Robotics Curriculum Guide

To the teacher

Today, one discovery leads to another. Since the advent of telecommunications, each scientific discovery is shared with a network of innovators across the world, which in turn leads to other discoveries; information grows exponentially. There have been studies that suggest that the total knowledge of humankind doubles every four years. This growth of information has led to changes in the way that teachers teach, as well as the organizers they must select to prepare children to be contributors in the modern world. Robotics, as an organizer of content, offers educators a unit of study that implicitly demonstrates the application of math, science, and technology as well as introduces children to the following work related competencies: project and time management, resource allocation, information accessing, systems understanding, team work, and problem solving. Robotics allows teachers to introduce academic concepts in contexts that make sense to children. Children begin to gain an understanding of the digital world that they grow up in.

This curriculum guide is designed to use the LEGO(R) Challenge Set, the Robotics Educator, and the Robotics Educator companion workbook. The Robotics Academy also offers additional open-ended design problems and challenges that teachers can access by visiting our academy website at www.rec.ri.cmu.edu/education.

This set of instructional units focuses on:

- Introduction to robotics
- Basic programming
- Electronic control
- Advanced programming using variables
- Mechanics
- Design engineering
- Scientific processes
- Project management, problem solving, teaming, and the development of workplace competencies.

Mathematic Teachers

Robotics enables students to “do” mathematics rather than study it. When robots are programmed to move specific distances students are required to apply geometry, measurement, conversion of units, ratios and proportions... Robotics is a motivational tool that brings math to life in the classroom.

Science Teachers

The mechanics modules in the Robotics Educator are designed to help students understand the iterative nature of scientific exploration. Many of the modules in the mechanics section of the Robotics Educator include an independent, dependent, and control variable. This method of inquiry-based learning begins to teach students that science, unlike mathematics, has many more variables that affect the anticipated outcome or result of what a scientist studies.

Technology Teachers
Robotics as a content area teaches children technological literacy. Children learn how electronics, feedback from sensors, conditional statement, loops, and wait states control the digital technologies that direct the world that they grow up in. In this unit of study, children are organized in teams and placed in situations where they are able to design solutions and solve problems.

Robotics As A Content Organizer
Robotics as a content organizer will allow the teacher to give students meaningful exercises that introduce or reinforce the following applied physics and mathematics concepts: ratios; diameter, radius, and circumference; friction; measurement of distance, time, angles, and speed; light and the electromagnetic spectrum, and basic electricity and circuits. At the same time students will be engaged in design activities that challenge them to develop their own original solution for each open-ended problem presented, thereby developing the “out of the box” thinking that is important for innovators.

Lesson Design

Each lesson is designed to be student centered. The teacher becomes the facilitator and presents information on a need to know basis. Lessons are designed to reinforce math, science, and technology concepts. As the lessons evolve, students begin to recognize the importance of the academic concepts used because they are delivered in context.

An important part of every lesson is documentation. Students are expected to keep a notebook that documents their work. They are required to keep all completed worksheets in their notebook. All written documentation in the notebook will be in full sentences. All sketches will be evaluated using the standard sketch evaluation found in the appendix of this document.

Students will be immersed in activities that require the integration of software and hardware. Students are required to write programs to control their robots. They are introduced to conditional statements, loops, and logic. Part of the project requires each team to document their work on a homepage and to prepare for a presentation using presentation software.

Students will be required to work in teams. The teams will consist of a project manager, a programmer, an engineer, and a communications specialist. Students will be encouraged to work together and exchange roles so they are able to experience each of the roles. The students will be immersed in activities that require them to effectively manage time and materials to complete the project in a successful manner.

Index for Robotics Curriculum Guide

Standards Addressed
.

NATIONAL SCIENCE STANDARDS
1. Systems, Order, and Organization
2. Evidence, Models, and Explanation
3. Constancy, Change, and Measurement
4. Evolution and Equilibrium
5. Form Follows Function
6. Content standards A, B, and C.

NCTM MATHEMATICS STANDARDS
1. Numbers and Operations
2. Algebra
3. Geometry
4. Measurement
5. Problem Solving
6. Reasoning and Proof
7. Communications
8. Connections

ITEA TECHNOLOGY STANDARDS
1. The Nature of Technology
2. Technology and Society
3. Design
4. Abilities for a Technological World
5. The Designed World

The standards that robotics is able to address are linked in the Introduction Module of the Robotics Educator. They are organized under:

- Science Standards
- Math Standards
- Technology Standards

It was an oversight on the developers' part was not to include the communications standards in this document.

Organizers for using the Robotics Educator with this curriculum guide

Quickstarts	Quickstarts are used to explain the functionality of the product. If you are a first time user, take the time to click through the Quickstarts since many setup questions are answered there.
Content Organizers	The Robotics Academy development staff has used the following organizers to introduce robotics: • Introduction • Hardware • Mechanics • Sensors • Programming • Robots. Clicking the blue colored icons across the top of the Robotics Educator accesses these content organizers.
Slide Shows	Slide shows are available throughout the product. In the Hardware content area, the new user will find a series of slideshows designed to help the build structurally sound robots. In the Hardware, Mechanics, Sensors, and Programming content organizers teachers will find a slideshow that can be used as an introductory presentation for that area of study. For instance, in the Sensors content area there is a slide show named "Sensors for Robotics" designed to show the new programmer what sensors are. PowerPoint editions of these slide shows can be found on the root directory of this CD.
Helper Links	Helper links are used throughout the CD. They are easily identified by their brown color. If the text is brown, then there is a link at that point that will take you to more information about that topic. At the bottom of each page of content organizers, the user will find Helper Links about that specific content.
Lessons	Lessons are found in the Mechanics Module. The lessons are designed to teach or reinforce a concept. They will be spelled out in greater detail in the body of this curriculum guide.
Rubrics for assessment	Project based learning requires a different type of assessment models. Customizable rubrics for assessment can be found throughout the CD. They include: • Work Habit Evaluation Form • Workplace Competencies Evaluation Form • Robotic Design Rubric • Presentation Rubrics • Daily Log Editable copies of these can be found on the root directory of this CD.
Editable Materials	In the root directory of the CDROM the user will find a directory called extra resources. The resource directory has subdirectories named PowerPoint shows, Word Documents, and Programs.

Unit One **Introduction to Technology**

Note to the teacher: This introductory lesson is designed to help students understand the current exponential change in the growth of technology. Many of the innovations that play a significant role in children's lives (the Internet, cell phones, GPS, telecommunications, the personal computer, pagers and beepers) are very recent technologies. Most of the inventions and innovations that we as a society find we cannot live without were developed in the past 100 years. A timeline does an excellent job illustrating this. Based on the amount of time you have to spend on this topic this is an excellent research project for students: Develop a timeline that illustrates the advancement of the technological society that we live in.

<u>It is suggested that teachers complete the following:</u>
Prepare for a robotics demonstration to excite your class.

1. Install ROBOLAB(R) software and test IR communications with the RCX.
2. Build several Tankbots and download programs that demonstrate the capability of sensors. This demonstration is to be used as an anticipatory set to excite the class about the robotics activities they will be exploring in the days and weeks ahead. Suggested robots:
   A. Tankbot with a touch sensor attachment – use the programs from Exercises 1, 2, & 3 of the Touch Sensor Exercises in the Programming Module.
   B. Tankbot with light sensor attachment – use the programs from Exercises 1, 2, & 3 of the Light Sensor Exercises in the Programming Module.

Technological Concepts taught in this lesson

Evolution of technology

Relationship between science, technology, and society

Resources

Notebooks, paper, and pencils

LEGO® robotics kits

ROBOLAB software

Robotics Educator workbooks

From the Robotics Educator CD ROM (To build and program your first robot)

1. Hardware Module/Tankbot Slideshow
2. Hardware Module/ Light Sensor Slideshow
3. Hardware Module/ Touch Sensor Slideshow
4. Hardware Module/Rotational Sensor Slideshow
5. Programming Module/Motors and Timers through Rotational Sensors Sample programs

http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontorobotics.htm

(The link above has many resources, including worksheets, tests, and PowerPoint presentations to support introduction to robotics)

http://www.rec.ri.cmu.edu/education/teachertraining/tankbotbldginstr.pdf

Adobe acrobat set of instruction for assembling Tankbot.

Teacher will:

Prepare a presentation on the evolution of technology. This can be done in the form of a timeline. Emphasis is on the evolution of technology.

Help students prepare a list of robots they've seen in the movies and contrast them with today's robotic technology. The key concept is that although the field of robotics is rapidly advancing, Hollywood's robots are more advanced than the robots available on the market today.

Build and program several robots that demonstrate sensors and control. Demonstrate to the students how the LEGO(R) robots and sensors work.

Prepare a short demonstration of ROBOLAB software.

Discuss the sequential nature of programming.

Students will:

Participate in a teacher led discussion contrasting robots students have seen on TV and the movies with real world robots and presentation of robotics.

Observe a demonstration of several teacher-built robots.

Observe a demonstration of ROBOLAB™ software controlling mobile robots.

Discuss the sequential nature of programming. Write a step-by-step set of instructions to accomplish a simple task and then have someone follow that set of instruction to complete the task. Instructions must be very specific – as with programming, general instructions will not work.

Enrichment Activities:

Write a "how to" manual describing in a logical, step-by-step, manner how to accomplish a simple task (program a VCR, wash clothes etc.)

Develop a timeline that illustrates the development of technology.

Evaluation

In class participation

Teacher observation

Completion of enrichment activities
Completion of writing assignments
Work Habit Evaluation

Unit Two **Introduction to Hardware**

Note to the teacher: One drawback to working with LEGOs® is the amount of organization required of the teacher when working with kits with hundreds of parts. Most investigations in Robotics Educator were designed to use a limited number of LEGO parts in to help teachers with their organizational issues. Tankbot, the primary robot for most labs, was developed for classroom use for two reasons:

- First, it is robust, students can drop it and for the most part it maintains its integrity.
- Second, it consists of very few parts.

It is suggested that for all of the initial sensor and programming exercises as well as the mechanics investigations the students build a Tankbot and a set of sensor attachments and then the teacher can put the rest of the kits away until they begin their open-ended design challenges.

It is suggested that teachers complete the following:
Become familiar with the Hardware Module of the Robotics Educator
Go to:
http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontohardware.htm:
Review and download the following Adobe Acrobat files for classroom use:

- Tankbot PDF
- LEGO Parts Reference PDF
- LEGO Identification Worksheet PDF
- LEGO Parts Quiz PDF

Robotip - All LEGO parts from the robotics kits are important, but the expensive parts to replace are the RCX, motors, and sensors. Make sure that each team of students understands their responsibility for the parts from their robotics kits.

Technological Concepts taught in this lesson

Teamwork

Drawing Interpretation

Measurement

Following Instructions

Resources

LEGO robotics kits

The Robotics Educator on each computer

Hardware Module that shows and names all LEGO parts

Robotics Educator workbooks

The following handouts:

- Tankbot PDF
- LEGO® Parts Reference PDF
- LEGO Identification Worksheet PDF
- LEGO Parts Quiz

Teacher will:

Prepare a demonstration on how to identify LEGO parts.

Prepare all relevant handouts for student use.

Demonstrate how to read and interpret the plans from either the Robotics Educator or the Tankbot PDF to the class.

Assign students to organize and inventory their robotics kits.

Assign student partners to work on robotics activities.

Help student teams identify parts to build a Tankbot chassis, touch sensor attachment, and light sensor attachment for future labs.

Students will:

Participate in a teacher led discussion on how to identify LEGO parts.

Participate in a teacher led discussion on how to read and interpret Tankbot plans.

Familiarize themselves with the Robotics Educator

Organize and inventory their robotics kits.

Build Tankbot and sensor attachments.

Enrichment Activities:
Organize, identify, and inventory all LEGO parts from their robotics kits.

Research the field of robotics, visit the Robots Module in the Robotics Educator and see the types of robots currently being developed at Carnegie Mellon University.

Write a one-page report on how robots are being used today.

Evaluation:
Teacher observation
Completion of Tankbot and sensor attachments
Completion of part identification worksheet from workbook
Work Habit Evaluation form

Unit Three **Introduction to the RCX**

Note to the teacher: This is a short instructional unit and may be combined with another unit. The RCX is the programmable controller used by LEGO® robots. In the Robotics Educator Hardware module you will find a helper link named the RCX on the left hand column of the page. This can be used to show students the various parts of the RCX.

<u>It is suggested that teachers complete the following:</u>
Become familiar with the RCX. Understand that the input ports 1,2,&3 can either receive analog or digital feedback. For the RCX to understand how it should read the information, the program must be downloaded and run one time.

Technological Concepts taught in this lesson

Basic operation of the RCX

Sketching technique

Resources

The following pages on the Robotics Educator:

- Hardware Module/The RCX
- Programming Module/Helper Pages/Sensor Ports
- Programming Module/Helper Pages/View Screen
- Programming Module/Helper Pages/Calculating Thresholds
- Sensor Ports page – helper link section found in the Sensors Module
- Sketching and dimensioning worksheets in the Robotics Educator workbook

Teacher will:

Prepare a demonstration of the parts of the RCX

Assign the students to sketch the RCX and label all of its parts

Students will:

Actively participate in a question and answer session using the RCX

Sketch the RCX and label all of its parts

Enrichment Activities:
Depending on the focus of the course, technical sketching is an important skill for all engineers. Assign the students to sketch Tankbot, measure and dimension it. Use the sketching and dimensioning worksheet included in the Robotics Educator workbook.

Evaluation:

Teacher observation
Completion of assigned work
Sketch Evaluation Form
Work Habits Evaluation form

Unit Four **Introduction to Electronic Control**

Note to the teacher: This is an introductory lesson using ROBOLAB™ for simple electronic control. In this lesson, the teacher will demonstrate how to turn things on and off using ROBOLAB's timing features. Timing is the least accurate method of controlling motors when programming robots because of battery power. (A weak battery will move a robot slower than a fully charged battery. This affects the distance the robot will travel.) A common problem beginning roboticists have deals with polarity; the direction the current flows. When students design, program, and build a robotic system they must have the program correct, the mechanics correct, all inputs and outputs secured correctly, and use the proper polarity if they want their robot to behave correctly. In this lesson, the teacher has the opportunity to discuss continuity, polarity, and power generation; all important topics for the future scientist. This lesson will teach students how to use ROBOLAB software to turn motors and sounds on and off using the "motors, timers, and sounds" control features built into ROBOLAB.

Robotip - *Demonstrating ROBOLAB software.* Each teacher will have his or her own style of teaching. It is suggested that students be taught rudimentary ROBOLAB rules. Teach them how to use the help screen, and then let them begin to explore by themselves. Each day more complex ROBOLAB control features will be added. If you are new to ROBOLAB, a ROBOLAB Programming Slideshow is available at the top level of the Programming Module in the Robotics Educator

Robotip – Many students do not understand the concepts of polarity, generation, and continuity. The RCX, motors, and wheels quickly demonstrate these important science concepts.

<u>It is suggested that teachers complete the following:</u>
Go to the root directory of the Robotics Educator and get the following PowerPoint presentations

- RCX maintenance PowerPoint presentation
- Electronic Generator PowerPoint presentation

Modify the presentation to meet your classroom needs.
Go to:
http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontosoftwareandelectroniccontrol.htm
This site is at Carnegie Mellon's Robotics Academy. We will be referencing this site often in this guide because there are resources there that you may want to use. Review the following presentations and worksheets:

- ROBOLAB PowerPoint presentation 1 "motors, speeds, and sounds"
- Programming worksheet and worksheet solutions
- Basic ROBOLAB worksheet and worksheet solutions

Prepare a presentation demonstrating electronic control systems.

Robotip - It is extremely important to have students cover their IR tower and robot whenever multiple users are beaming programs. When not in use, robots should be turned off so they don't pick up stray signals. If a robot receives a signal from multiple IR towers at the same time the firmware becomes corrupt and needs to be downloaded again. (Open the "Download" Helper Link on the Robotics Educator for details.)

Technological Concepts taught in this lesson

Electronic Control

Polarity

Basic Programming

Scientific Method

Logical Thinking

Proportion

Estimation

Measurement

Resources:

Robots, ROBOLAB software, working IR tower

Downloading helper link – found in the Programming Module helper pages section

IR Tower helper link – found in the Programmer Module helper pages section

ROBOLAB Programming Tutorial – Robotics Educator/Programming Module top level

Functions Palette helper link – found in the Programmer Module helper pages section

Simple programming exercises worksheet from the Robotics Academy site

ROBOLAB PowerPoint presentation 1 "motors, speeds, and sounds"

Basic ROBOLAB worksheet and quiz

RCX maintenance PowerPoint presentation

Electronic Generator PowerPoint presentation

Human to Robot programming challenge from the Robotics Educator Workbook

Teacher will:

Prepare a presentation demonstrating continuity and polarity.

Describe the parts of an electric circuit.

Ask students to list examples of electronic control in their lives. Divide the list into two categories: examples of simple electronic control and examples that involve programming and logic.

Prepare a presentation that demonstrates how to use ROBOLAB™ to program the RCX to turn motors, timers, and sounds on and off.

Demonstrate the following ROBOLAB features to students before they start their programs:

- Programmer mode
- Help Screen
- Functions Palette
- Tool Palette
- IR Tower
- Downloading a program

Have the students go to the “Motors and Timers” exercises in the Programming Module and write the programs, then download them to Tankbot and test their programs. Each exercise is designed with a simple challenge. Demonstrate how to build the programs. The focus of the exercise is explained below:

Robotip – wiring the icons is tricky at first. Open the tools palette under window in ROBOLAB. Demonstrate how to wire icons. Icons cannot be place on top of wires and have them work. Demonstrate to the class how to move from one icon in the tools palette to another. Hit the space bar when you are working with the tools palette and you will toggle between the wiring tool and the selection tool. This can be a great timesaver.

- Moving Forward – This is the simplest of programs. The student will wire up motors A&C with the arrow on the icon pointing toward the right (forward) they will place a 4-second timer next to the icon and then an ABC stop icon. The exercise is designed to introduce simple electronic control; turning on an actuator (motor) for a fixed amount of time and then turning it off. *Ask the students what would happen if there was no ABC stop icon at the end of the program.* Try it. The robot would continue to move forward forever. This is the first lesson on the importance of implicitly telling the robot everything you would like it to do when programming.

Robotip – students need to be taught how to save their work. If they do not setup a file structure then their programs will be all over the computer. This is a good time to review how you want them to save their work, where you want them to save their work, and how often their work should be backed up. In the next exercise the student will modify their first program. This is an excellent time to teach them about the save-as function when saving edited programs.

- Moving Forward then Reverse – In this program the student learns how to change polarity using ROBOLAB. They are required to add to their previous program by inserting a reverse (arrow pointing left) motor icon for A&C, they will add a 4 second timer and a stop. Students are required to select and delete a wire from the previous program, move the red stoplight icon to the right to make room for the new icons and then begin adding and wiring the new icons.

- Point Turns – In this exercise students learn the most efficient way to turn Tankbot; the point turn. They will start this program from scratch, build and test it. Although the student has only written three ROBOLAB™ programs they are ready for their first programming challenge. In the Mechanics Module under Design Challenges there is a challenged called "Robo500" which is an extension of this activity. Students will quickly learn that there needs to be a better way to control motors other than timing.
- Modifiers – Modifiers are used in ROBOLAB to change power levels, adjust time, and with sensors and containers. This exercise helps students to begin to use them. Go to: Modifier helper link in the Programmers helper link section.
- Loops – Loops are a very important programming concept. ROBOLAB uses two types of loops; jumps-and-lands and loop icons. This program uses a loop icon and has the robot travel in a square shape for a predetermined amount of time.

Students will:

List examples of electronic control in their lives

Complete the "Generator Lab" from the Robotics Educator Workbook.

Observe a teacher prepared robot demonstration.

Observe a demonstration of ROBOLAB

Build the programs from the "Motors and Timers" section in the Programming Module of the Robotics Educator. Upload them to Tankbot and test their robots.

Complete the "Simple Programming Exercises" worksheet.

Enrichment Activities:

Divide students into pairs. Have each pair of students write a descriptive program to accomplish a simple task i.e. Pick up a pencil and follow a line through a maze, or pick up a pencil and draw a line, etc. and then have another team execute the program exactly as written. Have the students execute the code with their eyes closed. Have them execute the code with work gloves on. Discuss what worked and what didn't. Write a one-paragraph summary of what was learned.

Use estimation and basic proportions to predict distance when changing the variable time in the program. Write programs to travel specific distances. Develop a PowerPoint presentation on the data your team developed when completing the two enrichment activities.

Evaluation

Teacher observation

Completion of assigned work

Work Habit Evaluation form

Teacher designed quiz

Unit Five **Introduction to Programming**

Note to the teacher:

In this instructional unit students add the "conditional statement" programming concept to loops and begin to "teach" their robot more complex behaviors. They learn what makes a robot different from a machine. They learn how to use feedback from sensors to make their robot complete a simple task. This unit will be separated into six parts:

- Introduction to Programming
- Touch Sensors
- Light Sensors
- Rotational Sensors
- Temperature Sensors
- Advanced Programming Concepts

The initial programming exercises will use the light and touch sensor attachments previously built, and couple them with Tankbot. The rotational and temperature sensors are not included in the Robotics Challenge set, but can be purchased separately. They will be covered last. In the advanced programming section, students will be taught how to convert the light sensor into a tool that can act as a rotational sensor. There are many resources available on the Robotics Educator and at the Robotics Academy website: www.rec.ri.cmu.edu/education. The intent in the next couple of paragraphs is to help you to identify where they are located and how you may use them.

When a programmer programs a robot they "teach" their robot a simple behavior. Examples of simple behaviors are:

1. Turn a motor on for a time and then turn off,
2. Turn two motors on in the forward direction until a sensor is activated and then turn both motors off.
3. When tracking the edge of a line, a programmer turns one motor on in a forward direction and turns another motor off until a sensor sees a value and then turn the first motor off and the second motor on in the forward direction until a sensor sees a different value… do this forever.

These simple behaviors are coupled with feedback from sensors and then added together to make robots do complicated behaviors. Advanced programming involves combining:

- Simple behaviors
- Boolean logic and
- Feedback from sensors

The perfect example of putting together simple behaviors to accomplish complex and even amazing tasks is the computer. The computer is based on "1's" and "0's" and by organizing these bits of code in particular patterns we see what we see on the computer screen. Go to http://computer.howstuffworks.com for more on this.

One way to teach students to program their robots is to use the Robotics Educator and the Programming Module. This module shows the user to build the code used to "teach" simple behaviors. Students build the code with ROBOLAB™ software, upload it to the RCX, and test it on Tankbot. After the code is tested on Tankbot, they are given a challenge to change the code to have the robot perform another behavior. The Programming Modules includes videos of what the robot may look like when it is programmed correctly.

If you have more time to devote to teaching programming and would like additional programming challenges and solutions you may want to visit: http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontoprogramming.htm. At that link you will find six PowerPoint presentations and supporting activities designed to sequentially teach both how to program using ROBOLAB and the logic behind using loops, conditional statements, multitasking and containers.

It is suggested that teachers complete the following:
Practice using the Robotics Educator and ROBOLAB together so that you are able to demonstrate to your students how to multitask between both programs as they are learning to program.

Review the Robotics Academy site listed above to gain more insight how on how to teach programming concepts. The PowerPoint presentations are downloadable for your convenience.

Robotip An important concept for students to understand is how the RCX input ports 1,2, &3 work. Ports 1,2 &3 receive feedback from sensors. These ports can be connected to touch, light, rotation, and temperature sensors. A touch sensor sends digital feedback; the other sensors send analog feedback (a range of numbers). Before a port is able to correctly read feedback from a sensor it has to be activated by uploading a program to the RCX. Once the program has been uploaded to the RCX it needs to be run one time. If the program is not run (executed), the sensor does not know what type of feedback to prepare for. In order to see the feedback that the sensor sees the user will use the view button. More on this will de described when it is appropriate.

Technological Concepts taught in this lesson

Electronic Control

Analog and Digital Feedback

Polarity

Programming Concepts

Logical thinking

Teamwork

Resources:

Sense-Plan-Act slideshow from the Robotics Educator root directory, this slideshow teaches what makes a robot a robot. A non-editable copy can be found at the top level of the programming module. It is named “Sensors for Robotics.”

Programming Module – light sensors and touch sensors exercises

Light Sensor engineering and programming pages from the Robotics Educator Sensors Module

Touch Sensor engineering and programming pages from the Robotics Educator Sensors Module

Robotics Educator workbook assignments for touch and light sensors

Helper pages from both the Sensors and Programming Modules on the Robotics Educator

Robotics Academy website. English curriculum “Introduction to programming” module. There are six downloadable PowerPoint presentations and also several programming exercises that go with the presentations

Teacher will:

(Remember this is only the first of six units involving programming, there are two points of emphasis here: what makes a robot a robot, Sense-Plan-Act, and how to program using the ROBOLAB™ environment.)

Prepare a presentation and lead a discussion on “What Is A Sensor?”

Compare and contrast the 5 human senses to sensors available in robotics. Ask if they know of any more human senses (Sense of Balance and Sense of Kinesthetics [knowing where parts of your body are at any time]). How do you build a robot to compensate for these senses?

Show the PowerPoint presentation “Sensors for Robotics” from the Robotics Educator Sensors Module. The teacher should emphasis the Sense-Plan-Act nature of a robot. In other words, a robot senses what is in its environment, makes a plan based on feedback from its sensors and the logics of its program, and then acts accordingly.

Demonstrate step-by-step to students how to build a program using ROBOLAB software.

Review the following ROBOLAB concepts:

- Programmer mode
- Help Screen
- Functions Palette
- Tool Palette
- IR Tower
- Downloading a program

Help students identify new ROBOLAB icons.

Assign students ROBOLAB worksheet from the Robotics Educator workbook.

Assign students practice in building the simple problems from the Programming Module in the Robotics Educator. More detail about each of the sensors and the point of emphasis covered in each program will be covered in the next unit.

Students will:

Actively participate in a teacher led discussion on what makes a robot a robot.

Identify and contrast human senses and sensors.

Work in teams to write ROBOLAB programs that are assigned by the teacher.

Use the basic programming concepts learned to program their robots to complete simple behaviors.

Complete all assignments from the Robotics Educator workbook.

Enrichment Activities:

Select, research, and write a report on the sensor of their choice.

Evaluation:

Teacher evaluation
Completion of assigned work
Work Habits Evaluation
ROBOLAB quiz.
Teacher developed test

Unit Introduction to the Touch Sensor

Note to the teacher:
In this unit it will be explained in more detail how a touch sensor acts like a switch. Touch sensors can be programmed to be normally closed (wait for let-go) or normally open (wait for push). You will see those two icons and the exercises to teach those concepts in the Programming Module under touch sensor exercises. The first module is "Wait for Push." Students will also learn what "digital" feedback is. The feedback from a touch sensor is either a "1" or a "0". If the feedback on the view screen of the RCX is either "1" or "0" it is a digital feedback. Analog feedback on the other hand is a range of numbers. Examples of analog feedback will be given when we work with the other sensors.

As we cover the investigations from the Programming Module we will also start to teach students what conditional statements and loops are. The logic taught in each programming exercise will be explained in detail when each example is covered in the "Teacher will" section of this guide.

<u>It is suggested that teachers complete the following:</u>
Open ROBOLAB™ and build and test the four programs from the touch sensor exercises from the Programming Module in the Robotics Educator

- Wait for Push
- Wait for Let Go
- Bug Bot
- Remote Control

Examine the "Tablebot" challenge found in the Mechanics Module under "Design Challenges" – Moderate level.

Examine the "RoboMazing" challenge found in the Mechanics Module under "Design Challenges" – Difficult level

Practice using the view button on the RCX to see the types of input received from each port.

Visit http://www.rec.ri.cmu.edu/education/roboticscurriculum/touchsensor.htm and look at the worksheets with solutions available for touch sensors and either use them or develop your own for classroom use.

Technological Concepts taught in this lesson

Systems

Electronic Control/Controlling Current

Using Touch Sensors

Electric Circuits

Digital Feedback

Logical Thinking

Programming Concepts

Resources

Touch Sensor Building slideshow in the Hardware Module

Touch Senor engineering and programming pages found in the Sensors Module

Sensor Ports helper page – helper link section found in the Sensors Module

View Button helper page – helper link section found in the Sensors Module

Touch Sensor helper page – helper link section found in the Sensors Module

Teacher will:

Discuss the types of feedback that robots use to make decisions. (Analog and Digital)

Show the students the "Touch Sensor programming page" found in the Sensors Module. Move the mouse over the touch sensor yellow button and show them what happens on the view screen of the RCX when you are in "View Mode."

Show the students the second page from the "Touch Sensor Programming page" found in the sensors module. Have the students identify why the program will not work. Roll over the program and see the answer.

Discuss the term "system". A system is a group of parts that works together to accomplish a task.

Discuss how all parts of the robotics system has to be correct to get the results that you expect when designing, building, and programming a robot. That is all of the following needs to be correct if the robot is to work as expected:

- The mechanics have to be correct
- The polarity has to be connected correctly
- The outputs and inputs have to be connected to the correct ports
- The program has to be correct
- The logic has to be correct

Prepare a demonstration that demonstrates "wait for let go" and "wait for push."

Demonstrate how to use "view mode" on the RCX. In order to use the RCX view screen the program has to be "run" one time after it is uploaded to the RCX. That enables the port to know the type of feedback to expect from the sensor. Once the program has been run, the operator can press the view button until the indicator under the port aligns with the port that the operator wants to get feedback from.

Wait for Push/Touch Sensor exercise - The first programming exercise uses a wait for push icon. This program demonstrates how a *conditional statement* works in programming. In this example both motors will move in the forward direction until the touch sensor attachment is pressed. When the sensor is pressed, the condition is met, and the motors move in reverse for one second, turn randomly for three seconds and stop.

Students will:

List the parts of an electronic circuit.

Identify the difference between digital and analog feedback from sensors.

Describe how a touch sensor works.

Work independently in the ROBOLAB^{TM} environment to complete all teacher assigned work.

Test student written programs on Tankbot

Complete teacher assigned notebook activities.

Enrichment Activities:
Sketch a diagram of a simple circuit and label it.

Design a robot and write a program that allows you to use remote control.

Complete either the "Tablebot" or "Robomazing" design challenges described in the Mechanics Module under Design Challenges.

Evaluation:
Teacher Observation
Work Habit Evaluation
Completion of Assigned work
Notebooks
Sketches
Teacher Developed Quiz

Unit Introduction to Light Sensors

Note to the teacher:
Light sensors use analog feedback. The light sensors that come with the Robotics Challenge Kits have two main parts: an LED (Light Emitting Diode) and a phototransistor. This is shown at the Sensors Module on the Light Sensor Engineering page. This type of light sensor is an active light sensor. An active sensor supplies its own light source; in this case, the LED. The LED emits a beam of red light and the phototransistor detects the reflected value of that light. The reflected light is registered as a number and is fed back to the RCX. A passive light sensor does not have its own light-source; it reads the ambient light in its environment as its feedback.

The type of feedback that a light sensor gathers is known as analog feedback. Analog feedback consists of a range of numbers. The RCX light sensor has a range between 0-100. The lower numbers represent darkness and the higher numbers represent light. When using this sensor your feedback generally will be between 35 and 55. Whenever using a light sensor the programmer will need to calculate the threshold value of the sensor under those light conditions. The threshold value of the sensor is the average value between what the sensor sees when it sees light and what the sensor sees when it sees dark. Using the numbers above, 35 & 55, the threshold value would be 45. Add 35 + 55 and the result is 90. Divide 90 by 2 and the result is 45. Any light value less than 45 will be considered to be dark; any light value equal or greater than 45 will be considered to be light. More can be found about calculating threshold values at the Light Sensor Programming Page, page 2. You can find additional information about calculating threshold values on the Threshold helper page found at the Sensors helper link section.

Robotip – The ambient light in the room will affect the threshold value of the sensor. If your students are testing threshold values in a bright section of the room and then it gets darker their light sensors may not work accurately.

The light sensor gives a perfect example of how color absorbs and reflects light. When the light sensor is reflected towards a dark surface the surface absorbs the light and the value of feedback shown on the view screen is low. When the light sensor is reflected towards a bright or white surface there is a larger amount of light reflected back towards the sensor, and the value on the view screen is larger.

<u>It is suggested that teachers complete the following:</u>

Calculate the value of threshold and use modifiers to program your light sensor to work with all programs.

Write the programs demonstrated in the Programming Module of the Robotics Educator and test them for a classroom demonstration.

Go to: http://www.rec.ri.cmu.edu/education/roboticscurriculum/lightsensor.htm

You will find several worksheets with questions that check for students understanding about light sensors.

Technological Concepts taught in this lesson

Light, reflection of light

Measurement

Analog Feedback

Calculating Threshold

Logical Thinking

Problem Solving

Teamwork

Programming Concepts

Resources

Light Sensor Engineering and Programming helper pages

Calculating Threshold helper page

http://www.rec.ri.cmu.edu/education/roboticscurriculum/lightsensor.htm Light sensor curriculum from the Robotics Academy website.

RoboTracker, FireFly Bot, and Table Bot Design Challenges found at the Mechanics Module under Design Challenges

Light Sensor Building Slideshow in the Hardware Module

Teacher will:

Prepare a presentation on the various vision systems used in robotics today. Including sonar, laser, Infrared, and stereovision systems.

Contrast and compare the various vision systems.

Describe the parts of the light sensor; LED and phototransistor.

Demonstrate how to program with the light sensor.

Explain what threshold is and demonstrate how to calculate it.

Robotip – There are two light sensor icons that look very similar but act significantly different. It is worth pointing out to the student the difference between wait-for-dark and wait-for-darker. Wait-for-dark is used in all of the examples that are shown in the Robotics Educator. Wait-for-dark requires the user to calculate a threshold value and use it inside the conditional statement. Wait-for-darker will activate the conditional statement when the light sensor senses a value that is five percent less than the value the sensor is currently seeing. As a teacher you will see students using and confusing these icons daily. It is important that you read the help files in ROBOLAB so that you are able to recognize when they have confused the use of the two icons. There are also a set of icons named wait-for-light and wait-for-lighter, they use the same principles.

Robotip – Whenever using a light sensor mount the sensor so that it is as close as possible to the surface that it is to be measuring; approximately 2-3 mm if possible. If the light sensor is more than a decimeter away the feedback the robot gets from the sensor begins to vary and may impact how your robot behaves.

Assign students the following programming exercises from the Robotics Educator Programming Module:

Wait for Dark – This programming exercise introduces the following concepts:

- Conditional statements
- Calculating Thresholds
- Using a Light Sensor

The student starts with a program that turns motors A&C moving in the forward direction. They are required to add a wait-for-dark icon. The icon may or may not work without a modifier. The icon has a default setting of 55 and it depends on the ambient light in the room, the reflectance value of the surface, and the power level of the batteries. The student will need to calculate the threshold value and insert a modifier to make the program work correctly every time. The light sensor icon serves as a conditional statement. When the light sensor senses a value that is less than threshold the robot will stop.

Robotip – All icons in ROBOLAB have a default setting. The default setting is the setting that the icon is preset to use if the programmer doesn't use a modifier. The default setting for the light sensor is 55. Experience has told us that this is usually too high to be used as a threshold.

Wait for Light – In this exercise the programmer writes a program using a jump and land icon (the red up and down arrows) that will execute the program forever or until you press the button. As long as the robot sees light it will continue to move forward. When it registers dark it will backup, make a random turn up to 5 seconds, and then continue forward again until it sees dark. (It is up to five seconds because that is the default setting for the random command. Information about each of the icons can be seen by turning on the help screen and rolling over the icon. THIS IS A WAIT FOR DARK PROGRAM!

Line Track/Conditional Statements – This is the first program that requires the use of a fork and fork merge. Every time programmers uses a fork in ROBOLAB™ they are required to have a fork merge (green "y" shaped icon) In this program; the programmer embeds a conditional statement, the light sensor fork, inside a forever loop, the jump and the land. This program demonstrates how to write a program to track the edge of a line. The line track program either tracks the right or left edge of the line. It alternately turns one motor on and one motor off based on the feedback given to the light sensor. This particular program is not very useful, because the robot tracks the line forever.

Robotip – remember to place the proper modifier with the proper port. What type of feedback would you get if you wrote a program for a sensor from port one but you actually plugged the sensor into port three?

Line Track/Timer Sensor Fork – This program begins to show the user the power of ROBOLAB. The programmer has two nested conditional statements inside a loop. The first conditional statement checks the time on the timer. If the time is less than ten seconds it will continue to execute the code. As the program moves forward it checks the feedback from the light sensor. If the light sensor sees dark it will turn on motor C in the forward direction and turn off motor A. Otherwise it will take the top path off the fork. When the program reaches the jump (red up arrow) it jumps back to the land and rechecks the time. The program will continue to cycle through until the timer value is greater than ten seconds, at that point it will stop the program.

Robotip – ROBOLAB allows the programmer to use up to three timers. The timers are identified by the colored timer modifiers. Timers need to be reset before they are used. Timers are reset by using the Clear-Timer icon. A common mistake of programmers using a timer sensor fork and a light sensor fork is to place the land outside the Clear-timer icon. When that happens, the time is continually reset and the program never ends.

Students will:

Participate in a teacher led discussion on how light sensors can be used in conjunction with wait-for statements, program control loops, and light sensor conditional statements to make robots move autonomously.

Build the light sensor attachment for Tankbot and complete all assigned light sensor programming assignments.

Describe the parts of a light sensor

Calculate threshold values

Properly identify the difference between the wait-for-dark and the wait-for-darker light sensor icon.

Enrichment Activities:
Research various vision systems that are used in robotics today.

Write a program that will track one side of a line, executes a point turn and then returns on the same side of the line it started on.

Evaluation:
Teacher Observation
Completion of Assigned Work
Work Habit Evaluation form
Completion of enrichment activities
Teacher designed quiz
Notebook check

Unit Introduction to Rotation Sensors

Note to the teacher:
The rotational sensor (also called an angle sensor) allows the programmer to track how many times a wheel or axle spins. The resolution of the rotational sensor available from LEGO® is 16 ticks per revolution. That means that for every time a wheel directly attached to an axle through this sensor turns one complete revolution that the sensor will count to 16. This is illustrated in an animation on the Rotation Sensor Engineering page in the Sensors Module. The feedback from the rotational sensor is an absolute value, i.e., it makes no difference whether the axle is turning clockwise or counterclockwise.. Whenever the Rotation Sensor is used the Zero reset icon should be used first. The rotation sensor acts like a counter; if you want to get accurate feedback from the sensor, then you need to reset the counter before you begin monitoring the feedback.

Robotip – Rotational sensors and the zero reset icon must be linked by a modifier that specifies the port that the rotation sensor is connected to.

Typically rotational sensors are connected to wheels or arms. The distance a robot travels is determined both by the rotational sensor and the diameter of the wheel. As the diameter of the wheel increases the distance traveled each rotation increases. In the Robotics Educator workbook there is a lab that asks the student to measure the diameter of each of the wheels in the Robotics kits, calculate the circumference of the wheels, and then calculate the distance traveled per tick with each wheel. This is an excellent way to immerse students in activities that involve conversion of units, measurement, and applied geometry, three areas that student do not test well in.

<u>It is suggested that teachers complete the following:</u>
Build and test each of the rotational sensor attachments shown in the Hardware Module. There are two slide shows that show how to attach a rotational sensor to Tankbot.

- One of the shows has a rotational sensor attachment with a 1 to 1 gear ratio between the axle that is being turned and the sensor that is being read.
- The other robot has a 24-tooth gear turning the drive axle on the wheel, which turns a 40-tooth gear that turns the axle attached to the sensor. This gives the sensor a 3 to 5 gear ratio between the axle being driven and the rotational sensor. Each time the axle on the wheel spins five times the rotational sensor axle only spins three times. The resolution of the feedback on this robot is 3/5*16 or approximately 9 ticks per rotation of the wheel.

Visit the Robotics Academy website below and download the PowerPoint presentation and the worksheets and modify them for your classroom use.
http://www.rec.ri.cmu.edu/education/roboticscurriculum/rotationalsensors.htm

Review the following helper links at the Sensor Module page: Calculating Ticks, Circumference, Rotational Sensors, Rotational Sensor Programming, Rotation Sensor & Gear Ratios

Technological Concepts taught in this lesson

Parts of a Circle

Measurement

Rotational Speed

Angular Rotation

Distance

Logical Thinking

Programming Concepts

Resources:

Rotation Sensor engineering and programming pages at the Sensors Module

The following helper links at the Sensor Module page: Calculating Ticks, Circumference, Rotational Sensors, Rotational Sensor Programming, Rotation Sensor & Gear Ratios

Rotation Sensor exercises in the Programming Module: Straight & Point Turn

Worksheets and handouts from the Robotics Educator workbook

Worksheets and handouts from the Robotics Academy site listed on the next line http://www.rec.ri.cmu.edu/education/roboticscurriculum/rotationalsensors.htm

Using Robots to Teach Math and Science PowerPoint presentation in the root directory of the Robotics Educator

Rotational Sensor Worksheet and Solutions PDF root directory of the Robotics Educator

Teacher will:

Review the parts of a circle.

Prepare a demonstration on how robots use rotational sensors to accurately move from point to point.

Locate and identify the new programming icons available to be used with the rotational sensor.

Assign students the two simple programming exercises from the Programming Module.

Straight – The rotation sensor exercise straight teaches students that they must use a zero reset icon before they use their rotation sensor. A point of emphasis in this exercise is to make sure that the student links a port modifier to both the zero reset icon and the rotational sensor icon.

Turn – In this exercise students are asked to use feedback from a rotational sensor to accurately turn 45, 90, and 180 degree angles. From this exercise students begin to understand the advantages and disadvantages of this sensor. When using a rotation sensor to accurately execute a point turn whether the programmer is going to activate the motor to move forward or backward will depend on which axle the sensor is connected to. For accurate feedback, the rotational sensor MUST be connected to an active wheel/motor.

Students will:

Participate in a teacher led discussion on what a rotational sensor measures and how rotational sensors can be used in conjunction with wait-for statements, program control loops, and rotation sensor conditional statements to control autonomous robots.

Participate in a teacher led discussion on the use of rotational sensors to measure the speed of rotating axles using compound gear ratios.

Complete all assigned worksheets.

Complete the two rotational sensor programming exercises from the Programming Module section of the Robotics Educator.

Enrichment Activities:
Complete one of the following open-ended design problems from the Design Challenge Section on the Mechanics Module page:

- BotMower
- RoboMazing
- LinePainter Bot

Use a rotation sensor fork coupled with a light sensor fork to accurately track a line for a given distance.
Robotip – Remember that timing is the least accurate method of programming a robot to move a distance. As the batteries become weaker, the robot does not move as fast. A better way to track a line for a specific distance is to use a rotation sensor fork and a light sensor fork.

Attach a rotational sensor to each wheel on your robot. You probably have discovered by now that when you program your robot to move forward in a straight direction that it may drift to one side or the other. Write a program using feedback from the two rotation sensors to monitor the feedback from both wheels at the same time to make sure that the robot moves forward in a straight line. You will need to use containers to do this.

Evaluation:
Teacher Observation
Completion of Assigned Work
Work Habit Evaluation form
Completion of enrichment activities
Teacher designed quiz
Notebook check

Unit Introduction to Temperature Sensors

Note to the teacher:
The temperature sensor does not come with the Robotics Challenge set, but can be purchased separately at LEGO®. The temperature sensor sends an analog feedback to the controller. The temperature sensor reads in both Fahrenheit and Celsius and the programmers have to specify which in their program.

<u>It is suggested that teachers complete the following:</u>
Test feedback from the sensor in a simple program

Technological Concepts taught in this lesson

Measurement

Logical Thinking

Programming

Resources

Temperature sensor

Teacher will:

Demonstrate the Temperature Sensor

Students will:

Write a simple program that uses feedback from the temperature sensor to have the robot do something.

Enrichment Activities:
Explore the investigator mode of ROBOLAB.

Evaluation:
Teacher Observation
Completion of Assigned Work
Work Habit Evaluation form
Completion of enrichment activities
Teacher designed quiz
Notebook check

Unit Introduction to Advance Programming Concepts

Note to the teacher:
ROBOLAB™ is an incredibly powerful programming language. In the initial exercises we have only begun to use the potential of this program. The robotics guide that we've created was not to teach ROBOLAB, but to introduce the student to the concept of robotics.

In this unit we will begin to show the programmer how to use some of the more powerful features of ROBOLAB, specifically:

- Containers, which are analogous to variables in other programming languages.
- Multitasking, which allows the processor to monitor two separate programs as the robot completes a task and.
- Subroutines, which allows the programmer to organize his/her code into organized units and call those units as they are needed.

These topics will be covered in a general manner and for the students to become proficient in the use of containers, multitasking, and subroutines they will need practice.

It is suggested that teachers complete the following:
Download ROBOLAB PowerPoint presentations 5 and 6 from the following site:
http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontoprogramming.htm

The PowerPoint presentations are used with the ROBOLAB programming problems assignment 5 and 6 from the same page.

ROBOLAB sells a training manual through LEGO®
**Teachers, if this is confusing to you, join the club. I have been training teachers to program robots for four years. Some of these advanced concepts are hard to get. In today's world when you are teaching advanced technology its alright if you are not the expert at everything … YOU ARE TRYING and that has to amount for something.*

Technological Concepts taught in this lesson

Electronic Control

Measurement

Conversion of Units

Basic Algebra

Geometry

Logical Thinking

Programming Concepts

Resources

Containers and Timers PowerPoint presentations from Robotics Academy site

Programming examples of multitasking, containers, and subroutines

Advanced Programming Worksheet PDF in the root directory of the Robotics Educator ROBOLAB™ tutorial purchased from LEGO®

Teacher will:

Prepare presentations on:

- Containers
- Timers
- Multitasking
- Subroutines

Write several sample programs that demonstrate how these concepts work and walk the students through them step-by-step.

Robotip - Containers are places to store values; variables. The values in the containers can be added to, subtracted from, multiplied, or divided. Whatever you can do to a real number, you can do to a container. The bewildering part about containers for beginning ROBOLAB programmers is confusing the two types of containers available: one container is to store a value, and the other container is to place a value. Keep this in mind when you are using containers. Use the Help screen if any confusion arises.

Containers – The containers example in the Advanced Programming exercises section does the following:

- The first icon resets the rotational sensor connected to port three to zero
- The next icon resets the red container to zero
- The next five icons turn on A & C in the forward direction until the value of the light sensor on port 1 is less than 40; in other words the motors move forward until it sees dark. At that point we will stop all motors and pause for one second
- The new icon sets the container value to the value of the rotational sensor connected to port three; in other words whatever the value the rotational sensor saw at that point was placed in the red container.
- Now, the rotational sensor should be set to zero. The red container will still contain the value that was in the rotational sensor.
- Both motors are then reversed until the rotational sensor moves whatever value was stored in the red container. Notice the color of the container is different. The white outlined container is the numerical value of the container and so it can be used as a numerical constant.
- The robot stops

Rotational Sensor Fork – This probably doesn't need to be in the advanced programming concepts but here it is. This challenge was placed in the "Introduction to Rotational Sensors" unit. Here is a working solution. If it doesn't need to be here, where should it be?

Multitasking – This is a valuable lesson for teachers and students that do not have a rotation sensor.

Robotip – If you would like to demonstrate to your student how an encoder works you could attach a black and white disk to a gear and attach the whole assembly to an axle. Point the light sensor at the gear/pinwheel assembly to begin to record feedback. The multitasking program and PowerPoint presentation demonstrate how feedback from a light sensor can be converted into data that controls how far a wheel turns.

This exercise shows how to convert a light sensor into a rotation sensor. In the root directory of the Robotics Educator in the PowerPoint section there is a presentation that can accompany this lesson. The program starts with a task split, the yellow "y" shaped icon. From that point on there are two separate programs.

The top program:

- Sets the blue container to zero and then begins counting forever because of the jump and the land.
- When the sensor at port 1 sees a value less than 40 it will add "1" to the red container.
- When the sensor at port 1 sees a value greater than 40 it will add "1" to the red container.
- The top program will continue to execute that code. Remember that the light sensor attached to port 1 is looking at a black and white disk that spins when the axle spins,

The bottom program:

- begins by checking the value of the blue container, which the top program continues to add to. If the value is less than 40 it will turn motors A & C in the forward direction.
- There is a jump after the motor icons, which takes the program back to check the value of the blue container. When the value of the blue container is larger than 40 the robot stops.

Subroutines – Subroutines allow the programmer to write snippets of code and assign the code to a subroutine. ROBOLAB 2.5 uses two new icons to accomplish that. The first icon is the blue sideways "h" shaped icon. This is found in the structures palette of ROBOLAB. Help on using subroutines can be found in ROBOLAB 2.5. They are very straightforward to use.

Students will:

Work independently or in groups to solve the programming challenges.

Unit **Introduction to Mechanics**

Note to the teacher:

Over the past three years, the State of Pennsylvania has tracked the results of children that take the state's standardized assessment tests. The studies provide empirical data that students do not fully understand (they don't test well on the PSSA standardized tests) the following concepts: ratios and proportions, conversion of units, measurement, and applied geometry. The mechanics module in the Robotics Educator is rich with applied math and science investigations that use discovery based learning principles and reinforce the academic concept students struggle with. In the Mechanics Modules children are engaged in a hands-on-mind-on investigation where they are required to "do" science. They learn the academic concepts in contexts that they can relate to.

The Mechanics consists of the following investigations:

- Gears and Speed
- Gears and Distance
- Gears and Torque
- Measurement
- And Belts and Pulleys.

Many of the investigations begin by identifying the independent, dependent, and control variables. All of the investigations require the student to conduct experiments, track data, and summarize their results. A point that students discover immediately is that mathematics is pure and science on the other hand is impacted by many variables. These investigations use discovery based learning and authentic assessment.

The Mechanics Module will be broken into three sections. The First section will consist of investigations involving gears, the second section will cover measurement, and the third section will cover belts and pulleys.

<u>It is suggested that teachers complete the following:</u>
Review the following:

- "Teaching Mach and Science with Robots" PowerPoint presentation from the root directory
- "Gears" PowerPoint presentation from the root directory.
- Review the mechanics module in the Robotics Educator.
- http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontogears.htm

Present students with contextual examples of what they are about to learn.

Read and study each of the investigations presented at the Mechanics Module. Each of the investigations can be modified to teach concepts the teacher may find interesting and relevant to their classroom needs.

Technological Concepts taught in this lesson

Scientific Process

Force

Power

Work

Torque

Speed

Gear Ratios

Mechanical Advantage

Circumference

Conversion of Units

Measurement

Simple Machines

Reading Diagrams and Following Directions

Resources

"All About Gears" Slideshow Mechanics Module

The Following Helper Links from the Mechanics Module:

- Compound Gears
- Fractions
- Gears and Speed
- Gears and Strength
- Idler Gear
- Mechanical Advantage
- Power
- Ratio and Proportion
- Slip Clutch
- Spur Gear
- Torque
- Worm Gear

The following links are from the Robotics Academy website and include valuable information for any student or teacher studying the mechanics of a robot:

http://www.rec.ri.cmu.edu/education/roboticscurriculum/introductiontogears.htm

http://www.rec.ri.cmu.edu/education/roboticscurriculum/gearsspeedtorque.htm

http://www.rec.ri.cmu.edu/education/roboticscurriculum/simplemachines.htm

Unit Introduction to Gears

Note to the Teacher:

The Unit "Introduction to Gears" will use resources from the Mechanics Module of the Robotics Educator. This unit begin a study of the following concepts:

Gears and Speed

Gears and Torque

Gears and Distance

As students begin to work through this unit of study they will work begin to understand the mathematic concept of ratios and proportions. They will also discover the inversely proportional relationship between what is gained using gears and what is lost. This set of investigations emphasizes the Unifying Scientific Concept of "Constancy, Change, and Measurement."

As you read and study each of these investigations the teacher may find that they would like to have the students conduct multiple investigations at the same time. When a student investigates the relationship between gears and speed in "Gears and Speed Investigation One" they may also be able to get their data for "Gears and Distance Investigation One." The setup is the same for each investigation, what changes is the topic of study. The teacher may find it important, based on the amount of time the teacher is able to allocate for this activity, to have the students gather information for more than one experiment while the test bed is built.

The Gear Investigations in the Mechanics Module consist of the following topics:

Gears and Speed Investigation One

In this investigation the control variable will be the distance the robot travels, the independent variable will be the gear ratio, and the dependent variable will be the time it takes to travel this distance. Students will make modifications of the drive train on Tankbot that will change how fast it travels. They are required to measure the distance the robot travels given as set amount of time. They will use this distance to calculate speed.

Gears and Speed Investigation Two

In this investigation the control variable will be the distance the robot travels again. In this investigation the distance is fixed and the robot moves until it activates a touch sensor. The independent variable will be the gear ratios and the dependent variable will be the time it takes to travel his distance.

Gears and Speed Investigation Three

This investigation involves having the students build a gearbox. The control variable will be the initial motor speed connected from a pulley to the gearbox. The independent variable will be the gear ration and the dependent variable will be the rotational speed at each measured point on the gearbox. Student will be required to use a rotational sensor to collect the feedback.

Gears and Torque Investigation One

In this investigation students will build a drive train that has the following gear ratios: 1/1. 3/1, 15/1, and 75/1. The investigation involves testing the lift capacity of a gearbox and comparing the actual lift capacity with the predicted lift capacity.

Gears and Torque Investigation Two

Students build a test bed that allows them to investigate worm gears. They will build test beds that theoretically should have 8/1, 16/1 24/1 and 40/1 mechanical advantages. Students build the test beds, test them, and write up their results.

Gears and Distance Investigation One

This investigation is similar to the Gears and Speed Investigation One. The difference is that in this investigation the control variable is the robots travel time, the independent variable is the gear ratio, and the dependent variable is the distance the robot travels.

Gears and Distance Investigation Two

This investigation is set up similar to Gears and Distance Investigation One. The difference is that this robot will use a rotation sensor to help track the distance traveled. The control variable will be a program that uses a rotation sensor to track the distance traveled, the independent variable will be the gear ratio and the dependent variable will be the distance the robot travels.

Teacher will:

Use LEGO to develop a simple presentation that shows an example of how gears can be used to obtain a mechanical advantage.

Asked the students to describe in their own words the advantages and disadvantages of mechanical advantage.

Demonstrate how to calculate compound gear ratios.

Develop worksheets that students can use to practice these calculations.

Demonstrate the gearbox to students.

Assign students investigations to complete.

Students will:

Observe a demonstration of teacher-selected materials that demonstrate how gears are used.

Participate in a teacher led discussion on that describes: work, force, torque, & power.

Be able to describe the work, force, torque, and power.

Complete all teacher assigned challenges

Enrichment Activities:
Use the investigator module of ROBOLAB to gather feedback from mechanics investigations.

Evaluation:
Teacher Observation
Completion of Assigned Work
Work Habit Evaluation form
Completion of enrichment activities
Teacher designed quiz

Notebook check – all worksheet from labs complete and in the notebook

Unit Introduction to Measurement Using Robotics

Note to the Teacher:

Measurement is an area of study that many children have trouble with. The goal of this instructional unit is to put measurement in context and make the exercise something that is important for the student to understand to accurately program their robot.

The following investigations are found in the Mechanics Section of the Robotics Educator.

<u>Measurement Investigation One</u>

In this lesson students are required to:

- Measure the diameter of the wheels on their robots
- Calculate the circumference
- Use a rational sensor

so that they can accurately have their robots travel a specific distance.

<u>Measurement Investigation Two</u>

In this investigation students are required to use a rotation sensor to control their robot. They will discover how changing the size of the wheel will change the input required to accurately program their robot to travel a specific distance.

<u>Measurement Investigation Three</u>

In this lesson students will learn how to get feedback from their rotation sensor.

Teacher will:

Discuss the topic of measurement and how important it has been in the development of technology.

Work with students in their classroom and develop a list of things that are measured in today's world.

Modify the Circumference PowerPoint presentation to fit their classroom needs and show it to their class.

Demonstrate how the rotational sensor is programmed.

Show students how to use the rotation sensor on their robot to travel a specific distance.

Assign students and investigation to complete.

Students will:

Actively participate in a teacher led discussion on measurement.

List as many things as possible that are measured.

Complete all teacher assigned classwork.

Enrichment Activities:
Complete one of the following Design Challenges:
Robo500
RoboMazing
A teacher-designed problem that emphasizes the Rotation Sensor and measurement.

Evaluation:
Teacher Observation
Completion of Assigned Work
Work Habit Evaluation form
Completion of enrichment activities
Teacher designed quiz
Notebook check – all worksheet from labs complete and in the notebook

Unit Introduction to Belts and Pulleys

Note to the Teacher:

In this lesson students will learn how changing pulley sizes affects the distance Tankbot travels using a rotation sensor.

Teacher will:

Prepare a demonstration.

Students will:

Complete teacher assigned work

Enrichment Activities:
Research example of pulleys.

Evaluation:
Teacher Observation
Completion of Assigned Work
Work Habit Evaluation form
Completion of enrichment activities
Teacher designed quiz
Notebook check – all worksheet from labs complete and in the notebook