ROBOTICS
ASSISTIVE DESIGN for the FUTURE

Spring 2007 Edition

Curriculum Resources

Contributors: Morgan Hynes, Tufts University
Haruna Tada, TechBoston
Northeastern University

The preparation of this curriculum was partially supported by grant DRL-0423059 from the National Science Foundation. Grantees undertaking such projects are encouraged to express freely their professional judgment. This curriculum, therefore, does not necessarily represent the position or policies of the National Science Foundation.

Not for distribution without the authors’ permission

Copyright 2008 Tufts University Center for Engineering Educational Outreach and LEGOengineering.com
Table of Contents for this Resource Packet

Section 1. Teacher's Guide

Section 2. Student Handouts and Homework for All Lessons

Section 3. Supplemental Teacher Resources
Section 1:
Teacher’s Guide
# Table of Contents

- Introduction ................................................................. 2
- Lesson 0: Introduction to Engineering Design Process ............ 7
- Lesson 1: Wheelchair Design Challenge .......................... 14
- Lesson 2: Selecting a Solution ......................................... 19
- Lesson 2.5: More About Gears (OPTIONAL) ....................... 27
- Lesson 3: Programming with ROBOLAB ......................... 31
- Lesson 4: Programming with Sensors ............................. 37
- Lesson 5: Begin Final Project ........................................... 45
  Identify Need, Research Problem, and Develop Possible Solutions
- Lesson 6: Continue Final Project ...................................... 49
  Select the Best Possible Solution, Begin Constructing Prototype
- Lesson 7: Continue Final Project ...................................... 53
  Finish Constructing Prototype, Test and Evaluate Solution
- Lesson 8: Communicating the Solution ............................ 56
- Lesson 9: Complete Final Project ..................................... 60
- Lesson 10: Final Presentation ........................................... 63
Introduction

The Robotics: Fundamentals of Information Technology and Engineering program is designed to excite middle-school students about engineering topics, and integrate engineering and technology into their curriculum.

The curriculum engages students in designing, building and programming a LEGO robot, challenging them to work cooperatively and think creatively. The students form teams or “companies” that work together to create a LEGO robot prototype. The program culminates in a final presentation where student teams present their products to a panel of judges. Project themes involve assistive devices for the physically challenged population.

THE ROBOTICS METHOD

Our overall goal in the course is to provide teachers with materials to help them engage their middle-school students in thinking about engineering problems, and to integrate engineering and technology into after-school programs and in-school curricula.

This curriculum addresses a number of the Massachusetts State Curriculum Frameworks for Technology/Engineering in Grades 6-8. Students are required to pursue engineering questions and technological solutions that emphasize research and problem solving as a way of meeting their goals. Throughout, this program uses the Engineering Design Process (EDP) as a basic organizing principle. This general process, which can be seen as analogous to the scientific method for understanding and carrying out scientific studies, is a logical, step-wise approach to solving a problem. The material used in this program involves having students actively learn and successfully apply the EDP, and therefore become better problem-solvers.

The basic pedagogy for the course is the 5E Learning Cycle: Engage, Explore, Explain, Elaborate and Evaluate. Through this cycle, students will learn through active engagement with the material, exploration and problem solving. In each meeting the students will work through this cycle by taking on various “challenges.” Teacher-led questioning and discussions to introduce the lesson’s material and the challenge presented at each lesson will provide motivation for the students and will carry them into active learning. The groups will work cooperatively and creatively to solve the problem of the challenge, and discussions and presentations will allow students to voice their ideas and practice communication skills. Whole group discussions and analysis of output encourage students to reflect on their work and evaluate it. This approach is compatible with recent emphases on active learning and constructivist philosophical perspectives surrounding the understanding of science and technology.

The teacher should use a variety of instructional approaches during each lesson: short lecture and demonstration (to introduce topics and engage students, to wrap-up lesson), discussion and questioning (identify and connect with students’ prior knowledge; share ideas and solutions), and challenge activities (allowing students to explore ideas with LEGOS, elaborate on their learning by extending to new problems, perform research, present their ideas and
work, and evaluate their progress). Using a variety of instructional methods and working in various grouping structures (e.g. whole-group, pairs, teams) helps to maintain motivation, and makes the content accessible to students with varied learning styles.

The 5E Learning Cycle

<table>
<thead>
<tr>
<th>Learning Cycle Phase</th>
<th>Goal</th>
<th>Instructional Methods and Examples</th>
</tr>
</thead>
</table>
| Engage               | Motivate students, capture their interest, elicit previous knowledge | • Introductory discussion  
                      |                                                                      | • Questioning about prior experience with assistive devices   |
|                      |                                                                      | • Exciting demonstrations                                            |
|                      |                                                                      | • Posing the lesson’s challenge                                      |
| Explore              | Students engage in inquiry, discover new ideas, generate and test hypotheses | • Challenge activity  
                      |                                                                      | • Researching problem                                               |
|                      |                                                                      | • Working with LEGO to build prototypes                              |
| Explain              | Students explain their results, defend their decisions, practice communication skills | • Presentations  
                      |                                                                      | • Teacher-led discussions                                           |
|                      |                                                                      | • Questioning by teacher and other students                           |
| Elaborate            | Students transfer new skills and ideas to different problem         | • Re-design challenge  
                      |                                                                      | • Using new skills and knowledge to address the final assistive device project |
| Evaluate             | Students reflect on their project and their learning                | • Team discussions  
                      |                                                                      | • Team re-design                                                    |

Each class meeting moves through the 5E cycle, typically beginning with a short lecture/demonstration and/or a discussion to introduce the day’s focus on the EDP. At this point the teacher can also review and connect with the previous meeting. These activities act to engage the students, and assist them in getting ready to proceed through the lesson. The class generally moves on to a challenge, where teams of students explore the day’s topic, and apply what they know to various problems. The classes end with a wrap-up session where students explain what they’ve learned and evaluate where they are in the EDP process.

Throughout this course students will also be working on literacy and communication skills. There are many occasions on which students can be asked to write about their work, giving them practice reflecting about their learning as well as practice writing. In many lessons student small groups report back to the entire class; these reports offer opportunities for students to practice their communication skills to lead up to the final presentation.

The teacher should make it clear that taking advantage of these opportunities will be helpful to the students during their final presentation for this course, but also in the future. By rotating student roles on a regular basis, each student should be provided with multiple chances of practicing presentations. Teacher should also provide feedback to students after informal presentations to help students improve their presentation skills prior to the final presentation.
STUDENTS WORKING IN COLLABORATIVE GROUPS

Working collaboratively is essential to the success of students in this program—as it is for people in many employment environments today. Working with the students early in the program to build a learning community is critical. The first lesson provides teachers the opportunity to develop classroom rules with the students, set standards for teamwork, model appropriate discussion behavior, and set the tone for the rest of the program.

Forming teams

Teachers will help students form their teams ("corporations"). Teachers can adjust the composition of teams depending upon the number of students in the program and the number of kits available. Teachers should assign groups based on their knowledge of the students. In many cases, heterogeneous groups will be most suitable, but for classrooms with students varying greatly in ability, some consideration of grouping by level may be appropriate. Ideally the teams will consist of 2-4 students, and we suggest that each teacher have a maximum of 6 teams. In programs or classes with more students, the activities may require more than one teacher (or trained teacher aid) in the classroom.

Team assignments should remain the same once the project begins (Lesson 5); however, students can change teams until that time. We recommend that teachers experiment with different team assignments in the first few weeks to see what works best prior to making permanent team assignments at or before Lesson 5.

In order for all students to have the opportunity to learn, they must be able to fully participate in all team activities. Depending upon their prior experience and knowledge, temperament, and learning styles, some students may be more comfortable with the material, be quicker to understand the concepts, and potentially dominate the team. It is critical that teachers monitor teams and make sure that no team member is behaving in ways that exclude others; instead, modeling of peer learning and teaching is important.

Assigning roles to team members

To ensure that each member of team contributes, it may be necessary to assign roles to team members. In one example, teacher can assign easily remembered labels to each student in the teams (e.g. number or color), and then systematically assign tasks or roles to the various members. In one challenge, student #1 in each team might be responsible for getting and returning materials, and student #2 for guiding the LEGO building. On the next challenge those roles can be switched, so that each child has the opportunity to participate, and be responsible for, each part of the process. Specific suggestions for facilitating teamwork are included with some of the lessons.

When programming, it is very easy for one student to take control of the keyboard/mouse. One strategy to make sure everyone gets a chance to program is to have students switch seats every five to ten minutes to make sure each student gets time in front of the keyboard. Alternatively, icon printouts can be used away from the computer so that the students can design the program as a team before entering it into the computer.

ASSESSMENT

Although this curriculum was initially developed as an after-school program, a number of worksheets and homework have been added in this version of the curriculum to support in-class implementation. These assignments and worksheets can also be used within the after-school setting if time permits. There is also an evaluation rubric provided for the project and final presentation.
TIMING

Each lesson takes between 1 and 2 hours to accomplish. Guidance is provided to split lessons designed to run for about one hour. However, some student groups will need more time to complete the challenges, and some less. The curriculum is designed to allow teachers to differentiate their teaching in various ways. Suggested extensions are given at the end of each lesson to provide enrichment opportunities for teams who complete the tasks early or for teachers to adjust the timing of the lessons. Teams that require more time and help will be assisted in choosing appropriate challenges that will allow them to be successful.

The lessons in the curriculum are sequential, so teachers can go as far as is meaningful for their particular class.

MATERIALS

- **LEGO Robotics kits:** You will need one kit for each group of 2 to 4 students you have in the program. The recommended kit is the ROBO Technology Set, supplemented by Resource Set (about 1 Resource Set for every 3-4 ROBO Technology Sets); however, other types of kits such as Team Challenge Set may also be used. Kits must contain RCX, motors, and sensors, as well as pieces for building such as beams, gears, and axles.

- **Computers:** You will need access to at least one computer per each group of 2 to 4 students. Computers with Internet access are preferred for the research portion of the program. A USB port is needed for the Infrared Towers included with your kits.

- **ROBOLAB software:** ROBOLAB V2.5.4 (or above) must be installed on each computer the students will be using.

- **Microsoft PowerPoint** (or other presentation software): Presentation software is needed for the final presentation.

- **Locked/safe storage:** The LEGO kits are expensive and should be stored in a locked closet or safe storage area. You will also want boxes or bins to store the students’ works-in-progress.

- **Batteries:** Each RCX takes six AA batteries. Depending on the level of use, batteries may need to be replaced every few weeks. Always have a supply of fresh batteries ready in case the battery runs low. Turning the RCX off when not in use helps conserve battery power.

- **Other materials:** Other materials needed, such as tape or weights, are listed in each lesson plan.

- **Use of Non-LEGO materials in final project:** Teachers may choose to whether or not to allow students to use items other than LEGOS for their final project. If allowing other materials, students will be integrating LEGO motors and sensors with other materials such as construction paper, popsicle stick, etc. This allows for more creativity and flexibility in design. Have some craft supplies and tools available so that students can get more creative.

PREPARING FOR AN AFTER-SCHOOL PROGRAM

- **Reserve classroom space:** Reserve the required classroom space and computer lab/computers for each of your sessions. If possible, use a room with large tables. Avoid rooms with slanted desks, as small pieces tend to roll off.

- **Install ROBOLAB Software:** You or your computer administrator must install the software on each machine and connect the infrared tower. (Note: Administrator access is needed for installing ROBOLAB.)
• **Reserve a computer projector:** For a number of the lessons, a computer projector is helpful (although not mandatory) so that the students can all see what is being done on the screen. For the final presentation session, a projector is required.

• **Recruit the students:** You may want to create a flier and possibly do a presentation or demonstration so the students get a sense of what they will be doing. You may also want a previous participant to show their creation to other students. Some students need to be approached individually. Recruiting female and members of under-represented groups may be challenging and special attention should be made to attract and recruit under-represented students.

• **Permissions:** You will need to communicate with your principal/headmaster to be sure you are following school guidelines in running an after-school program and are getting parent or guardian permission for student participation. Be sure to ask on the permission slip how the students are getting home after the sessions.

• **Recruit a panel of judges:** The final session is designed for the students to deliver a presentation of their product to a panel of professionals. You should begin recruiting this panel of judges ahead of time and set a date for the final presentations. The judges could be engineers, marketing professionals, handicapped individuals, school officials, teachers, etc.

### LEGO KIT MANAGEMENT

Students must have adequate room to work with the materials in this program. A clear working space will allow teams to better monitor their LEGO parts; removing all other materials from the working space is important.

Each teacher will have to determine the best method for managing and controlling the LEGO kits they will be using. The LEGO kits are costly, and the success of the project depends on the kits being complete. The teacher should develop classroom routines for getting kits, stowing LEGO pieces, and returning kits, so that students are clear on what is required of them, and these tasks can be completed quickly and efficiently. Several suggestions for managing kits follow.

- Have students sweep the floor of the room before each session and then again at the end. The first sweeping will clean the floor, which will make it easier to find dropped pieces during the second sweeping.
- Kits should be numbered and checked out to a team only as they are needed. One of the rotating roles of a student in each group will be responsibility for checking out and returning the kit.
- Emphasize that they will need to have all of their pieces to complete the challenges.
- After early challenges (before teams begin final projects), leave some time for clean up and inspecting each kit. When teams have final projects in progress, have bucket for storage of each work-in-progress.

### DIFFERENTIATING TEACHING

Many of the characteristics of this program make it appropriate for students with special needs and English language learners; for example, the curriculum has many hands-on activities, varied instructional methods, chances for students to be physically active, and opportunities for students to work in small groups. Teachers can further assist students with various learning styles and abilities by regularly recording discussions on the board, appropriately grouping students, and providing enrichment activities or extra guidance when needed.
Lesson 0

Introduction to Engineering Design Process

OBJECTIVES

- Introduce the Robotics program to students.
- Establish classroom rules and routines for team work, kit management, etc.
- Gauge students’ prior knowledge.
- Explore different LEGO pieces and their functions, and how to make strong structures.
- Experience the Engineering Design Process.

BACKGROUND FOR TEACHERS

The Engineering Design Process (EDP) is a framework used in nearly all engineering problems and solutions. EDP is used in the development of new products, new processes, new systems, or in the optimization and improvement of existing products, processes and systems. While it may sound like a very technical process, the engineering design process is a common-sense approach to solving a problem. It is a process that, in some form or another, we have probably all used.

The Massachusetts Science and Technology/Engineering Frameworks describe the engineering design process as the 8-step process below:

1. Identify the Need or Problem
2. Research the Need or Problem
3. Develop Possible Solution(s)
4. Select the Best Possible Solution(s)
5. Construct a Prototype
6. Test and Evaluate the Solution(s)
7. Communicate the Solution(s)
8. Redesign

NOTE: Included in students and teachers materials will be a more detailed handout of the Engineering Design Process.

The steps of the Engineering Design Process help engineers produce functional, safe, reliable, competitive, usable, manufacturable, and marketable solutions.

TIMING

- Introductory discussion 10 minutes
- Develop classroom rules 10 minutes
- Prototyping and Testing 30 minutes
- Redesign and Testing (Optional) 15 minutes
- Engineering Design Process discussion 15 minutes
- Wrap-up and clean up 10 minutes

Total: 90 minutes
STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multiview drawings.

2.3 Describe and explain the purpose of a given prototype.

MATERIALS

- ROBO Technology Set (or Team Challenge Set) without RCX, motors, and sensors – 1 per team
- Discrete weights (e.g. coins, mass weights, washers, expended batteries, marbles, etc.)
- A small container to load weights. Container should be able to slide on LEGO plate.
- Ruler or tape measure.
- Team folder or binder to keep all handouts, etc. – 1 per team

Handouts & Homework:

- Spatula Challenge Handout – 1 per team
- Engineering Design Process Handout: 1 per student
- Engineering Design Process Homework: 1 or 2 per student

PREPARATION

- Have enough weights for each team to test their design. You may use any small discrete weight, such as standard mass weights, expended batteries, marbles, bolts, etc.
- Photocopy Spatula Challenge Handout (1 per team)
- Photocopy Engineering Design Process Handout and Homework (1 per student).
- Remove all electrical components – RCX, tower, motors, sensors, and cables – from the kits, and set aside. You will not be using these parts for this lesson.
- This lesson works better if the number of pieces are limited. If using Team Challenge Set, you may want to remove some pieces from the kit so that the set is similar to the ROBO Technology Sets (components in the ROBO Technology sets is shown in a separate sheet).

LESSON GUIDE

Introductory Discussion

“You are starting a project where you will be acting as engineers. You will be designing and building assistive devices.”

“Does anyone know what an assistive device is?”

Have students answer and give examples of assistive devices. Help the students to think outside of the box, beyond wheelchairs, crutches, walkers, etc (special jar openers, automatic
doors, grabber, drinking straws, hearing aids, eye glasses, etc).

“During these sessions you will have a number of engineering challenges. You will be using the Engineering Design Process throughout as you solve challenges and, ultimately, design and build your own assistive device prototype.”

“Can anyone tell me what a **prototype** is?”

Write students’ responses on the board. Key points to cover are:

- A prototype is a model of something.
- A prototype has one or more of the functionality of the final product.
- A prototype is not exactly what the final product will be.
- A prototype is made to demonstrate and test the design concept.
- A prototype is not always the same size as the final product.

Ask, “Why would a company create a prototype?”

Again, ask for students’ responses. Key points to cover are:

- to test their ideas before they invest in mass-producing the product.
- to make sure the design will work.
- to communicate to others the design and idea (for example, showing it to potential consumers to get feedback)
- to test for functionality.
- to perform durability or reliability tests (for example, a crash test)

“Today you will be constructing a prototype. Before you can build a prototype, what do you think you’ll need?”

Elicit from them that they will need a purpose, reason, need, or problem. This will be the Need or Problem (Step 1 from EDP).

“Exactly. The problem that you’ll help solve today is this: There are a number of people out there who are in wheelchairs and cannot easily cook at their stove. One of their biggest challenges is being able to reach their back burners with a spatula. Today your job is to design and build a spatula that is long enough to reach these back burners. You will be building the prototypes with the LEGO kit.”

“The spatula has to extend beyond the edge of the testing platform at least 12 inches, and it should **hold as much weight as possible**. Again this is a prototype to **test out your concept**.”

![Diagram of spatula setup](image)

**Scoring formula** = \((L - 11") \times (# \text{ batteries})\)

(batteries can be substituted with other discrete weights)
Pass out the Spatula Challenge handout. Explain the testing method and scoring formula to your students. To test, the spatula will be held on a desk or other support while a weight is placed into a small container placed at the end of the spatula. The spatula must be able to hold a small bin or container (use the 8×16 plate in the ROBO Technology Set at tip). Show the container to the students so that they know what their spatulas need to hold. Weights will be added until the spatula breaks or until the container slides off – whichever that happens first. Be sure to choose a container that will slide on a LEGO plate.

Discuss the implications of the scoring formula. There is a trade-off that is built into the formula – shorter spatulas tend to be more rigid and strong (therefore holding more weight); however, longer spatula will get you more points for the given number of batteries. Teams need to decide whether they want to maximize length, maximize strength/rigidity, or take the middle road.

Form teams of 2 or 3 (depending on #students and #kits). It may be helpful to make up the teams ahead of time. These teams will not be permanent until Lesson 5, so experiment with different team dynamics to see what works best.

**Develop classroom rules**

Before distributing the kits, it may be a good idea to discuss classroom rules and routines. It is up to you and your students to establish rules; however, some points to cover:

- Set up routines for getting and putting back kits. You may want to designate one student per team to be in charge of kits.
- You may want to establish tasks and roles – example: a scribe to be in charge of filling in team handouts, a material handler to be in charge of getting and returning materials, etc. If you establish roles, be sure to rotate roles each session so that each student experiences each role.
- Set up other classroom rules such as taking turns to speak, respecting classmates, etc.

Once the rules are established, you should write them down on the board or on a large piece of paper and post the rule in a visible location.

**Prototyping and testing**

After students have settled into their teams and gotten kits, give them about 15 minutes to build a prototype. Walk around the classroom and assist students in identifying and using pieces. Remind them that the spatula needs to be at least 12 inches long (plus some extra to hold), and to use the 8×16 plate (or any large plate) at the end to support the bin.

During this first lesson, keep a close eye on how construction is being shared in the teams. If one student seem to be dominating, tell the students to take turns building (allow students to add one new piece at a time, switching between each piece).

After 15 minutes, announce, "It’s now time to test your prototypes!"

Have the teams come up one or two at a time and test their designs (depending on # of weights available). To test the spatula, use the edge of a desk to support the spatula and then place a small bin on the end of the spatula. Measure the length from the table edge to the tip and record it on a score sheet. Gently place weights into the bin until the spatula breaks or the bin falls off from the bending. Record the number of batteries and calculate the score. Have a score keeper write down everyone’s score on the board.

As they test their designs, ask the students to comment what building techniques made their designs strong or weak. When you see something that held together particularly well, be sure to share that with the rest of the class. Have the teams complete the Spatula Challenge Handout.

**Redesign and testing (Optional)**

After every team has tested, give the class 5 more minutes to modify their design to improve
the strength. Re-test after five minutes and have each team talk about what improvements they made, and why.

**Engineering Design Process discussion**

Pass out the Engineering Design Process Handout (1 per student).

“This is called the Engineering Design Process. Why do you think engineers would use a process like this?”

Key Points:
- systematic procedures to create a product.
- make sure they make a high quality product.
- consider multiple ideas and concepts.
- get feedback often.

“Why do you think this process is a circle or cycle?”

Key points:
- to make the best product possible.
- iterate to improve (make improvements based on the feedback over and over again).
- because products are rarely perfect and there is always room for improvement.

“Let’s go through each step of the Engineering Design Process and see if you followed these steps when making your prototype.”

Go through each step with the students and have them discuss whether they did each step and how. See the example below if necessary.

- **Step 1**: Identify the Need or Problem (given at the beginning of the challenge).
- **Step 2**: Research the Need or Problem (given at the beginning of the challenge).
- **Step 3**: Develop Possible Solution(s) — think about and discuss possible designs in light of what is in the LEGO kit and the objective set for the spatula.
- **Step 4**: Select the Best Possible Solution(s) — pick one of the discussed designs to build.
- **Step 5**: Construct a Prototype — build the spatula.
- **Step 6**: Test and Evaluate the Solution(s) — test how much weight the spatula could hold.
- **Step 7**: Communicate the Solution(s) — each team will present its results.
- **Step 8**: Redesign — improving upon their existing design given the results of their tests.

“Just like you followed the Engineering Design Process to design and prototype your spatula today, engineers follow this process when designing ANY product. For example, let’s take a look at this chair (or any other object that’s in the classroom)”

As an exercise with the whole class, go through the engineering design process for an object. If you wish, you can hand out a blank Engineering Design Process form and have the students fill out each step during the discussion. The students will be doing the same thing as a homework.

An example for a simple classroom chair is shown below:

**Step 1**: Identify the Need or Problem:
- We all need to sit down.
- Existing designs did not meet some requirement or company saw an opportunity for
Step 2: Research the Need or Problem

- Look at existing designs and how they are made, how much they sell for, etc.
- Talk to consumers and find out what type of chairs schools are looking for.
- Come up with design constraints, such as chair height and desired features, cost to manufacture (related to price and profit), safety requirements, features such as stackability, etc.

Step 3: Develop Possible Solution(s)

- Brainstorm different chair designs – different design (shape and features), material (plastic vs. wood vs. fabric), color, assembly methods (welded vs. screws, etc)...

Step 4: Select the Best Possible Solution(s)

- Choose the best solution based on market needs, cost to manufacture and price (profit), target consumer, etc. For example, if trying to make an inexpensive chair to market to schools, you might choose a low-cost option like a molded plastic design.

Step 5: Construct a Prototype

- Construct a model before going into mass-production.
- Molded plastic shapes are often prototyped in something else first, because molds are expensive to make and is not economical to invest in a mold until you are ready to mass-produce the product.
- Depending on the chair design, you may need prototypes of certain parts of it... for example, prototype of the height-adjustment mechanism before it gets installed in all chairs.

Step 6: Test and Evaluate the Solution(s)

- Test durability and strength by putting it under repetitive stress.
- Sit on it to test for comfort.
- Test all design features... for example, if the chair was designed to roll, then roll it to make sure it moves smoothly. If it was made to stack, then it should stack well without falling over or getting stuck.
- Show to consumers to get feedback about comfort, looks, etc.

Step 7: Communicate the Solution(s)

- Brochure/flyer to market the chair to stores and consumers.
- User manual and/or assembly instructions for consumers.
- Manufacturing instructions (drawings, etc) for manufacturers.

Step 8: Redesign

- Get feedback from users to make design improvements in the future.
- Redesign to adapt to new material or technology – for example, if a new, cheaper and stronger plastic was developed, you might want to redesign your chair to use the new material.

“You will each get a blank Engineering Design Process worksheet. At home, pick a simple product – like a pencil or pen, or a hairbrush – and write down what the designer of the product was thinking as he/she went through the engineering design process.”

Alternatively, you can have students complete the worksheet in class. Have each team pick a different object (examples: pen, cell phone, backpack, desk, etc) and complete a
worksheet. Report back to share their ideas with the rest of the class.

**Wrap-up and clean up**

Ask your students to take apart their spatulas and put all pieces back into the kits. Reinforce the classroom rules that were established earlier about kit management. Establish a routine of checking kit content before students leave.

**EXTENSIONS**

- Spend some time at the next session discussing the students’ work on the Engineering Design Process homework.

**HOMEWORK**

- Engineering Design Process Homework

**WORDS TO KNOW**

- **Assistive device** – A device or technology that is designed to improve the quality of lives of the disabled (or any person, if defined loosely).
- **Engineering Design Process** (EDP) – The cyclical process used by engineers when developing a new product or invention.
- **Prototype** – A model of a solution that demonstrates one or more of the design features.

**EXTRA RESOURCES**

- **Engineering Design Process Poster** is a larger-format, color version of the Engineering Design Process handout that can be printed and used as a poster in the classroom.
Lesson 1
Wheelchair Design Challenge

OBJECTIVES

• Begin focusing on wheelchair as an assistive device.
• Learn about building strong LEGO structures. Explore different LEGO pieces and their functions.
• Learn to assess failure and to learn from failure.
• Experience the Engineering Design Process.

BACKGROUND FOR TEACHERS

Engineers design products to meet all kinds of criteria. The product may need to be durable, cost-effective, safe, user-friendly, etc. Engineers also consider whether the product is environmentally friendly, attractive and eye-catching, easy to ship, able to withstand the possible environments it will be exposed to, and an appropriate weight. With all these constraints in mind, the engineer will then research the available materials, designs, and processes, brainstorm possible solutions, choose the best solution and make prototypes of the solution.

The engineers then test their prototypes. Many companies have instruments that simulate the possible environments and stresses a product might encounter in shipping, normal use, extreme use, earthquake, etc.

Often, multiple prototypes are made of the same product to test features separately – for example, if designing an oven, the heating components may undergo its own prototype and testing for efficiency, while the door and outer casing undergo testing for strength, durability, and ease-of-use.

After testing, the engineer will make any necessary changes to the design and repeat the prototyping, testing, and redesigning steps until a satisfactory end product is produced.

This lesson uses a “drop test,” a very common test that looks at a product’s ability to withstand a drop or fall. Products such as computers, cellphones, mp3 players, alarm clocks, suitcases, and toys are all put under the drop test to ensure that the product will remain intact and function after an accidental drop.

TIMING

• Introductory discussion 10 minutes
• Wheelchair design challenge 30 minutes
• Testing 15 minutes
• Redesign and testing (Optional) 10 minutes
• Discussion & Clean-up 15 minutes

Total: 80 minutes
STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

1.1* Given a design task, identify appropriate materials (e.g., wood, paper, plastic, aggregates, ceramics, metals, solvents, adhesives) based on specific properties and characteristics (e.g., strength, hardness, and flexibility).

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multiview drawings.

2.3 Describe and explain the purpose of a given prototype.

7.1 Explain examples of adaptive or assistive devices, e.g., prosthetic devices, wheelchairs, eyeglasses, grab bars, hearing aids, lifts, braces.

* - with the use of optional homework: Putting Things Together.

MATERIALS

- ROBO Technology Set (or Team Challenge Set) without RCX, motors, sensors, and tower – 1 per team
- Small object to be a “passenger” for the wheelchair. An 8 oz. water bottle or a small stuffed animal (size of Beanie Babies) work well, but any other object of similar size is acceptable.
- OPTIONAL – Box or bin at least 1’ x 1’ to catch the dropped wheelchairs.

Handouts & Homework:

- Wheelchair Design Challenge Handout: 1 per team
- OPTIONAL – Putting Things Together homework: 1 per student

PREPARATION

- Photocopy Wheelchair Design Challenge Handout (1 per team).
- Remove all electrical components – RCX, tower, motors, sensors, and cables – from the kits, and set aside. You will not be using these parts for this lesson.
- This lesson works better if the number of pieces are limited. If using Team Challenge Set, you may want to remove some pieces from the kit so that the set is similar to the ROBO Technology Sets (components in the ROBO Technology sets is shown in a separate sheet).
- Build a simple wheelchair out of LEGOs that will break apart when dropped.
- OPTIONAL – Photocopy Putting Things Together homework (1 per student).

LESSON GUIDE

Introductory discussion

Hold up a poorly designed LEGO wheelchair so that all students can see it, and drop it so that explodes in many pieces. Interact with any questions or comments students bring up. To prod discussion you may ask, “What do you think we’ll be doing today?” or “Do you think you could build one that doesn’t break?”
Ask several general questions about wheelchairs. For example, ask them to raise their hand if they have ever ridden in a wheelchair, or to raise their hand if they know someone who uses a wheelchair. Then ask more open-ended questions. Possible questions are:

“What is important to think of when designing a wheelchair?”

“If you were riding in a wheelchair, what would be important to you?”

“What would be important for your grandmother/father if she/he needed a wheelchair?”

Write responses on the board. If the students do not raise the following factors you may prod or lead them to these factors including a brief description of why they are important:

- **Size**: the wheelchair should be appropriately sized for the person it will be transporting and not so big that it won’t fit through doorways.
- **Weight**: the wheelchair shouldn’t be excessively heavy, but should be heavy enough for stability.
- **Application/Use/Purpose/What used for**: it is important to know what the purpose of the wheelchair is so you can design for any environmental or use constraints (i.e., will it travel in water, mud, be manually or automatically powered, etc.).
- **Durable/Reliable**: the wheelchair should be able to resist damage or breakage during normal use and work reliably.
- **Cost-effective/Inexpensive**: As with everything, keeping it less expensive is always a good thing.
- **Safety**: the wheelchair should be safe to use so people don’t get injured using it.
- **User Friendly**: the wheelchair should be easy to use and have intuitive controls. It should also be ergonomic so that the user will not be strained during use.
- **Attractive/Eye-Catching**: as with many products people would probably prefer a more eye-pleasing design.

“Great. All of these are called **design requirements** for a product. Anytime someone designs a new product, there is a set of design requirements that engineers must follow.”

**Wheelchair design challenge**

“Your challenge today is to build a wheelchair that will meet all the design requirements on your handout. You can use anything in the LEGO Robotics kit (except for RCX, motors, sensors, cables, and tower – these should have been removed prior to class).”

Have students get into teams of 2 or three and pass out the Wheelchair Design Challenge Handout (1 per team). These teams do not need to be the same teams from Lesson 0. Since team assignments are not permanent until Lesson 5, you may want to experiment with different team dynamics to see what works best for your class.

As a class, review the requirements of the challenge.

- **Wheelchair must be at least 8 inches tall.**
- **Chair must be able to roll while holding an object of teacher’s choice (the object should be about the size of a small water bottle).**
- **The chair can be made from any LEGO pieces in the kit, except for RCX, motors, sensors, and the tower.**
- **The wheelchair must pass a drop test from 3 feet (teacher’s waist).**

Explain that the wheelchair will be dropped from the teachers waist height. Ask the students what factors will be important to them for this activity. Write the responses on the board or circle the responses they gave earlier.

Pass out LEGO Robotics kits and announce, “You have 25 minutes to complete your design!”
Give students about 25 minutes to build. Avoid answering any questions that lead them to any specific type of design. They may be uncertain of what to do, keep encouraging them to test things out and play with the LEGO pieces. Direct them to look at the things around them for ideas (i.e. bricks or cinderblocks in walls, braces in desks/chairs, cars in the parking lot with frames/wheels/etc.). Also have them refer to what they have written down on the Spatula Challenge handout from previous class.

If some students come up with particularly good way to put pieces together, stop the class and have the student share the idea with the class.

**Testing**

“Now it’s time to test the wheelchairs.”

Have the teams come up one at a time, show that their chair is at least 8 inches tall and can hold the required object. Ask the students to describe their design and how they made it strong. Drop their wheelchair from your waist into a box or bin (to avoid having the pieces scatter in the room). Allow them to tell you exactly how they want to have the wheelchair dropped (i.e. with the wheels facing down).

Students may be embarrassed that their design broke. Discourage them from immediately taking apart the remaining pieces and/or from grabbing the broken pieces and hurrying back to their desk. Instead, use the time to ask them how/why their design did/didn’t break. Point out weak points in the design as well as techniques that provided strength. Focus on the parts that held together, not the parts that broke off.

**Redesign and test (Optional)**

Have the teams go back and redesign their chairs to make them even stronger. Give them 5 more minutes to come up with a stronger design and have them retest them on their own.

**Discussion**

Have the students comment on what made their designs work/not work. Ask them what they learned about building with LEGOs that they will remember for their final projects.

Some key points that you may want to point are are:

- Staggering pieces – like laying bricks – help keep pieces together.
- Using pins to connect beams is a very strong way to hold them together.
- Triangles (trusses) are very strong, particularly when held together by pins at the ends.
- Beams and bricks are thicker, and thus can be stronger than plates which can flex.

Have the students complete the second page of the Wheelchair Design Challenge handout. Take apart the wheelchairs and put all pieces back to where they belong. Be sure to check the kit content before students leave.

**Optional Homework**

The optional homework, Putting Things Together, asks students to explore their surroundings to look for different ways of fastening materials together (i.e., screws, nails, glue, etc). While this homework is not directly related to robotics or assistive devices, it is an extension from what students learned during this lesson and addresses the Massachusetts Technology/Engineering Framework standard 1.1.

**EXTENSIONS**

- Spend some time at the beginning of this session discussing the students’ work on the Engineering Design Process homework.
- Spend time to go through the optional Putting Things Together homework as a class. Relate
the fastening methods to ways of putting LEGO\textregistered s together.

- Discuss and demonstrate ways of using some of the specialized LEGO pieces, such as pins, connectors, axle joiners, etc.
- Additional building activities such as the "Kentucky Do-Nothing Machine" (from *Crazy Action Contraptions* booklet) can be included before or after this lesson to further improve students’ construction skills.

**HOMEWORK**

- OPTIONAL – Putting Thing Together Homework

**WORDS TO KNOW**

- **Design requirements** – A set of requirements that a design must follow in order to serve its function and be marketable.

**EXTRA RESOURCES**

- Additional tips for building strong structures are available in *Constructopedia*, which is included on the CD as a PDF document.
Lesson 2
Selecting a Solution

OBJECTIVES

- Go through the process of selecting the best solution from a set of possible solutions (EDP Step 4).
- Experience looking at and working with isometric and orthographic drawings.
- Introduce students to working with gears.
- Understand the subsystems of a transportation vehicle.

BACKGROUND FOR TEACHERS

Transportation Systems

Most transportation vehicles – including wheelchairs – can be broken down into subsystems, including structural, propulsion, guidance, suspension, control, and support.

- **Structural** – the mechanical components that keep the vehicle intact – outer casing, support bars, etc.
- **Propulsion** – the component that gives it motion – engine, motors, etc.
- **Guidance** – GPS, for example, that help guide the vehicle
- **Suspension** – component that absorbs shock and protects the vehicle and passenger.
- **Control** – component that allows passenger to control the vehicle, for example, a steering wheel or joystick
- **Support** – components that provides structural support for the vehicle – safety cage, etc.

These subsystems are often designed and manufactured separately and brought together as a complete system. This allows engineers to specialize and work within their field of expertise. For example, when designing a car, experts in ergonomics can design the seat, while fluid dynamics experts can design the outer shape, and power systems experts can design the engine block. It is also possible to purchase different subsystems and assemble into your own product.

Engineering Drawings

A picture is worth a thousand words. It may be a cliché, but it certainly rings true in the field of engineering. Engineering drawings are crucial to communicating ideas, products, solutions, etc. Drawings can effectively communicate the design of a complex system or product. For example, an engineer could design something and then send the drawings to a machine shop to construct a prototype, to marketing to decide on the colors and logo placement, to purchasing to purchase the materials, and to their boss to get a pat on the back (or approval).

Engineers mainly use two types of drawings to communicate: *isometric* drawings and *orthographic* drawings.

*Isometric view* is a 3-D view of the object, with X, Y, and Z axes appearing 120 degrees apart on the paper. The drawing on the left shows the three axes of an isometric drawing, along with an isometric drawing of a cube. While isometric drawings are useful for getting the overall picture of an object, it is not always the best drawing for communicating specific design details.
For design details such as dimensions, *orthographic* drawings are typically used. An orthographic view of an object is just one side of it. There are six orthographic views to an object: front, back, top, bottom, left, and right. The figures below show how four of the orthographic views are generated for a stacked cube object, as well as the orthographic views for all six views of the stacked cube.

Often times, the isometric drawing, along with one, two, or more orthographic views, are necessary to communicate all aspects of the design. Furthermore, drawings that show various stages of construction and assembly is necessary for putting together multi-part designs.

**Gears**

Gears are used in many mechanical devices. Gears are a type of simple machine that can give you a mechanical advantage (the ratio of the force exerted by a machine to the force applied to it). The components of any mechanical devices could be broken down into simple machines. There are six basic simple machines: lever, pulley, wheel and axle, inclined plane, wedge, and screw. The gear is considered to be within the wheel and axle category of simple machines.

Gears are used to:

1) increase/decrease speed
2) increase/decrease power (torque)
3) change the direction of rotation
4) move the axis of rotation
5) change the axis of rotation by 90°

These effects come hand-in-hand; for example, if you increase the speed, power will inevitably decrease, and vice versa.

Many everyday objects use gears. Gears are used in hand-mixers or eggbeaters to increase the speed of the mixers so that a person can turn the handle at a leisurely pace while the egg beaters spin rapidly to whip cream or eggs. The gears in your bicycle can help you to transfer more or less torque to your wheels. Here, you trade off speed for power – when you go up a steep hill, the lower gear gives you more power and helps you climb the hill although at a slower speed. When on a level surface, you shift to a higher gear so that you can gain a lot of speed with very little pedaling.

More details about gear ratios and different types of gears can be found in "More about Gears – Supplement for Teachers" and the optional Lesson 2.5: More about Gears.

**TIMING**

- Introductory discussion 10 minutes
- Selecting the best solution 15 minutes
- Building the gear train 45 minutes
- Design comparison and gears discussion 30 minutes

**Total: 100 minutes**
STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multiview drawings.

2.3 Describe and explain the purpose of a given prototype.

6.3 Identify and describe three subsystems of a transportation vehicle or device, i.e., structural, propulsion, guidance, suspension, control, and support.

MATERIALS

- ROBO Technology Set (or Team Challenge Set) – 1 per team
- Resource Kit, or extra LEGO bricks and gears (not necessary if you are using the Team Challenge Set)
- Large box or container to hold LEGO constructions: 1 per team
- OPTIONAL – Overhead projector
- OPTIONAL – Ramp and/or bumpy terrain for testing cars. A ramp can be created using a large poster board propped up on a box or chair. To test a car’s ability to drive on a bumpy terrain, have it drive over a book or LEGO bricks.
- Extra batteries.

Handouts & Homework:

- Handout 2-1 – 1 per team
- Handout 2-2 – 1 per team; cut apart by columns
- Handout 2-3 a~c (Assembly Instructions) – For each type of geartrain, make enough copies for about half the number of teams in class.
- OPTIONAL – Handout 2-4 – 1 per team
- Orthographic Drawing Homework – 1 per student

PREPARATION

- Cut Handout 2-2 by columns so that each team only gets the information that they asked for.
- Review Background for Teachers section above and “More About Gears – Supplement for Teachers.”
- If using an overhead projector, print out “Orthographic Drawings of Model A” on a transparency. Otherwise, print on normal paper to show class or make copies to distribute.
- Be sure that all RCX’s have batteries and have the default Program 1 loaded. (To test, turn the RCX on, select Program 1, and press Run. Small arrows should appear above A and C on the display. If the arrows do not appear, use ROBOLAB to install firmware on the RCX.)
Lesson 2 – Selecting a Solution

LESSON GUIDE

Introductory Discussion

“Today, you will be working as a team of engineers who were hired to design a new, ergonomic motorized wheelchair. What are some of the different parts that you need to design?”

Write down responses on the board. Some examples are:

- seat
- wheels
- engine/motors
- frame
- controls
- extras (places to carry things, foot rest, etc)

“Very good. Any type of vehicle – car, airplane, train, bicycle, wheelchair, etc – have many components. We can break them up into subsystems: structural, propulsion, guidance, suspension, control, and support.”

Discuss how each of the features above relate to these categories.

- **Structural** – frame, seat, some of the extras
- **Propulsion** – engine/motor, wheel
- **Guidance** – GPS
- **Suspension** – suspension system (if any), tyre
- **Control** – computer interface and the control buttons/joystick
- **Support** – seat

“Often, these different systems are designed by different teams of engineers, or even different companies. Why do you think this is the case?”

Guide the discussion to lead to these answers:

- The expertise needed for these areas are very different, so it’s difficult for one team to do everything. For example, structural components are related to materials and structures, while propulsion is related to power and electricity.
- It helps companies specialize. It is difficult for one company to collect expertise in all fields. In today’s economy, it is sometimes better for companies to specialize in one field, while relying on other companies for other areas.
- If some other company already makes good components, it is more economical to purchase the component than to develop it from scratch.

“Your company decided that, as a cost- and time-saving measure, you will be purchasing the drive-train (engine/motors and wheels – part of the propulsion system) from a different company. Your responsibility will be to choose a drive-train around which you will design the chair.”

“What are some of the things you might want to look for in the drive-train of a motorized wheelchair?”

Have the class list out what they should look for. Prod them for answers until you get a good list on the board. You may want to remind them about Lesson 1 discussion about what you look for in a wheelchair. Some ideas are:
Lesson 2 – Selecting a Solution

- strong
- reliability/maintenance needs
- durability; lasts a long time
- speed
- size and weight
- aesthetics
- cost
- traction/maneuverability
- power source/battery life

“Which one of these do you think is the most important?”

“Not all consumers and companies agree on which criteria is the most important, because we each have our own preferences and priorities. That’s why there are always many different models of the same product on the market.

Discuss how consumer choice depends on what you consider to be the most important criteria. For example, of the numerous cellphone models that are on the market, one person may choose a particular model because it looks nice, while another person may choose something else because of some special feature, and a different person may choose yet another model based on price.

**Selecting the best solution**

Pass out Handout 1. The handout has three drive-train models and some general information.

“The handout shows the three wheelchair drive-trains that you will be choosing from. Some information are already given about the three models. In your teams, discuss and choose three other criteria that are most important to you. When you’ve decided on your three criteria, report back to me and I will share those information with your team.”

Based on the three top criteria that the students choose, give each team appropriate additional information (columns from Handout 2, separated). Have the students write down their information on the three empty rows in Handout 1. Instruct the teams to decide on their top choice of drive-train based on their information. Remind the students that they are selecting a drive-train to be used in the new wheelchair design, and that these are NOT the actual wheelchairs.

Give students about 10 more minutes to discuss the three designs and pick the best choice based on information given. Share the results with the rest of the class.

Discuss how different criteria give you different choices, and you often need to compromise on one criteria to get the best result on a more important criteria (trade-offs).

“Now, you will get a set of instructions for building the drive-train that you selected. The instructions will have different drawings. Engineers use different types of drawings to communicate a design.”

Hold up the handout with three wheelchairs.

“This type of drawing is called an isometric drawing. Can anyone guess what ‘isometric’ means?”

“What does this drawing tell you? Does it give you enough information to actually build the design? What might you need to find out what’s going on in parts that you can’t see?”

Guide student response until someone mentions “showing different views.”

“Exactly. You need different views to see all the parts. An object has six views: front, back, top, bottom, right, and left (hold up an object and rotate as you say this). These drawings
are called ‘orthographic’ drawings.”

Show the orthographic drawings of Model A, which is provided.

“One, two, or even more views are needed to show all of the details that are in an object. Your instructions will contain some of these different views and drawing types.”

**Building the gear train**

Give each team building instructions for their top choice drive-train. Ideally, each of the three models will be represented at least once so that all three models can be compared at the end. If one or more models does not get selected, you should: 1) assign the model to a team even if the team picked another model, 2) have a team that finishes first build a second model, or 3) build it yourself to have as an example.

Plan on giving students about 45 minutes to build the drive-train. (Note: Have extra pieces ready because some of the designs require more pieces than what are in the RoboTechnology Sets.)

As a team-building or communications exercise, try one of the following strategies:

- After a student complete one step of the instruction, hand the model to another student to complete the next step. Repeat until model is done. This method ensures that everyone gets a chance to build.
- Assign three roles to students: one “reader”, one “getter”, and one “builder.” Any remaining members will be observers. The “reader” is in charge of instructions, and no other students can see the instruction packet. The reader must verbally describe the instruction to the team. The “getter” will then get the correct piece from the kit and hands it to the builder. The “builder” will put the piece in the correct place. Switch roles after each step so that everyone gets a chance to be each of the three roles. This method reinforces the use of correct part names and communication. (Allow extra building time if you are using this method.)

As they are building, walk around the classroom and occasionally ask students, “What type of a drawing is this? What view is this?” to reinforce the idea of engineering drawings.

As some teams get close to finishing, stop the class and explain how the motors, RCX, and electrical cables connect together. (Note: Cables are not included in the assembly instructions. Instruct the students to add cables when adding motors and RCX.)

Key facts to cover are:

- RCX is the “brain” or the computer. It holds the programs, connects to motors and sensors, and holds batteries.
- RCX has three output ports for motors/lights, and three input ports for sensors.
- The motors change the electrical power of the batteries to motion – they are similar to car engines, which takes in gasoline and produces motion. The motors connect to ports A,B,C using black cables.
- The On/Off button turns the RCX on and off. When not in use, turn off the RCX to save batteries.
- Press Prgm button to switch between programs – the RCX holds five different programs.
- Go to Program 1 and hit the Run button to make the motors A and C move.
- Electrical components are expensive – treat them with care, and NEVER drop them.
- **NEVER DROP THE RCX.** Once the RCX and motor are hooked up, DO NOT test the cars on the table. Always test on the floor to prevent accidental falling.

Once some of the teams begin testing their cars, you will most likely see some cars that travel
backward or move in circles. When this happens, stop the class and point out that motor direction can be changed by turning the black connector by 180°.

The construction of three models have varying degrees of difficulty, with Model C being the most complex. If some teams finish earlier than others, see Extensions below for suggested additional activities.

**Design comparison & gears discussion**

The three drive-train models have different gearing and speeds (Model B is fastest; Model C is slowest). As different teams test their designs, encourage students to compare their designs and look at other teams’ designs. Ask, “Why do you think one is faster/slower than the other?” Emphasize that faster is not necessarily better – slower cars have advantages too, such as being able to climb a ramp and being more maneuverable.

If you wish, you may use the optional handout “Wheelchair Drive-train Test & Evaluation” and have students compare their cars with other teams’ cars to complete the form. See Extensions.

After everyone is done with the building, stop the class and spend some time to talk about general characteristics of gears. Key points to discuss are:

- Small gear to big gear decreases the speed.
- Large gear to small gear increases the speed.
- Speed increase/decrease is accompanied by power decrease/increase, respectively.
- Two gears that are engaged will turn in opposite directions.

At the end of the lesson, keep the cars intact because you will be using it for Lesson 3. Store them in a box in a safe location.

If you wish to discuss gears in greater detail, proceed to optional lesson: *Lesson 2.5: More About Gears*. This lesson can be used between Lesson 2 and 3 for a more in-depth look at gear ratios and different types of LEGO gears.

**EXTENSIONS**

- Since these models were just the drive-train for an ergonomic wheelchair, you may ask the students to build the rest of the chair around their drive-train. Have a small object that’s about the size of a soda can (small stuffed animal, or any object that’s about the size of a soda can) and have students build the chair to carry the object.

Using the optional handout, “Wheelchair Drive-train Test & Evaluation,” have students test the following and document the results on the handout:

- The pre-loaded Program 2 will allow you to control the car using two touch sensors as a remote. Attach two touch sensors using long cables (use ports 1 and 3) and try driving through an obstacle course. See which model is the most/least maneuverable. Note: Depending on the wheels on the car and the type of flooring, some cars may not turn very well. Encourage students to try different wheels or take off the rubber on the non-driving wheel to improve turning ability.
- Test each car’s ability to go over bumps or up a ramp. Challenge teams to try different wheels or modify their gear train to improve the design.
- Test each car’s durability by seeing how the car withstands a bump into the wall. Students should NOT crash their cars with other cars; only crash them into the wall.
HOMEWORK

• Orthographic Drawing Homework
• OPTIONAL – Gears Exploration Homework

WORDS TO KNOW

• Ergonomics – “Ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance.” (official definition adopted by the International Ergonomics Association, http://www.iea.cc)

In simpler terms, ergonomics engineers design products that are comfortable, intuitive, and safe, keeping human anatomy and psychology in mind. The ergonomics discipline is also called human factors engineering or engineering psychology.

• Isometric drawing – “Three-dimensional” drawing, seen from an angle so that, if viewing a cube, three sides are seen equally.

• Orthographic drawing – Drawing of one side of an object as seen from front, back, left, right, top, or bottom.

• RCX – The big yellow brick that contains the controller behind LEGO robotics.

EXTRA RESOURCES

• More About Gears – Supplement for Teachers: A document describing gear-related concepts in more details. Also contains sample demo-models for various gear types.

Related activities from Robolab@CEEO

• Snail Car [ http://130.64.87.22/robolabatceeo/k12/activities/3/activity.html ]
An activity to design and build the slowest car possible using gears.

• Exploring Gear Trains [ http://130.64.87.22/robolabatceeo/k12/activities/38/activity.html ]
An activity that allows students to practice working with gears in an open-ended activity.
Lesson 2.5
More About Gears (OPTIONAL)

OBJECTIVES

• Learn how to calculate gear ratios.
• Understand what happens to power and speed as you gear up or gear down.
• Explore different types of gears, such as spur gear, bevel gear, crown gear, and worm gear, and understand what each type is used for.

BACKGROUND FOR TEACHERS

While the previous lesson allowed students to experience gears qualitatively, this lesson will go into more details about gears so that students learn how to calculate gear ratios and to work with different types of gears.

Most students should be learning about ratios in 6th grade mathematics. Check to make sure that they have learned about ratios prior to going into this lesson (Some parts of the lesson can still be done even if the students have not learned about ratios yet).

TIMING

• Exploring Gears Handout 30 minutes
• Discussion 10 minutes
• Compound gears 20 minutes
• Other types of gears 10 minutes

Total: 70 minutes

MATERIALS

• LEGOos – have lots of gears, axles, bushings, beams, etc. Electronic parts (RCX, sensors) are not used in this lesson.

Handouts & Homework:

• Exploring Gears Handout – 1 per team
• OPTIONAL – Gears Homework – 1 per student
• OPTIONAL – More Gears Homework – 1 per student

PREPARATION

• Photocopy Exploring Gears Handout – 1 per team
• Read through “More About Gears – Supplement for Teachers” and build models for bevel gears, crown gear, worm gear, rack, a pulley system, and a chain system as shown on the last page of the supplement.
• Make the 2-gear model that is on the first page of Exploring Gears handout.
LESSON GUIDE

Exploring Gears Handout
The Exploring Gears Handout is a self-guided handout that introduces students to gears and gear ratios. Give students about 30 minutes to work through this handout.

Discussion
Gather the class’s attention, and discuss what they have found while working on the handout. Ask probing questions such as:

“Do bigger gears turn faster, or smaller ones?”
“What would you do if you wanted use gears to lift up something that’s really heavy?”
“Who can tell me what an idler gear is? Why would anyone need idler gears?”

As the discussion progresses, list the following characteristics of gears on the board:

• small gears turn faster; large gears turn slower.
• small driver and a big follower results in reduced speed and increased power.
• big driver and a small follower results in increased speed and reduced power.
• adjacent gears turn in opposite directions.
• idler gears can be used adjust gear spacing and to fill in the space between driver and follower gears.
• gear ratio describes the amount by which speed (rotation) is increased between two gears, and is defined as:

\[
\frac{\text{# of teeth on driver gear}}{\text{# of teeth on follower gear}} = \frac{\text{# of turns (speed) of follower gear}}{\text{# of turns (speed) of driver gear}}
\]

Compound Gears
Hold up the two-gear model that you have made ahead of time. Ask,

“Which one should I use as the driving gear if I wanted to make a car go slow?”

“What if that’s not slow enough? How can I make this go even slower?”

A common mistake here is to simply add more pairs of 40-tooth and 8-tooth gears so that you get a gear train that looks like 40-8-40-8-… However, because everything except for the first and last gears are idler gears, this configuration does not help the follower gear go slower.

Guide a discussion until the students see that you need to make compound gear train, with more than one gear on an axle.

Challenge the students to make a compound gear train with at least 5 gears on one long beam. Have them count the number of turns on the driver gear and the follower gear, and determine the gear ratio for the new gear train. If your students are familiar with ratio multiplication, you may want to teach them how to calculate compound gear ratios by multiplying individual gear ratios (see "More About Gears – Supplement for Teachers").

Other types of gears
Using sample constructions (see last page of "More About Gears – Supplement for Teachers"), explain the following concepts to the students. The students do not need to learn how to build with these gears at this time; they should, however, know that these gears exist.

• The type of gears that the students have been using so far is called spur gears – they are characterized by straight teeth.
• **Bevel gears** are used to change the axis of rotation by 90 degrees. Bevel gears are used only with another bevel gear – not with spur gears.

• **Crown gears** also do the same thing as bevel gears. Crown gears have curved teeth, and are used with spur gears or other gears to change the axis of rotation by 90 degrees.

• **Worm screw** looks like a screw, and is used for slowing things down dramatically or for increasing power. It meshes with a spur gear so that one turn of the worm turns the spur gear by one tooth.

  The gear box allow you to mount a worm screw and a 24-tooth spur gear in a compact manner (gear ratio of 1:24). Although the gear box makes worm gear assembly very easy, it is not necessary – you can make your own assembly using beams and axles.

  Worm screw can only be the driver gear, never the follower gear – if you try to move the worm screw by turning the spur gear, you will not be able to.

  Worm drive also turn the axis of rotation by 90°, but this is not the primary purpose of the worm drive – it is used typically for reducing speed or for gaining power (or both).

• **Racks** are used with worm gears or spur gears. They are used to change rotation into linear motion.

• **Pulleys and chains** are similar to gears. Pulleys tend to slip and can only be used for low-power applications; however, they have a distinct advantage that it does not suffer from alignment issues that sometimes make gear assembly difficult. Chains work with spur gears and they also don’t require precise alignment. Another difference between gears and pulley/chain is that both chains and pulleys make the axles of two adjacent gears/pulleys turn in the same direction.

At the end of the lesson, have students disassemble their gear trains and put all pieces back to their appropriate locations.

**EXTENSIONS**

• Challenge the students to make a large structure with as many gears as they can incorporate. Encourage them to use various types of gears.

• See Extra Resources below for suggested additional activities.

**HOMEWORK**

• **OPTIONAL** – Gears Exploration Homework

**WORDS TO KNOW**

• **gear ratio** – The ratio of the number of teeth on driver gear to the number of teeth on the driven gear. The ratio describes the amount by which the speed is increased, or power is reduced. For example, if the gear ratio was 2, then the speed is doubled and the power is halved.

• **driver gear (or driving gear)** – The gear to which power is applied. In LEGO, it’s the gear that is attached to the motor.

• **follower gear (or driven gear)** – The gear to which output is connected to. The output could be wheel, arm, etc.

• **idler gear** – Any gears that are between driver and follower gears that have no effect on the overall gear ratio. They are used to adjust the direction and the spacing of input and output.
output gears.

**EXTRA RESOURCES**

- **More About Gears – Supplement for Teachers:** A document describing gear-related concepts in more details. Also contains sample demo-models for various gear types.

  Related activities from Robolab@CEEO

- **Snail Car** [http://130.64.87.22/robolabatceeo/k12/activities/3/activity.html]
  An activity to design and build the slowest car possible using gears.

- **Exploring Gear Trains**
  [http://130.64.87.22/robolabatceeo/k12/activities/38/activity.html]
  An activity that allows students to practice working with gears in an open-ended activity.
Lesson 3
Programming with ROBOLAB

OBJECTIVES

• Begin learning about programming in ROBOLAB.
• Begin to develop the terminology and basics of computer programming.
• Complete simple programming challenges using ROBOLAB.

BACKGROUND FOR TEACHERS

Robots are stupid! While a robot can be programmed to perform some very sophisticated tasks even tasks a human cannot do, they can ONLY do what the human programming them tells them to do. Thus, it is very important to learn how to effectively tell your robot what to do.

ROBOLAB is a graphical programming language specifically designed for the LEGO RCX programmable brick. The programmer uses icons to develop programs that send commands to the RCX outputs (A, B, C) and reads data from the RCX inputs (1, 2, 3). The ROBOLAB software allows the user to write a program and then download it via an infrared signal to the RCX programmable brick. More details about the ROBOLAB programming language can be found in several PDF documents on your CD.

This lesson plan uses Inventor Level 3; however, you may begin at other levels if you wish. Pilot Level 4 may work better for younger students, while Inventor Level 4 may be more suitable for advanced students. ROBOLAB icons are provided for commonly used icons in Inventor 3. These icons can be printed out and be used by students to write the programs as a team before writing it on the computer.

TIMING

• Introductory discussion 15 minutes
• Creating a “Human Program” 15 minutes
• Programming with the computer 45 minutes
• Discussion 5 minutes

Total: 80 minutes

STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multi-view drawings.

2.5 Explain how such design features as size, shape, weight, function, and cost limitations would affect the construction of a given prototype.
MATERIALS

• Computer with RoboLAB 2.5.4 (or comparable version) – 1 per team
• USB or serial Infrared Tower: 1 per team (included in LEGO Robotics kit)
• Two-motored car (from Lesson 2) – 1 per team
• Masking tape (for dark floor, or black electrical tape for light floor)
• Large box or container to hold LEGO constructions: 1 per team
• OPTIONAL – Projector or large monitor for showing instructor’s computer screen to the class
• Extra batteries.

Handouts & Homework:

• OPTIONAL – RoboLAB icon print-outs, cut apart – 1 set per team
• OPTIONAL – RoboLAB Programming Homework – 1 per student

PREPARATION

• Install RoboLAB software on computers. (Note: You may need administrative access to the computer in order to install the USB driver for the tower on PC’s.) BE SURE TO TEST THE TOWER-to-RCX COMMUNICATION PRIOR TO CLASS by either downloading a simple program to the RCX or going into RoboLAB → Administrator → Test RCX Communication.
• Make sure that all RCX’s have batteries and the firmware that matches your RoboLAB version. This will save you time during the class. Have extra batteries in case you need to replace them during class.
• Students will use their two-motored cars from Lesson 2. If they don’t have one, print out Model A instructions from Lesson 2 and have students build it. Plan for ~20 minutes additional time for the students to build the cars.
• OPTIONAL – Print out RoboLAB icons and cut them apart.
• OPTIONAL – Photocopy RoboLAB Programming Homework.

LESSON GUIDE

Introductory discussion

Ask the students: “How are we going to tell your wheelchairs/devices what to do?”

After some responses, introduce RoboLAB.

“We are going to use a software called RoboLAB to program our robots to move. RoboLAB uses icons/pictures that we can string together to create a program. A program is just a list of commands that tells the robot what to do.

Hold up the different icons you have printed out. Have them try and guess what each icon stands for.

- Begin – Begins Inventor program. Must be at the beginning of every program. This icon tells the RCX to start here.
- End – Ends Inventor program. Must be at the end of every program. The begin and end icons must be in every program or else the program will not be recognized by the RCX and will not run.
Motor A Forward – Commands the RCX to power a motor connected to output A to move forward. NOTE: “Forward” does not always mean “forward” for your car, depending on the orientation of the motor and the cables.

Motor A Reverse – Commands the RCX to power a motor connected to output A to move in reverse. NOTE: “Reverse” does not always mean “backward” for your car, depending on the orientation of the motor and the cables.

Motor C Forward / Motor C Reverse – Same as above, but powers motor connected to output C. There are also motors for output B.

Wait for 1 sec – Tells the RCX to wait 1 sec before moving on to the next operation. Similarly, you can choose wait for commands for 2 sec, 4 sec, 6 sec, 8 sec, 10 sec, ? sec (which allows you to set any time including decimal times)

Stop A – Commands the RCX to stop sending power to output A. Similarly, there are stop commands for outputs B and C.

Stop All Outputs – Commands the RCX to stop sending power to all outputs.

Creating a “Human Program”
During this exercise you will be using the programming icon cards in your kit. You will tape the cards to the blackboard or wall to create a program that a student (or yourself) will have to act out as a robot would.

“We are going to program a robot to go from one point in the classroom to another point. Where should we start? (Students pick point in the room) Where should we end?”
Let students pick point in the room; make sure the path includes some turns.

“Ok, what is the first thing we have to put on the board?”
Wait for students to say:

“What’s next?”
Let the students construct an entire program that they think will work. Have the class work together arranging the icon printouts on the board. Even if you know there are errors or it won’t work, don’t correct them.

“What is the last thing that has to be in the program?”
Wait for students to say:

Ask for a student volunteer to be the “robot.” Have the student stand in the starting position. Explain that he/she will be a robot with two motors: left leg will represent Motor A, and right leg will represent Motor C. One second can correspond to one step.

Slowly go through each step of the program, having the human robot follow the commands exactly.

Have students “troubleshoot” any errors that may have occurred and redesign their program.

“Remember, a robot can only do what the program tells it to do – but it will always do exactly what the program tells it to do.”
Lesson 3 – Programming with ROBO-LAB

Programming with the computer

Distribute the vehicles from Lesson 2 to each team. Have each team work at a computer. (Note: You may use the optional icons printout to have students program at their table before going to the computer.)

First, instruct the students to connect their USB or Serial tower to their computer. Then guide them to open ROBO-LAB. Once the opening screen loads, click “Programmer” and select “Inventor Level 3”.

Go to “Window” menu at top of screen and click “Show Tools Palette” and then “Show Functions Palette” (Note: the Functions Palette may already be open).

If you have a projector or monitor that all the students can see, demonstrate constructing the following program. Otherwise construct this on the board with the laminated icons.

Show how to place icons onto the page, and how to connect the icons with wiring tool. The pink wire must connect from the top-right corner (End) to the top-left corner (Begin). The pink wire tells ROBO-LAB the order in which to execute these commands.

Show the students how to use the “Context Help” feature. They can access this by pressing Ctrl-H or by going to the “Help” menu and selecting “Show Context Help.” When the context help window is open, you can point to any icon and the window will give you information about how to use it.

Have each team construct the simple program that will make their car go forward for some time, then stop. When the students are ready to download their programs, direct their attention to the download program arrow. Explain that when the arrow is broken the program won’t be downloadable and when the arrow is white the program is ready to download.

Draw students’ attention, and point out the USB or serial tower.

“This device is called the ‘tower’. This device sends the program from the computer to the RCX.”

“When your program is ready, place the RCX in front of the tower, turn the RCX on, set it to Programs 3, 4, or 5, and click on Run – the white arrow at the top left of the screen.”

Then have each team download the program. (Note: On some PC, the USB tower needs to be connected to the computer before starting up ROBO-LAB. If a computer is not recognizing the tower, save the program, quit out of ROBO-LAB, and restart ROBO-LAB while the tower is plugged in.)

“When the RCX makes a sweeping beep, that means that the program was successfully transferred, and you can now run the program by pressing the green button on the RCX.”

Also point out the following:

- The tower and RCX communicates by infrared, similar to how a TV remote works. That means that the RCX (the black part) and the tower needs to be able to see each other.

- If you have two (or more) RCX’s in the vicinity, it is possible for the tower to get confused over which RCX to talk to. If this happens, place a paper ‘tent’ over your RCX and the tower to prevent the signals from getting out. Turning off the RCX when not in use will prevent nearby towers from accidentally overwriting your program.

- The RCX is capable of holding five programs. However, programs 1 and 2 are locked and can not be used. The students’ programs must go into programs 3, 4, or 5.

Have students test their program. When done, present the students with the following programming challenge.
Challenge 1: Make a program that will have your car go forward past a line (5-10 ft away) and then travel in reverse past the starting line and stop.

Possible solution:

There are several variations of the program that will also work:

- “Stop All Outputs” can be replaced by “Stop A” and “Stop B”.
- “Flip Direction” may be used instead of “Motor A Reverse” and “Motor C Reverse”.
- Wait times will vary depending on gearing, power of the motor, and battery level of the RCX.

The “Wait for Time” command (shown on right) lets you set any time. Use “Numeric Constant” in Modifiers menu to enter a time.

As the students work on programming, help them debug the programs. At times, stop the class to point out some details that everyone should be aware of:

- You can undo by typing Ctrl-Z or going to Edit → Undo. This is very useful when you accidentally erased something you didn't want to erase.
- Broken wires are bad. You must erase them and re-wire the icons together. Select them individually and delete, or type Ctrl-B to remove all broken wires at once.
- The space-bar lets you toggle between the wiring tool and the arrow.
- If you get a black screen when trying to maximize a window, type Ctrl-E.

Discussion

“What did you learn about programming with ROBOLAB today?”

“What kinds of things might you be able to do now that you know how to program? What kinds of things could you do for your final projects?”

Remind the students to keep their cars intact, as they will be using it for the next lesson.

EXTENSIONS

Students who finish early can try some of the following programs:

- Challenge 2: Write a program which will make the two-motored car go forward for 2 seconds, make a right turn for 5 seconds, make a left turn for 5 seconds, and stop.
  
  Sample Solution (Motor A is left motor; Motor C is right motor):

- Challenge 3: Draw a line on the floor using tape. The line should consist of two or three straight segments, connected at some angle. Have the students program the car to follow the line (go straight for some time, turn, go straight again, and so on...).

- Challenge 4: Program the car to go forward for 5 seconds, stop, then play a sound. Challenge the students to explore the Functions tool palette to figure out which icon(s) will make the robot play a note or sound.

- Challenge 5: Program the car to go forward for 1 second, back for 1 second, and repeat indefinitely. You will need to teach them about jump/land.
HOMEWORK

- OPTIONAL – ROBOLAB Programming Homework

WORDS TO KNOW

- **Tower** – a device that sends the program from the computer to the RCX. The information travels via Infrared signal – the same type of signal used for TV remote. Tower is connected to the computer via USB or serial cable.
- **Program** – a list of commands that tells a robot what to do.

EXTRA RESOURCES

- **ROBOLAB Reference Guide** contains descriptions about all ROBOLAB icons. (PDF document is on CD)
- **Using ROBOLAB** is a comprehensive guide for LEGO Robotics system including hardware and software. (PDF document is on CD)
Lesson 4
Programming with Sensors

OBJECTIVES

• Learn about components of a communications system, and how LEGO sensors and RCX relates to communications systems.
• Learn how LEGO sensors work and how they can be used on a robot to make it communicate with the surroundings.
• Work with RoboLab touch sensor and light sensor.
• Learn how to program the RCX to react to sensors.

BACKGROUND FOR TEACHERS

Humans have five senses (sight, touch, hearing, smell and taste). Much like humans, machines and robots can be given sensors to detect various external conditions. A human’s senses work much the same way as a machine’s sensors.

The human eye has millions of tiny light receptors known as rods and cones. Each receptor receives light from the surroundings and sends a specific message to the brain based on the wavelength (color) and intensity of the light. The brain then decodes the message from the millions of receptors and translate it into what you interpret as the image.

A light sensor (the eye) receives light and a light-sensitive material in it produces a voltage that is proportional to the amount of light energy that it receives. The voltage signal (the message) is then sent to the computer (the brain). The computer decodes this signal and uses the information to make the robot react in a pre-programmed manner. The machine can thus be programmed to perform certain actions depending on the light conditions.

An example of using a light sensor with a Lego wheelchair might look something like the following. A wheelchair has a light sensor attached to the front of it pointing down towards the ground. As the wheelchair travels on a white floor and encounters a dark line, the light sensor value drops. This tells the wheelchair to stop and reverse direction. A program like this can be used to keep a robot or wheelchair confined in an area that is defined by a dark line on a light floor.

Much like how you receive information from your surroundings and react to it accordingly, robots must use sensors to communicate with the surroundings. The Massachusetts Technology/Engineering standards describe a communication (or sensor) system as including the following components: source, encoder, transmitter, receiver, decoder, storage and retrieval.

Below is a breakdown of how this relates to the LEGO light sensor on a robot:

• Source – external light that enters the light sensor.
• Encoder – the light sensor itself, which takes the light and produces a voltage.
• Transmitter – the wire connecting the light sensor to the RCX which transmits the voltage from the light sensor to RCX.
• Receiver – the RCX, or more specifically, the microprocessor chip within the RCX.
• Decoder – algorithm programmed within the RCX that takes the voltage and changes it to a number.
• Storage – the memory within the RCX.
• **Retrieval** – the user program that is stored on the RCX which retrieves and uses the information to determine the robot’s next course of action.

• **Destination** – message sent to output (i.e. motors, computer, display screen).

**More about LEGO light sensor**

LEGO Light sensor consists of two parts: the lens, and the LED. The LED (red) emits light. The lens (white/clear) collects light from the surrounding. The light that is collected by the lens hits a photo sensitive material, which produces a voltage that is proportional to the amount of light it receives. The voltage signal is sent to the RCX, and the RCX translates the voltage into a value between 0 and 100 (0 = no light, 100 = very bright).

The LED light source is used to illuminate the area that the light sensor is looking at. This is both good and bad. The advantage is that it allows the light sensor to differentiate colors or surfaces even when it is completely dark. The disadvantage is that this extra source of light sometimes result in counter-intuitive light values. For example, a dark-but-smooth surface (dark tile floor) will sometimes give a higher reading than a light-but-rough surface (such as a white carpet) because the smooth surface reflects more of the LED light into the lens while the rough surface scatters the light.

Light sensors can be used for various applications:

• Detecting a light or dark line on the floor (light sensor must be pointing down).

• Detecting when a hand or wall is nearby – as you approach a wall with a light sensor, the light value will increase due to the LED’s light reflecting back into the lens. Light sensor can be used as a touch sensor in this way.

• Differentiating colors.

• Use as a rotation sensor, by placing dark/light patterns on a wheel and looking at the wheel with the light sensor.

• Detecting changes in surrounding lighting by pointing it up toward the ceiling.

**Other LEGO sensors**

• **Touch Sensor**: Two touch sensors are included in most LEGO kits. The touch sensor can be used in two modes: detecting when it is pushed, or detecting when it is let go. Touch sensors are useful for application such as button switches, remote control for cars, or bumpers (to make a robot stop when it hits something).

• **Rotation Sensor**: Rotation sensors detect how much an axle has turned in increments of 1/16 rotations (16 “clicks” per rotation). They are more consistant and reliable than using time to move for a certain amount. Rotation sensor is not part of the kit and is sold separately (Part# W979891, $18.50 from LEGO Education).

• **Temperature Sensor**: Temperature sensors is used to measure temperatures between -20 °C to 50 °C. It is most useful for data collection in ROBOLAB Investigator, but can also be used in Programmer to make a robot that reacts to temperature changes. The temperature sensor is not part of the kit and is sold separately (Part# W979889, $30 from LEGO Education).

**TIMING**

• Introductory discussion 20 minutes

• Programming with Sensors 45 minutes
STANDARDS ADDRESSED IN THIS LESSON

*Massachusetts Science & Technology/Engineering Frameworks (6-8)*

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multi-view drawings.

2.5 Explain how such design features as size, shape, weight, function, and cost limitations would affect the construction of a given prototype.

MATERIALS

- Computer with ROBOlab 2.5.4 (or comparable version) – 1 per team
- USB Infrared Tower: 1 per team (included in LEGO Robotics kit)
- Two-motored car (from Lesson 2 & 3) – 1 per team
- Masking tape or black electrical tape (choose one that provides more contrast with your floor)
- OPTIONAL – Projector or large monitor for showing instructor’s computer screen to the class
- Extra batteries.

**Handouts & Homework:**

- Communications Systems Homework – 1 per student
- OPTIONAL – ROBOlab Cheat Sheet – 1 per student
- OPTIONAL – ROBOlab icon print-outs, separated – 1 set per team
- OPTIONAL – Programming with Sensors Homework

PREPARATION

- OPTIONAL – Photocopy ROBOlab Cheat Sheet and Communications Systems Homework – 1 per student.
- OPTIONAL – Print out ROBOlab sensor icons and cut them apart.

LESSON GUIDE

*Introductory discussion*

“Who can tell us what we did last time?”

Review basics of programming – turning motors on, stopping motors, and waiting.

“Today, we will add something new… sensors.”

“How many senses do humans have?” List all five (sight, smell, touch, hearing, and taste) on the board.
“There are two types of sensors in your LEGO kits – light sensor, and a touch sensor. Which of the two human senses these sensors represent?”

Relate light sensor to sight, and touch sensor to touch.

“Exactly. Let’s talk a little bit about how our eye works. What do you think is actually going on when you ‘see’ something?”

Guide a discussion until the students get the following general points:

- light enters the eye
- eye turns the light into some sort of a signal
- signal goes to the brain
- brain constructs an image
- you understand the image as your physical surroundings

Write this list on the board.

“Now let’s talk about how the LEGO light sensor work (hold up a light sensor). A LEGO light sensor takes in light, and sends a voltage to the RCX. The RCX then takes the voltage and outputs a number between 0 and 100 – 0 being very dark and 100 being very bright. How does this device compare to your eye?”

On the board, next to each item above, write:

- light enters the sensor
- sensor turns the light into some sort of a signal (a voltage)
- signal goes to the RCX
- RCX interprets the signal (outputs a number between 0 and 100)
- RCX uses the signal to determine next course of action

“The sensor allows RCX to communicate with the surroundings, and vice versa. So, in a way, it is a communications system. Let’s see how this works.”

“A communication system consists of these components: source, encoder, transmitter, receiver, decoder, storage, and retrieval.”

Write these on the board, and guide a discussion to define each one. Then relate each step to the LEGO light sensor until you get the following chart on the board:

<table>
<thead>
<tr>
<th>Source</th>
<th>The starting point of a message.</th>
<th>External light that enters the light sensor.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder</td>
<td>Message is translated into a signal.</td>
<td>The light sensor itself, which takes the light and produces a voltage.</td>
</tr>
<tr>
<td>Transmitter</td>
<td>Signal is sent out.</td>
<td>The wire connecting the light sensor to the RCX which transmits the voltage from the light sensor to RCX.</td>
</tr>
<tr>
<td>Receiver</td>
<td>Signal is received.</td>
<td>The RCX, or more specifically, the microprocessor chip within the RCX.</td>
</tr>
<tr>
<td>Decoder</td>
<td>Signal is translated into a message.</td>
<td>Algorithm programmed within the RCX that takes the voltage and changes it to a number.</td>
</tr>
</tbody>
</table>
Lesson 4 – Programming with Sensors

**Storage**
Message (or signal) is stored.
The memory within the RCX.

**Retrieval**
Message is retrieved from storage.
The user program that is stored on the RCX which retrieves and uses the information to determine the robot’s next course of action

**Destination**
The end point for a message.
Message sent to output (i.e. motors, computer, display screen).

“How do you think sensors can help your car/robot?”
Sensors make a robot more “intelligent” – able to take in surrounding information, and react to it in one way or another. It allows robots to interact with the surroundings – both the environment and humans.

**Programming with sensors**
Introduce the different icons you have printed out. Have them try and guess what each icon stands for.

- **Wait for Push / Wait for LetGo** – Tells the RCX to wait until the touch sensor is pressed, or the pressed button is let go.

- **Wait for Brighter / Wait for Darker** – Tells the RCX to wait until an increase or decrease in light sensor value is detected.

- **Input 1 / Input 2 / Input 3** – Sensor can be connected to one of three inputs. These modifiers are wired into the sensor icons to tell the program which sensor port is in use.

“These icons are used in the same way that you used the Wait-for time icons last class, but instead of just waiting for a certain amount of time, it allows you to wait until something happens externally.”

Distribute the vehicles from previous lesson to each team. Have each team work at a computer. Instruct the students to connect the USB or serial tower to the computer and open up RoboLab. Go to Programmer, then Inventor Level 3. (Note: You may use the optional icons printout to have students program at their table before going to the computer.) Pass out the optional RoboLab Cheat Sheet to each student.

Introduce the first challenge:

**Challenge 1**: Make a program that will have your car go forward until it sees a line on the floor then stop. (Use white masking tape on dark floor or black electrical tape on light floor to make a line).

Before programming, students must to attach a light sensor to their car using a right-angle bracket. Be sure to position the light sensor so that it is pointing down and is close to the floor.

Possible solution (for light-colored line on a dark floor, with light sensor connected to port 2):

- The Wait for Brighter/Darker icon can be wired with an optional modifier to adjust the % change in light to wait for. The default value is 5%, which means that a Wait for Brighter icon is waiting to detect a 5% increase in the light value. This
value may be too small or too big, depending on the lighting condition and the floor – for example, if the floor was light gray and you had a white line on it, then the sensor may only see a 3% change in light when it gets to the line. The optional numeric modifier can be used to adjust the Wait for Brighter icon so that the 3% change is adequate.

When students complete the first challenge, proceed to the following challenges:

**Challenge 2:** Put two parallel lines on the floor, about 6 feet apart. Start the car in between the two lines. The car should stop when it sees one line, then back up, and stop when it sees the second line.

Possible Solution (for light-colored line on a dark floor, with light sensor connected to port 2):

Note: The Wait for Time icon is inserted to ensure that the car backs up past the first line before beginning to look for the second line. Without this, the car will go forward, stop when it sees a line, back up, and stop immediately when it sees the line again.

**Challenge 3:** Add a touch sensor to the car using a long cable. Program the car so that it waits until a touch sensor is pressed, then goes forward, and stops when you press the button again.

Possible Solution (for touch sensor on port 1):

For additional challenges, see Extensions.

**Wrap-up and clean up**

“Can you give me examples of ways in which you can use sensors to make your robot more useful?”

“What are some things you can do with sensors?”

“Start thinking about what kind of robot you are going to design to help people. You will begin the project next week.”

**IMPORTANT:** At the end of this lesson, students should take apart the cars and put pieces back into their appropriate place.

**EXTENSIONS**

Students who finish early can try some of the following programs:

- **Bumper Car Challenge:** Build a bumper for the car with a touch sensor built-in, and program the car so that it will go forward until the bumper is struck, then backs up. The bumper should be robust enough so that it doesn’t fall off when the car bumps into a wall. The touch sensor should be triggered even when the car hits the wall at some angle.

  Sample Solution:
• **Bumper Car Challenge 2**: Modify the program from Challenge 4 so that the car goes forward, hits a wall, reverses for 2 seconds, and repeats over and over again. *(Note: Students need to learn about Jump/Land loop.)*

Sample Solution:

• **Find the Exit! Challenge**: Place the light sensor on the car again. Draw a box on the floor with tape, about 3 feet by 3 feet, with an opening on one side about 10 inches wide. Program the car so that, when placed inside the box at random directions, it will stay in the box until it finds its way out through the opening. *(Note: Students need to learn about Jump/Land loop.)*

Sample Solution:

The car goes forward, sees the line, stops, backs up, turns slightly, then goes forward again. The process is repeated until the car exits the box without encountering the line.

• **Line Follower Robot**: Draw a line on the floor using tape. The line should have a few curves, but not too tight. Write a program that will make a robot follow the line. Students will need to learn about Jump/Land. *(Hint: The light sensor should either stay entirely on top of the line if the line is fairly wide, or follow one edge of the line.)*

Sample Solution:

(This solution is for following the left-side edge of a dark line on light background, with motor A being the left motor and motor C being the right motor.)

If you have additional time, you may introduce more advanced programming concepts such as loops, jumps/lands, and forks.

**HOMEWORK**

• Communications Systems Homework
• OPTIONAL – Programming with Sensors Homework

**WORDS TO KNOW**

• **Source** – The source of a message.
• **Encoder** – The component that translates the message into a signal that can be transmitted.
• **Transmitter** – The component (medium or device) that transmits the signal.
• **Receiver** – The component that receives the transmitted signal.
• **Decoder** – The component that takes the signal and translates it into something that can be understood by the destination.

• **Storage** – The component that stores the signal or the decoded signal.

• **Retrieval** – The component that retrieves the signal or message from storage.

• **Destination** – The final destination of a message.

**EXTRA RESOURCES**

• **ROBOLAB Reference Guide** contains descriptions about all ROBOLAB icons. (PDF document is on CD)

• **Using ROBOLAB** is a comprehensive guide for LEGO Robotics system including hardware and software. (PDF document is on CD)
Lesson 5
Begin Final Project

Identify Need, Research Problem, and Develop Possible Solutions

OBJECTIVES

- Identify the need/problem to address for their final project.
- Research the need/problem.
- Develop possible solutions for the need/problem.

BACKGROUND FOR TEACHERS

The first three steps of the Engineering Design Process (highlighted below) allow an engineer to begin focus in on the need or problem they are going to address and then brainstorm the different ways of solving that need or problem.

The Massachusetts Science and Technology/Engineering Frameworks describe the Engineering Design Process as the 8-step process indicated below. Today we will focus on the first three steps:

Selecting a final project that is personally meaningful to the students is critical. During this lesson you want to guide the students to choose a need or problem that is meaningful to them, preferably one that is real for them in their lives (i.e. a need or problem that a family member or friend has). This is no easy task. The students will be working in teams and may want to pursue different ideas. Have them work together and try to combine their ideas and compromise.

The lesson plan focuses on designing an assistive device for the disabled. However, you may change the project focus if you wish – for example, you can open up to wider range of possibilities by defining “assistive device” as anything that helps any person, not just the disabled. You can also narrow down the choices by limiting the project to specific types of assistive devices – for example, devices to help someone who is in a wheelchair.

TIMING

- Introductory discussion: 5 minutes
- Identify the need or problem: 10 minutes
- Research the need/problem: 25 minutes
- Develop possible solutions: 15 minutes
- Discussion: 15 minutes

Total: 70 minutes
STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multi-view drawings.

MATERIALS

- Computers with Internet access: 1 per team
- Folder or Envelope: 1 per team
- Extra batteries.

Handouts & Homework:
- Final Project Development Form Handout – 1 per team
- Brainstorming Homework – 1 per student

PREPARATION

- Photocopy handouts and insert them into a folder or envelope for each team. You can either copy and distribute the entire packet (6 pages), or only the parts that are needed for this lesson (first two pages).
- Photocopy Brainstorming Homework (1 per student)
- Determine student team assignments. Teams will remain the same from this point through the remainder of the curriculum.

LESSON GUIDE

Introductory discussion

“Today you will start work on your final project! Your project will be to invent an assistive device that helps a disabled person.” (Note: If you wish, you may define the project in a different way to open up more choices or to restrict them.)

Direct the students to take out their engineering design process handout and ask them:

“What is the first step for starting an engineering project?”

Take student answers – Step 1: Identify the need or problem.

“Then what will you do?”

Take student answers – Step 2: Research the problem.

“As you work on your final project you will be working through the different steps of the engineering design process. Today, you will complete Steps 1 through 3 of the Engineering Design Process. You will identify the need or problem that you will solve, then research this need or problem, and then develop possible solutions.”

“In your folders, you will find a Final Project Development Form. You will be filling out this form in the next few sessions as you go along the project. Also, in the folder, will be some ideas for needs/problems that you could address. You don’t have to use one of these; they
are just suggestions.”

Have students get into teams – the teams will remain the same for the remainder of the program.

“I am going to pass out a folder to each team. You should look at the materials and then start by identifying the need or problem you are going to address with your final project. You may use the Internet to search for ideas or information about your project.”

**Identify the need or problem**

“First, identify the need or problem you want to address. What are some types of disabilities that may make a person’s life difficult?”

Write down students’ answers. Answers can include:

- a person can be confined to a wheelchair or bed (paralysis, amputation, etc)
- impairment with fine-motor control
- visual impairment
- loss of hearing
- uses crutches, canes, walkers, braces, casts, etc.
- limited strength

“Good. Think about how these disabilities can make your life more challenging, and think of inventions that could make it a little easier. You have five minutes to think of a need or a problem to solve.”

Encourage the students to pick a need/problem or final project idea that is personally meaningful to them – has any students experienced being in crutches or casts? Do they know of anyone with disabilities and if yes, ask what that person may have difficulties doing. Have the students imagine doing their daily tasks or hobbies while having one of these disabilities (i.e., brushing your teeth, taking a shower, drying your hair, opening a soda bottle, playing video games, etc.). Let them be creative with their ideas.

If you wish, you can begin the brainstorming as a class discussion, then break into teams after some ideas begin flowing. This may help students get into a creative state of mind.

Give them 5 minutes for the first step. If some teams are having difficulties deciding on a topic, step in and guide their discussion so that they can agree on a problem to address. (Note: They can move on to steps 2 and 3 if they have decided on a problem.)

At the end of 5~10 minutes, stop the class and ask teams to share their need/problem. Be sure that each team has an idea that is specific and addressable. Instruct the students to write down their need/problem on the Final Project Development Form.

**Research the need/problem**

“OK, now begin researching your need or problem. Use your Final Project Development Form as a guideline for what you should be researching. Write down everything you find on your Final Project Development Form, and don’t forget to write down URL’s if you use a website.”

If you have access to a printer, then you may have students print out the information rather than writing them down. If students print out any information or write down information on a separate page, instruct them to keep this in their team folders.

Give them 20~30 minutes for this with time warnings.

**Develop possible solutions**

“If you are satisfied with your research, begin developing possible solutions. Think of the different innovative ways you could solve your problem and record them in the next section of the handout. Remember, you are just brainstorming now – don’t criticize each others’
ideas, and write down everything that comes to mind, even if it seems impossible.”

Emphasize that there may be many possible solutions at this early stage and that they should not try to decide on just one solution at this point.

Give them 15 minutes for this with time warnings.

Discussion

“Let’s stop for now and I would like each team to come up and briefly tell us the need/problem you are going to address, what you found in your research, and what possible solutions you have thought of. You’re going to continue brainstorming until we meet next time, so it’s okay if you don’t have all of your ideas yet.”

Have the students share their ideas.

Give each student the Brainstorming Homework, and have them copy the first line (the need/problem) from the Final Project Development Form so that they don’t forget what their topic was. Explain that they will continue to brainstorm individually at home (or in team, if they have time to meet as a team). Also encourage them to talk to friends and family for ideas.

Collect all folders and Final Project Development Form handout.

EXTENSIONS

- You may have each team come up with a “company name” and logo. You may also have each team decorate their team folder or envelope with their company name and logo.

HOMEWORK

- Brainstorming Homework
Lesson 6

Continue Final Project

Select the Best Possible Solution, Begin Constructing Prototype

OBJECTIVES

- Select the best possible solution for the final project by assessing the pros and cons of possible solutions.
- Begin constructing a prototype for their final project.

BACKGROUND FOR TEACHERS

The Massachusetts Science and Technology/Engineering Frameworks describe the Engineering Design Process as the 8-step process indicated below. Today we will focus on Steps 4 and 5:

During this lesson, students will select the best possible solution and begin building their prototype. To select a solution, it is important that the teams select solutions based on how well the idea addresses the design requirement and constraints, as well as feasibility.

About prototypes

Prototypes are used in engineering to test ideas before going into full-scale production. Below are some characteristics of prototypes:

- A prototype is a model of something.
- A prototype has one or more of the functionality of the final product.
- A prototype is not exactly what the final product will be.
- A prototype is made to demonstrate and test the design concept.
- A prototype is not always the same size as the final product.
- Multiple prototypes are often used, each addressing a different product feature or aspect.
- A product will often undergo multiple stages of prototyping, testing, and redesigning. The first prototype is often very crude; it may be made by hand out of cardboard and styrofoam.
- At times, computer models are used in place of physical prototypes when appropriate. This saves the company a lot of time and money.

Prototypes are used to:

- test ideas before they invest in mass-producing the product.
- compare several potential solutions to see which one is best.
- make sure the design will work.
- communicate to others the design and idea (for example, showing it to investors to convince them of the feasibility, showing it to potential consumers to get feedback, or
showing it to your boss for approval).

- test for functionality.
- perform durability or reliability tests (for example, a crash test).

For this project, the prototype is to be made out of LEGO bricks; therefore, they are limited by what LEGO bricks, sensors, and RoboLab can do.

**TIMING**

- Introductory discussion 5 minutes
- Share brainstorming results & Select the best possible solution 15 minutes
- Begin prototyping 40 minutes
- Wrap-up and clean up 10 minutes

**Total: 70 minutes**

**STANDARDS ADDRESSED IN THIS LESSON**

*Massachusetts Science & Technology/Engineering Frameworks (6-8)*

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multi-view drawings.

**MATERIALS**

- LEGO Robotics kit – 1 per team
- Extra LEGO pieces
- Folder or Envelope: 1 per team (from previous lesson)
- Large box or container to hold LEGO constructions: 1 per team
- OPTIONAL – Digital camera
- Depending on students’ projects, you may need additional materials such as tape, ramps, water bottles, etc.
- Extra batteries.

**Handouts & Homework:**

- Final Project Development Form Handout – 1 per team (from previous lesson)

**PREPARATION**

- Check that team folders contain all pages of the Final Project Development Form.

**LESSON GUIDE**

*Introductory discussion*

“Today you will continue work on your final project. Can someone tell me what we did last
time and for the homework?”

Direct the students to the Engineering Design Process handout, and remind them that they have just completed steps 1~3 of EDP.

“What is your next step?”

The next step is Step 4: Select the best possible solution(s).

“How would you choose the best possible solution?”

Lead a discussion and talk about these strategies:

• List out the pros and cons for each solution.
• Think about how each solution addresses the design requirements and constraints.
• The solution should be feasible using existing technologies (i.e., no ‘time machine’).
• The solution should be marketable and profitable.

**Share brainstorming results & Select the best possible solution**

“Get into your team, and first share all the ideas you came up with at home. We’re still just sharing ideas, so be open to anything and don’t be critical.”

Redistribute team folders with the Final Project Development Forms. Give students 5-10 minutes to share the ideas that they have come up with at home.

“Now, from all of the possible solutions that you came up with during last class and at home, you will choose one solution AS A TEAM. You will begin with picking the three best solutions, then writing down the pro’s and con’s of the three solutions on your Final Project Development Form.”

Give students 5-10 minutes to decide on a solution. Walk around the classroom to watch how the teams are progressing. If a team is having difficulties reaching a consensus, help them narrow their choices down.

“When you’ve agreed on a solution, raise your hand so that I can come and approve your decision before moving on to the next step.”

As teams come to a decision, approve them based on the following criteria:

• The solution actually addresses the initial problem/need stated on the Final Project Development Form. If it doesn’t, you may suggest the students to rephrase their initial need/problem or to change their solution.
• The solution is novel.
• The solution is feasible using existing technologies.
• The students can realistically model one (or more) aspect of the design using LEGO and ROBOLAB.

As you approve the solution, ask the students, “What will your prototype look like? What aspect of the solution are you planning to model?” Direct them to page 3 of the Final Project Development Form and ask them to plan their prototype and fill out Steps 4 and 5 before proceeding.

**Begin prototyping**

After most teams have gotten approval, gather everyone’s attention and have each team share their idea with the rest of the class.

“Now you will begin prototyping. Who can remind us what a prototype is?” – Take student answers.

“Your prototypes must be completed by the end of next class. We have a lot of work and very little time, so make sure you stay on task and you all pitch in.”
Distribute the LEGO kits. Walk around the class, and make sure the teams are on task. Make sure they are working on both the constructing and programming portions of their prototypes. If possible, take some pictures of work-in-progress so that it may be included in the final presentation.

Some teams may need to learn about programming concepts that were not covered in Lessons 3 and 4, such as loops and forks. Introduce these concepts as the need arises.

**Wrap-up and clean up**

Give the students time updates, and stop the class when there is 10-15 minutes left.

“It’s time to clean up! We’ll keep working on this next class, so be sure to put your work in a labeled box so that you can find it next time!”

Collect all Final Project Development Forms and folders as well as works-in-progress. Prototypes should be stored in separate and clearly labeled box or container for each team. Have each team say a few words about where they are and what their next step is.

**EXTENSIONS**

- You may have each team come up with a “company name” and logo. You may also have each team decorate their team folder or envelope with their company name and logo.

**WORDS TO KNOW**

- **Prototype** – A model of a solution that demonstrates one or more of the design features.

**EXTRA RESOURCES**

- **RoboLab Reference Guide** contains descriptions about all RoboLab icons. (PDF document is on CD)
- **Using RoboLab** is a comprehensive guide for LEGO Robotics system including hardware and software. (PDF document is on CD)
- **Constructopedia** is a collection of step-by-step building instructions for common assemblies such as a rack-gear system or a lift arm. (PDF document is on CD)
Lesson 7
Continue Final Project

Finish Constructing Prototype, Test and Evaluate Solution

OBJECTIVES

- Finish building a prototype.
- Test and Evaluate the prototype.

BACKGROUND FOR TEACHERS

The Massachusetts Science and Technology/Engineering Frameworks describe the Engineering Design Process as the 8-step process indicated below. Today we will focus on Steps 5 and 6:

![Engineering Design Process Diagram]

The EDP steps 5 and 6 are really a small cycle by themselves. It is appropriate that teams should be testing and modifying their designs with incremental changes as deemed by their evaluation of test results.

It is important that the students finish constructing their prototype by the end of the lesson. It is also important for the students to be clear what a prototype is. The prototype is a model and may not incorporate all the features they want for the final end product that would be manufactured on a large scale.

As the teams progress, you may need to suggest the teams to trim down their features list in order to finish on time. Remind them that this is a prototype, and not a finished product. In many engineering projects, it is more important to deliver a ‘good’ solution on time than to attempt a ‘brilliant’ solution that doesn’t get completed on time.

TIMING

- Introductory discussion: 5 minutes
- Finish constructing prototype, and test & evaluate: 50 minutes
- Wrap-up and clean up: 15 minutes

Total: 70 minutes

STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches,
orthographic projections, multi-view drawings.

MATERIALS

- LEGO Robotics kit – 1 per team
- Extra LEGO pieces
- Computer with RoboLab – 1 per team
- Folder or Envelope: 1 per team (from previous lesson)
- Depending on students’ projects, you may need additional materials such as tape, ramps, water bottles, etc.
- Extra batteries.

Handouts & Homework:
- Final Project Development Form Handout – 1 per team (from previous lesson)

PREPARATION

- Check that team folders contain all pages of the Final Project Development Form.

LESSON GUIDE

Introductory discussion

“Can someone remind us what we were doing last time?” – Building a prototype.

“Today, we will finish construction of your prototype. What is the next step after constructing the prototype?”

Direct the students to their EDP handout. The next step is Step 6: Test and evaluate the solution.

“Why would you want to test and evaluate your prototype?”

“How will you be testing your prototypes?”

Test will vary with each team – remind them that they should be testing for whichever feature they were trying to demonstrate.

Finish constructing prototype, and test & evaluate

Distribute the LEGO kits, team folders with the Final Project Development Form, and the works-in-progress.

“You will have the remainder of the time today to complete your prototype and test/evaluate it.”

Give the students time updates every 10 or 15 minutes. Remember to work on both constructing and the programming. Teams should be very close to completing the prototype after 30 minutes so that they can begin testing.

Some teams may need to learn about additional programming features that were not covered in Lessons 3 and 4, such as loops (jump/land) and forks. These concepts can be taught individually as the need arises.

During this time you want to make sure the students are on track to have some working parts of their prototypes by the end of the session. If the teams are struggling to finish, encourage them to trim down the features so that they will have some working feature for the final presentation. Remind them that this is a prototype only, and does not need to have every
feature of the final product included.

Some students may want to add things that aren’t feasible or won’t have time for. Encourage these students to have a working prototype first, and then add other features at the end if there is time, or explain those features in their final presentations.

After 30 minutes:

“Everyone should be about ready to test your prototypes if you haven’t already started. For the remainder of today, you will be testing your models and modifying it if you need to.”

Wrap-up and clean up

With 15 minutes remaining in the lesson, stop the students. Have each team share a few words about how they tested their prototype and what the result was.

Direct them to Page 4 of the Final Project Development Form and give teams a few minutes to fill out Step 6.

When teams finish writing in the Final Project Development Form, instruct them to clean up and put away their prototypes safely and to place the Final Project Development Form in their team folders. Collect all materials.

EXTENSIONS

- You give as much time as you need to allow students to complete their prototype – you may even choose to add a day or two to the lesson.

EXTRA RESOURCES

- RoboLab Reference Guide contains descriptions about all RoboLab icons. (PDF document is on CD)
- Using RoboLab is a comprehensive guide for LEGO Robotics system including hardware and software. (PDF document is on CD)
- Constructopedia is a collection of step-by-step building instructions for common assemblies such as a rack-gear system or a lift arm. (PDF document is on CD)
Lesson 8
Communicating the Solution

OBJECTIVES

• Understand the purposes and methods for communication in the Engineering Design Process.
• Create a PowerPoint presentation to communicate the solution to potential consumers.

BACKGROUND FOR TEACHERS

The seventh step of the engineering design process allows an engineer to communicate the progress of the solution/prototype. This step also allows the other people involved in the development of the product to learn what is happening and give any appropriate feedback. This communication is usually two-way. That is, the engineers or product developers will present what they have done and give the audience an opportunity to ask questions and provide feedback.

The Massachusetts Science and Technology/Engineering Frameworks describe the Engineering Design Process as the 8-step process indicated below. Today we will focus on Step 7:

[Diagram of the 8-step process]

Communicating the solution involves multiple audiences and purposes:
• sending technical drawings to a manufacturer so that the part can be made correctly.
• presenting an idea to a panel of potential investors.
• presenting an idea to your boss or experts to get feedback.
• if multiple teams are working on subsystems of the same product, you need to communicate your design to other design teams to make sure that all the subsystems work together seamlessly
• communicating the design to marketing agents so that they can accurately pitch the product to potential customers.
• marketing the idea to potential customers using presentations, ads, informational brochures, websites, etc.
• educating the users about the product so that it can be used safely and effectively.

There are many forms of communicating a solution, and different methods are used for different audiences. Examples are:
• brochure, manual, or booklet
• posters, printed ads, or billboards
• presentation
• web pages
• word-of-mouth
• audio and video

In some cases, multiple methods are used to make the largest impact to the biggest audience – for example, you may run ads on the TV and radio, put a poster on a subway, place an ad on the newspaper, etc. to make more potential customers aware of your product. Even the product packaging is often designed in such way that it contains a lot of information about the product.

Two PowerPoint presentation templates are provided for this lesson – a four-page version and a six-page version. You may choose either one to use, depending on how much time you have for the final presentation. You may also choose to have students make a presentation from scratch, without using a template.

The final presentation should be an opportunity for students to show their work to their friends and family – try to schedule it during a time when parents, principal, their classmates, or other members of the community can come and see the presentations.

Alternative ideas for this lesson

As described earlier, PowerPoint is not the only method to communicate the solution. Poster presentation is a good alternative if computer access is an issue, and has an advantage that it can be displayed around the classroom for a long time for everyone in the school to see. A poster can also be used to accompany the project as a Science Fair entry. Creating a webpage, brochure, a user manual, or a TV commercial (using multimedia) can also be a fun way for students to document their inventions.

This lesson plan focuses on PowerPoint presentation as the method for communication; however, you may tailor this lesson in a way that best suits your class.

TIMING

• Introductory discussion 5 minutes
• Creating PowerPoint slides 30 minutes
• Presentation skills discussion 10 minutes
• Finishing up and practicing 25 minutes

Total: 70 minutes

STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

3.3 Identify and compare communication technologies and systems, i.e., audio, visual, printed, and mass communication.

MATERIALS

• Computer with PowerPoint – 1 per team
• OPTIONAL – Digital camera and/or scanner

Handouts & Homework:

• Final Project Development Form – 1 per team (from previous lesson)
• OPTIONAL – Presentation Template Handout – 1 per team
• OPTIONAL – Presentation Do’s and Don’ts Handout – 1 per team

PREPARATION

• Load a copy of the Presentation Template on each computer, or have it at an online location that is accessible from all computers.
• Check that team folders contain all pages of the Final Project Development Form.
• Photocopy optional handouts if appropriate, and insert them into the team folders.

LESSON GUIDE

Introductory discussion

“Where were we in the Engineering Design Process?” – Step 6 Test and Evaluate the Solution.

“Today, we’re moving on to the next step: Communicating the Solution. Why and how do you communicate a solution to someone? Who do we communicate to?”

Discuss some ideas with students. Ideas may include:
• creating a detailed technical drawing so that the manufacturers can make the part.
• making a brochure, a TV ad, or presentation to inform potential customers.
• making a User’s Manual so that the product can be used safely and effectively.
• presenting the idea to investors so that you can get more money to develop the product.
• presenting the idea to your boss or other people in the company.

“Today, your team will create a PowerPoint presentation that will inform potential customers about your product. What are some things you might want to include in your presentation if you were trying to get your audience interested in buying the product?

Write down student responses on the board. Ideas include:
• What the product does (what need it is addressing?)
• What features it has.
• How it is an improvement over existing products.
• How it has been designed/made and tested.
• What future improvements the company is working on.
• What the product looks like (drawing or photo, or an actual product demo)

Creating the PowerPoint slides

“You will use a template for the PowerPoint presentation so that each team covers these points. You have 30 minutes to create your slides today. Use your Final Project Development Form as reminders of what you did earlier.”

(Note: The templates are provided for convenience; however, you may have students create the presentation from scratch if you prefer.)

Pass out the team folders with the Final Project Development Form and have teams work on computers to create their presentations. Give them time warnings to keep them on task. Gather students’ attention after 30 minutes (they don’t need to be completely done yet).
**Presentation style discussion**

“Remember, you will be presenting these slides to your classmates (and parents and/or other teachers) on ____ (date). What are some things you should keep in mind when speaking to an audience?”

Write student responses on the board. Examples are:

- speak clearly and with good volume
- don’t talk too fast
- have good eye contact
- appear confident and professional
- don’t fidget

“What should your slides look like? What are some things you should look out for?”

Write student responses on the board. Examples are:

- check for typos
- don’t use colors that are difficult to read
- use large fonts that are easy to read
- don’t use “chat language” (for example, using ‘u’ instead of ‘you’)
- use pictures and graphics if it helps
- make it look pleasant
- try not to make the slides too wordy

If using the Presentation Do’s and Don’ts handout, pass them out at this time and discuss each point. Some of the points should already be on the board.

Emphasize that slides meant to accompany the presentation and not a stand-alone slide show. Therefore, students should avoid putting down paragraphs of text to be read word-for-word. Slides should contain bulleted items that highlight what the speaker is going to say.

**Finishing up and practicing**

“Good. So as you finish up your presentations, keep these things in mind. Make sure to spell check! When you are done making slides, practice your presentations. Each person on the team must speak at least once.”

In the remaining time, students should finish their presentations and practice. The students will have more time to finish up during the next lesson; however, you should encourage them to at least complete the slides during this lesson.

At the end of the session, ask everyone to save their slides. They should write down the computer name and filename on the Final Project Development Form. The teacher should keep a backup copy on a USB key drive or at another location. (Note: For the final presentation, all presentations should be on one computer that is hooked up to a projector.) Collect all team folders.

If teams are done making their slides, print them out (print as a handout to put multiple slides per page) and have students take home the printouts. This will help them practice the presentations at home.

**EXTENSIONS**

- If you have a digital camera and/or scanner available, you can help students include photos of their project or scanned images of their sketches on the presentation.
• You may show students how to customize their slides by applying templates or changing colors. You should only do this if the students are done with practicing, as students can get carried away making their slides look exactly the way they want.

• You can also have students use additional communication methods such as poster presentation, a brochure, a web page, user manual, or product packaging. This can be done as an alternative or in addition to the PowerPoint presentation.

• Students’ projects, along with a poster, can be entered into the Science Fair under the Robotics category. See the Massachusetts State Science Fair website for more information (http://www.scifair.com/).

• If you have extra time to spend on practicing presentations, do so. One very effective way of practicing is to video-tape each group’s practice presentation and have the teams look at the video for a self-critique of the presentation.

HOMEWORK

• Continue practicing the presentation. (Print out the presentations as handouts so that students can practice at home.)

• If you are inviting members of the community to see the presentation, then have the students inform parents and friends about the presentation date and time.

EXTRA RESOURCES

• Massachusetts State Science Fair: http://www.scifair.com/
  Describes the guidelines and procedures for entering a robotics project into the State Science Fair.
Lesson 9
Complete Final Project

OBJECTIVES

• Complete the project and presentation, and prepare for presentation day.

BACKGROUND FOR TEACHERS

The eighth step of the engineering design process (highlighted below) allows an engineer to redesign or rework their solution. This step follows the step of communicating the solution so that they can use the feedback from such communication to improve their solution.

It is important that the students are fully prepared with their PowerPoint presentation and have prototype to demo.

TIMING

• Introductory discussion 10 minutes
• Finishing up project 60 minutes

Total: 70 minutes

STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multi-view drawings.

MATERIALS

• LEGO Robotics kit – 1 per team
• Extra LEGO pieces
• Computer with ROBOLAB and PowerPoint – 1 per team
• Depending on students’ projects, you may need additional materials such as tape, ramps, water bottles, etc.
• Extra batteries.
**Handouts & Homework:**
- Final Project Development Form Handout – 1 per team (from previous lesson)

**PREPARATION**
- Check that team folders contain all pages of the Final Project Development Form.

**LESSON GUIDE**

*Introductory discussion*

“Today is a day to catch-up and finish up any loose ends that we may have. In your teams, decide what needs to be done today.”

Have teams spend a few minutes to discuss, then share with the class. Make sure that their to-do list is feasible and on-task – some things they may be doing are:

- finishing up the prototype and/or program
- finishing up presentation
- practicing
- if all else is done, then adding new features to the prototype
- improving the presentation
- helping other teams

*Finish up project*

“Great. Let’s get to work! You have until ___ (time/date) to complete your project. Be sure to fill in the last step in the Final Project Development Form – I will be collecting these forms at the end of the day today.”

Assist the students in finishing their projects and manage the time of this final session accordingly. The students may be doing some final testing and evaluating and may need to do some small redesign work.

It may be necessary to direct some teams to trim down the features on the prototype if they are not going to be able to finish on time. Encourage them that compromise between time and product features is an important part of engineering – engineers are always working against the clock!

At the end of the lesson, collect and keep all of the Final Project Development Forms for evaluation.

**EXTENSIONS**

- You may give as much time as you need to allow students to complete their project – you may even choose to add a day or two to the lesson.

**HOMEWORK**

- Continue to practice the presentation

**EXTRA RESOURCES**

- **RoboLab Reference Guide** contains descriptions about all RoboLab icons. (PDF document...
• **Using RoboLAB** is a comprehensive guide for LEGO Robotics system including hardware and software. (PDF document is on CD)

• **Constructopedia** is a collection of step-by-step building instructions for common assemblies such as a rack-gear system or a lift arm. (PDF document is on CD)
Lesson 10
Final Presentation

OBJECTIVES

• Present their project to a panel of peers, teachers, and/or parents.

BACKGROUND FOR TEACHERS

The final session is not actually a lesson, but a time for the students to present their work in front of an audience. There is a rubric provided the final presentation and project if you wish to grade this part of the project.

Planning

Plan in advance so that you can invite parents, teachers, administrators, other students, and other members of the community (professional engineers, etc.) to attend this session. Invitation flyers should go out several weeks in advance. Plan for a room that is large enough for the number of attendees that you anticipate, and have refreshments and food if appropriate.

Setting up

Be sure to have an overhead projector and a computer. All presentations should be loaded on one computer, or put on a USB drive. Keep all prototypes at the front of the room during the session so that students won’t be playing with their model during other teams’ presentations. Have an extra table for the presenters so that they can put the prototype on the table when they present.

TIMING

• Introduce the project 5 minutes
• Presentations 5-10 minutes per team
• Debrief 10 minutes

STANDARDS ADDRESSED IN THIS LESSON

Massachusetts Science & Technology/Engineering Frameworks (6-8)

2.1 Identify and explain the steps of the engineering design process, i.e., identify the need or problem, research the problem, develop possible solutions, select the best possible solution(s), construct a prototype, test and evaluate, communicate the solution(s), and redesign.

2.2 Demonstrate methods of representing solutions to a design problem, e.g., sketches, orthographic projections, multi-view drawings.

MATERIALS

• Overhead projector hooked up to a computer
• OPTIONAL – Food and refreshments for students and guests.
• Final Project Rubric and Score Sheet
PREPARATION

• Become familiar with the Final Project Rubric and make copies of the score sheet as necessary.
• Several weeks before the presentation, begin inviting teachers, parents, students, etc. to the presentation. You may want to create a flyer or an invitation email.

LESSON GUIDE

Introduce the project
Introduce the curriculum and project to the audience. Points to cover are:

• Today’s presentations are the results of hard work by the students.
• Students spent ~10 weeks to learn engineering and robotics, and they designed and built a prototype of an assistive device of their own conception.
• Hold questions and comments until after each presentation.

Presentations
Introduce each team as they come up. Each team will present their PowerPoint slides, followed by a demonstration of their prototype. Take questions from the audience as time permits.

If using the rubric, take a minute after each presentation to enter the team’s score into the rubric score sheet.

Debrief
Thank the audience for coming. Give the audience some time to look at the devices up close and take pictures.
Section 2:
Student Handouts and Homework for All Lessons
**The Challenge:**

- Design and construct a spatula that is at least 12 inches long using LEGO®s.
- The tip of the spatula must be made using a 8×16 plate.
- The spatula must be able to support a container at the end, which will hold weights.

Type of weight used: ____________________________

- Spatula will be scored based on length and weight held using the following formula:
  
  \[ \text{Your Score} = (\text{Length} - 11) \times \# \text{ of weights held} \]

**Sketch your design below:**

**Calculate your score:**

\[ \text{Your Score} = ( \underline{\text{_________}} - 11) \times \underline{\text{_______}} = \underline{\text{_________}} \]

length in inches # of weights your score

What were some techniques used to make the spatula strong?

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

Copyright 2008 Tufts University Center for Engineering Educational Outreach and LEGOengineering.com HANDOUT 0-1
ENGINEERING DESIGN PROCESS

STEP 1
Identify the Need or Problem
• How can someone’s quality of life be improved?
• How can we make a certain task easier?
• How can we improve upon an existing product?

STEP 2
Research the Need or Problem
• How have other people addressed the need or problem?
• What existing technologies (materials, methods, parts) can we use as part of the solution?
• What are the requirements and constraints?

STEP 3
Develop Possible Solution(s)
• Brainstorm possible solutions to the need or problem.
• Draw on mathematics and science.
• Use your imagination.
• Describe and refine the possible solutions.

STEP 4
Select the Best Possible Solution(s)
• Which solution(s) best meets the requirements and constraints?
• Is the solution realistically feasible?
• Which solution is the most marketable solution?
• Which solution(s) deserve further exploration?

STEP 5
Construct a Prototype
• Model the selected solution(s) in two and three dimensions.
• Model may be a physical model or a computer model, and may not be the same size as actual product.
• Prototype can address some or all of the features in the solution(s).

STEP 6
Test and Evaluate the Solution(s)
• Does the solution work?
• Does the solution meet the original design requirements and constraints?
• Is it safe, reliable, and durable?
• Does the solution appeal to the consumers?

STEP 7
Communicate the Solution(s)
• Make a presentation that discusses the original need/problem and how the solution best meets it.
• How would the product impact the society?
• Communicate the design to manufacturers.
• Describe the product to consumers and users.

STEP 8
Redesign
• How can the design be improved based on things you learned from test and usage?
• Redesign the solution to adapt to newly available technologies or methods.

Copyright 2008 Tufts University Center for Engineering Educational Outreach and LEGOengineering.com
HANDOUT 0-2
Engineering Design Process

1. Choose a simple device that is familiar to you (example: pen, toothbrush, kitchen utensils).
   Name of the device: ____________________________
   Briefly describe the device: what its purpose is, what it is made of, special features, etc.: ____________________________

2. Imagine that you are part of the engineering team that developed the device. In the diagram below, describe what you may have done in each step of the Engineering Design Process as you designed this device.

   Step 1: Identify the Need or Problem
   Step 2: Research the Need or Problem
   Step 3: Develop Possible Solution(s)
   Step 4: Select the Best Possible Solution(s)
   Step 5: Construct a Prototype
   Step 6: Test and Evaluate the Solutions
   Step 7: Communicate the Solution(s)
**DESIGN REQUIREMENTS:**

- The wheelchair must be at least 8 inches tall.
- The chair must be able to roll while holding a ________________.
- The chair must pass a drop test from 3 feet.
- The wheelchair can be made from any pieces in your kit EXCEPT FOR the RCX, motors, sensors, and the LEGO tower.
Sketch your design below:

Which parts stayed together during the drop test? How were these parts put together?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

What parts did not stay together during the drop test? How can these connections be strengthened?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
Putting Things Together

Just like there are different ways to put LEGO pieces together, there are many different ways to put an object together. Look around your school and your house and find examples of different types of fasteners.

Complete the table below for screws, nails, nuts and bolts, and glue. Find another method of fastening and complete the blank row on the bottom.

<table>
<thead>
<tr>
<th>Fastening method</th>
<th>List at least three objects or places where you found this method used</th>
<th>What materials or joints is this method good for?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screws</td>
<td><img src="image" alt="Screws" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nails</td>
<td><img src="image" alt="Nails" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nuts and Bolts</td>
<td><img src="image" alt="Nuts and Bolts" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glue, Cement, or other Adhesives</td>
<td><img src="image" alt="Glue" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Another Method</td>
<td><img src="image" alt="Another Method" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison of Wheelchair Drive Train Models

<table>
<thead>
<tr>
<th>General Characteristics</th>
<th>Criteria #1</th>
<th>Criteria #2</th>
<th>Criteria #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model A</td>
<td>- Compact, low-profile design.</td>
<td>- Four-wheel drive; good on rough or slippery surfaces.</td>
<td>- More power than other designs.</td>
</tr>
<tr>
<td></td>
<td>- Plastic and metal construction.</td>
<td>- Sturdy all-metal construction.</td>
<td>- Large wheels can go over bumps and rough terrains.</td>
</tr>
<tr>
<td></td>
<td>- Small wheels; not good on rough terrains – recommended for mainly indoor use only.</td>
<td>- Fast.</td>
<td>- Plastic and metal construction.</td>
</tr>
<tr>
<td>Model B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Price of Drive-train</td>
<td>Expected lifetime (excluding batteries)</td>
<td>Size</td>
<td>Weight</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Model A</td>
<td>Model A</td>
<td>Model A</td>
<td>Model A</td>
</tr>
<tr>
<td>$490</td>
<td>12 years</td>
<td>40 cm wide</td>
<td>4.5 kg</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60 cm long</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>26 cm high</td>
<td></td>
</tr>
<tr>
<td>Model B</td>
<td>Model B</td>
<td>Model B</td>
<td>Model B</td>
</tr>
<tr>
<td>$700</td>
<td>6 years</td>
<td>40 cm wide</td>
<td>7 kg</td>
</tr>
<tr>
<td></td>
<td></td>
<td>49 cm long</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>30 cm high</td>
<td></td>
</tr>
<tr>
<td>Model C</td>
<td>Model C</td>
<td>Model C</td>
<td>Model C</td>
</tr>
<tr>
<td>$610</td>
<td>10 years</td>
<td>46 cm wide</td>
<td>6.2 kg</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70 cm long</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>45 cm high</td>
<td></td>
</tr>
<tr>
<td>Speed &amp; Maneuverability</td>
<td>Reliability</td>
<td>Safety</td>
<td>Battery Life &amp; Cost</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>--------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Model A</td>
<td>Model A</td>
<td>Model A</td>
<td>Model A</td>
</tr>
<tr>
<td>4 mph</td>
<td>10</td>
<td>2</td>
<td>Charge lasts approx.</td>
</tr>
<tr>
<td></td>
<td>Breakage of plastic parts during falls or hard hits.</td>
<td>No serious injuries.</td>
<td>12 hours.</td>
</tr>
<tr>
<td></td>
<td>Moderate turning radius; fairly maneuverable.</td>
<td></td>
<td>Suggested replacement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>every 10 months at $280</td>
</tr>
<tr>
<td>Model B</td>
<td>Model B</td>
<td>Model B</td>
<td>Model B</td>
</tr>
<tr>
<td>6 mph</td>
<td>16</td>
<td>5</td>
<td>Charge lasts approx.</td>
</tr>
<tr>
<td></td>
<td>Rust and corrosion when used in humid areas.</td>
<td>No serious injuries.</td>
<td>10 hours.</td>
</tr>
<tr>
<td></td>
<td>Gears need lubrication if used in high-dust area.</td>
<td></td>
<td>Suggested replacement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>every 12 months at $400</td>
</tr>
<tr>
<td>Model C</td>
<td>Model C</td>
<td>Model C</td>
<td>Model C</td>
</tr>
<tr>
<td>3 mph</td>
<td>24</td>
<td>4</td>
<td>Charge lasts approx.</td>
</tr>
<tr>
<td></td>
<td>Breakage of plastic parts during falls or hard hits.</td>
<td>One serious injury caused</td>
<td>8 hours.</td>
</tr>
<tr>
<td></td>
<td>Gears need lubrication if used in high-dust area.</td>
<td>by falling.</td>
<td>Suggested replacement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>every 12 months at $340</td>
</tr>
</tbody>
</table>
Materials

Large Parts
- (2) 1×8 plates
- (2) 2×2 plates
- (2) 2×4 plates
- (5) 2×8 plates
- (4) 1×6 beams
- (2) 1×16 beams
- (1) RCX
- (2) motors
- (2) 40-tooth gears
- (2) medium wheels
- (2) pulley wheels

Small Parts
- (2) ½ bushings
- (2) 8-tooth gears
- (2) axle pins
- (2) friction pins
- (2) ¾ pins
- (2) axle 6

- (2) 1×6 beams
- (2) 1×16 beams
- (3) 2×8 plates
- (2) axle 6
- (2) ½ bushings
- (2) 2×8 plates
• (2) 2×2 plates
• (2) 1×8 plates

• (2) motors
• (1) 2×4 plate

• (2) 8-tooth gear
• (2) 40-tooth gear
• (2) axle pins

• (2) medium wheels
• (2) pulley wheels

• (1) RCX
• (1) 2×4 plate
• (2) short cables (not shown)
• (2) friction pins
• (2) ¾ pins

• (2) 1×6 beams

COMPLETED MODEL
Model B Instructions

Large Parts

- (1) RCX
- (2) motors
- (4) 40-tooth gears
- (2) 24-tooth gears
- (4) medium wheels

- (4) 2×8 plates
- (4) 2×4 plates
- (8) 2×2 plates
- (2) 1×4 plates
- (2) 2×6 bricks
- (12) 1×4 beams

Small Parts

- (4) bushings
- (8) 1×2 plate w/ rails
- (4) axle 6

Copyright 2008 Tufts University Center for Engineering Educational Outreach and LEGOengineering.com
Model B Instructions

Page 2

Slide onto motor as shown

• (2) motors

Page 3

• (4) 2×8 plates
• (2) 1×4 plates

• (4) axle 5
• (4) bushings
Model B Instructions

Page 4

• (2) 2×6 bricks
• (4) 40-tooth gears
• (2) 24-tooth gears

Page 5

• (4) medium wheels
• (1) RCX
• (2) short cables (not shown)

COMPLETED MODEL
Model C Instructions

Materials

Large Parts

• (1) RCX
• (2) motors
• (4) 40-tooth gears
• (2) 24-tooth gears
• (2) medium wheels
• (2) large wheels
• (2) 1×16 beams
• (2) 1×8 beams
• (1) 1×6 beam
• (1) 1×2 beams
• (1) 2×4 plates
• (4) 2×8 plates
• (5) 2×4 bricks
• (1) 2×8 brick

Small Parts

• (4) bushings
• (4) axle pins
• (4) 8-tooth gears
• (4) axle 7
• (2) axle 4

Model C Instructions

(1) 1×16 beam
(1) axle 7
(1) axle 4
(2) 40-tooth gear
(1) 24-tooth gear
(1) 8-tooth gear
(1) bushing
(1) axle pin
• (1) 1×16 beam
• (1) axle 7
• (1) axle 4
• (2) 40-tooth gear
• (1) 24-tooth gear
• (1) 8-tooth gear
• (1) bushing
• (1) axle pin

• (2) 1×8 beams
• (2) bushings

• (1) 2×4 plate
• (3) 2×8 plates
• (1) 2×8 plate
• (1) 1×6 beam
• (1) 1×2 beam

• (2) motors

• (2) 8-tooth gears
• (1) 2×8 brick
• (5) 2×4 bricks
• (2) large wheels
• (2) axle pins
• (2) medium wheels

Model C Instructions  Page 6

COMPLETED MODEL
<table>
<thead>
<tr>
<th>Other similarities and differences</th>
<th>Durability and ability to withstand crash</th>
<th>Ability to go around corners and obstacles</th>
<th>Ability to go up a ramp</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Orthograpic Drawings of Model A

LEFT

FRONT

RIGHT

BOTTOM

BACK

TOP
Orthographic Drawings

1. Choose a simple three-dimensional object that appears differently from different views. Some examples are:
   - cellphone
   - camera
   - stapler
   - computer monitor or TV
   - remote control
   - shoe or sneaker
   - alarm clock

Name of the object: ____________________________

Brief description of the object: ____________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

2. Orthographic drawings are drawings of the object seen from different sides. The example below shows the orthographic drawings of a stacked-cube.

   Isometric view of stacked cube and orthographic projections

   Orthographic drawings of the stacked cube

In the space provided, sketch the six orthographic views of the object chosen above.

FRONT VIEW

BACK (REAR) VIEW
Gears Exploration

1. Look around your house, school, and the neighborhood. Find as many devices as you can with gears. List the names of devices that you found:

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

2. Choose one object from the list above, and briefly describe what it is and what it does:

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

   What type(s) of gears did it have? What material(s) were they made of?

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

   How do the gears help this device function properly?

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

   ___________________________________________________________

3. Sketch the device and the gears below. Use multiple orthographic views if necessary.
Construct the following model and use it to answer Questions 1 ~ 6 below:

**Step 1:** Assemble one long beam, two axles, one 40-tooth gear, one 8-tooth gear, and two pulleys as shown.

**Step 2:** Insert one ¾-pins (pins that have one full-length side and a short side) into each pulley. The pins can serve as a handle, and will also help you keep track of how many times a gear has turned.

1. Experiment turning the axles. Write down any observations you make.

   __________________________________________________________
   __________________________________________________________
   __________________________________________________________
   __________________________________________________________
   __________________________________________________________

2. Which gear turns faster? __________________________________________________________

3. Which axle is easier to turn (try turning the pulleys by using the pins as a handle)? ______________________

   Have your teammate hold the pulley wheel on the 8-tooth gear while you try to turn the pulley wheel on the 40-tooth gear. Now switch so that you hold the 40-tooth gear while your teammate turns the 8-tooth gear. What do you notice?

   __________________________________________________________
   __________________________________________________________
   __________________________________________________________

   Which gear gives you more power? __________________________________________________________

4. When you turn the 40-tooth gear (larger one) by one rotation, how many times does the 8-tooth gear (smaller one) rotate?

5. The **driver gear** is the gear to which power is applied, and the **follower gear** is the gear that produces the output motion.

   If you are making a car that goes very fast, which gear should you attach the motor to? (This is your driver gear)
6. The **gear ratio** is defined as the number of teeth on the driver gear divided by the number of teeth on the follower gear. Find the gear ratio for your gear train.

\[
gear \ ratio = \frac{\text{# of teeth on the driver gear}}{\text{# of teeth on the follower gear}}
\]

How is the gear ratio related to your answer in Question 4?

---

◆ Change the 8-tooth gear to a 24-tooth gear (you will need to move the axle to a different hole). Answer Question 7 below.

7. When you turn the 40-tooth gear by three rotations, how many times does the 24-tooth gear rotate?

Based on your answer above, how many times do you think the 24-tooth gear will rotate if you rotate the 40-tooth gear once?

What is the gear ratio for this gear train?

◆ Add an 8-tooth gear next to the 24-tooth gear. Move the pulley wheel from the 24-tooth gear to the 8-tooth gear so that you get a gear train that looks like the picture on the right (You may need to switch to a longer beam). Place a bushing on the 24-tooth gear axle so that it stays in place.

Answer the following questions:

8. When you turn the 40-tooth gear by one rotation, how many times does the 8-tooth gear rotate?

What is the gear ratio for this gear train?
9. How does your answers to Question 8 compare to your answers in Questions 4 and 6? What does this tell you about the 24-tooth gear?

In this gear train, the 24-tooth gear is an idler gear, and has no effect on the overall gear ratio.

10. Try turning the two pulley wheels – which one is easier to turn? Why?

◆ On a long beam, build a gear train with at least four gears.

Sketch the gear train below. Write the number of teeth on each gear, and indicate which is the driving gear and which is the follower gear.

11. Complete the following table using your gear train:

<table>
<thead>
<tr>
<th># of turns on the Driver Gear</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td># of turns on the Follower Gear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12. Calculate the gear ratio for the gear train using the following formula:

\[
gear \text{ ratio} = \frac{\text{# of teeth on the driver gear}}{\text{# of teeth on the follower gear}} = \frac{\text{# of teeth on the driver gear}}{\text{# of teeth on the follower gear}}
\]

How are your answers in Question 11 and 2 related? ________________________________

Copyright 2008 Tufts University Center for Engineering Educational Outreach and LEGOengineering.com
ROBOLAB Programming

The following program is for a two-motor car with the left motor connected to A and right motor connected to C.

1. Describe what the car will do when you run this program:

2. Rearrange the following icons to make a program that will make the car go forward for a certain time, reverse until it’s halfway back to the start line, then stop. Connect the top icons to the appropriate place on the bottom row. The car has two motors – the left motor is connected to A, and the right motor is connected to C. Each icon is used once. The first icon is already placed for you.
**RoboLAB Cheat Sheet**

### Keyboard Short-Cuts

- **Ctrl-B** Removes all broken wires in the program.
- **Ctrl-E** Brings up the program diagram.
- **Ctrl-H** Opens/closes the Context Help window.
- **Ctrl-L** Opens the Error List (same as clicking on a broken arrow).
- **Ctrl-R** Downloads the program to the RCX.
- **Ctrl-Z** Undo.
- **Space Bar** Toggles between an arrow and a wire spool.
- **Tab** Scrolls through all tools – hand, arrow, text, wire spool.

### General Information

- Use Context Help (Ctrl-H) and point to any icon to get information about it. In Context Help window, click on “Click here for more help” to see an example of how the icon is used.
- To download program to RCX, click on the white arrow on top-left corner of the window or press Ctrl-R. Make sure that the RCX is ON and is set to Programs 3, 4, or 5, and the tower and RCX are in direct view.
- Use the arrow cursor to select/move icons and wires. Use the wire spool cursor to wire icons together. Use the hand or text tool to modify a number in a box. The text tool can also be used to make notes or comments on the program.

### Frequently Asked Questions & Answers

<table>
<thead>
<tr>
<th>Q: My Functions Palette or Tools Palette disappeared! What do I do?</th>
<th>A: Go to Window → Show Functions Palette or Show Tools Palette. You can also make the Functions Palette appear by right-clicking anywhere on the page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: I see a broken arrow. What is wrong with my program?</td>
<td>A: Click on the broken arrow (or type Ctrl-L) to open the Error List. Click on the error(s) to identify the location of the error.</td>
</tr>
<tr>
<td>Q: Why isn’t my sensor working?</td>
<td>A: Check that the sensors are connected to one of the input ports (1, 2, or 3) and that the correct port number is wired into each of the program’s sensor icons. If using a touch sensor, make sure that the cable is connected to the top side of the sensor.</td>
</tr>
<tr>
<td>Q: My program window disappeared! What do I do?</td>
<td>A: Type Ctrl-E or go to Window → Show Diagram to make the program reappear.</td>
</tr>
<tr>
<td>Q: There really is nothing wrong with my program but I still get a broken arrow. What should I do?</td>
<td>A: First, check the Error List by clicking on the broken arrow and click on the error that appears. If it identifies an icon, then delete the icon and re-wire it even if it looks like there’s nothing wrong with it. If the Error List does not recognize any icons, try typing Ctrl-B to remove all broken wires – sometimes, broken wires get hidden behind icons in places you can’t see. Also try moving the Tools/Functions palettes to see if there is an unused icons hidden behind the windows. If all else fails, open a new window and rewrite the program.</td>
</tr>
</tbody>
</table>
Programming with Sensors

1. A car with a light sensor mounted in the front travels down a path with three black lines. The car has two motors (A and B), and the light sensor is connected to input port 1.

![Diagram of a car with light sensor and three lines]

The following program controls the car above.

![Program control diagram]

Describe what the car will do when you run this program:
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

2. The robot on the right has one hand-held touch sensor attached by a long cable, and another touch sensor that is attached to the front of the car. The car has two motors – motors A and B.

We want to program the car to do the following:
- When you press touch sensor 1, the car moves forward.
- Then, when the car hits the wall, it reverses direction.
- When touch sensor 1 is pressed again, the car stops and the program ends.

![Diagram of a robot with touch sensors and a wall]

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Use the icons above to program the car. Use the letters above each icon to symbolize the icon. Icons may be used more than once.
Final Project Development Form

Team Members: ________________________________
Company Name (Optional): ________________________________

Step 1: Identify a need or problem that your project will address. Describe the need/problem:
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

Step 2: Research the need or problem by answering these questions below (not everyone will have answers to all questions, depending on the need/problem that you identified).
- Are there existing products that address the same or similar need/problem? If yes, describe them below and explain why this solution is not the perfect solution.
- Describe the nature of the disability that causes the need/problem. Is it caused by a disease, injury, both, or something else? What age groups are affected? How many people are impacted by this disability?
- Are there some existing technologies (parts, devices, etc) that can be used as part of your solution?
- What could potential customers be looking for?
- Write down all of your sources – websites, books, etc.
- Please write down any other important facts you found during the research.
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

List the design requirements and constraints that your solution must meet:


Step 3: Develop possible solutions by brainstorming. Write down everything that comes to your mind, even if it seems impossible. You can always take it off the list later. You may also sketch an idea (use the back of the paper).
**Step 4: Select the best possible solution** by first choosing the top three ideas from your brainstorming, then completing the following table:

<table>
<thead>
<tr>
<th>Brief description</th>
<th>Pro’s</th>
<th>Con’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idea 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idea 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idea 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Describe your best possible solution below, and explain why this is the best solution.

Sketch your solution below:
**Step 5: Construct a prototype** of your solution using LEGO. A prototype does not need to show all of the product features. Write down below what product feature(s) you want to include in your prototype:

________________________________________________________________________

________________________________________________________________________

Describe how you plan on prototyping the feature(s) listed above. Be as specific as possible: ______________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Sketch what your prototype will look like:

__________________________

__________________________

__________________________

**Step 6: Test and evaluate the solution** and answer the following questions:

Did your solution meet the original design requirements that were stated in Step 2? Please explain.

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Did your prototype successfully demonstrate the feature(s) that you wanted to test? Please explain.

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
Please describe some of the tests/evaluations that you performed on the prototype, and the results. Were there any modifications you made as a result of the test?

__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

**Step 7: Communicate the solution** by creating a PowerPoint presentation.
Write down your file location: _______________________________________________

**Step 8: Redesign** is a crucial step in finalizing a design. Describe some of the changes you might make to your original design plan (Step 4) based on what you found during your prototyping and testing.

__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

Sketch and describe the new/modified solution below:
Final Project Ideas

Wheelchair Modifications
• a wheelchair that would let people stand up or turn into a bed
• a wheelchair that would climb stairs
• self-guided wheelchair

Everyday Convenience
• something that helps a person reach or grab something
• assistance with things that require fine motor skills
• tools to help the disabled use the kitchen or cook something
• tools to help the disabled use the shower or bathroom

Fun & Games
• allow users to participate in sports or games that they otherwise would have difficulties participating in
• tools to help a person with impaired motor skills use devices such as an i-Pod or game controller
Brainstorming

1. Write down the need or problem that your team is going to address (from Step 1 of the Final Project Development Form):

________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________

2. Brainstorm possible solutions for the need or problem. Be creative, and write down anything that comes to your mind, even if it seems impossible. Use sketches if it helps express your imagination.

________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________

3. Interview a friend or family member to ask for their ideas on how they might solve your problem/need.

________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
Presentation Title or Name of Assistive Device

Company name
Names of people in the company
Date of presentation

Introduction

• The need/problem you are addressing (EDP Step 1)
• Background research (EDP Step 2)
  – who will need your invention?
  – existing solutions and their shortfalls
• Your solution
  – why this is the best solution

The Prototype

• Features of the prototype (EDP Step 5)
  – feature 1
  – feature 2 …
• Programming
  – program features
• Design requirements & constraints

Future

• Testing & Evaluation (EDP Step 6)
  – what tests were done
  – what tests to perform
• Proposed redesign (EDP Step 8)
  – points of improvement
  – features that were not part of this prototype
Presentation Title or Name of Assistive Device

Company name
Names of people in the company
Date of presentation

Introduction
• The need/problem you are addressing (EDP Step 1)
• Background research (EDP Step 2)
  – who will need your invention?
  – existing solutions and their shortfalls

Our Solution
• Ideas of possible solutions (EDP Step 3)
  – Idea 1
  – Idea 2 …
  – Best idea
• Why you chose your idea (EDP Step 4)
  – what features it will have
  – design requirements and constraints

The Prototype
• Features of the prototype (EDP Step 5)
  – feature 1
  – feature 2 …
• Uses of the prototype
• Limitations of the prototype
  – features not shown in prototype but will be a part of actual product

Programming
• Programming features (ROBOLAB)

Future
• Testing & Evaluation (EDP Step 6)
  – what tests were done
  – what tests to perform
• Proposed redesign (EDP Step 8)
  – points of improvement
• Future considerations for production & manufacture, marketing, etc.
# PRESENTATION DO’S AND DON’TS

<table>
<thead>
<tr>
<th>Do’s</th>
<th>Don’ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Speak clearly, with good volume and pace.</td>
<td>• Speeding through too many slides in a short time: 1 minute per slide (except for the title slide) is usually a good estimate.</td>
</tr>
<tr>
<td>• Introduce all speakers at the beginning.</td>
<td>• Using too much notes – all the cues that you need should be on the slides.</td>
</tr>
<tr>
<td>• Ask for questions at the end of the presentation.</td>
<td>• Using too much humor – the presentation becomes too distracting and you may lose credibility.</td>
</tr>
<tr>
<td>• Look at the audience, and look around to address the whole room.</td>
<td>• Standing in front of the slide; blocking your presentation.</td>
</tr>
<tr>
<td>• Practice and time yourselves.</td>
<td>• Talking to one person in the audience – you should look around the audience, not stare at just one person.</td>
</tr>
<tr>
<td>• Keep your hands out of your pockets.</td>
<td>• Talking too fast, too slow, or too softly.</td>
</tr>
<tr>
<td>• It’s okay to say &quot;I don’t know&quot; when asked a question that you don’t know the answer to. This is far better than trying to make something up.</td>
<td>• Talking or looking distracted when your group member is presenting. Be attentive when your partner is presenting.</td>
</tr>
<tr>
<td>• Use humor (to a certain extent), but only if you feel comfortable doing so.</td>
<td>• Memorizing the presentation – if you memorize the entire speech, then 1) it doesn’t sound natural, and 2) you are more likely to get stuck if someone stops you with a question. It’s better to memorize the content but not the exact words.</td>
</tr>
<tr>
<td>• Use demonstrations or props where relevant.</td>
<td>• Overcrowding your slides with information. The absolute smallest font size that you should be using is 14 pt, but you should try not to go below 18 pt.</td>
</tr>
<tr>
<td>• Involve the audience if appropriate (ask questions, ask for a volunteer, etc).</td>
<td>• Using fancy fonts that are difficult to read.</td>
</tr>
<tr>
<td>• When presenting with multiple presenters, plan in advance about who is presenting each slide.</td>
<td>• Using paragraphs and full sentences in the slides.</td>
</tr>
<tr>
<td>• Pay attention to what your team member is saying and look attentive even when it is not your turn to speak.</td>
<td>• Using too much irrelevant animations and sound effects that may distract the audience.</td>
</tr>
<tr>
<td>• Check available equipment – make sure that your presentation is compatible with the equipment that you will be using. This includes checking what PowerPoint version is available, checking for internet or appropriate disk drive, speakers, etc.</td>
<td>• Including irrelevant figures or pictures.</td>
</tr>
<tr>
<td>• Spell-check all slides.</td>
<td>• Using conflicting colors – for example, red font on gray background will blind the audience.</td>
</tr>
<tr>
<td>• Make your slides look &quot;pleasant&quot; – audience is much more likely to pay attention to a visually attractive slide. Choose a design based on what the audience will like, not what you personally like. A neutral design works better than a flashy design.</td>
<td>• Using “chat language”</td>
</tr>
<tr>
<td>• Use bullets and short phrases.</td>
<td>• Pictures and figures are good, but only when they’re relevant.</td>
</tr>
<tr>
<td>• Keep texts concise. Use short bullet points, not paragraphs.</td>
<td>• Use neutral colors to appeal to the entire audience.</td>
</tr>
<tr>
<td>• Use the same background, font color, and font style on all slides (unless it’s necessary to change them to make a point).</td>
<td>• Overcrowding your slides with information. The absolute smallest font size that you should be using is 14 pt, but you should try not to go below 18 pt.</td>
</tr>
<tr>
<td>• Pictures and figures are good, but only when they’re relevant.</td>
<td>• Using fancy fonts that are difficult to read.</td>
</tr>
<tr>
<td>• Use neutral colors to appeal to the entire audience.</td>
<td>• Using paragraphs and full sentences in the slides.</td>
</tr>
</tbody>
</table>

---

**Speech Tip:**

- **Speak clearly, with good volume and pace.**
- **Introduce all speakers at the beginning.**
- **Ask for questions at the end of the presentation.**
- **Look at the audience, and look around to address the whole room.**
- **Practice and time yourselves.**
- **Keep your hands out of your pockets.**
- **It’s okay to say "I don’t know" when asked a question that you don’t know the answer to. This is far better than trying to make something up.**
- **Use humor (to a certain extent), but only if you feel comfortable doing so.**
- **Use demonstrations or props where relevant.**
- **Involve the audience if appropriate (ask questions, ask for a volunteer, etc).**
- **When presenting with multiple presenters, plan in advance about who is presenting each slide.**
- **Pay attention to what your team member is saying and look attentive even when it is not your turn to speak.**
- **Check available equipment – make sure that your presentation is compatible with the equipment that you will be using. This includes checking what PowerPoint version is available, checking for internet or appropriate disk drive, speakers, etc.**

**Slides & Visuals Tip:**

- **Spell-check all slides.**
- **Make your slides look "pleasant" – audience is much more likely to pay attention to a visually attractive slide. Choose a design based on what the audience will like, not what you personally like. A neutral design works better than a flashy design.**
- **Use bullets and short phrases.**
- **Keep texts concise. Use short bullet points, not paragraphs.**
- **Use the same background, font color, and font style on all slides (unless it’s necessary to change them to make a point).**
- **Pictures and figures are good, but only when they’re relevant.**
- **Use neutral colors to appeal to the entire audience.**
## Final Presentation Rubric

<table>
<thead>
<tr>
<th>No (1)</th>
<th>Yes, but... (2)</th>
<th>Yes (3)</th>
<th>Yes, and... (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Robot Design</strong></td>
<td><strong>Robot Design</strong></td>
<td><strong>Robot Design</strong></td>
<td><strong>Robot Design</strong></td>
</tr>
<tr>
<td>- The design does not meet the guidelines in the &quot;Final Project Guidelines&quot;</td>
<td>- The device meets all the guidelines in the &quot;Final Project Guidelines&quot;</td>
<td>- The device meets all the guidelines in the &quot;Final Project Guidelines&quot;</td>
<td>- The device meets all the guidelines in the &quot;Final Project Guidelines&quot; and exceeds the requirements (i.e., demonstrates multiple features, contains multiple sensors, etc.)</td>
</tr>
<tr>
<td>- The prototype does not work, and the team is unable to explain why it doesn’t work and how to fix it.</td>
<td>- The prototype does not work consistently, but the team is able to describe why it does not work and suggest ways to fix it.</td>
<td>- The prototype works and works consistently (i.e., is repeatable)</td>
<td>- The prototype works and works consistently (i.e., is repeatable)</td>
</tr>
<tr>
<td>- The prototype is not structurally sound and breaks or falls apart easily.</td>
<td>- The prototype has some components that break or fall apart easily, but the team is able to describe the problem and suggest ways to fix it.</td>
<td>- The prototype is structurally sound.</td>
<td>- The prototype is structurally sound.</td>
</tr>
<tr>
<td><strong>Engineering Design Process</strong></td>
<td><strong>Engineering Design Process</strong></td>
<td><strong>Engineering Design Process</strong></td>
<td><strong>Engineering Design Process</strong></td>
</tr>
<tr>
<td>- The team has not documented their use of the EDP in developing the prototype.</td>
<td>- The team has some documentation about the EDP steps used in developing their prototype.</td>
<td>- The team has documented their device using each step of the EDP.</td>
<td>- The team has documented their device using each step of the EDP and has gone above and beyond in documenting some steps of the EDP (i.e. testing protocols and results).</td>
</tr>
<tr>
<td>- The team did not follow the EDP.</td>
<td>- The team has used the EDP in developing the prototype.</td>
<td>- The team has used the EDP in developing the prototype.</td>
<td>- The team has used the EDP in developing the prototype.</td>
</tr>
<tr>
<td>- The team did not share the work effectively and did not show good teamwork.</td>
<td>- The team developed good teamwork and shared work effectively most of the time.</td>
<td>- The team developed good teamwork and shared work effectively.</td>
<td>- The team developed good teamwork and shared work effectively.</td>
</tr>
<tr>
<td><strong>Presentation</strong></td>
<td><strong>Presentation</strong></td>
<td><strong>Presentation</strong></td>
<td><strong>Presentation</strong></td>
</tr>
<tr>
<td>- The final presentation does not meet all of the guidelines.</td>
<td>- The final presentation meets all the guidelines.</td>
<td>- The final presentation meets all the guidelines.</td>
<td>- The final presentation meets all the guidelines.</td>
</tr>
<tr>
<td>- The slides were unclear or not well organized.</td>
<td>- Some aspects of the presentation (i.e., slides or speech) were unclear or not well organized.</td>
<td>- The slides were organized, clear, and effective.</td>
<td>- The slides were organized, clear, and effective.</td>
</tr>
<tr>
<td>- The speakers were not clear.</td>
<td></td>
<td>- The team communicated their ideas effectively and clearly.</td>
<td>- The team communicated their ideas effectively and clearly.</td>
</tr>
<tr>
<td><strong>Analysis</strong></td>
<td><strong>Analysis</strong></td>
<td><strong>Analysis</strong></td>
<td><strong>Analysis</strong></td>
</tr>
<tr>
<td>- The team does not have or can not explain the rationale behind design decisions.</td>
<td>- The team can explain some of the rationale behind design decisions.</td>
<td>- The team can explain the rationale behind design decisions.</td>
<td>- The team can explain the rationale behind design decisions.</td>
</tr>
<tr>
<td>- The team can not explain the limitations of their prototype.</td>
<td>- The team can explain some limitations of their prototype.</td>
<td>- The team can explain the limitations of their prototype.</td>
<td>- The team can explain the limitations of their prototype.</td>
</tr>
<tr>
<td>- The team can not suggest areas of future improvement.</td>
<td>- The team has some ideas about future improvements on the prototype.</td>
<td>- The team has ideas about future improvements on the prototype.</td>
<td>- The team has ideas about future improvements on the prototype.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The team demonstrates deep understanding of the problem/need, the disability that is associated with it, and how their innovation can help this problem.</td>
</tr>
</tbody>
</table>
Section 3:
Supplemental Teacher Resources
# Supplemental Teacher Resources

## Table of Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROBOLAB Installation Guide for PC’s</td>
<td>1</td>
</tr>
<tr>
<td>LEGO Mindstorms Electric Components</td>
<td>5</td>
</tr>
<tr>
<td>Guide to ROBOLAB Pilot 1-4</td>
<td>9</td>
</tr>
<tr>
<td>ROBOLAB Pilot Challenges</td>
<td>15</td>
</tr>
<tr>
<td>Guide to Robolab Inventor 3</td>
<td>22</td>
</tr>
<tr>
<td>ROBOLAB Inventor 3 Icons</td>
<td>29</td>
</tr>
<tr>
<td>ROBOLAB Inventor 3 Challenges</td>
<td>38</td>
</tr>
<tr>
<td>ROBOLAB Troubleshooting Guide</td>
<td>40</td>
</tr>
<tr>
<td>ROBOLAB Cheat Sheet</td>
<td>47</td>
</tr>
<tr>
<td>More About Gears</td>
<td>49</td>
</tr>
<tr>
<td>Robo Technology Set Contents</td>
<td>55</td>
</tr>
<tr>
<td>Useful Web Links</td>
<td>57</td>
</tr>
</tbody>
</table>
ROBOLAB 2.5.4 Installation Instructions for PC

1. Log in as an Administrator (or any account with Admin privileges). ROBOLAB WILL INSTALL CORRECTLY ONLY IF YOU HAVE ADMIN PRIVILEGES! If you do not have an admin account, please contact the tech support person at your school.

2. Locate a file called “PC-Install” on the CD, and double-click it to run it. This will begin the installation process.

3. Follow prompts until installation begins. Let the installation process run by itself... this may take a while.

4. Eventually, a new window will open up and will ask you to install WinVDIG – say Yes (or OK) and follow prompts to install WinVDIG.

5. Another prompt will ask you if you want to install the LEGO USB Tower driver files – click on OK.

6. After the USB Tower driver is installed, you will get the following message:

7. After the installation is complete, it will prompt you to restart the computer. Click on OK to restart (if the Restart prompt doesn’t come up, then the installation is not complete yet – let it run until the prompt comes up).

8. After the computer restarts, log on as Administrator again. Plug in a USB tower into one of the USB ports. If you get a window saying “Found New Hardware
Wizard”, it means that the tower driver did not install properly and you need to manually install it. Follow steps 10~13 below to do this.

If you don’t get the “Found New Hardware Wizard” window, then skip to Step 14.

9. To install the Tower driver... In the Found New Hardware Wizard, select “No, not this time” and click Next.

10. Select “Install from a list or specific location (Advanced)” and click Next.

11. - Click on “Search for the best driver in these locations”
- Un-select “Search removable media (floppy, CD-ROM...)”
- Select “Include this location in the search” and click on Browse.

In the window that pops up, go into the ROBOLAB CD, locate a folder called "Tower" and select it. Click Next.
12. The Wizard will locate the tower driver. Follow prompts to install the driver. When installation is complete, you should get a prompt saying “Driver successfully installed” (or something like that).

13. To check that the USB Tower is correctly installed, open up ROBOLAB 2.5.4 and go into Administrator.

![ROBOLAB menu](image1)

14. Click on SELECT COM PORT. In the window that opens up, make sure that “USB #1” is one of the available options. Click on the green check mark to make that your default port.

![SELECT COM PORT](image2)

---

**Notes:**

- The USB tower should be plugged in before starting ROBOLAB.

- In some PC’s, the USB tower will only work with the USB port that was used when you installed the driver. If you want it to work in any USB port, you must repeat Steps 10~13 for each USB port on your computer.
LEGO Mindstorms Electric Components

**RCX**
RCX is the main “brain” for LEGO-based robots. Its main components are shown in the diagram below.

- **Buttons:**
  - On·Off: Turns RCX on and off.
  - Prgm: Scrolls through Programs 1~5.
  - Run: Press to Run the program or to Abort.
  - View: Scrolls through various view modes.

- **Inputs (1,2,3):**
  Attach sensors here.

- **Outputs (A,B,C):**
  Attach motors and/or lights here.

- **IR window:**
  Used to communicate with tower.

- **Cover:**
  Remove cover to replace batteries. RCX uses 6 AA batteries.

**RCX Display**
- Display contains a little person. The person will appear to be walking if the RCX is running a program.
- The number on the right side of the screen is the program number (1~5).
- The number to the left can mean different things, depending on the View setting. Press View button to see sensor values.
- RCX will show a low battery sign (battery mark with an “X” on it) when the battery begins to run low.

**Changing Batteries**
- Remove the rear cover to replace batteries. Batteries should be changed one at a time to prevent the RCX from losing its memory (the RCX takes six AA batteries).

**CAUTION:**
- **NEVER drop an RCX.** When testing cars with RCX on board, be sure to test on the floor or while holding it up so that it does not accidentally fall off the table.

**Motors**
Motor are used for creating any kind of motion for LEGO robots.
- Use a cable to connect the motor to A, B, or C.
- If the motor is running in the wrong direction (i.e., the car is not moving in the “forward” direction), flip the orientation of the cable lead by 180°. This will change the motor direction. Alternatively, motor direction can also be changed inside the program.
- Motor power can be varied from 1 to 5 (1 to 100 if using ROBOLAB 2.9).
- Due to speed variations between individual motors, two-motored car will have a tendency to curve slightly to one direction. If you need a car that runs straight, try a different pairs of motors until you find a pair with matching speed, or use a lower power setting on the faster motor.
**Touch Sensor**

There are two **touch sensors** included in a kit.

To use a touch sensor, connect a cable to the top side of the sensor. Touch sensors can be used as buttons, switches, smart bumpers, etc.

**Sample Touch Sensor Icons**

- **Wait for Push**
  Waits until the touch sensor button is pressed in.

- **Wait for LetGo**
  Waits until the touch sensor button is let go.

- **Touch Sensor Fork**
  Goes to the top path if touch sensor is not pressed and the bottom path if touch sensor is pressed in.

**Light Sensor**

Each kit contains one **light sensor**.

Light sensor contains two things: a LED light source, and a light detector. The **light source** turns on when the sensor is in use to illuminate the area that is in question. The **sensor** collects light and returns a value between 0 (dark) and 100 (bright).

The sensor’s reading is a combination of: 1) ambient light and 2) the LED’s light that is reflected off the surface and bounces back into the sensor. Because of this, a light sensor’s reading is not always intuitive – a smooth dark-colored surface (such as a dark floor tile) may return a higher number than a rough light-colored surface (such as a light-colored carpet).

Light sensors are most often used for line-following applications. Other uses include differentiating different colors or darkness and detecting an object passing in front of the sensor (i.e., counting how many people have passed though a doorway).

**Sample Light Sensor Icons**

- **Wait for Brighter**
  Waits until light sensor sees a change of +5 from initial value.

- **Wait for Darker**
  Waits until light sensor sees a change of -5 from initial value.

- **Light Sensor Fork**
  Compares light sensor value to a prescribed number (default is 55) and goes to top path if light is brighter than this number and bottom path if the light is darker than this number.

- **Wait for Light** *(Inventor 4 only)*
  Waits until the light sensor value is equal to or greater than the specified number (default is 55).

- **Wait for Dark** *(Inventor 4 only)*
  Waits until the light sensor value is equal to or less than the specified number (default is 55).
**Rotation Sensor**

**Rotation sensors** can count the amount that an axle has turned. To use a rotation sensor, insert the axle of interest into the hole in the sensor. The rotation sensor returns a number to the RCX. One full rotation equals 16 counts. Rotation sensor must be “zero-ed” before use.

Rotation sensor offers a good alternative to wait-for-time commands because the travel distance is not affected by motor performance variation due to battery levels.

Rotation sensors are not part of a kit, and are sold separately.

**Sample Rotation Sensor Icons** *(Note: Rotation sensor can be used with Inventor 4 only.)*

- **Zero Angle Sensor**
  - Initializes the angle sensor to zero. This icon MUST be used before using other rotation sensor to ensure that the sensor begins counting at zero.

- **Wait for Rotation**
  - Waits until rotation sensor count equals or exceeds the number specified in either direction. 1 rotation = 16 counts.

**General Comments about Sensors**

- All sensors must be connected to input ports 1, 2, or 3 on the RCX (or 1~4 on NXT).
- All sensor icons in Inventor have a default port setting of port 1. If the sensor is connected to any other port, a modifier must be strung into the sensor icon to specify the port.
- With the light sensor and rotation sensor, the value that the sensor reads can be shown on the RCX display by pressing the View button on the RCX until a small arrow appears below the appropriate input port.
**RoboLAB Pilot 1~4**

**Starting Up**

1. Connect the USB tower to the computer.
2. Open RoboLAB, and click on Programmer.
3. In the next screen, double-click on the appropriate Pilot level.

**Pilot 1: One Output, One Step**

**Choose One output (Port A):**
- Motor Forward
- Motor Reverse
- Light On

**Choose One Wait-for Command:**
- 1 second
- 2 seconds
- 4 seconds
- 6 seconds
- 8 seconds
- ? seconds

**Basic Instructions for Pilot:**

1. Click on the left icon, and choose an output device for Port A. Output (motor or light) must be connected to port A.
2. Click on the right icon and choose a Wait-for command. If you choose “?” you must input a number (in seconds) in the box under the icon.
3. Download the program to the RCX – see “Downloading the Program to RCX” section below.
Sample Programming Challenges for Level 1:

- Program a one-motored car move forward for four seconds and stop.
- Program a one-motored merry-go-round turn for 20 seconds and stop.
- Program a one-motored car to travel 2m and stop.

**Downloading the Program to RCX**

The RCX can hold five programs. Programs 1 and 2 are locked with default programs:

- Program 1: Turn on motors A and C
- Program 2: Two-button remote control program using motors A and C, and touch sensors in ports 1 and 3.

Your programs are saved in Programs 3, 4, or 5. (Note: Programs 1 and 2 can be unlocked through the Administrator panel if necessary.)

**Downloading Instructions:**

1. Turn on the RCX and set it to Program 3, 4, or 5. Be sure that the USB tower is plugged into the computer.
2. Place the RCX in front of the tower and click the white Run arrow on the screen. You should see a green light in the tower.
3. Wait for the progress bar on screen to reach 100% and the RCX to give out a sweeping beep sound. The program is now on the RCX, and can be executed by pressing the green Run button on the RCX.

**Interference:**

Tower to RCX communication happens via infrared, similar to a TV remote. Other infrared in the room (such as another tower/RCX pair) may interfere with the communication. Some people have also experienced interference from overhead lights in the room.

To prevent this, it is recommended that you place the RCX and tower in a “garage” while downloading the program. A garage can be a box or simply a paper tent that blocks other infrared sources from view.

**Note about Firmware:**

Once in a while, when downloading a program, ROBOLAB will prompt you that the Firmware must be downloaded. A prompt similar to one shown on next page will appear.

When this prompt shows up, click on Download. Do not move the RCX away from the tower until: 1) the progress bar reaches 100%, 2) five default programs are downloaded, 3) you get a message saying “Communication Successful”, AND 4) your own program has been downloaded.

It will take about five minutes to download the program – keep the RCX and tower close during this time, and make sure that the computer does not go to sleep during this time.

Firmware is similar to an operating system on a computer and is necessary component of RCX operation. Firmware download is typically necessary only when: 1) RCX batteries have just been replaced, or 2) you are using a version of ROBOLAB that is different than the one that the RCX was previously used with.
Pilot 2: Two Outputs, One Step

Choose Two outputs (Ports A and C):
- Motor Forward
- Motor Reverse
- Light On
- Stop (do nothing)

Specify Power Setting (1~5) below each icon.

Choose One Wait-for command:
- 1 second
- 2 seconds
- 4 seconds
- 6 seconds
- 8 seconds
- ? seconds
- Touch Sensor In
- Touch Sensor Out

If using Touch Sensor, specify Ports 1, 2, 3.

Added Features for Level 2:
- Outputs on Ports A and C. You can disable one – or both – by using the stop sign.
- Option to change power setting on both outputs. Power varies from 1 to 5, with 5 being the highest power.
- “Wait for touch sensor” option – you must specify Port 1, 2, or 3 if using a touch sensor.

Sample Programming Challenges for Level 2:
- Program a two-motored car to spin in place for ten seconds and stop.
- Program a two-motored car move forward until you press a touch sensor, then stop.
- Program a one-motored car move forward, with a light turned on, for 10 seconds.
### Pilot 3: Three Outputs, Two Steps

#### Choose Three Outputs (Ports A, B, C):
- Motor Forward
- Motor Reverse
- Light On
- Stop (do nothing)

Specify Power Setting (1~5) below each icon.

#### Choose Two Wait-for commands:
- 1 second
- 2 seconds
- 4 seconds
- 6 seconds
- 8 seconds
- ? seconds
- Touch Sensor In
- Touch Sensor Out
- Wait for Brighter
- Wait for Darker

If using Touch Sensor or Light Sensor, specify Ports 1,2,3.

#### Continuous Run:
Loops the program indefinitely.

### Added Features for Level 3:
- Outputs on Ports A, B, and C. You can disable one or more by using the stop sign.
- Option to change power setting on both outputs. Power varies from 1 to 5.
- “Wait for darker” or “wait for brighter” option.
- **Continuous Run** feature, which loops the program indefinitely. Press the green Run button on RCX to stop the program.

### Sample Programming Challenges for Level 3:
- Program a car to go forward for four seconds, stop, and reverse for four seconds.
- Build a car with a light sensor pointing down to the floor. Program a car to move forward until it sees a line on the floor, then back up for 5 seconds.
- Build a car with a light sensor pointing down to the floor. Program a car to remain still until a touch sensor is pressed, then go forward until it sees a line on the floor.
- Program a car that moves forward when touch sensor is pressed, stop when it is let go, and repeats indefinitely.
Pilot 4: Three Outputs, Infinite Steps

Choose Three outputs (Ports A, B, C):  
- Motor Forward  
- Motor Reverse  
- Light On  
- Stop (do nothing)  
Specify Power Setting (1~5) below each icon.

Choose Wait-for command:  
- ? seconds  
- Touch Sensor (choose In or Out)  
- Light Sensor (choose Brighter or Darker)  
  If using Touch Sensor or Light Sensor, specify Ports 1,2,3.

Added Features for Level 4:  
- Lets you add as little or as many steps as you need by clicking “insert page” button. The default program has two steps.  
- Each page displays one step; click on right or left arrow to see next or previous step, respectively.

Sample Programming Challenges for Level 4:  
- Program a car that remains still until a touch sensor is pressed then goes forward for 2 seconds, stops, reverses for two seconds, then repeats indefinitely.  
- Program a car to drive in a square pattern (move forward, turn 90 degrees, move forward again, turn, and so on).  
- Build and program a car that uses a light sensor to follow an arbitrary line on the floor.
**ROBOLAB PILOT Programming Challenges**

**Pilot Level 3**

☐ 1. Build a car with one or two motors and a touch sensor (touch sensor may be handheld or attached as a bumper).
   
   Program the car so that it goes forward until the sensor is pressed, then reverses for two seconds, and stops.

☐ 2. Build a car with one or two motors and a touch sensor used as a handheld button.
   
   Program the car so that it starts moving when the touch sensor is pressed, and stops when it is let go.
   
   Bonus: Use the Continuous Run feature to make this program indefinitely.

☐ 3. Build a car with a light and a light sensor. Light sensor should be pointing up or to the side.
   
   Program the car so that it moves forward until the room becomes dark or the car goes into a dark section. When the room becomes dark, the car will stop and the light will turn on. Light will turn off when the room becomes bright again.
   
   (If you can not shut off the room light, you can simulate a dark room by holding a dark paper over the light sensor.)

☐ 4. Build a car with two motors and a light sensor on the front. The light sensor should be pointing down to the floor.
   
   Program the car so that it can follow a line – or an edge – on the floor.
   
   (Hint: You will need to use the Continuous Run mode to do this.)

**Pilot Level 4**

☐ 5. Build a car with two motors. Using tape, draw a large triangle on the floor (the three sides should have different lengths).
   
   Program the car to follow the triangle.

☐ 6. Build a car that has one touch sensor as a handheld switch and a second touch sensor as a bumper.
   
   When you run the program, the car will remain still until you press the handheld switch. It then moves forward until the front bumper is depressed. When the bumper is depressed, the car will move back. When you press the handheld switch again, the car stops.

☐ 7. Build a car with a lift-arm that will lift up a small object. Attach a light sensor to the front of the car, pointing down.
   
   Program the car so that it picks up the object, goes forward until it encounters a line on the floor. Stop, lower the object, then move backward for 2 seconds.
1. Build a car with one or two motors and a touch sensor (touch sensor may be handheld or attached as a bumper).

Program the car so that it goes forward until the sensor is pressed, then reverses for two seconds, and stops.

Sample Solution

2. Build a car with one or two motors and a touch sensor used as a handheld button.

Program the car so that it starts moving when the touch sensor is pressed, and stops when it is let go.

Bonus: Use the Continuous Run feature to make this program indefinitely.

Sample Solution
3. Build a car with a light and a light sensor. Light sensor should be pointing up or to the side.

Program the car so that it moves forward until the room becomes dark or the car goes into a dark section. When the room becomes dark, the car will stop and the light will turn on. Light will turn off when the room becomes bright again.

If you can not shut off the room light, you can simulate a dark room by holding a dark paper over the light sensor.

Sample Solution

4. Build a car with two motors and a light sensor on the front. The light sensor should be pointing down to the floor.

Program the car so that it can follow a line – or an edge – on the floor. (Hint: You will need to use the Continuous Run mode to do this.)

Sample Solution

(Note: This program is for dark line on light floor, with motor A being the left wheel and motor C being the right wheel. The car will travel in a zigzag manner along the left edge of a line.)
5. Build a car with two motors. Using tape, draw a large triangle on the floor (the three sides should have different lengths).

Program the car to follow the triangle.

Sample Solution

(Note: Times will differ depending on the shape of your triangle.)
6. Build a car that has one touch sensor as a handheld switch and a second touch sensor as a bumper.

When you run the program, the car will remain still until you press the handheld switch. It then moves forward until the front bumper is depressed. When the bumper is depressed, the car will move back. When you press the handheld switch again, the car stops.

Sample Solution

(Note: In this example, sensor 1 is attached to the handheld switch and sensor 2 is attached to the bumper.)
7. Build a car with a lift-arm that will lift up a small object. Attach a light sensor to the front of the car, pointing down.

Program the car so that it picks up the object, goes forward until it encounters a line on the floor. Stop, lower the object, then move backward for 2 seconds.

Sample Solution

(Note: In this example, motor A and C are attached to the wheels, and motor B is attached to the lift-arm. Time to move motor B will change depending on the design of your lift-arm.)
**RoboLAB Inventor 3**

**Starting Up**

1. Connect the USB tower to the computer.
2. Open RoboLAB, and click on Programmer.
3. In the next screen, double-click on Inventor 3.

**Windows & Palettes**

Inventor environment has two windows and two palettes. *(Not all windows will open on default – if you are missing any windows, see “Missing palettes/windows?” section on the next page.)*

- **DIAGRAM WINDOW** is where all the programming happens.
• **FRONT PANEL** is an inherent artifact from LabVIEW, the programming language that RoboLab was written in. It needs to stay open; however, you will not be using this window. If you accidentally maximize this window, press **Ctrl-E** to bring back the Diagram.

• **TOOLS PALETTE** lets you choose appropriate tools for your operation. In the Tools Palette, the four most-commonly used tools are:
  - **Select tool**: Use this tool when selecting, moving, or deleting an icon or wire.
  - **Text tool and Operate Value tool**: Use either of these tools to change numerical or text values.
  - **Wiring tool**: Use the wire tool to connect icons together.
**Keyboard Short-cuts:**
  - **Space-bar** toggles between Select tool and Wiring tool.
  - **Tab-key** scrolls through all tools.

• **FUNCTIONS PALETTE** contains all icons (commands) for programming in Inventor. Inventor 3 icons are shown below.

![Functions Palette Diagram]

  - **Main Menu**: Contains motor/light output icons as well as stop signs for stopping output power.
  - **Wait-For**: Contains Wait-for commands using time, touch sensor, or light sensor.
  - **Structures**: Contains forks, jumps/lands, while loops, and task split.
  - **Modifiers**: Modifiers are used to assign attributes to other icons, such as port or power setting.
  - **Music**: Music icons are used to make RCX play sound or music.

Please refer to *Using RoboLab* and *RoboLab Reference Guide* for more information about each of these icons.
**Missing palettes/windows?**

The Function Palette and/or the Tools Palette may not open automatically. If either of them are missing, go to Window and select “Show Functions Palette” or “Show Tools Palette” to open them. The Functions Palette can also be opened by right-clicking anywhere inside the Diagram window.

If the Diagram window is missing, go to Window and select “Show Block Diagram” or simply type Ctrl-E to bring it back.

---

**Writing a Program**

**Basic Steps:**

1. Select **icons** (commands) from **Functions Palette** and drag them onto the **Diagram**.
2. **Wire** the icons together in order of execution.
3. Add **modifiers** to icons where necessary.

**Rules:**

- Each program must begin with a Begin icon and end with an End icon.
- Each icon, except for Modifiers, has a Begin (upper left quadrant) and an End (upper right quadrant) connector. When connecting icons in order, wire must go from End of one icon to the Begin of the next icon.

**Wiring the icons**

All icons must be strung together using "wires" in order to run the program. Wires have several purposes:

- Tells the program the order in which to execute the commands
- Assigns "modifiers," or attributes, to icons

**To manually wire the program:**

1. Select the wiring tool by either pressing the Space-bar or clicking on the wiring tool on the Tools Palette.
2. Point to one icon using the wiring tool until the correct connector is blinking. Click ONCE to begin the wire.
3. Move the mouse to the second icon and point to the appropriate connector so that it blinks. Click ONCE to end the wire.

**NOTE:** The connector on an icon must be blinking in order to begin/end a wire.

**Auto-wiring feature** will automatically wire a new icon onto the nearest neighbor. Press the Space-bar while dragging the icon to enable/disable Auto-wiring.

**Broken Wires**

Invalid wires will appear as “broken wires.” All broken wires must be fixed or removed before the program can be executed. To remove broken wires, click on the wire and delete, or type Ctrl-B to remove all broken wires.

In the example on the right, the top wire is a good wire and the bottom wire is a broken wire. (This broken wire came about because the wire was connected to the End side of motor A icon)
Context Help Window (Ctrl-H)

If you are unsure about how to use an icon, use the **Context Help Window**. Context Help can be opened by typing **Ctrl-H** or going to Help → Show Context Help. Point to any icon to get a description of the icon as well as an example of how it is used.

### Debugging

A “**broken arrow**” at the top left corner of Front Panel and Diagram indicates that the program is not complete or there is an error in the program.

Look over your program for any errors. If you can not find any, click on the broken arrow to open the **Error List**.

The Error List describes the errors in the program, and double-clicking on any of the errors will highlight exactly where the error occurs on the program.

Once the program is free of any syntax errors, the broken arrow will turn into a solid white arrow. The program can now be downloaded onto the RCX.

This process will only remove syntax errors in the program, which prepares you to download and run the program on the RCX. Other errors (such as robot not behaving in the way you had intended) must be debugged by examining the robot behavior and program.

### Downloading the Program to RCX

When the program is ready, the **Run arrow** on top left corner of the window will turn to a solid white arrow. (Note: This only means that the program’s syntax is correct; it does not guarantee that the program will do what you meant it to do.)

**Downloading Instructions:**

1. Turn on the RCX and set it to Program 3, 4, or 5. Be sure that the USB tower is plugged into the computer.
2. Place the RCX in front of the tower and click the white Run arrow on the screen. You should see a green light in the tower.
3. Wait for the progress bar on screen to reach 100% and the RCX to give out a sweeping beep sound. The program is now on the RCX, and can be executed by pressing the green Run button on the RCX.

**Interference:**

Tower to RCX communication happens via infrared, similar to a TV remote. Other infrared in the room (such as another tower/RCX pair) may interfere with the communication. Some people have also experienced interference from overhead lights in the room.

To prevent this, it is recommended that you place the RCX and tower in a “garage” while downloading the program. A garage can be a box or simply a paper tent that blocks other infrared sources from view.
**Note about Firmware:**

Once in a while, when downloading a program, ROBOLAB will prompt you that the **Firmware** must be downloaded. A prompt similar to one shown on next page will appear.

When this prompt shows up, click on Download. Do not move the RCX away from the tower until: 1) the progress bar reaches 100%, 2) five default programs are downloaded, 3) you get a message saying "Communication Successful", AND 4) your own program has been downloaded.

It will take about five minutes to download the program – keep the RCX and tower close during this time, and make sure that the computer does not go to sleep during this time.

Firmware is similar to an operating system on a computer and is necessary component of RCX operation. Firmware download is typically necessary only when: 1) RCX batteries have just been replaced, or 2) you are using a version of ROBOLAB that is different than the one that the RCX was previously used with.

---

**Programming Tips & General Comments**

**Programming Tips:**

- Program in segments and check periodically.
  
  For example, if writing a program to make a robot drive forward a certain distance, grab and lift something, and return to starting point, break it up in four steps: 1) drive forward a certain distance, 2) grab, 3) lift, 4) drive back.
  
  Start with the first step, and check the code as soon as it is done. Repeat with the other steps.

- The "**Undo**" command is VERY useful. Go to Edit → Undo, or type **Ctrl-Z** to undo the last action.

- When modifying the program to fine-tune robot's motion, change one thing at a time and test each time. For example, do not change motor power and time at once.

**General Comments:**

- There are generally more than one way to program a robot to do the same thing – don’t assume something is wrong just because it’s not the way you are used to doing it.

- It is generally a good practice to write RoboLAB programs in a well-organized, linear manner. Avoid crossing wires whenever possible.
What’s Next: Inventor 4

Inventor 4 works in the same way as Inventor 3, only with more icons and commands as the Function Palette on the right shows. (Note: This Functions Palette is taken from ROBOLAB 2.9. Function palettes from other versions may vary slightly.) Below are some of the notable features; please refer to Using Robolab and ROBOLAB Reference Guide for more information about Inventor 4 icons.

Added Features for Inventor 4:

- **Additional sensors**: Inventor 4 includes icons for Rotation Sensor and Temperature Sensor.
- **Float Output**: The yellow stop sign removes power to the output, but does not brake the motors. Using this, a car can coast to a stop rather than stopping abruptly.
- **Wait for Bright/Dark**: Rather than waiting for a percent change in light as you did in Wait for Brighter/Darker icon in Inventor 3, you can now wait for a specific light value.
- **Events & Tasks**: New structures allow you to program multiple simultaneous tasks and gives you greater control over the priorities by which commands are executed.
- **Containers**: Containers are like variables – they let you store and manipulate numbers.
- **RCX Communications**: You can have one RCX talk to another RCX, enabling you to synchronize more than one RCX or have them work together.
RoboLAB Inventor 3: Icon Descriptions

Main Menu

Begin: Any Robolab program must start with this icon.

End: Any Robolab program must end with this icon.

Outputs

Motor $A/B/C$ Forward: Use these icons to turn designated motor on in the forward direction.

Motor $A/B/C$ Reverse: Use these icons to turn designated motor on in the reverse direction.

Generic Motor Forward/Reverse: Use these icons to turn any motor on in forward/reverse. Use modifiers to specify which motor to turn on.

Flip Direction: This icon flips the direction of a motor.

Lamp $A/B/C$: Use these icons to turn on a light bulb. If using the generic icon, use modifiers to specify the port.

Stop $A/B/C$: Stops sending power to the specified port.

Stop All Outputs: Stops sending power to all three ports.

Generic Stop Output: Stops sending power to any port(s) specified by modifiers.

Wait For Sub-Menu

Whenever a program encounters a wait-for icon, it does not go on to the next command until the action that is specified by the wait-for icon is accomplished.

Wait for 1/2/4/6/8/10 Seconds: Waits for specified number of seconds before proceeding to the next command.

Wait for Time: Use a Numeric Constant modifier to tell this icon how long to wait for.

Wait for Random Time: Waits for a random amount of time (between 0 and 5 seconds) before proceeding to the next command.

Wait for Push/LetGo: Waits until a touch sensor is pressed (or let go) before proceeding to the next command.

Wait for Brighter/Darker: Waits until a light sensor detects a change in light level before proceeding to the next command.
**Sample Program 1:**

This program turns motor A and C on forward, waits 4 seconds, turns motors A and C on reverse, waits 4 seconds, and stops all power. If you program a two-motored car using this code, it will move forward for 4 seconds, reverse directions and travel for 4 seconds, and stop.

**Sample Program 2:**

This program will first wait for a touch sensor to be pressed. Once a touch sensor is pressed, then it turns motors A and C on forward. When the touch sensor is pressed the second time, then all motors will stop.

**Structures Sub-Menu**

**Forks**

A fork in RoboLab is equivalent to if/then scenario. If A, then do one thing; if B, then do something else. When the program encounters a fork, it tests a condition (for example, is the touch sensor pressed in?) and chooses a path that matches the condition.

**Touch Sensor Fork/Light Sensor Fork:** Chooses one of two actions, depending on the state of the touch or light sensor.

**Fork Merge:** All forks must be merged back together before the end of the program.

In general, a fork icon has one pink wire going in and two coming out. Of the two that comes out, only one is actually executed (at a time). The two pink wires must be merged back together using a Fork Merge icon at some later point in the program.

Inventor 3 has two forks: Touch Sensor Fork and Light Sensor Fork. A touch sensor fork does one thing when the button is pressed in and another when button is not pressed in. Light sensor fork does one thing when light is brighter than a certain value, and does another thing when light is darker than the value.

**Sample Program:**

This program turns motors A and C on forward. It then checks the light sensor value. If it is dark (light sensor value less than or equal to 55) then it turns on a light at port B. If it is
bright (light sensor value greater than 55) then it does nothing to port B. After 4 seconds, it shuts off power to all three ports.

**Jump/Land**

**Jump/Land:** When ROBOLAB program encounters a Jump icon, the sequence of executing the icons literally “jumps” to the corresponding Land icon. If you place the Land icon at any point BEFORE the corresponding Jump icon, then the program will repeat indefinitely.

Jumps and Lands must always be used as a pair. There are five sets of Jumps/Lands in Inventor – red, blue, yellow, black, and green. A Jump of a certain color will go to the Land with the same color.

When a Land icon is placed in the program anywhere before the Jump icon, an endless loop is created. When the program encounters the Jump icon, it goes back to the Land icon position. Whatever commands that are placed between the Jump and Land are repeated over and over again.

**Sample Program:**

This program moves motor A and C forward if the touch sensor is pressed in, and shuts off power if touch sensor is not pressed. If used on a two-motored car, the car will move forward while you hold down the button and stop moving when you let go. This goes on indefinitely until you stop the program by pressing the Run button on the RCX or turning the RCX off.

**Start/End of Loop**

**Start/End of Loop:** This pair of icons is used to repeat a certain set of instructions a set number of times.

Start of Loop and End of Loop icons are always used as a pair. Icons placed between the two are repeated for a prescribed number of times (the default is “twice”).

The difference between Jump/Land and Start/End of Loop is that Jump/Land loop goes on forever, while Start/End of Loop allows the user to specify how many times to perform the loop.

**Sample Program:**

In this program, motors A turns on forward. After 4 seconds, the motor reverses direction. After 4 more seconds, the motor reverses direction again, then stops. (Note: You will probably not notice the second “Flip Direction” because the motor stops immediately following the flip direction command.)
**Task Split**

Task split is similar to multitasking in everyday life. For example, rather than moving forward, THEN playing a song, you can have a program move forward while playing a song at the same time.

Task Split, like a fork, has one pink wire going in and two coming out. However, the two icons behave in a different way. When using a fork, only ONE of the two possible paths is executed. With task split, both paths are executed simultaneously.

Split tasks are NOT merged back together. Each path ends with its own End (red light) icon.

**Sample Program:**

In this program, two things happen at once. The top task turns on motor A forward, then flips directions every second. The red Jump/Land keeps this going indefinitely.

The bottom task turns on a light in port B for a random duration then shuts it off for random duration (the effect is that you get the light to flash on and off randomly). The yellow Jump/Land keeps this going indefinitely.

Note that two pairs of Jump/Land were used. Also note that after a task split, each task gets its own End icon to finish the program.

---

**Modifiers Sub-Menu**

Modifiers are used to specify optional parameters for other commands.

- **Output A/B/C**: Use these icons to specify output port. Commonly used with the generic Motor Forward/Reverse icons. Two or more can be strung together to turn on two/three motors at the same time.

- **Input 1/2/3**: These icons are used to specify which ports the sensors are connected to. Use with all icons related to sensors (forks and wait-for’s).

- **Numeric Constant**: Use this icon to enter a number – for example, in the Wait for Time command, the amount of time to wait is specified using this modifier. The number can be changed by clicking inside the box while in the Text tool. (To choose Text tool, hit Tab key several times until you see a text cursor.)

- **Power Level 1/2/3/4/5**: Use with Motor Forward/Reverse or Lamp commands to specify the power level. Level 1 is weakest and 5 is maximum power.

- **Random Number**: Outputs a random number between zero and a maximum number that is specified using the Numeric Constant modifier.
Sample Program 1:

In this program, the generic Motor Forward icon is used with Output A and C modifiers to turn motor A and C on simultaneously. The motors stay on for 2.5 seconds (as specified by the Numeric Constant modifier in the Wait for Time icon).
Not that Output A/B/C modifiers can be strung together to turn on more than one motor at the same time.

Sample Program 2:

In this program, a touch sensor is connected to input port 2. The robot will move forward when the touch sensor is pressed, and stops when the sensor is pressed again.
Notice that Input 1/2/3 modifier must be anytime a sensor icon shows up in the program, even if it has been specified earlier in the program (unless you are using the Port 1, which is default).
Music Sub-Menu

Music icons are used to make RCX play a note or music. For example, you can make the RCX make a sound after finishing a task, or play a song while moving forward.

Music Notes: There are icons for C, C#, D, D#, E, F, F#, G, G#, A, A#, B. Use the appropriate icon for the note that you want to play. The default duration is a quarter note in the standard octave.

Rest: Use this icon to insert a rest. Standard duration is quarter note.

Note Duration Modifiers: These icons are wired into the Music Notes or Rest icon to specify the duration of the note. There are modifiers for sixteenth, eights, quarter, half, and whole notes.

Up/Down an Octave: This modifier is wired into Music Notes icon to change the octave of the note.

Sample Program:

The program above will play the song “Twinkle Twinkle Little Star.”
RoboLAB Inventor 3 Programming Challenges

Use a simple two-motored car for the following challenges. Challenges should be completed in order, at your own pace. Use Inventor Level 3 or higher.

1. Program the car to go forward for 4 seconds, then stop.

2. Program the car to go forward for 4 seconds, stop, go backward for 4 seconds, then stop.

3. Program the car so that it turns in place clockwise for 3 seconds, spins in place counter-clockwise for 5 seconds, then stop. (Note: You may need to modify the car to make it turn properly.)

4. Add a touch sensor to the car using a long cable. Program the car to start moving when the touch sensor is pressed, and stop after 3 seconds.

5. Program the car to start moving when the touch sensor is pressed, and stop when the touch sensor is pressed again.

6. Modify Program 4 using Jump/Land so that the program repeats itself indefinitely.

7. Remove the touch sensor, and add a light sensor to the front of the car. Light sensor should be pointing down at the floor. Program the car so that the car moves forward until it sees a dark line, then stops.

8. Using a fork structure, program the car so that it turns left when the light sensor sees a bright color, and turns right when light sensor sees a dark color. This process should go on indefinitely. What can this program be used for?

9. Remove the light sensor, and add two touch sensors to the car using two long cables. Using task split and forks, program the car so that one touch sensor controls one motor, and the other touch sensor controls the other motor (similar to default Program 2).

10. Re-write Program 9 using embedded forks instead of task split.
The following solutions are samples only. There are more than one way to solve many of these challenges, so your program may look different than the one shown here. These solutions are for a two-motored car with Motor A on the left and Motor C on the right.

1. Program the car to go forward for 4 seconds, then stop.

2. Program the car to go forward for 4 seconds, stop, go backward for 4 seconds, then stop.

3. Program the car so that it turns in place clockwise for 3 seconds, spins in place counter-clockwise for 5 seconds, then stop. (Note: You may need to modify the car to make it turn properly.)

4. Add a touch sensor to the car using a long cable.
   Program the car to start moving when the touch sensor is pressed, and stop after 3 seconds.

   Note: Touch sensor is in Port 1. If using a different port, it must be specified using an input modifier.

5. Program the car to start moving when the touch sensor is pressed, and stop when the touch sensor is pressed again.

6. Modify Program 4 using Jump/Land so that the program repeats itself indefinitely.

7. Remove the touch sensor, and add a light sensor to the front of the car. Light sensor should be pointing down at the floor.
   Program the car so that the car moves forward until it sees a dark line, then stops.

   Note: Light sensor is in Port 1. If using a different port, it must be specified using an input modifier.
8. Using a *fork* structure, program the car so that it turns left when the light sensor sees a bright color, and turns right when light sensor sees a dark color. This process should go on indefinitely.

What can this program be used for?

![Diagram of fork structure](image)

This program can be used as a line-follower (or an edge follower) to follow the right-side edge of a dark line. Notice that a fork merge is needed at the end of a fork.

9. Remove the light sensor, and add two *touch sensors* to the car using two long cables.

Using *task split* and *forks*, program the car so that one touch sensor controls one motor, and the other touch sensor controls the other motor (similar to default Program 2).

![Diagram of task split and forks](image)

**Note:** Touch sensor 1 turns Motor A on and off; Touch sensor 3 turns Motor C on and off. Notice that each split task ends with its own red light.

10. Re-write Program 9 using embedded forks instead of task split.

![Diagram of embedded forks](image)

**Note:** Embedded forks create four scenarios – (from top) 1) no sensors pressed in → no motors on, 2) Sensor 3 pressed in → Motor C on, 3) Sensor 1 pressed in → Motor A on, and 4) both sensors pressed in → both motors on.
**ROBOLAB Troubleshooting Guide**

**Useful Tips**
- Type Ctrl-H to open the Help window. The Help window will show you a detailed description of any icon that you point to.
- Error List – Click on the broken arrow at any time to see an Error List. Clicking on an error on the list will highlight the problem area in your program.
- Use the Space-bar to toggle between two most-commonly used tools in the diagram: the arrow (used for selecting/moving icons and wires) and the wire spool (used for connecting icons). Use Tab to switch to other tools such as the text tool.

---

**General Problems**

**Q: I can’t download the program into the RCX. What’s going on?**

**A1:** RCX may be turned off. Press the On/Off button to turn it on.

**A2:** ROBOLAB may not be recognizing the USB tower. Save the program, shut down ROBOLAB, connect the tower to a USB port, then restart ROBOLAB. Typically (on a PC), the tower must be connected to the computer before starting ROBOLAB.

**A3:** On some PC’s, the USB tower can only be used on the USB port that was used when installing the tower driver. Try plugging the tower into a different port.

**A4:** If using an older serial tower, check the battery. The serial tower takes one 9V battery.

**Q: The firmware download fails before reaching 100%. What’s happening?**

**A1:** Make sure that the computer does not “go to sleep” during the firmware download by moving the mouse occasionally.

**A2:** There may be too much stray infrared signals in the room (i.e., from other towers or from lights). Place the tower and RCX in a box or cover them with thick paper to shield them from stray infrared.

**A3:** The RCX battery may be low (even if the low battery indicator has not turned on). Replace with fresh batteries.

**Q: The RCX keeps on losing power and/or firmware. Why?**

**A1:** Batteries may be getting loose inside the RCX, particularly with cheaper batteries which are sometimes a bit smaller. Replace with higher quality batteries.

**A2:** There may be stray infrared signal in the room. Be sure to place a paper “tent” over RCX and towers when downloading programs to prevent affecting other RCX’s.

**Q: I just put fresh batteries in the RCX, but it’s not turning on. Why?**

**A1:** Electrical contact on the battery may be a bit loose. Open the RCX again and turn each of the batteries a few times, or try shaking the RCX.

**A2:** Some cheaper batteries seem to be a bit smaller. Use higher quality batteries to make sure they make good contact with the battery holder.
**General Problems**

**Q:** My program window disappeared. How do I get it back?

**A:** Type Ctrl-E to show the diagram. Alternatively, go to Window and Show Diagram.

---

**Q:** The Functions Palette disappeared. How do I get it back?

**A:** On the program, go to Windows → Show Functions Palette. Alternatively, you may also Right-click anywhere inside the program (click on the thumb tack icon at the upper left of the palette to make it stay up).

---

**Q:** The Tools Palette disappeared. How do I get it back?

**A1:** On the program, go to Windows → Show Tools Palette.

**A2:** You can also use the Space Bar to toggle between the selection tool and wiring tool, or use Tab key to scroll through all tools.

---

**Q:** I am running the default Program 1 (or 2) but nothing is happening. Why?

**A1:** The RCX may not have firmware. Download firmware using ROBOLAB and try again.

**A2:** Be sure that the motors are connected to port A and C. If using program 2, touch sensors should be connected to ports 1 and 3.

---

**Q:** I want to store than three programs on the RCX. Can I do that?

**A:** Yes. Programs 1 and 2 can be unlocked using the RCX Settings tab in Administrator. Click on Administrator at ROBOLAB start-up screen and go to RCX Setting tab to find this option.

---

**Q:** I saved a program on the computer, but I can’t find it. Where did it go?

**A:** Your program is most likely saved in ROBOLAB’s default folder. Go to My Documents → ROBOLAB Data → Program Vault → Pilot or Inventor → My Programs. The program should be in the Pilot or Inventor folder that corresponds to the level that you were working with.

---

**Q:** Program download keeps on repeating over and over again. How do I make it stop?

**A:** Most likely, the "Run Continuously" button (next to the Run button) was clicked by accident. If the button that is directly to the right of the Run button shows two black arrows in a circle, click this button to disable it. The two arrows will turn white if the feature is disabled. You may need to click it a few times, since the mouse click will only be detected after one download completes, before the next one begins.
Building & Programming Issues

Q: Program says Turn Motor On, but nothing moves. What’s going on?
A1: The RCX battery might be low. Check the RCX’s display to see if there is an icon that looks like a battery with an X on it. If there is, then change battery. If there isn’t, then the battery is fine.
A2: The motor is turning on, but for a very brief time. Inspect the program to make sure that there is an appropriate wait-for function after the motor-on command.
A3: Make sure the electrical cable between the motor and RCX are attached securely. Also, try a different cable – it might be a defective cable.
A4: Check the program and the wiring of the motor to make sure that the motor is connected to the correct port as specified in the program.
A5: The motor may not have enough power to move. Change the power level to maximum. If it still doesn’t move, then you may need to use gears to give more power (this is especially the case when you have a heavy car going uphill or you are trying to move a very long or large arm).

Q: Touch sensor doesn’t seem to be working. How can I fix it?
A1: The sensor port may not be correctly specified on the program. Use appropriate modifier to assign sensor port to ALL sensor icons in the program.
A2: The cable may be connected to the wrong side of the sensor. Be sure that the black cable attaches to the TOP side of the touch sensor.
A3: Cable may be broken – try replacing the cable with another one.

Q: Light sensor is not working properly. How can I fix it?
A1: The sensor port may not be correctly specified on the program. Use appropriate modifier to assign sensor port to ALL sensor icons in the program.
A2: The light level in the room may not be suitable for the default setting of the light sensor icon. Check the light sensor reading on the RCX display and modify the Light Sensor icon to change the light level that the sensor reacts to. You may need to add a modifier to the light sensor icons to change its default settings.
A3: Make sure that the light sensor is pointing toward the object that you are looking at – for example, if you are looking at a tape on the floor, then the light sensor needs to be pointing down.

Q: Rotation sensor is not working properly. How can I fix it?
A1: The sensor port may not be correctly specified on the program. Use appropriate modifier to assign sensor port to ALL sensor icons in the program.
A2: Rotation sensor needs to be zero-ed before use to make sure that the program begins counting from zero. Make sure that “Zero Rotation Sensor” icon is inserted before using rotation sensors.
Building & Programming Issues

Q: Program looks fine, but there is a broken arrow. What’s wrong?
A1: There may be a hidden broken wire somewhere (such as underneath an icon). Press Ctrl-B to remove all broken wires.
A2: There may be a mistake in wiring. Click on the broken arrow to open the Error List. Double click on an error. If an icon is highlighted, delete that icon. Replace the icon and re-wire. Repeat until all errors on the error list disappears.

Q: Car is going the wrong way or is spinning in place. How do I make it go forward?
A1: One or both of the cables that connect the motor(s) to the RCX is flipped. Pick up the car and run it to find out which of the wheels is moving the wrong direction. Remove the wire from the appropriate motor port, turn it 180°, and reattach. Alternatively, you can change the motor direction in your program.

Q: Motors are running, but the car is not moving. What’s going on?
A1: Pick up the car and check to see if the wheels are actually turning. If using gears, make sure that they are properly meshed so that the power from the motors is transferred to the wheels.
A2: Lift the car up and check to see which directions the wheels are turning. If both front and rear wheels are powered, and they are connected by gears, it is possible that the front and rear wheels are turning in opposite directions. If this is the case, change the gears so that the wheels turn in the same direction.
A3: The wheels may not be touching the floor. Check carefully to make sure that the powered wheels are touching the floor.

Q: Car is programmed to turn, but it is not turning. How do I make it turn?
A1: The non-driving tires may have too much traction, making it difficult for the car to drag that tire. Use a different type of tire for the non-driving wheel – try skid plate, castor, or remove the rubber from a wheel. Try running the car on a different surface.
A2: If the car had been going forward before turning, the forward momentum may be too great to overcome. Add a pause (wait-for time) in the program so that the car comes to a stop before entering a turn.
A3: Even if the car is having difficulties making tight turns, it may be able to make gradual turns. Rather than programming two motors to turn in the opposite direction (which would make the car turn in place), try turning one motor off while moving the other one, or setting one motor to be at a slower speed than the other one.

Q: The motors fall off when I run my car. Why?
A1: The axles on two-motored cars need to be able to turn independently. If the axles on left and right are connected, then the motor and wheel may want to go in different direction/speed, causing motors to fall off.
A2: The motors may need to be attached more securely. Be sure to have reinforcements on top and bottom of the motors to keep them from twisting off.
### Building & Programming Issues

#### Q: The motors do not stop running. How do I make them stop?
**A1:** Make sure there is a stop sign icon at the end of the program. The red light at the end of the program does not stop the motors.
**A2:** Simply press the Run button on the RCX to stop the car.

#### Q: My car does not go straight. How can I fix this?
**A1:** One or more of the tire may be rubbing against something. Check all tires to make sure they are turning freely without rubbing on something. Make sure that the electrical cables are out of the way.
**A2:** Axle might be attached too tightly. Check the spacers and wheels on the axles to make sure all axles turn freely (including the powered and non-powered wheels).
**A3:** Two motors are running at different power settings. In the program, set all motors to the same power.
**A4:** Even with the same power setting, each motor will run at slightly different speeds, causing the car to turn slightly. If you need the car that travels very straight, try different motors until you find a pair that has similar speeds. If your car does not need to turn, then a one-motored car may be a better option.

#### Q: The hub for tank treads are spinning freely. How do I secure the hub so that it turns with the axle?
**A:** A 16-tooth gear fits inside the hub and holds the hub in place.

#### Q: When using gears, the car is moving intermittently or is making clicking or grinding noises. What’s going on?
**A:** Gears are not correctly aligned, and are "slipping". Realign gears so that they are all properly meshed. There should be no space between gear teeth.

#### Q: I geared up my car to make it go really fast, but it does not move at all. Why?
**A:** Gearing up to increase speed also decreases the power. If the power is decreased too much, then the car would not start. Rearrange the gear train so that the gear ratio is not so high.

#### Q: I can’t get two spur gears to mesh properly. What can I do?
**A1:** Use different size gears or add one or more idler gears until you get a good spacing.
**A2:** Rather than meshing two gears side by side, use a chain to connect the two gears. Chains can work with any spacing.
**A3:** If using the 9794 Team Challenge Set, there are two 1x2 brick with two holes (typically dark gray). This brick can help adjust the spacing between gears.
Building & Programming Issues

Q: I used Jump/Land to make a loop, but the program isn’t looping. How do I fix it?
A: Make sure that the Jump/Land icons are in the correct place. Typically, Land (down-arrow) comes BEFORE Jump (up-arrow) in a program. If Jump is placed before Land, then the program skips over everything that comes between the two icons.

Q: The gears keep on slipping out of place. How can I fix this?
A: Use bushings as spacers to lock gears and axles in position. Bushings and gears have the same thickness.

Q: Pins are stuck in a hole and I can’t get them out. How can I get them out without damaging them?
A: Use an axle and push the pin out.

Q: The short cable is too short, but the long cable is too long. What should I do?
A: Two or more cables can be connected end-to-end to make a longer cable.

Q: My car flips over when I put it on a ramp. How can I make it stay down on the ramp?
A: You probably have too much of the car’s weight at the back. Either change the direction of the car (so that the “front” is now the “back”) or modify the design so that the heavy components – RCX and motors – are toward the front of the car. You can also try to balance the car by putting a weighted brick at the front end.
**Keyboard Short-Cuts**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl-B</td>
<td>Removes all broken wires in the program.</td>
</tr>
<tr>
<td>Ctrl-E</td>
<td>Brings up the program diagram.</td>
</tr>
<tr>
<td>Ctrl-H</td>
<td>Opens/closes the Context Help window.</td>
</tr>
<tr>
<td>Ctrl-L</td>
<td>Opens the Error List (same as clicking on a broken arrow).</td>
</tr>
<tr>
<td>Ctrl-R</td>
<td>Downloads the program to the RCX.</td>
</tr>
<tr>
<td>Ctrl-Z</td>
<td>Undo.</td>
</tr>
<tr>
<td>Space Bar</td>
<td>Toggles between an arrow and a wire spool.</td>
</tr>
<tr>
<td>Tab</td>
<td>Scrolls through all tools – hand, arrow, text, wire spool.</td>
</tr>
</tbody>
</table>

**General Information**

- Use Context Help (Ctrl-H) and point to any icon to get information about it. In Context Help window, click on “Click here for more help” to see an example of how the icon is used.

- To download program to RCX, click on the white arrow on top-left corner of the window or press Ctrl-R. Make sure that the RCX is ON and is set to Programs 3, 4, or 5, and the tower and RCX are in direct view.

- Use the arrow cursor to select/move icons and wires. Use the wire spool cursor to wire icons together. Use the hand or text tool to modify a number in a box. The text tool can also be used to make notes or comments on the program.

**Frequently Asked Questions & Answers**

<table>
<thead>
<tr>
<th>Q:</th>
<th>A:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q:</td>
<td>My Functions Palette or Tools Palette disappeared! What do I do?</td>
</tr>
<tr>
<td>A:</td>
<td>Go to Window \ Show Functions Palette or Show Tools Palette. You can also make the Functions Palette appear by right-clicking anywhere on the page.</td>
</tr>
<tr>
<td>Q:</td>
<td>I see a broken arrow. What is wrong with my program?</td>
</tr>
<tr>
<td>A:</td>
<td>Click on the broken arrow (or type Ctrl-L) to open the Error List. Click on the error(s) to identify the location of the error.</td>
</tr>
<tr>
<td>Q:</td>
<td>Why isn't my sensor working?</td>
</tr>
<tr>
<td>A:</td>
<td>Check that the sensors are connected to one of the input ports (1, 2, or 3) and that the correct port number is wired into each of the program’s sensor icons. If using a touch sensor, make sure that the cable is connected to the top side of the sensor.</td>
</tr>
<tr>
<td>Q:</td>
<td>My program window disappeared! What do I do?</td>
</tr>
<tr>
<td>A:</td>
<td>Type Ctrl-E or go to Window \ Show Diagram to make the program reappear.</td>
</tr>
<tr>
<td>Q:</td>
<td>There really is nothing wrong with my program but I still get a broken arrow.</td>
</tr>
<tr>
<td>A:</td>
<td>First, check the Error List by clicking on the broken arrow and click on the error that appears. If it identifies an icon, then delete the icon and re-wire it even if it looks like there’s nothing wrong with it. If the Error List does not recognize any icons, try typing Ctrl-B to remove all broken wires — sometimes, broken wires get hidden behind icons in places you can’t see. Also try moving the Tools/Functions palettes to see if there is an unused icons hidden behind the windows. If all else fails, open a new window and rewrite the program.</td>
</tr>
</tbody>
</table>
• **Driving gear** refers to the gear that is connected to the power (motor, crank, etc).

• **Driven gear** refers to the gear that is connected to the output (wheels, arm, etc).

• Gears that are neither driving or driven are called **idler gears**, and are used to change the direction of rotation or to adjust the distance between driving and driven gears.

• Two gears that are “meshed,” or in contact, will turn in opposite directions.

• In general, when two gears are meshed, the larger gear turns at a slower speed than the smaller gear.

• LEGO kits typically come with four sizes of **spur gears**.

Gears are used to accomplish several things...

• Increase/decrease the speed.

• Decrease/increase the power.

• Change the direction of rotation.

• Move the axis of rotation to a different spot.

• Change the axis of rotation by 90° – for example, change horizontal rotation to vertical rotation.

• Change rotary motion to linear motion.

Increasing/decreasing the power and decreasing/increasing the speed come hand-in-hand: increased speed comes with decreased power, and increased power comes with decreased speed.

Change of axis can be accomplished by using bevel gear, crown gear, or a worm gear.
Simple Gear Trains

Gear ratio compares the driven gear’s speed to the driving gear’s speed.

For example, if the driving gear (A) has 8 teeth and the driven gear (B) has 24 teeth, the Gear Ratio is 8:24 or 1:3. The driven gear (B) rotates at a third of the speed of the driving gear (A) – for every rotation that A makes, B turns through one-third of a revolution.

As gear B loses speed, it gains power. Gear B will have three times more power than gear A, which is useful when doing tasks such as going up an incline or lifting up something heavy.

Consider the gear train shown above with five gears. To calculate the gear ratio for this gear train, we multiply the gear ratios for each place where two gears are meshed:

\[
\frac{A}{B} \times \frac{B}{C} \times \frac{C}{D} \times \frac{D}{E} = \frac{16}{40} \times \frac{40}{24} \times \frac{24}{8} \times \frac{8}{40} = \frac{16}{40}
\]

This example illustrates that when a gear train consists of multiple gears in a line with just one gear on each axle, you only need to consider the first and the last gear of the gear train to calculate the gear ratio.

The final gear turn in a clockwise direction, at 16/40 or 2/5 the speed of the first gear.
The figure on the left shows four gears, A~D. A is the driving gear. B and C share an axle, and therefore will turn at the same speed.

Again, we will multiply the gear ratio for every place two gears mesh.

\[
\frac{A}{B} \times \frac{C}{D} = \frac{8}{24} \times \frac{8}{40} = \frac{1}{3} \times \frac{1}{5} = \frac{1}{15}
\]

This time, because there are two gears on the second axle, we must take these gears into account when calculating the final gear ratio.

Gear D will rotate at 1/15 the speed of gear A, in the same clockwise direction.

Other gear facts...

- Because the circumference of a circle is proportional to the diameter of the circle, the number of teeth on a gear is proportional to the diameter – so you can calculate the gear ratio using gear diameters rather than using the number of teeth, if it’s more convenient to do so (for example, if you have a gear that is 12” diameter and you don’t know how many teeth are on it without counting).

- Gears work well only if they are properly meshed – no extra spaces between teeth. If there are gaps between two gears, then the motion on the driver gear may not be transmitted to the follower gear. This is called “slipping.”

- Gears are not perfect, so adding gears to a system also introduces friction and some loss of power. If you put in too many gears, the motor will eventually stop moving because it will not have enough power to overcome the friction.

- The thickness of LEGO gears is the same as width of one beam or one bushing.

- Spur gears can also be connected by **chains**, which eliminates the tricky alignment issue. Gear ratio for a chain system is calculated in the same way as a gear system. Two gears connected by one chain will rotate in the same direction.

- **Pulleys** work in a similar manner to gear-and-chain system. Pulleys, however, can slip easily and therefore can not transfer too much power.
**Other Types of Gears**

Two **bevel gears** combine at 90° angle to change the axis of rotation. Bevel gear can only mesh with itself.

A **crown gear** (on right) has curved teeth, and can mesh with itself or with a spur gear to change the axis of rotation by 90°.

These types of gears are necessary for devices such as the egg beater where a vertical rotation of your arm becomes horizontal rotation of the beaters.

A **worm screw** meshed with a spur gear also changes the axis of rotation by 90°. This type of gearing is called a worm drive.

One rotation of the worm screw will advance the spur gear by one tooth. Therefore, the gear ratio for a worm-to-spur gear system is always 1-to-# of teeth on the spur gear.

A worm can only be used as the driving gear to slow down a system and to increase power. It can never be used as the driven gear. (Try rotating the axle on the spur gear – you won’t be able to!)

Worm drives are often used for slow-moving mechanisms or things that require a lot of power – for example, lifting a heavy object or opening/closing a heavy gate.

A **rack** is used to produce linear motion from the rotary motion of a gear.

Racks can be used with a spur gear or a worm screw.

Racks are commonly seen in things that move linearly – for example, the CD tray in your computer that slides in and out.
Sample Constructions for Various Gear Types (for Class Demo)

**Bevel gear model**
- (6) 1×4 beams
- (1) baseplate
- (1) 2×4 plate
- (2) medium axles
- (2) bevel gears
- (1) bushing

**Crown gear model**
- (8) 1×4 beams
- (1) baseplate
- (1) 2×6 plate
- (2) medium axles
- (1) 24-tooth gear
- (1) crown gear
- (1) bushing

**Gear box model**
- (1) gear box
- (1) 24-tooth gear
- (1) worm screw
- (2) axles

**Rack model**
- (2) 1×4 beams
- (2) 1×8 beams
- (2) 1×4 racks
- (2) 1×4 tiles
- (4) 2×4 bricks
- (4) 2×4 plates
- (1) axle
- (1) 8-tooth gear
- (1) baseplate
Useful Web Links

Lego Engineering: www.legoengineering.com - Contains activities and lesson plans, FAQ's, software updates, etc.

High Tech Kids: www.hightechkids.org - A site for "in Science and Technology Education" (INSciTE). Contains training material for Minnesota FIRST LEGO League, including building tutorials, programming guides, general FLL tips, etc.

Robot Building 101 (and many other training documents): A presentation containing a subset of the "Building LEGO Robots for FLL" document, focusing on building techniques.
(http://www.hightechkids.org/index.php?2-1-1054)

Lego Education: www.legoeducation.com - Vendor for educational LEGO Mindstrom sets. Look for ROBO Technology Set (W979786), Technology Resource Set (W979649), NXT Education Base Set (W979797), etc.