



Robotics Educator 2.5 Curriculum Guide

To be used with The Robotics Educator CD-Rom

Developed by Carnegie Mellon University's Robotics Academy

Written by Robin Shoop

Index for Robotics Curriculum Guide

Robotics Educator 2.5 Curriculum Guide	1
Introduction	3
To the Teacher	3
Mathematic Teachers.....	3
Science Teachers.....	3
Technology Teachers.....	4
Robotics As A Content Organizer	4
Lesson Design	4
Standards Addressed.....	5
National Science Standards	5
NCTM Mathematics Standards	5
ITEA Technology Standards	5
Quickstarts	6
Content Organizers	6
Slideshows	6
Helper Links	6
Lessons.....	6
Rubrics for Assessment	6
Editable Materials.....	6
Unit One: Introduction to Technology	7
Unit Two: Introduction to Hardware.....	9
Unit Three: Introduction to the RCX	11
Unit Four: Introduction to Electricity/Electromagnetism	12
Unit Five: Introduction to ROBOLAB™	14
Unit Six: Behaviors/Pseudocode/Flowcharts.....	17
Unit Seven: Introduction to Programming.....	19
Introduction to Touch Sensors	22
Introduction to Light Sensors.....	25
Introduction to Rotation Sensors	29
Introduction to Temperature Sensors	32
Introduction to Advanced Programming Concepts	33
Unit Eight: Introduction to Mechanics	36
Gears and Speed/Constant Time	38
Gears and Speed/Constant Distance	38
Gearbox Hoist/Torque	38
Belts/Ratios	39
Worm Gears/Properties.....	39
Pulleys/Force, Distance, Work	39
Unit Nine: Introduction to Measurement.....	40
Wheel Size/Distance	41
Compound Gears/Ratios.....	41
Fractional Parts/Containers.....	42
Probability Wheel/Chance	42
Equivalent Fractions/Predictions	42
Unit Ten: Motion Planning	43

Discovering Pi/Circumference	43
Measured Turns/Radius	43
Building Behaviors/Programming	44

Introduction

To the Teacher

Today, one discovery leads to another. Each new 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. It introduces children to technological literacy as they develop 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 understand the digital world that they grow up in.

This curriculum guide is designed to use the LEGO® Challenge Set, the Robotics Educator, the ROBOLAB™ video trainer, and the Robotics Educator companion workbook. The Robotics Academy continues to develop additional resources which can be found at the 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.

Mathematics 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 investigations Module in the Robotics Educator is designed to help students learn to apply scientific process. All of the investigations in the Investigations Module include

independent, dependent, and control variables. Students are required to form a hypothesis, test it and either reject it or accept it based on test results. Students quickly discover that science has many more variables which affect the anticipated outcome than mathematics has.

Technology Teachers

Robotics teaches children technological literacy. When students program, they learn how electronics, feedback from sensors, conditional statement, loops, and wait states manage the digital technologies that control the world they live in. They learn about systems, resource allocation, and time management. In robotics competitions they learn to work as a team, they are confronted with open-ended challenges where they develop the innovative problem solving skills needed to compete in the global economy.

Robotics As A Content Organizer

Robotics as a content organizer allows teachers 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 develop work related competencies as they learn to work in teams and manage projects.

Lesson Design

Each lesson is designed to be student centered. Teachers become facilitators; they introduce background information preparing students for investigations. Lessons are designed to introduce/reinforce math, science, and technology concepts. As the lessons evolve, students begin to recognize the relevance of the academic concepts used because they are delivered in context.

An important part of every investigation is documentation. Students are expected to keep a notebook where they track the data collected in their investigations. At the end of each investigation they are required to summarize the data that they collect for each investigation.

Students are immersed in activities that require the integration of software, hardware, and project management. Students are required to write programs to control their robots. They are introduced to conditional statements, loops, and logic. They are required to present the lessons they learned in a presentation format.

Students work in teams. Team sizes change based on the complexity of the project and the age of the student. Younger students should be placed on smaller teams. Teams could 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. They will be immersed in activities that require them to effectively manage time and materials to complete their project in a successful manner.

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

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	<p>The Robotics Academy development staff has used the following organizers to introduce robotics:</p> <ul style="list-style-type: none"> • Introduction • Hardware • Investigations • Sensors • Programming • Robots. <p>Clicking the blue colored icons across the top of the Robotics Educator accesses these content organizers.</p>
Slideshows	Slideshows are available throughout the product. In the Hardware Module, the new user will find a series of slideshows designed to help them build structurally sound robots. In the Hardware, Investigations, Sensors, and Programming content organizers, teachers will find slideshows 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. Editable PowerPoint editions of these Slideshows can be found on the root directory of this CD for teacher use.
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 Investigations Module. The lessons are designed to teach or reinforce an academic concept. They will be spelled out in greater detail in the body of this curriculum guide.
Rubrics for Assessment	<p>Project based learning requires different assessment models. Customizable rubrics for assessment can be found throughout the CD. They include:</p> <ul style="list-style-type: none"> • Work Habit Evaluation Form • Workplace Competencies Evaluation Form • Robotic Design Rubric • Presentation Rubrics • Daily Log <p>Editable copies of these can be found on the root directory of this CD.</p>
Editable Materials	In the root directory of this CD 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 lesson is designed to help students understand the current exponential change in the growth of technology. Many 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 depend on daily were invented in the last 20 years. A timeline does an excellent job of 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.

It is suggested that teachers complete the following:

Prepare a robotics demonstration to excite your class.

1. Install ROBOLAB™ software and test IR communications with the RCX.
2. Build several Tankbots and download programs that demonstrate the capability of sensors. This demonstration is designed to excite the class about robotics.

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

The ROBOLAB video trainer

Robotics Educator workbooks

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

1. Hardware Module/ROBOT 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.

Present a presentation on the "State of Robotics Today." There is a PowerPoint presentation in the root directory of the product that shows robot projects currently being developed at Carnegie Mellon. Visit Carnegie Mellon Robotics Institute's homepage to see current robot projects. <http://www.ri.cmu.edu/>

Build and program several robots that demonstrate sensors and control. Demonstrate to the students how the LEGO[®] 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.

Research what is new in the field of robotics. Students can use the internet, books, or the robot projects linked to the Robotics Educator to begin their research.

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, (pick up a pencil and place it on the table, etc.) and then have someone follow that set of instructions 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

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 the Robotics Educator were designed to use a limited number of LEGO parts in order 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 the initial sensor and programming exercises, as well as for the Investigation Modules, that students build a Tankbot and a set of sensor attachments and use them. The teacher can put the rest of the kits away until they begin their open-ended design challenges.

Teachers should complete the following:

Become familiar with the Hardware Module of the Robotics Educator

Review the following slideshows and pages from the Hardware Module:

- Standard parts
- Specialty parts
- The RCX
- Tankbot
- Sensor attachments

Robotip - All LEGO parts from the robotics kits are important, but the expensive parts 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 ROBOLAB video trainer

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 Hardware Module

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.

Teachers should complete the following:

Become familiar with the RCX. Understand that the input ports 1,2,&3 can receive either analog or digital feedback. For the RCX to understand how it should interpret the signal from the sensor, the program must be downloaded and run one time.

Technological Concepts taught in this lesson

Basic operation of the RCX

Sketching technique

Resources

The ROBOLAB video trainer – RCX preflight, tower setup, downloading

The following link from 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

Robotics Educator workbook - Sketching and dimensioning worksheets

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 Electricity/Electromagnetism**

Note to the teacher: This investigation steps away from LEGO technology. Students are given instructions on how to build an electromagnet, a DC motor, and a generator. In this lesson, the teacher has the opportunity to discuss continuity, polarity, and power generation; all important topics for the future scientist. This investigation is designed to introduce students to four important basic electricity concepts:

1. Basic circuit
2. Electromagnet
3. Simple motor
4. Simple generator

The activities are introductory in nature. They will need to be supplemented with outside research and reading for full understanding. This unit is designed to supplement other instruction as it compliments the robotics instruction.

Technological Concepts taught in this lesson

Basic Circuits

Electromagnetism

Generators

DC Motor

Polarity

Resources:

Two feet (60 centimeters) of enamel-covered wire

D cell battery or 6-volt battery

Two jumbo paper clips

5 Five small disc-shaped magnets

Plastic cup

Scissors

Robotics Educator, Investigations Module, Basic Electricity

Teacher will:

Prepare a presentation demonstrating continuity and polarity.

Describe the parts of an electric circuit.

Prepare a presentation that demonstrates how a circuit works - flick a light switch in the room to demonstrate a circuit opening and closing, or use a touch sensor program with Tankbot.

Build examples of: a circuit, an electromagnet, a DC motor, and a generator. Describe the basic electricity properties that make the circuits work

Assign students outside reading materials to supplement this instruction

Students will:

List examples of electronic control in their lives

Actively participate in a teacher lead discussion on basic electronics.

Complete teacher assigned labs on basic circuits, electromagnetism, DC motors, or generators.

Observe a teacher prepared basic electricity demonstration.

Prepare a presentation on a teacher assigned topic.

Evaluation

Teacher observation

Completion of assigned work

Work Habit Evaluation form

Teacher designed quiz

Unit Five: **Introduction to ROBOLAB™**

Note to the teacher: This lesson will give students practice working in the ROBOLAB™ programming environment. The idea is to let students build a couple simple programs without having to worry about difficult programming logic. The teacher will demonstrate how to turn things on and off using ROBOLAB motors, timing features, and stop icons. Teachers should stress that timing is the least accurate method of controlling 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. Another important concept for students to understand is that when they design, program, and build a robotic system they must have the program correct, the mechanics correct, all inputs and outputs secured correctly, and use proper polarity if they want their robot to behave correctly.

Robotip - Demonstrating ROBOLAB software. We have made a conscious decision to begin programming in programming level 4. For some students that may be too much. Each programming level from 1-4 adds a degree of complexity. You may choose to teach a lower programming level. Teach students how to navigate in the ROBOLAB environment, 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.

Teachers should complete the following:

Build the simple robot behaviors shown in the Programming Module.

Watch the videos from the ROBOLAB Video Trainer.

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

This site is hosted by Carnegie Mellon's Robotics Academy. There are many resources there that you may find helpful in your classroom. Review the following Units:

- Introduction to Hardware
- Introduction to Software
- Introduction to Programming

Look in the Extra Resources folder in the root directory, there are PowerPoint presentations and worksheets in a folder named ROBOLAB Training Materials designed to support teachers teaching basic programming using ROBOLAB.

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

Basic Programming

Scientific Method

Logical Thinking

Resources:

Robots, ROBOLAB software, working IR tower

The ROBOLAB video trainer – RCX preflight, tower setup, downloading, basic programming with timing

ROBOLAB Programming Tutorial – Robotics Educator/Programming Module/Motors and Timers

ROBOLAB Training Materials – Root Directory/Extra Resources/Training Materials

Human to Robot programming challenge from the Robotics Educator Workbook

Teacher will:

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, write the programs, 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 placed 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. The ROBOLAB Video trainer has a 4 minute video that explains wiring.

- 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 wait-for-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 reverse motor icons for A&C, they will add a wait-for-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 challenge called “Robo500” which is an extension of this activity. Students will quickly learn that there needs to be a better way to control their robot 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:

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.

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

Unit Six: **Behaviors/Pseudocode/Flowcharts**

Note to the teacher:

Programming allows your robot to make decisions based on feedback. This allows your robot to complete tasks autonomously. One definition of a robot is “SPA” sense – plan - act. A robot senses what’s in its environment, makes a plan based on what it finds, and executes the plan. When programmers program a LEGO robot they “teach” their robot simple behaviors. They combine the simple behaviors to complete complex tasks. (See the “Behaviors” worksheet in the Investigator Module under the Motion Planning, Building Behaviors Investigation) Examples of simple behaviors are:

1. Turn a motor on for one second and then turn off.
2. Turn two motors on in the forward direction until a light sensor senses a change in feedback and then turn both motors off.
3. Tracking the edge of a line, the robot uses feedback from a light sensor to turn one motor on in a forward direction and turn the other motor off until the feedback from the sensor changes which turns the on motor off and the off motor on, this behavior is placed in a loop.

These simple behaviors are coupled with feedback from sensors and then joined together to build complex robot behaviors. Advanced programming involves combining:

- Simple behaviors
- Boolean logic and
- Feedback from sensors

Lessons from this unit will be applied in all student programming. This unit is introductory in nature and will be applied to the next programming unit.

Teachers should complete the following:

Review the following:

- The behaviors (Investigations Module, Motion Planning, Behaviors Worksheet)
- The pseudocode worksheet (Investigations Module, Motion Planning, Pseudocode Worksheet)
- The Flowcharts Worksheet

Complete several simple programming challenges to prepare for a student demonstration.

Robotip An important concept for students to understand is how the RCX input ports 1, 2, & 3 work. Ports 1, 2 & 3 are inputs and 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 be described when it is appropriate.

Technological Concepts taught in this lesson

Logic

Digital communications

Resources:

The following worksheets:

Pseudocode

Behaviors

Flowcharts

The Building Behaviors Lesson (Investigations Module/Motion Planning/Building Behaviors)

Teacher will:

Introduce basic programming logic using the Robotics Academy worksheets

Assign students several simple programs and have them write the pseudocode and draw the flowchart before the student attempts to write the code.

Students will:

Work cooperatively with their partner completing all teacher assigned work.

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 Seven: **Introduction to Programming**

Note to the teacher:

Make sure that your students understand the concept of pseudocode, behaviors, and flowcharts. That will make the concepts in this chapter easier to understand. In this instructional unit students will apply conditional statements, loops, variables, and multitasking to begin to “teach” their robot more complex behaviors. 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 also called Angle Sensors
- Temperature Sensors
- Advanced Programming Concepts

The initial programming exercises will use Robotics Educator light and touch sensor designs and attach them to 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 these resources are located and how you may use them.

When programmers program a robot they “teach” their robot a simple behavior. See the “Behaviors” worksheet in the Investigator Module under the Motion Planning, Building Behaviors Investigation. Examples of simple behaviors are:

4. Turn a motor on for a time and then turn off,
5. Turn two motors on in the forward direction until a sensor is activated and then turn both motors off.
6. 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 turns the first motor off and the second motor on in the forward direction until a sensor sees a different value... do this forever.

A quick way to get students programming their robots is to use the Robotics Educator’s Programming Module. This Programming Module gives the student examples of simple behaviors. The Programming Modules includes videos of what the robot behavior should look like when it is programmed correctly and then challenges the student to modify the program. The teachers roll is to help the student understand how to combine simple behaviors to make complex behaviors.

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.

Teachers should 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 be 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 emphasize 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 and robotic 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.

Introduction to Touch Sensors

Note to the teacher:

In this unit, students will see 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). They will find examples on how to program “wait for let-go” and “wait for push” 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”. A “1” or a “0” indicates 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.

It is suggested that teachers complete the following:

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 Investigation Module under “Design Challenges” – Moderate level.

Examine the “RoboMazing” challenge found in the Investigation 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

ROBOLAB Video Trainer, Wait for Push, Tablebot, BugBot, Remote Control

Touch Sensor Building slideshow in the Hardware Module

Touch Sensor 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 have to be correct to get the results that you expect when designing, building, and programming a robot. The following needs to be integrated and correct if the robot is to work as expected:

- The mechanics have to be correct
- The wires have to be connected correctly so that polarity is correct
- 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™ 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

Introduction to Light Sensors

Note to the teacher:

Light sensors detect the value of reflected light from a surface. The light sensors that come with the Robotics Challenge Kits have two main parts: an LED (Light Emitting Diode) and a phototransistor. The light sensor system is demonstrated in 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 1-100. The lower numbers represent darker and the higher numbers represent lighter. 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. If the light sensor had a low reading of 35 and a high reading of 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 run their robot in a darker section of the room, their robot may not work accurately.

The light sensor demonstrates how materials absorb and reflect light. When the LED in the light sensor reflects light onto a dark surface the surface absorbs the light and the phototransistor records a low value on the view screen. When the LED is reflected towards a bright or white surface there is a larger amount of reflected light and the phototransistor reads a high value; the value on the view screen is larger.

It is suggested that teachers complete the following:

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

Test the light sensor programs from the Programming Module of the Robotics Educator and demonstrate them.

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

ROBOLAB Video Trainer - Wait for Dark, Wait for Light, Line Tracking, Timed Line Tracking

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 in the Investigation Module under Design Challenges

Light Sensor Building Slideshow in the Hardware Module

Light Sensor Worksheet – Investigations Module, Motion Planning, Building Behaviors

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 reflected 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 Dark – In this exercise the programmer writes a program using a jump and land icon (the colored up and down arrows) that will execute the program forever or until you press the run 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.

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 of 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.

Complete the “Light Sensor worksheet” activities.

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

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 every time the axle spins once through this sensor it 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 can be negative or positive depending on the direction of the rotation. 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 Investigation Module there is a lab, *Wheels/Distance* 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.

It is suggested that teachers complete the following:

Build and test each of the rotational sensor attachments shown in the Hardware Module. There are two Slideshows 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 $3/5 \cdot 16$ 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:

ROBOLAB VideoTrainer – Measured Movement, Measured Turns

Rotation Sensor engineering and programming pages in 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 below:

<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 in the 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 degrees. 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.

Conditional Loops – This example introduces students to the conditional loops section of ROBOLAB. The conditional loops allow the programmer to write programs without jumps and lands. This will become very useful when you introduce your students to SUBVis.

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 in the Investigations Module:

- 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 line 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

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 one they are using in their program.

It is suggested that teachers complete the following:

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

Introduction to Advanced 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 engineering through 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.
- Events, which allow a programmer to activate a switch that is triggered when something happens

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.

Technological Concepts taught in this lesson

Electronic Control

Measurement

Conversion of Units

Basic Algebra

Geometry

Logical Thinking

Programming Concepts

Resources

ROBOLAB Video Trainer – Containers, Multitasking, Subroutines

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

Teacher will:

Prepare presentations on:

- Containers
- Timers
- Multitasking
- Subroutines
- Events

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 icon represents the container and the other container icon represents what is in the container. 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; it can be used as a numerical constant.
- The robot stops

Rotational Sensor Fork – This example demonstrates one way to control a the length of time to track a line using a rotation sensor

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 Eight: **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/Constant Time
- Gears and Speed/Constant Distance
- Gearbox Hoist/ Torque
- Worm Gears/Properties
- Pulleys/Equilibrium
- Belts/Ratios

All 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 while science, on the other hand, is impacted by many variables. These investigations use discovery based learning and authentic assessment.

It is suggested that teachers complete the following:

Review the following:

- "Teaching Mach and Science with Robots" PowerPoint presentation from the root directory
- "Gears" PowerPoint presentation from the root directory.
- Review the Investigator 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 teachers' 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>

Gears and Speed/Constant Time

Approximate classroom time: 1-2 periods depending on students' background

In this investigation students will learn how changing the gear ratio on a robot's drive train affects how far it will travel. The control variable in this lesson is the amount of time the robot's motors remain on. The motors will be powered for three seconds and turned off. The first time the robot is run, using a 1-1 gear ratio, the student will measure the distance traveled and record that value. They will follow this same procedure for the next two conditions; 5-1 & 3-1 gear ratio. They will compare the distances traveled in each condition to the gear ratio for the condition and see if there is a mathematical relationship.

The surface that you run your robots on will affect the distance traveled.

Extension activity – After the first condition, have the students measure the distance the robot traveled and then use that data to extrapolate the distance the robot will travel in the 5-1 condition and the 3-1 condition.

Gears and Speed/Constant Distance

Approximate classroom time: 1-2 periods depending on students' background

In this investigation students will learn about the inversely proportional relationship between gears and speed when distance is a constant. As the gear ratio increases the speed of the robot decreases. Students will use a light sensor to accurately control and measure when the robot stops. The light sensor will be focused on the ground; when the light sensor sees the end line it will stop and record the amount of time it took to travel that distance. Students will record the time, change the gear ratio on the drive train, and run the experiment again. At the end of the investigation, students should see a mathematical relationship between distance traveled, time, and gear ratio.

Gearbox Hoist/Torque

Approximate classroom time: 1-2 periods depending on students' background

In this investigation students will see the relationship between torque and the gear ratio applied to the driven axle. You will notice that the design of the gearbox uses a belt between the motor and the driven axles. The belt serves as a slip point so that the motor isn't damaged. The belt, a rubber band in this experiment, will have a significant impact on the amount of weight the gearbox is able to lift. A new-tight rubber band will lift much more weight than an old worn rubber band.

When you study this gearbox carefully you will recognize that the experiment starts out with a 1-1 gear ratio. In the second condition there is an 8 tooth gear turning a 24 tooth gear which generates a 3-1 gear ratio. In the last condition, there is an 8 tooth gear turning a 24 tooth gear which is coupled to another 24 tooth gear via the axle. The second 24 tooth gear turns a 40 tooth gear. As students begin to apply the formula for

calculating compound gear ratios they will discover that the two 24 tooth gears cancel each other out leaving an 8 tooth gear driving a 40 tooth gear leaving a 5-1 mechanical advantage. We ended up choosing the last gear ratio because it would have taken an extraordinary number of pennies to stall the motor for this experiment had we gone to the 15-1 gear ratio.

Belts/Ratios

Approximate classroom time: 1-2 periods depending on students' background

In this investigation, students learn that belts behave just like spur gears; they have "gear ratios." This is the ratio of the diameter of the output pulley to the diameter of the input pulley. There are some major differences, however, between belts and pulleys and gears. The first difference is the direction of rotation. With a pair of spur gears, the input axle and the output axle rotate in opposite directions; with a belt and pulley, both the input and output axles rotate in the same direction. Students will also learn that the second major difference is belts cannot transfer as much force as gears. The pulleys move because of the friction between the belt and the pulley. If the output axle were to get stuck, then there would be too much force for the belt friction to overcome, and the belt will slip. You should use a belt and pulley instead of a gear if there is a chance that the output axle, like the robot's wheels, might get stuck. This will keep the motor from locking up and failing.

Worm Gears/Properties

Approximate classroom time: 1-2 periods

Worm gears have several important characteristics that are uniquely different from conventional spur gears. First, worm gears change the direction of the rotation. The driving axle is perpendicular to the driven axle. Second, worm gears provide a large mechanical advantage. And third, worm gears cannot be back driven; when the power is turned off the gear doesn't slip. This is a large advantage when building robots. This investigation is designed to allow students to discover these characteristics.

This experiment is different than the other experiments in the fact that it asks for a qualitative analysis rather than a quantitative analysis. This may be a good time for the teacher to discuss the difference between the two.

Pulleys/Force, Distance, Work

Approximate classroom time: 1-2 periods

Even though this is not a robotic investigation it was added to complement the mechanics section. In this investigation, students will discover the properties of pulleys. This investigation reinforces the scientific concept of equilibrium; the offsetting forces become evident to the student. Even though the pulley system the students test is able to lift twice as much weight with two pulleys, they will have to pull the weight for twice the distance.

Unit Nine: **Introduction to Measurement**

Note to the Teacher:

Measurement is an area of study that many children have trouble with. The goal of this set of investigations is to put measurement in context. At the same time students will learn scientific processes as they accurately measure, track and analyze data.

The following investigations are found in the Measurement Section of the Investigations Module

- Wheel Size/Distance
- Compound Gears/Ratios
- Fractional Parts/Containers
- Probability Wheel/Chance
- Equivalent Fractions/Predictions
- Unit Conversions/Factors

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 an 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

Wheel Size/Distance

Approximate classroom time 1-2 periods

In this investigation students will identify a relationship between the size of the wheel and the distance traveled when the number of turns of the driving axle remains constant. Students are required to measure the distance traveled in centimeters, record the value, and then convert it into inches, and record the value. Students are required to write a conclusion about what they learned. The surface the students test the robot on will have a significant impact on the data collected; testing on a rug will have more friction than a tile floor.

Extension activity 1

This investigation does not require the student to measure the diameter and calculate the circumference. An extension to this activity could require students to do this. This extension also gives the teacher an excellent opportunity to talk about and use the mathematical term interpolate. The dictionary definition of interpolate means to fill up intermediate terms of, as of a series, according to the law of the series; to introduce, as a number or quantity, in a partial series, according to the law of that part of the series. Students will collect data based on distance and time and could be asked to interpolate more data based on the initial sample collected.

Extension activity 2

Given the first wheel diameter, the number of clicks of the rotation sensor, and the distance the robot traveled, have the students calculate the diameter of the wheel based on the distance the robot traveled in each investigation.

Compound Gears/Ratios

Approximate classroom time: 1-2 periods depending on students' background

In this experiment students discover the relationship between the RPM of the driven axle and the gear ratio that drives that axle. They will use a rotation sensor to measure the RPM of each driven axle.

Extension activity

Have students vary the power level of the motor and conduct the experiment again. This data collected from this investigation would allow them to measure RPM of the spindle at various motor power levels. This may be important when trying to control robot speed in later experiments using power levels.

Fractional Parts/Containers

Approximate classroom time 1-2 periods to run the experiment, a longer amount of time to teach the concept of variables

This investigation requires students to use containers to pass variables. The concept of variables can be difficult for students. Look in the extra resources folder on the root directory and you will find a PowerPoint presentation that can be used to present the topic of containers.

In this investigation students will use sensors to collect data and assign the value to a variable; a container in ROBOLAB. They will use the “container operations” pallet to mathematically manipulate the values the sensor collected to have the robot travel fractional parts of the whole distance. They will measure the distance traveled to see if their distance traveled is the predicted distance. If it isn't, then enables the teacher to begin to discuss systematic and random error and challenge the students to refine their experimental procedures to improve their results.

Probability Wheel/Chance

Approximate classroom time: 1-2 periods depending on students' background

Probability is the study of chance. This is a fun experiment that allows students to build a robotic test bed, gather data, and analyze the results. They will use the random number generator built into ROBOLAB to turn the spinners. Students will begin by recording only their data. Initially, the results may be skewed because students will be using a limited number of data-points. The final step of the investigation has students integrate the whole class's results. This should normalize the experimental results.

Equivalent Fractions/Predictions

Approximate classroom time: 1-2 periods depending on students' background

Systematic and random error often plays a significant role in the data a scientist collects. In this investigation, students will learn about systematic and random error and will be challenged to analyze the variables in the investigation to improve their results. Systematic error is something in the experiment that always throws off the data in the exact same way. In this experiment, the momentum of the robot carries the robot further than the investigator wants. (This may be minimized by lowering the motor power level or the robot's speed) Random error is caused by small factors that constantly change and affect the experimental results. In this experiment, random error may be caused by too much friction, slippage, test surface, varying starting points, sloppy measuring procedures, or battery level. Students will study the effect of systematic and random error and will try to eliminate these factors when they repeat the experiment. When finished students are required to write a conclusion about what they learned. The surface that you are testing on will have a significant impact on your robots traveling distance.

Unit Ten: **Motion Planning**

The motion planning section of the Investigators Module was included to help students determine how to successfully get their robot to move from point to point using either feedback from sensors to measure distances traveled, or using feedback from sensors and geometry from environment to move point to point.

One definition of a robot is: Sense-Plan-Act. This module is designed to reinforce that concept. It also focuses on programming using behaviors. The unit is divided into three lessons:

- Discovering Pi/Circumference
- Measured Turns/Radius
- Building Behaviors/Programming

Discovering Pi/Circumference

Approximate classroom time: 1-2 periods depending on students' background

As the diameter of the wheel changes on the robot so do other attributes of the robot.

- The robot will move faster if the wheels are larger if everything else remains constant.
- Feedback from the rotation sensor will not be as accurate as the wheel diameter increases if the resolution of the encoder remains the same.
- The speed of the robot may cause problems with accuracy because of inertia

If students slow their robots down, they may get better results.

Measured Turns/Radius

Approximate classroom time: 1-2 periods depending on students' background

In this investigation students will identify the key elements considered when designing and programming a robot to turn. This investigation will deal specifically with a pivot turn, that is when one wheel remains stationary and the other wheel is driven forward or backward. The three major considerations are:

- Diameter of the drive wheels
- The resolution of the encoder
- The center to center distance across the drive wheels when using the robots for this investigation

This investigation reinforces applied geometry, conversion of units, design in a technological world, and form follows function.

Building Behaviors/Programming

Approximate classroom time 1-2 periods

In this investigation students will learn to identify the difference between a basic, simple, and complex behavior. Students will use the lessons learned from this investigation to write future programs. The resources required for the investigation will be a Tankbot, a light sensor, a roll of electrical tape, and the assigned behaviors. They will learn to break down robot behaviors into basic behaviors, simple behaviors, and complex behaviors. An example of a basic behavior is turning on a motor. An example of a simple behavior is turning on two motors to move in a forward direction, an example of a complex behavior is to turn on both motors and to move forward until the light sensor sees black.

Extension activity: There are many challenges that a teacher can design using black electrical tape and a whiteboard. Robomazing, roboslalom, and robotracker are simple examples that are on the Robotics Educator CD. There are many more complex challenges on the Robotics Explorer CD.