. This will assign an outgoing COM port to the NXT. Note down this number and the NXT you just connected. Then repeat the same steps to setup the second NXT and note down the outgoing COM port for this NXT. The COM ports are required for the next section.The precise steps to setup a Bluetooth-serial COM port may vary on different systems. One step-by-step instruction was found online from the University of Buffalo. Please refer to the link in the reference section for this guide.

## *A2. Configuration File*

Config.txt

This file contains parameters that the user can define to vary the control program without recompiling. However this file must be modified before the program is executed or else the changes will not take effect until you restart the program. The parameters are explained below.

*1. COM1* and *COM2*. These two parameters are the COM port number you noted from the section above. The NXT and COM port assignment is outlined in the table below.

Table 1. COM port assignments

| | |
|---|---|
| COM1 | Port A – Left Grippler Servo<br>Port B – Wheel Frame Servo<br>Port C – Horizontal Slider Servo |
| COM2 | Port A – Right Grippler Servo<br>Port B – Grippler leverage servo<br>Port C – Wheel Frame Support servo |

2. *display_positions* and *control_display*. These two parameters are to there to help new users who are unfamiliar with the controls or for debugging purpose. Setting *control_display* to 1 will allow the program to keep the information of key assignment on top. Setting *display_positions* to 1 when control_display is 1 will print out the positions of all servos, along with the control key mapping, whenever a key press is made. Note turning display positions may cause the program to be less responsive.

3. Speed configurations. Change any of these parameters will change the speed at which the servos rotate for the entire program. The higher the values, the faster they rotate. If you put down a negative value to a servo, it will reverse the direction of the motor, hence the controls for that servo are swapped.

## *A3. Code*

The package is archived and attached along with the submission of this report. The tool we used to develop the program is Dev C++ with Anders' C++ Communication Library. The code has enough comments for the future programmers to understand.

The NXT-G program, Bluetooth test, that drives the PFmate is also contained in the same archive as the code.

## *A4. How to use the control program*

The LEGO LineScount Bluetooth control program is very intuitive to use. At the initiation of the program, you will be asked if you want to change the COM ports from the default values in the config.txt file. If yes, press y and enter the first NXT's COM port for the first prompt and follow by the second NXT's COM port for the next prompt. Then the program will try to connect to the two NXTs. Once it is successful, it would print out the controls for once regardless of whether you have set *control_display* parameter to 1 in the config file. If it fails to connect, please check the two NXTs' battery level and Bluetooth connection and restart the program again.

Once the program successfully connects to the NXTs, you should notice the red LED on Pfmate at the back of the LEGO LineScout should be flashing. This means that the Power Function Motor is ready to be driven. If the LED light is not flashing, manually start the NXT-G program you downloaded to the NXT connecting to the Pfmate. Then the follow controls are used to control the motors of the robot.

Table 2. The control mapping for the Bluetooth Control Program

| Key | Control | Key | Control |
|---|---|---|---|
| 1 | Drive Power Function Motor forward | A | Raise both gripplers |
| 2 | Drive Power Function Motor backward | Z | Lower both gripplers |
| O | Open left grippler | < | Move wheel frame to the left |
| L | Close left grippler | > | Move wheel frame to the right |
| P | Open right grippler | S | Tilt wheel frame forward |
| : | Close right grippler | D | Tilt wheel frame backward |
| Q | Engage wheel frame support | W | Disengage wheel frame support |

## References

[1]*Andre's Minstorm Page.* (n.d.). Retrieved January 10, 2010, from http://www.norgesgade14.dk/bluetoothlibrary.php

Lea, C. (2008, October 2008). *Setting Up the Lego Mindstorms NXT with Bluetooth.* Retrieved January 10, 2010, from http://www.eng.buffalo.edu/~colinlea/Bluetooth_With_NXT.pdf

mindsensors.com. (n.d.). *PF Motor controller for NXT resource page*. Retrieved January 10, 2010, from http://www.mindsensors.com/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=107