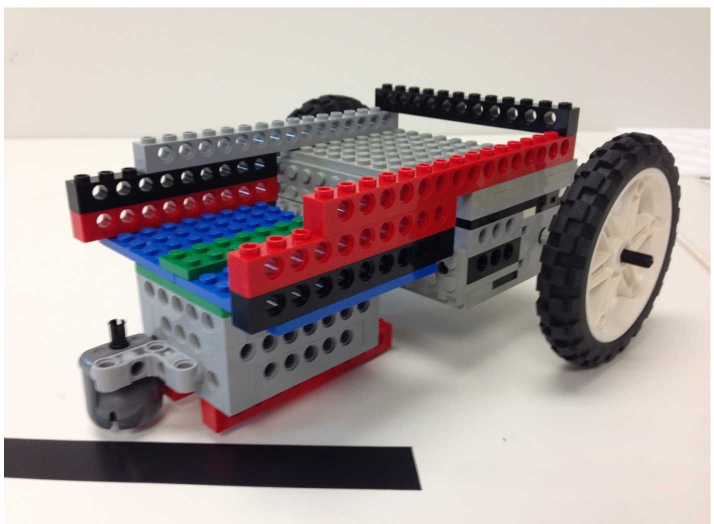
# Development of an Educational Robotics Platform Using Wireless Communication Between Components



**Primary Investigator: Matthew Mueller**

**ME96 - Senior Honors Thesis**

**Spring 2015**

**Faculty Advisor: Dr. Chris Rogers**

**Committee Members: Dr. Chris Rogers and Dr. Merredith Portsmore**

## Abstract

This study involves the development of an educational robotics platform with the sensors and actuators acting independently of each other and using wireless technology to communicate and how users interact with it differently than an existing platform. The LEGO Mindstorms platform regularly causes frustrations for novice students when they are trying to connect and wire the motors and sensor to the intelligent brick with the Technic building system. The new platform was developed as a way of allowing more students to have success in the project based learning environment as beginners by eliminating the need to have an intelligent, powered brick at the center of any robot creation. A pilot study was conducted to compare the new platform to the LEGO NXT when used to complete educational design challenges. Of the six undergraduate and graduate students tested, five participant responded that the new platform was easier to work with. Two case studies are presented in the paper, one of a participant with extensive experience with Mindstorms and one of a participant who had never built with LEGO Technic pieces before. These cases work to display the effectiveness of the platform at enabling beginner students more easily and the technology constraints of the platform that limit complexity. There are many engineering challenges such as powering each component separately and creating an autonomous robot of this nature that doesn't need to have a computer controlling it, but feedback from the pilot test hinted at the new design's potential in the educational setting.

# Table of Contents

# 1. Introduction

## 1.1 Background

As computer technology has begun to play a major role in society, educators are starting to realize the importance of integrating these technologies into their lessons. Teachers wanted to expose their students to the new tools at their disposal to enable their creativity and give them insight into how computers work. Robotic technologies have emerged as popular educational tools with 3 million units expected to be sold between 2014 and 2017 [1].

LEGO came out with their most popular robotics platform, Mindstorms, in 1998 when they released the RCX, a set consisting of motors, sensors, and an intelligent brick. Children were able to use a simple and user-friendly graphic programming language to come up with a program which they would then upload to the intelligent brick at the center of their robot. LEGO Education has come out with two more versions of the Mindstorms platform with the NXT and EV3 (Figure 1a), as well as their WeDo set (Figure 1b) targeted at a younger age group that controls separate motors and sensors connected to a computer with a USB cable.

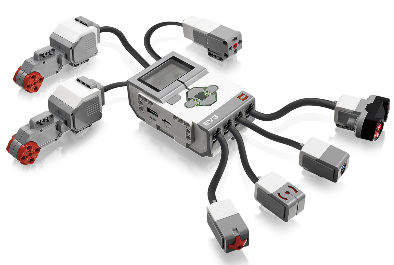


Figure 1a (left): The EV3 intelligent brick with motors and sensors wired in.

Figure 1b (right): The WeDo platform, with a USB cable to control the motors and sensors

Project-based techniques are often used to implement the robotic technology [2] in which students use a building platform to create a design to complete a challenge, learning many powerful concepts about problem solving, programming, and other disciplines along the way. This serves as a perfect way to give children a context for their learning and provides many new outlets for them to test out their knowledge in engaging ways. The educational potential for robotics technologies are vast, and as technology advances, the robotic technologies can reach a wider audience by lowering the barrier to entry and expanding what is possible with them.

### 1.2 Literature Review

Extensive research has been conducted into the effectiveness of robotic technologies in the educational setting over the past twenty years. The movement of using new computer technologies as teaching agents was coupled with the theory of constructivism, the idea that people generate knowledge and learn more effectively through real experiences [3]. Building with robotic technologies touches on many different principles in the STEM disciplines, allowing students to learn these topics in an experiential and engaging way.

Systems that enable students to create their own robots are one of the most common answers to the question of which technology tools are most effective in the educational setting [4]. These platforms generally consist of an easily buildable mix-and-match hardware combined with simple coding language that minimizes the complexity and development time of robotics, making the technology accessible to more students. Using these "user created robots" along with a project-based learning approach allows teachers to touch on arts and history lessons in addition to being focused on STEM and robotics topics [5]. In their study "Philosophy and Strategy of Minimalism-based User Created Robots for Educational Robotics- Education, Technology and

Business Viewpoint", Park and Kim propose that the more a the UCR minimizes its intricacy, cost, and development time, the more effective it will be as an educational tool.

In addition to enabling new and traditional lessons to be taught in a more interactive way, the act of building a robot teaches children very valuable lessons about problem solving, critical thinking, and creativity [6]. Posing challenges to students that can be solved in different ways and allowing them the freedom to design their own solution enables them to experience a problem-solving situation, and robotics' ease of testability allows the user to get instant and constructive feedback on their progress [7].

LEGO Mindstorms is one of the most common "user created robot" platforms used by researchers trying to evaluate the effectiveness of robotics as teaching agents [8, 9, 10]. There is general consensus in the academic field that the LEGO robotics sets have clear educational applications and there have been extensive investigations into their benefits and the best ways to implement the technology [11]. Despite the numerous studies, however, there has been limited research analyzing the relative effectiveness of the Mindstorms' hardware design of an intelligent brick with wires connecting motors and sensors.

### 1.3 Problem Statement

Through the STOMP program [12] and multiple academic classes at Tufts University, I have worked extensively both as a user and a teacher/facilitator of the LEGO Mindstorms platform. The fact that the same platform can be used as an educational tool at both the elementary and university level is certainly a testament to both the fundamental simplicity and power of the Mindstorms platform. Students generally have some level of success creating diverse and creative solutions to challenges, but I have also observed frustration with the building aspect of the platform when students are first introduced to the platform, at both the elementary and university levels.

The most common example of the difficulty students face that I have observed is when they attempt to put together a robot that can drive and change directions. This design is conducive to some of the best lessons for teaching programmatic concepts because basic movement control is one of the easiest ways to visualize the actuation of computer code. The primary challenge comes as users attempt to connect the motors to the intelligent brick that controls them. The central brick, motors, and sensors have holes that interface with the LEGO Technic building platform, allowing them to be connected with beams and pegs, but it is often times difficult to figure out how to attach the two in a way that allows for a sturdy vehicle with full range of motion. These building constraints limit students in most instances in which they are trying to connect a motor to the intelligent brick, even when using just one motor.

As a fellow in the STOMP program, I have consistently observed frustration as students struggle to build a sturdy structure for most of the class time, not leaving time for them to perform iterations on their design as they code. While building constraints offer students an experience closer to that of an engineer and can help promote creative thinking, they can also act as a deterrent to some students, raising the age of target users. Eliminating these constraints by minimizing the complexity of the building platform would allow beginner users to spend more time in the testing and redesigning phases of the engineering design process.

### 1.4 Thesis Questions

Based on many hours of observing the building patterns of students using Mindstorms in the educational setting, the aspect of the platform that seems to add the most complexity to the building system is the intelligent brick. In this study, I attempted to lower the barrier of entry into building with a LEGO robotics platform by designing and prototyping a robotics platform that utilizes wireless communication to eliminate the need for an intelligent brick. Motors and sensors were made to stand alone, but still be controlled by a program on the computer. In order to focus this study on the differences in the way that robots are constructed with each platform, a user interface was also developed to allow students to program the wireless technology in a similar way to the existing platforms.

Once the new platform was developed, it was used to produce two case studies to investigate the potential of a system that uses wireless communication between sensors and actuators as an educational tool. This study acted as an initial investigation into the following questions:

(i) How can an educational robotics platform use distributed intelligence to enable more students to experience success when used in a project-based learning environment?

(ii) What are the differences in the building and problem solving patterns of a platform with wireless communication and the traditional Mindstorms platform?

(iii) How can users benefit more from the newly developed technology and what limitations does it create?

(iv) How does the difference in technology change how students work on specific, goal-oriented challenges compare to how it changes how they work on open ended challenges?

# 2. Methodology

## 2.1 Scope and Specifications of New Technology

The development of a new technology culminated in a study investigating how students interact differently with the new building platform compared to the LEGO Mindstorms platform. In order to ease the difficulty that beginners face when building with Mindstorms, the new platform utilizes wireless technology to allow the separate components (motors and sensor) to be placed anywhere on their robot. This eliminated the constraints of needing to attach and wire the components to the intelligent brick, as well as the need to upload your code to the robot each time you make a change.

Considering the constraints of having the new platform be wireless and the goal of the study to compare the technologies, a few general specifications were brainstormed:

- Each of the separate components needs to have its own power source
- Each of the components needs to have some form of intelligence that can process, send, and receive commands
- There needs to be a way to connect the components using a standard building system
- The platform must allow for a diversity of solutions to any challenge
- The user interface for coding the robot must be as similar as possible to the one used by Mindstorms

With these main requirements for the new technology in mind, a spec chart (Table 1) was created to guide the development of the platform. The number of motors and sensors was determined based on the minimum number that I thought could produce an acceptable diversity of solutions and what was accomplishable in the time span of the project. Battery life was not the biggest concern in the very early stages of this investigation, and was chosen based on the maximum time the platform would be used in a day of user testing for this study. The latency requirement was chosen based on the "perceptual processing time constant" needed to make movements appear animated [13]. Similar to the speed and torque specifications, the size specification was based on the ~190 $cm^3$ volume of the NXT motor to capitalize on the ergonomics of its size.

| **Specification** | **Unit** | **Ideal** | **Marginal** |
|---|---|---|---|
| Number of individually controllable motors | Quantity | 4 | 2 |
| Number of different sensors | Quantity | 3 | 2 |
| Battery life (while in use) | Hours | 3 | 1 |
| Reaction time from sensor input to motor movement | Milliseconds | 50 | 0 |
| Motor torque must be similar to others used by LEGO platforms [14] | N-cm | 5 | 2 |
| Motor speed must be comparable to that of NXT [14] | RPM | 177 | 150 |
| Size of each of the components | $cm^3$ | 200 | 400 |

Table 1: Specifications for the technology that was developed in conjunction with this study

### 2.2 Description of User Test

A user test was conducted in order to test the effectiveness of the new, wireless platform compared to the effectiveness of the Mindstorms NXT. The goal of the study was to analyze how the different technologies could be used to complete educational design challenges of different natures. For this first investigation on the new technology, a pilot study of six users was used to collect data in the form of observation, images, and survey responses on the difference of use between the platforms.

All participants were required to have some experience working with LEGO Robotics to allow for more success and a slightly more challenging activity to be tested. Recruiting was done in the STOMP program and with grad students working at the CEEO at Tufts University. Despite the requisite experience, the test was designed to have each participant work with both technologies in one session in order to compare their solutions and have both experiences in mind when they were asked to compare the two. As a way of studying how the new technology is used to design solutions for a challenge with a specific goal, as well as a challenge with a more open ended task, the following prompts were given to the participants during the study:

- **Complete the Maze:** Design a robot that can complete the taped maze on the table (Figure 2a) using the motors, sensors, and LEGO pieces provided to you. Feel free to add objects to the maze if they would help you use different sensors to complete the challenge. Please ask the investigator any questions that you have as you work. You have up to 45 minutes, but there is no pressure whatsoever to finish the challenge.

Figure 2a (left): Maze designed for challenge. Figure 2b (right): Maze after user modification (Section 4.3).

- **Amusement Park Ride:** Design a fun amusement park ride using the motors and sensors provided to you, as well as any found materials available. There are no restrictions here. Have fun! Please ask the investigator any questions that you have as you work. You have up to 45 minutes, but there is no pressure whatsoever to finish the challenge.

At the beginning of the study, the participants were asked to complete the first page of the survey (Appendix A) answering questions about their prior experience working with LEGO Robots and programming. They were then randomly assigned a technology to use first (either the new technology or the NXT) and given one of the two prompts above at random, ensuring that a diversity of solutions could be created for each challenge and there would be no bias in the responses based on the order in which they are used. They were given a brief overview of the components of each technology and how it was programmed before they began each challenge. Along with the new technology, users were given a bin of assorted LEGO pieces including but not limited to pegged Technic beams, baseplates, and the EV3 rolling caster ball, but were told that they could request any other pieces.

The forty-five minute maximum was given for each activity to model the time limitations faced by students when doing an activity in a short lesson period. After each one of the activities was completed, the participant was asked to answer questions about their experience with the technology and to self-evaluate their success for the challenge. Observations were carried out as they worked noting how they constructed their robot and problem solved as they coded with each technology. Screen-shots were taken of all code that was written and photographs were taken of the final designs for each of the challenges. At the end of the study, the participants were asked questions about differences between the two technologies and how each platform helped or constrained them for the two types of challenges.

This collective data set was analyzed as case studies to evaluate the potential effectiveness of an educational robotics platform with distributed intelligence in the educational environment both for specific, goal-oriented challenges and ones with an open ended prompt.

# 3. Overview of the HiveMind* Technology

*The newly developed technology will be referred to as HiveMind from now on

## 3.1 HiveMind Platform Design

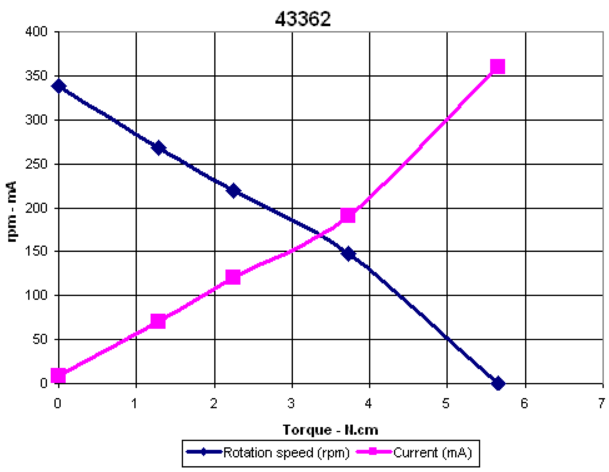
The goal of HiveMind was to enable a comparative user test looking at the benefits and limitations of a platform of independent, wirelessly communicating sensors and actuators. This chapter will give an overview of the hardware and software used to develop the new platform.

The platform needed to allow students to complete both the maze and the amusement park ride challenges, and needed to have two different sensors to allow for a diversity of solutions. Two separate motor blocks were created, one with two separately controllable motors, and one with a single motor. This was done to enable students to have more success with the building aspect of the challenge and allow more time for problem solving as they code. A touch sensor brick and a light sensor brick were designed for the platform because of their relevance to the challenges and ease of understanding to students.

To allow the wireless devices to use the same graphic coding language as Mindstorms, the platform was designed so the robot is controlled real time with code created in LabVIEW. A user interface (discussed in Section 3.7) was made to mimic the way that the Mindstorms platforms are programmed. HiveMind was designed to be constructed using the LEGO building brick system, but used beams with peg inserts compatible with Technic to make the platforms as similar as possible.

### 3.2 Parts Selection

The major constraint considered for the motor selection was size due to the desired design of having two separate motors in one brick. Combining size constraints with the specifications for torque and speed, LEGO's electric Technic mini-motor was selected for the platform (Figure 3a). 9V batteries were selected to power the motors because they would be capable of running both motors at their average current draw of 0.12 A (240 mA) for at least one hour [15] and allow for quick and easy battery changes.



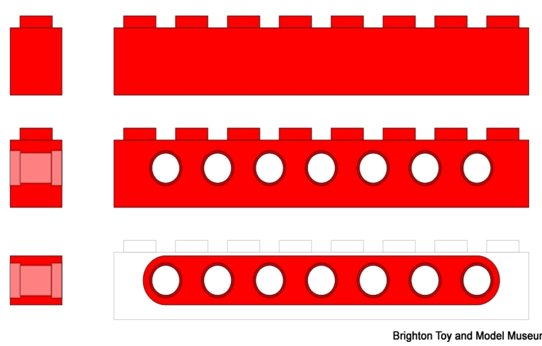
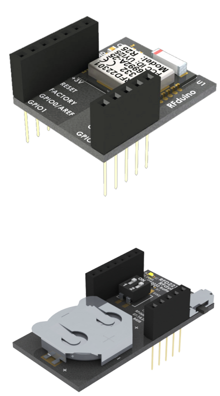




Figure 3a (left): Torque plotted against speed and current for the electric Technic mini-motor [13].

Figure 3b (right): The beam in the middle (compatible with Technic and traditional bricks) was used for the housings.

The housing of each component was constructed out of traditional LEGO building blocks, including some beams that are compatible with the LEGO Technic platform (Figure 3b). Pieces were designed to have baseplates on top and bottom to allow them to be connected easily with the traditional and familiar LEGO building platform. The components were also designed to allow for use totally separate from one another, so each had to be entirely self contained and self sufficient.

The most important decision in the selection process was which wireless technology to use as the controller and communicator for the separate pieces. After considering several options, the RFduino (Figure 4a), a miniature Arduino with six I/O pins that uses low energy bluetooth for communication [16], was selected to be used as the controller for each brick and as the host sending the commands from the computer code. The technology is small enough to fit into the dimension specifications, and can be powered with a 3V coin cell battery shield that lasts at least 2 hours under a use condition based on my use observations.



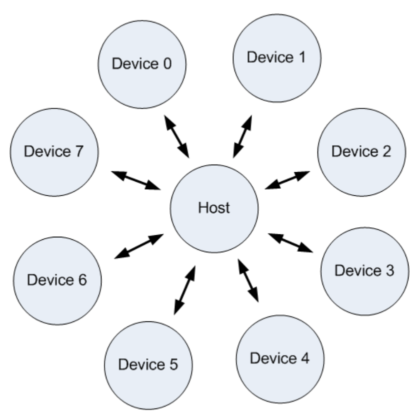



Figure 4a (left): The RFduino on top (23mm x 29mm) stacks on the 3V shield (23mm x 47mm).

Figure 4b (right): A star network created with GZLL communication, a host talking bi-directionally with up to 8 devices.

### 3.3 Communication Scheme

Considering what was supported by the RFduino platform, the communication network that was used to send and process signals between the different components was called the Gazell Link Layer [14], or GZLL. This system allows for a star network (Figure 4b above) in which a host can have bi-directional communication with up to eight devices at a time, enabling a robust wireless communication system to be established. The host, attached with a USB shield to the computer, used serial communication within the computer to allow LabVIEW code the ability to send, receive, and process commands to and from the devices.

### 3.4 Motor and Sensor Brick Hardware

In addition to the RFduino board and 3V coin cell shield used to power it, the two motor bricks will have a 9V battery to run the motors and an H bridge to control their directions and speeds. Figure 5 shows the schematic for the two motor housing, in which the I/O pins on the RFduino are used to enable the motors and pulse width modulation (pwm) to control their speed.

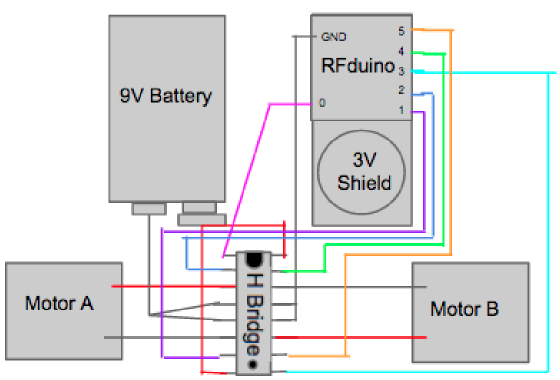



Figure 5: Wiring scheme for the two motor brick, with a 9V battery powering an H-bridge and the RFduino controlling the voltages going to each of the motor dictating direction and speed.

The pwm signals are used to limit the supply voltage coming from the 9V battery to either lead of the electric technic mini motor, a DC brush motor, allowing control over its speed and direction. The single motor brick for the HiveMind system has the same exact scheme as the double motor brick, but with only one side of the H bridge wired to the RFduino and the motor.

Rather than develop an entirely new touch sensor, the NXT touch sensor was connected to the controller and built into the sensor brick to capitalize on the backend development that LEGO put into the mechanics of the touch button. The circuit used the 3.3V output from the RFduino to run through a couple of resistors, and an input pin to read whether or not the circuit was completed indicating the touch sensor was compressed.

Similar to the touch sensor brick, the light sensor used a circuit (Figure 5) with two resistors and an input pin in the RFduino, but one of the resistors was a photoresistor that changed based on the amount of light it sensed, changing the voltage that would be read by the RFduino. The different voltage readings were correlated to different amplitudes of light, allowing the user to program the robot to react based on different light exposures.

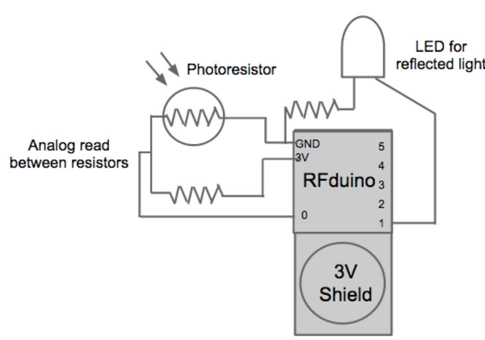



Figure 5: The wiring scheme for the light sensor brick, analog reading the node between the photoresistor, and digital writing the LED on and off

In addition to the input pin on the RFduino, there was an output pin turning on and off an LED to enable the sensor to be used for both ambient and reflected light. When the LED is turned on and pointed at a surface next to the sensor, the sensor will read a bright value if the surface is light in color, and a dark value if the surface is dark in color because the LED's light will not be reflected as well by dark colors. This would again allow for a greater diversity of solutions, and is conducive to solving the maze challenge.

### 3.5 Arduino Code

One major advantage of using the RFduino platform to construct HiveMind was that it is programmed with the familiar Arduino IDE and uploaded onto the board with a USB shield. Each of the motor bricks simply needed to receive a signal from the host, and interpret the commands to turn each of the motors the correct direction and speed. However, the GZLL communication network requires the devices to initiate the connection, so the motor bricks were set to continually send a null signal to the host, which would then respond with the desired command.

The sensor bricks were each set to continually send their readings to the host; a value of 1 or 0 in the case of the touch sensor based on the digitalRead() from the input pin to determine if the touch sensor was being pushed, and a mapped voltage value in the case of the light sensor based on the analogRead() from its input pin. The light sensor also needed to receive commands from the host, turning the LED on and off depending on if the user wants to read ambient or reflected light. The motor bricks only needed to receive and process commands, but were still required to initiate the connection with the host.

All devices were left in push mode, sending all available data when prompted. As shown in Figure 6a, the RFduino in the motor brick uses a "RFduinoGZLL.sendToHost()" command to initiate connection with the host. If the host (Figure 6b) receives any data from DEVICE3 with the "RFduinoGZLL_onReceive()" command, it sends the data input it read from serial back to that device with the "RFduinoGZLL.sendToDevice()" command. When the single motor brick receives data from the host with the same "_onReceive()" command (back to Figure 6a), it then uses digital and analog write commands to set the output pins connected to the Hbridge, controlling the motor's direction and speed.

```
ME94OneMotorCode
void loop() {
  delay(50);
  // request the state from the Host (send a 0 byte payload)
  RFduinoGZLL.sendToHost(NULL, 0);
}

void RFduinoGZLL_onReceive(device_t device, int rssi, char *data, int len)
{
  if (device == HOST) {
    cmd = *data;
    switch (cmd) { // Process Commands
    case 84: // T = STOP
      digitalWrite(mae, LOW);
      digitalWrite(ma1, LOW);
      digitalWrite(ma2, LOW);
    break;
```

```
ME94Host
void RFduinoGZLL_onReceive(device_t device, int rssi, char *data, int len)
{

  if (device == DEVICE1) { // Send command to two motor brick
    RFduinoGZLL.sendToDevice(DEVICE1, cmd);
  }
  if (device == DEVICE0) { // Receive current state from touch sensor, set var
    tch = *data;
  }
  if (device == DEVICE2) { // Receive value from light sensor, set var, send LED on/off command
    lts = *data;
    RFduinoGZLL.sendToDevice(DEVICE2, cmd);
  }
  if (device == DEVICE3) { // Send command to single motor brick
    RFduinoGZLL.sendToDevice(DEVICE3, cmd);
  }
```

Figure 6a (left): Code for the single motor brick, receiving a command, then setting direction and speed of the motor

Figure 6b (right): Host RFduino code, sending the commands written in serial, and parsing the sensor reading

The host of the system is controlled entirely by writing, reading, and processing commands using serial communication through the USB port. This meant that once the code had been uploaded from the Arduino IDE, the host could be controlled using any language through serial communication, including LabVIEW. The host was coded to send whatever command it read from serial communication to the two motors bricks and the light sensor, and to print the sensor values to serial based on the command being read from serial.

### 3.6 LabVIEW Code and User Interface

Using NI-VISA in LabVIEW to access serial communication through the USB port, commands could be written in a customizable order and the sensor values that were printed each time sensor commands were written could be processed within LabVIEW (Figure 7a). Users were able to place specific commands by dragging and dropping the graphic blocks in Figure 7b into a specific order and modify things like the direction and speed of the motors, and ambient/reflected or light/dark for the light sensor as shown in Figure 7c.

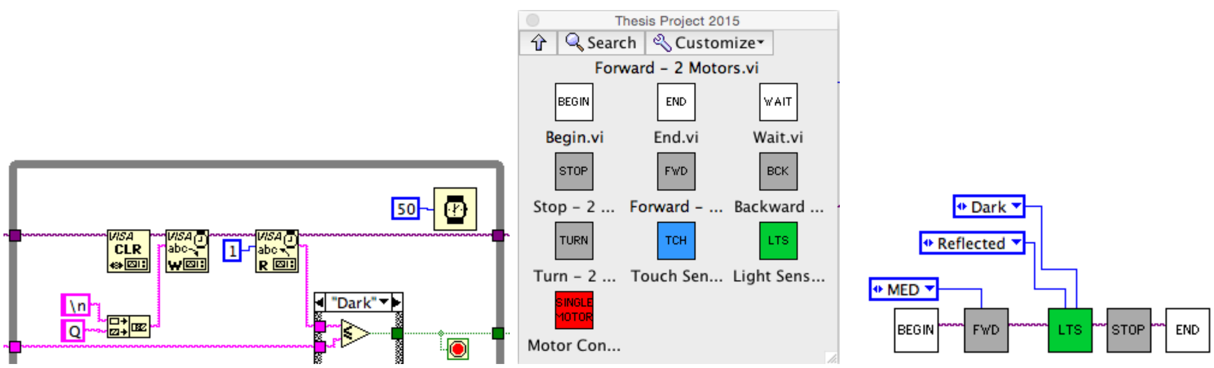



Figure 7a (left): LabVIEW code reading and writing to serial and processing data from the light sensor.

Figure 7b (center): The pallette of blocks available to the user to program the HiveMind platform.

Figure 7c (right): A code telling the motor to go forward at medium speed until it sees a dark surface, then stop.

Each of the blocks' colors was matched to the color of the brick it contorlled (Figures 8 & 9 below) to make it easier for the user to tell the story of what they want their robot to do as they troubleshoot their code. The wait block allowed users to set a delay in their code, and the turn block allowed users to control each individual motor's direction and speed on the two motor block. The red, single motor block allowed the user to stop and change the direction and speed. As it was explained to the participants at the beginning of the user study, both sensor blocks functioned as a “wait for” command, waiting for the touch sensor to be compressed, or light sensor to meet their specified parameters before going on.

### 3.7 Final Design of Actuator and Sensor Bricks

The motor and sensor bricks used baseplates as tops and bottoms, and were constructed mostly of studded LEGO Technic beams, allowing them to be compatible with both the traditional LEGO building bricks and the Technic pieces. Each of the bricks was color coded to match with the LabVIEW brick that controlled it. The grey, two motor brick (Figure 8a) had an eight by twelve peg baseplate on the top and bottom, and had a volume of roughly 240 $cm^3$. The motors were positioned towards the back of the brick to help the user know which direction the program would make the robot go, a common frustration among beginner programmers. The red, single motor brick (Figure 8b) was the only one of the four bricks to have a blueprint other than a box to allow space for the battery, with a volume of roughly 180 $cm^3$.

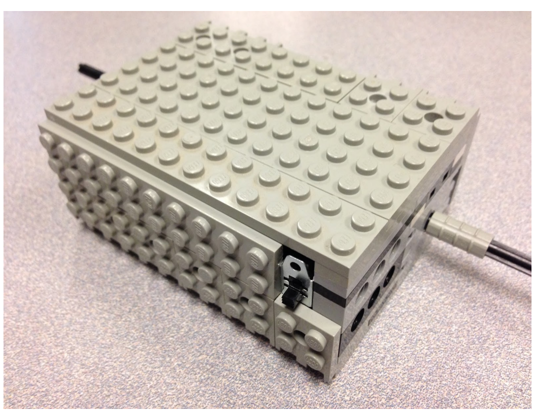
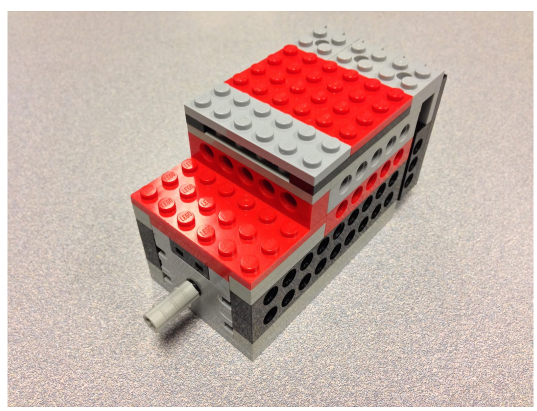



Figure 8a (left): Two motor brick has LEGO pegs on it’s top and front, and is peg compatible on its bottom and back

Figure 8b (right): The single motor brick has LEGO Technic compatible beams and has a peg compatible back

The sensor bricks both had a six by eight peg base plate area, and a volume of roughly 120 $cm^3$. The blue colored touch sensor (Figure 9a) left the NXT touch sensor that was wired to the RFduino sticking out of the brick, allowing the user to connect it to their robot in any orientation and attach things to the front using a Technic axle. For the green colored light sensor (Figure 9b), the LED and photoresistor were both placed in holes that were drilled out of a baseplate at the bottom of the light sensor, allowing the brick to be easily connected in a way that allows it to read the relative brightness of a surface.

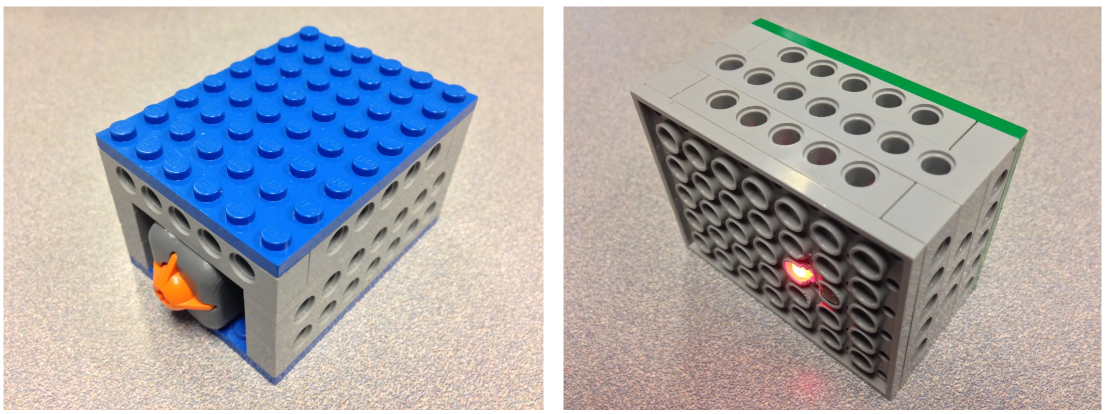


Figure 9a (left): The touch sensor brick is Technic and peg compatible, with the NXT sensor protruding from the brick

Figure 9b (right): The bottom of the light sensor, with the LED turned on next to the photoresistor one hole to the right

**3.8 Review of HiveMind**

Overall, the platform functioned as designed and was capable of being used to complete both the maze and amusement park ride challenges. The motors were able to communicate reliably with the sensors and react almost instantly based on LabVIEW code running real time. The latency from the computer to the robot was coded to be 100 milliseconds after shorter delays caused some unreliability with the communication. This was on the upper end of the specifications, but still perceived to be nearly instant. The new user interface successfully enabled HiveMind to be used in the comparison study because it could be used to code the new platform in a similar way to how Mindstorms is coded in LabVIEW.

Using this simple communication scheme created a few limitations on the system, but most of the issues with the technology were due to the limited time and scope of the project. One of the major technological limitations (discussed in detail in Section 4.2) was caused by a lack of reliability with the two motor brick. One of the two motors would start going full speed at seemingly random times when the commands were sent from LabVIEW as opposed to the Arduino serial monitor. It was difficult to quantify this behavior because it would happen for a short period and then act normally for longer stretches, but I would estimate the behavior happened roughly 5 percent of all the times code was run. This issue could not be resolved entirely, but any time this erratic behavior occurred during a user test, the investigator stepped in to help troubleshoot.

There were also some limitations with the functionality of the light sensor when used with the motors. The sensor was able to sense the difference between bright and dark surfaces very consistently, but it had to be within roughly two centimeters of the table and there was a very slight delay with the reaction. The electric Technic mini motors fit the specification for torque with 2.5 N-cm [13], but that ended up being too low to allow the two motor brick to move at a slow, constant speed. Due to the relatively high minimum speed, the light sensor had difficulty recognizing lines when driving.

A few of the constraints caused there to be a limitation on the complexity of the robots, but did not impact the functionality of the platform. The main such issue caused by a lack of time was the size of the bricks. By designing integrated circuits for all of the components, the bricks could have been made much smaller and the components could have been included in robots in more creative ways. Another issue was that the coding blocks could not be used with loops in LabVIEW due to the necessary sequential write-read process set up by the communication scheme used to be compatible with LabVIEW. The RFduino and communications network created a couple of minor limitations to the platform's functionality and complexity. It was tested to have a range of up to 120 feet of reliable communication across an open space and the GZLL communication could only channel hop reliably between up to 7 channels [17]. These limitations have huge implications on the scalability of a platform of this nature, but the initial prototype was able to be fully functional.

# 4. Results

### 4.1 Overview of Results

There was a wide range of feedback received from the participants about a comparison of the benefits and limitations of each platform (Appendix B). It was difficult to detect many reliable trends in the use case for the technologies or feedback across all of the participants. Each student interacted with each platform in their own way and liked or didn't like certain aspects of them, an idea supporting the goal of achieving a diversity of solutions. Despite these differences, one observed trend throughout the study was that many similarities could be found between each user's method of troubleshooting across the two platforms.

One interesting finding from this small case study of users was that five of the six users responded that HiveMind was easier to work with, although it could be argued that these results are biased because participants could have felt pressure to choose that answer knowing the developer was conducting the testing. Otherwise, there was no clearly discernable difference between the two challenges or the two technologies from the quantifiable data, responses to difficulty to complete the challenge and amount of success the participants felt they had as shown in Table 2. Even comparing the results for each participant, there was a mixture of responses about which challenge was more difficult and which one they had more success with.

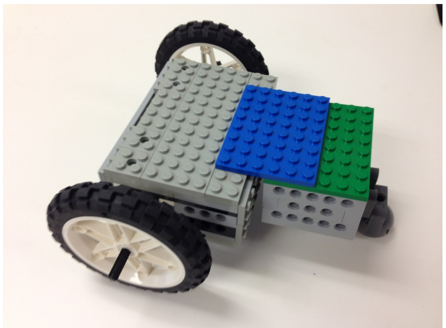
| Question | Maze | Ride |
|---|---|---|
| How challenging was it to complete the activity with the technology you were using? (5=”Very easy to complete”, 1=”Very difficult to work with”) | 3, 4, 3, 4, 4, 3 | 4, 4, 4, 3, 5, 3 |
| How successful was your design at completing the challenge? (5=”Total Success”, 1=”Total Failure”) | 4, 4, 3, 5, 5, 3 | 4, 4, 3, 3, 5, 2 |

Table 2: Distribution of responses to the quantifiable questions in the survey. Blue=Challenge completed with HiveMind Red=Challenge completed with NXT

**4.2 Case Study One**

The first of the two case studies was chosen as an example of a beginner user who the HiveMind system was designed to help the most. This participant had the least experience with Mindstorms of the group (responded two out of five), and had never used the LEGO Technic building platform before, responding “I've only used LEGO robotics through testing the vis [referring to LabVIEW], but have never built anything with them.”

To begin with, this participant was randomly given HiveMind and randomly selected the complete the maze challenge. The user began by attempting to connect the light sensor brick to the two motor brick with Technic pegs, before moving on and connecting the bricks with traditional LEGO building bricks. After difficulty remembering which direction was forward for the motor brick, the participant ended with the design shown in Figure 10a.



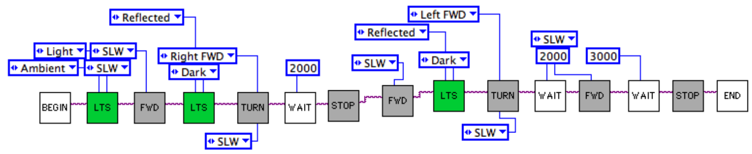



Figure 10a (left): The user ended with the light sensor connected to the two motor brick with a blue baseplate, and the EV3 rolling caster ball connected to the front. Figure 10b (right): Sequential code created to complete the task

Despite the relatively simple final design, it took the user many iterations before arriving at it due to the preconceived notion that Technic pieces had to be used and difficulty with figuring out what direction was forward. The participant created a new VI in LabVIEW to help troubleshooting by making a simple code and holding up the bricks to observe their reactions. This trial was the one mentioned earlier in which HiveMind exhibited the technical glitch with the two motor brick with roughly five minutes left in the forty-five minute challenge time limit. When asked to explain their criteria for success, the participant responded “I was able to build a design and code it, but the were software and hardware issues that made testing difficult. Sometimes, both of the motors would turn on and sometimes they didn't. Otherwise, the design seemed sound and so did the code (Figure 10b).” In response to the a question asking about the constraints of the technology, this user said, “It was very easy and quick to build with and coding was also very straight forward. All the pieces worked well together.”

For the second half, the participant attempted the amusement park ride challenge with the NXT platform. Similar to the first challenge, the user began by struggling to connect components with Technic pieces, building with the NXT intelligent brick as a base from the start. The same general use pattern of extensive building and unbuilding was observed as the design seen in Figure 11 was created by the end of the forty-five minutes. Just as the user had finally gotten the motor attached to the NXT brick in a way that fit her design, she had to rebuild a bit to allow room for the wire to be connected to the bottom of the motor. In response to a question about the constraints of the technology, she responded “There was one problem with my placement of the motor since I had not plugged in the motor and it's position made it impossible to do so, so i had to reposition it. Also using the brick can be cumbersome since you need to design around it.”

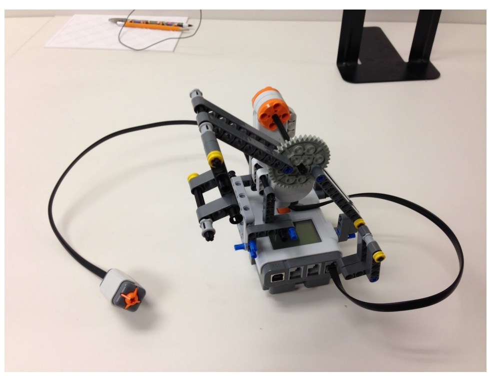


Figure 11: The ferris wheel designed by the user after forty-five minutes

At the end of the time limit, the user was still working on the connections between the seats of the ferris wheel and the beams, not leaving any time to code the robot to move. When asked about her criteria for success in this activity, the participant responded “I was very unfamiliar with the lego pieces that were provided, and only had prior experience with standard

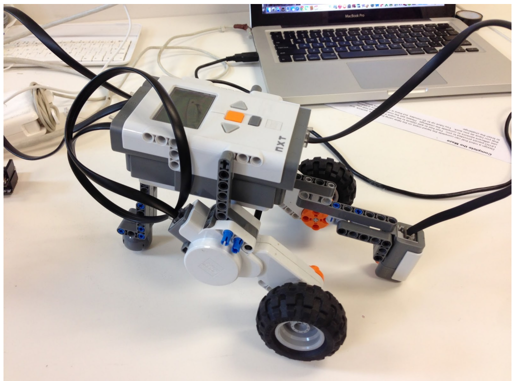
LEGOs. This made it difficult for me when I started building and in the end I completed a very rudimentary design, but didn't have time to code, which is the part I'm the most comfortable.”

In the comparison section of the survey, when asked about which technology would be prefered to use for which challenge, the participant said “I think it's easier to use the Hive Mind since it was a mixture of traditional LEGOs and the robotics LEGOs, which in the end made it easier for me,”. When asked to make a general comparison of the technologies, the user responded “I like how the HiveMind has no wires and no brick, and you can just focus on the sensors and motors that you are using.”

**4.3 Case Study Two**

This case study was selected due to the participant’s higher level of experience with LEGO Mindstorms, having used them for “line followers, block sorters, stair climbers, remote controlled vehicles, examples of advanced control systems.” This trial also posed each challenge with the opposite technologies used in case study one, hopefully allowing insight into the use of different aspects of each technology.

This participant began working with the LEGO NXT on the maze challenge. The user clearly had a plan in mind as he began building, connecting the motors to the brick in a sturdy and controllable way very easily. On the NXT brick itself, the user used a monitor to check the different values the light sensor was reading over the white and black sections of the table. They approached the problem by trying to make the robot follow the line of the maze, rather than in a sequential way as other participants did. The design seen in Figure 12 was created, but had some trouble getting around turns, so the user modified the maze as allowed in the rules (Figure 1b).



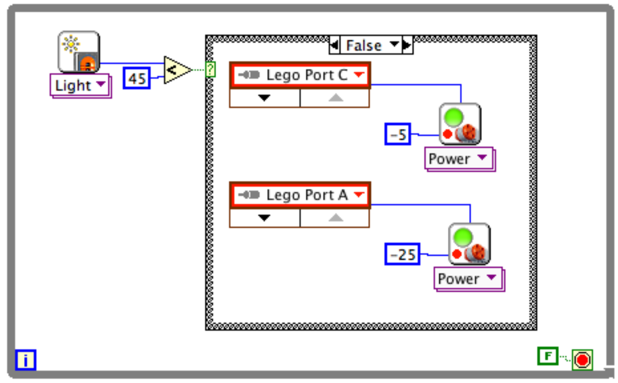
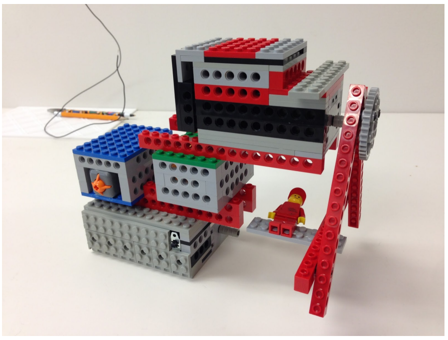



Figure 12a (left): The design used to complete the maze challenge, with the light sensor in front pointing at the table.

Figure 12b (right): The code written to make the NXT follow the line of the maze with a case statement and while loop

When asked if the technology constrained him in any way, the user said, “Not really. I could have reworked the robot to have a smaller turning radius but didn't think id have time to trial and error that…” continuing to list a few other ways that he could have solved the challenge better. This user used a more complex programming scheme to complete the challenge, and certainly had a good deal of experience troubleshooting issues with the Mindstorms platform as slight modifications were made to the robot to change the turning radius.

The next challenge was to create an amusement park ride with HiveMind. The user started by creating a simple program that made the two motor brick start and stop when a touch was sensed, and held the blocks up to test it out. The single motor brick was then chosen to create a swinging boat type ride as seen in Figure 13a. As discussed in the limitations section, the user said that “the motors weren't that powerful so trial and error adjusting the weights they carried.” The user responded, “I wish it used the technic/mind storms parts instead of traditional” when asked about constraints of the technology.



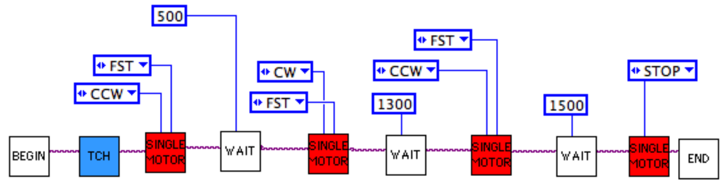



Figure 13a (left): Swinging ride design, using the single motor brick to make a lego platform spin around.

Figure 13b (right): The values of the wait commands were tuned to create the desired motion to run on a touch.

When asked about the benefits and limiting factors of HiveMind, the user said “It was very cool to not have to deal with wires. That's one of the first things I debug with student is whether or not they have the right ports selected or things plugged in.” They continued “They are very different but I see huge potential in hivemind,” in response to being prompted to make a general comparison between the platforms.

## 5. Discussion

### 5.1 Analysis and Comparison of Case Studies

In order to achieve some amount of success for either of the intro to robotics challenges used in the study, users needed to build an initial robot, program it, and iterate on the design, learning from how their robot performed during testing. Because the limiting factor for success in case one was the ability to build an initial robot design in the allotted time, the best guess at the cause for the differences in success across the two case studies was the lack of building experience of the user in case study one. Both users had some general robotics experience, but the participant in case study one had never built with the LEGO Technic platform.

User one's test showed an interesting example of how HiveMind could work to solve the issues laid out in Section 1.3. The inability of the user to even look at the coding aspect of the amusement park ride activity in the modeled class time displays an issue regularly faced by teachers in STEM due to a lack of time allocated to the activities. In this case, the new platform enabled the user to spend more time coding and redesigning their robot, a process that gives students experience with programming and problem solving lessons.

The second case study showed how the NXT could be used quickly and easily once it had been mastered and shows the complexity ceiling allowed by HiveMind with the current system. To complete the maze task, user two was able to use higher level programming concepts and pushed themselves to complete the challenge in a more complex way than others. Despite the limitations to HiveMind mentioned by the user, they still "would have been interested in trying the hivemind on the path follower but would have needed loop capabilities."

Differences in the complexity of each of the robots used in the case studies was to be one of the most interesting observations. The robot used to complete the maze in Figure 10a only used a couple of pieces in addition to the motor and sensor bricks after failing to create a more complex robot, whereas the NXT robot in Figure 12a had many complex connections, but both users felt they achieved success in the maze challenge. For the amusement park ride, user two had trouble adding complexity when using HiveMind and finished the challenge in twenty minutes, while user one could not figure out a simple way to connect the motor to the NXT and worked the whole time on an initial design.

Based on the HiveMind codes presented in Figures 8a and 11b, it is seen that both users understood how to use the sensor and wait blocks to act as a sequential timing controller for the motion of the bricks in their program. It was difficult to do an analysis of the differences between the Mindstorms code due to user one not doing any coding during that challenge. Overall, many more similarities were noted between the building and troubleshooting styles of each participant across the platforms, than across users for each platform; user one would start putting pieces together right away until she encountered an issue whereas user two would plan out what he wanted and build to execute the design.

### 5.2 Supporting Evidence from Other Users

The two case studies chosen represented the least experienced, and one of the most experienced users with LEGO Robotics. The other trials offered insight into how moderately experienced users interacted with the technologies and generally supported what was found in the two highlighted case studies. Overall, the other participants gave feedback on the new platform similar to the responses from the two case studies. Despite the previously discussed technology limitations to HiveMind, all users were able to use the platform to complete the designated challenges in a way similar to the NXT. All users in the study were able to use HiveMind to go through the same trouble shooting and learning process that they go through with the NXT.

Two more solutions to the maze challenge can be seen in Figures 14a and 14b, and two more solutions to the amusement park ride challenge in Figures 14c and 14d. The NXT presented building challenges to all participants, which prompted some participants to be more creative with their designs, whereas it encouraged some users to create the most basic robot possible. This brings up the question of how constraints can be viewed in a positive way in the educational setting, simulating the actual engineering process more closely.

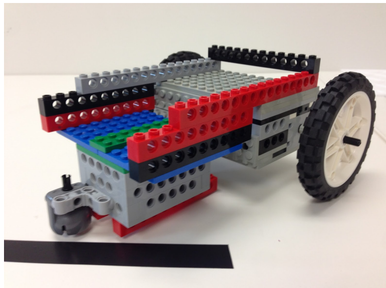
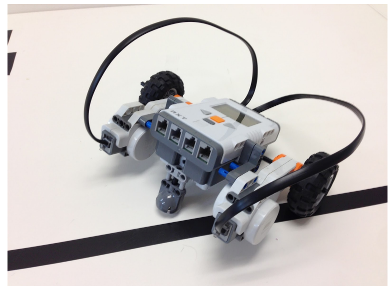
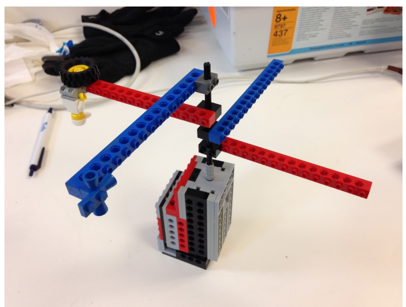
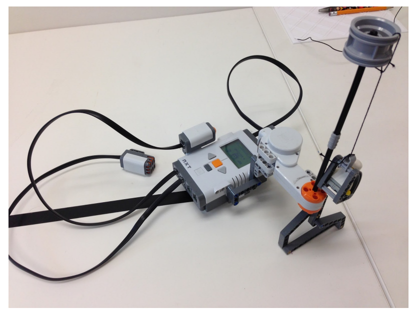





Figure 14a (top left): Design connects light sensor with two motor brick, trying to make light sensor more effective with brick on bottom. Figure 14b (top right): Design created to complete the maze challenge with just sequence and timing.
Figure 14c (bottom left): Amusement park ride designed with single motor placed on back, pointing up.
Figure 14d (bottom right): Similar ride design with a spinning, vertical axle, using a string make the seat swing out.

# 6. Conclusion

## 6.1 Conclusions

The HiveMind platform was developed using distributed intelligence and wireless communication as a solution to the use problems associated with the intelligent brick in the Mindstorms platform. Despite some limitations of the technology and scope of the study, the final system that was used in the pilot study was a fully functional prototype of an educational robotics platform that could be used in a project based learning environment. There was a latency of roughly 100 milliseconds from a computer input to actuation, and the communication system was entirely reliable with the exception of a glitch in the two motor block that occurred roughly one in twenty trials. With a technology successfully developed to fill the scope of the project, comparison user tests could be conducted to analyze how the platform with distributed intelligence is used.

One main conclusion about the use of HiveMind was that student users will attempt to complete educational challenges using the same engineering design process no matter what tool they are given to work. Some similarities were observed between how users built and problem solved suggesting that the different platforms encourage their own use pattern, but most of the consistencies in use were observed between each of the two challenges completed by the same user. This observation also supports the conclusion that a system using distributed intelligence could be used by higher level users for more complex challenges without limiting their design process.

One anticipated limitation with the wireless nature of HiveMind, that did not end up affecting any of the users, was confusion stemming from a lack of a physical connector between components (wire from intelligent brick to sensor or motor). Wireless technology is so prevalent that no users would think to question how the components communicate, and a simple program could be created to observe what happens while simply holding the components to understand the setup (as multiple users did).

Despite HiveMind still requiring a central brain (computer controlling host) to function, the new design successfully eliminated the need for the central intelligent brick while building, yet had a similar sequential programming environment in LavVIEW as the NXT. These factors also ended up serving as a limitation to the complexity of the robot, but a lot of the feedback on such limitations was associated with technology and scope aspects that could ultimately be solvable with another round of development.

### 6.2 Potential for Platform and Future Direction

This platform offers promise as a method of enabling a wider range of students to learn and benefit from robotics technologies as educational tools, but a user created robotics platform with wireless components would face a great deal of engineering challenges in further development. Some of the limitations of HiveMind such as size could be solved using the same RFduino technology, but something like a longer lasting and rechargeable power source for each brick would be much more difficult. In addition to this issue with batteries which could add a great deal of cost to the platform, the current system doesn't allow for the robot to be autonomous and run without a computer controlling it real time.

The next step would be to develop a new version of the platform that minimizes size of the sensor bricks to allow for more creative integration in the created robots, a programming scheme that allowed for a higher ceiling of capabilities, and a way to scale the system up to work with many different students working on their own project in the same class at the same time.

# References

[1] "Service Robot Statistics." IFR RSS. Web. 7 May 2015. <http://www.ifr.org/service-robots/statistics/>.

[2] Carbonaro, Mike, Marion Rex, and Joan Chambers. "IMEJ Article - Using LEGO Robotics in a Project-Based Learning Environment." *The Interactive Multimedia Electronic Journal of Computer-Enhanced Learning. Volume 6, Number 1*. 2004

[3] Papert, Seymour. "A Critique of Technocentrism in Thinking About the School of the Future." *M.I.T. Media Lab Epistemology and Learning Memo No. 2* September 1990.

[4] Park, Ill-Woo & Kim, Jin-Oh. "Philosophy and Strategy of Minimalism-based User Created Robots (UCRs) for Educational Robotics- Education, Technology and Business Viewpoint." *International Journal of Robots, Education and Art. Volume 1, Number 1* February 2011.

[5] Gaudiello, Ilaria; Zibetti, Elisabetta; Carrignon, Simon. "Representations to Go: Learning Robotics, Learning by Robotics." *Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots* November 15-16, 2010: 484-493.

[6] Alimisis, Dimitris. "Educational robotics: Open Questions and New Challenges." *Themes in Science & Technology Education, 6(1)* 2013: 63-71

[7] Blancharda, Samuel; Freimana, Viktor; Lirrete-Pitrea, Nicole. "Strategies Used by Elementary Schoolchildren Solving Robotics-Based Complex Tasks: Innovative Potential of Technology." *Procedia Social and Behavioral Sciences* 2010: 2851–2857

[8] Faisal, Akim; Kapila, Vikram; Iskander, Magued. "Using Robotics to Promote Learning in Elementary Grades." *ASEE Annual Conference* 2012: 5480

[9] Panadero, Carmen; Román, Julio; Kloos, Carlos . "Impact of Learning Experiences Using LEGO Mindstorms® in Engineering Courses." *IEEE EDUCON Education Engineering 2010- The Future of Global Learning Engineering Education* April 2010: 503-512.

[10] Hynes, Morgan. “LEGO Robotics in the Classroom: An Ethnographic Study of Implementing the Technology into the Classroom.” *Mid-term paper CD 145* April 6, 2005.

[11] Benitti, Fabiane. “Exploring the educational potential of robotics in schools: A systematic review.” *Computers and Education 58* 2012: 978-988.

[12] "Tufts STOMP | Student Teacher Outreach Mentorship Program." *Tufts STOMP*. Web. 22 Apr. 2015. <http://sites.tufts.edu/stomp>.

[13] "LEGO 9V Technic Motors Compared Characteristics." *LEGO 9V Technic Motors Compared Characteristics*. Web. 22 Apr. 2015. <http://www.philohome.com/motors/motorcomp.htm>.

[14] Card, S. K., Robertson, G. G., and Mackinlay, J. D. (1991). “The information visualizer: An information workspace”. *Proc. ACM CHI'91 Conf.* (New Orleans, LA, 28 April-2 May), 181-188.

[15] Duracell Batteries. “Alkaline-Manganese Dioxide Battery” MN9VCTUS0413 datasheet. Web. 22 Apr. 2015

[16] *RF Digital*. Web. 22 Apr. 2015. <http://www.rfduino.com/>.

[17] "Introduction." *Gazell Link Layer User Guide*. Web. 22 Apr. 2015. <https://devzone.nordicsemi.com/documentation/nrf51/4.3.0/html/group__gzll__02__user__guide.html>.